VOL 3 (2019) NO 4

e-ISSN : 2549-9904
ISSN : 2549-9610

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

Big Data Environment for Realtime Earthquake Data Acquisition and Visualization

Louis Nashih Uluwan Arif[#], Ali Ridho Barakbah[#], Amang Sudarsono[#], Renovita Edelani[#]

[#] *Department of Information and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia*
E-mail: louis@pasca.student.pens.ac.id, ridho@pens.ac.id, amang@pens.ac.id, redelani1410@gmail.com

***Abstract*— Indonesia is a country that has the highest level of earthquake risk in the world. In the past 10 years, there have been ± 90,000 earthquake events recorded and always increasing along with the explosion of earthquake data occurs at any time. The process of collecting and analyzing earthquake data requires more effort and takes a long computational time. In this paper, we propose a new system to acquire, store, manage and process earthquake data in Indonesia in real-time, fast and dynamic by utilizing features in the Big Data Environment. This system improves computational performance in the process of managing and analyzing earthquake data in Indonesia by combining and integrating earthquake data from several providers to form a complete unity of earthquake data. An additional function is the existence of an API (Application Programming Interface) embedded in this system to provide access to the results of earthquake data analysis such as density, probability density function and seismic data association between provinces in Indonesia. The process in this system has been carried out in parallel and improved computing performance. This is evidenced by the computational time in the preprocessing process on a single-core master node, which requires 55.6 minutes, but a distributed computing process using 15 cores can speeds up with only 4.82 minutes.**

***Keywords*— Earthquake Big Data Environment, Earthquake Big Data Acquisition and Visualization, Earthquake Data Analysis**

I. INTRODUCTION

Indonesia has a high level of earthquake risk in the world. Within 10 years there were ± 90,000 earthquake events. Throughout 2018, there has been a significant increase in earthquake activity in Indonesia compared to the previous year. This is based on earthquake data from Badan Meteorologi, Klimatologi dan Geofisika (BMKG-Indonesian Meteorology Climatology and Geophysics Centre), wherein 2017 the number of earthquake activities occurred as many as 6,929 times and in 2018 there was an increase to 11,577 times [1]. Some of the earthquakes in 2018 which caused heavy casualties and material losses were the Lombok earthquake (July and August 2018) and the Gorontalo earthquake (December 2018). The earthquake that occurred in Indonesia was caused by plate activity and located in the Ring of Fire Zone.

Earthquake activity in Indonesia increases every year so the number of earthquake data recorded by data providers increases too. Earthquake data that have been recorded from 1900 until now has accumulated increasingly days and it will explode at any time. The abundant earthquake data resources have not been managed properly so far. Because of the large amount of earthquake data, the analysis process of this data also requires a very long time. Critical information of earthquake data is really needed quickly, but the analysis process takes a long time caused by the process of analyzing data using just a single thread computation. Many earthquake data are processed statically, static data is formed by collecting all data, after that it is analyzed according to the specified time limit. Static earthquake data analysis cannot represent information about the latest earthquake.

This earthquake data requires a framework for storing and processing data. Each provider stores earthquake data that have different characteristics according to the way the sensors of each provider capture earthquake signals. If all available earthquake data can be combined and integrated into one, it can certainly be a complete source of earthquake data. This data can be used for a better seismic analysis process.

II. RELATED WORKS

Many researchers had built Big Data Environments to stored, managed and integrated data from various providers and accelerate computing performance for earthquake data analysis. Wang Xiuying et al. [2] introduced the basic idea of large data research, analyzed the need for the application of big data in seismic observation, investigating certain problems and solutions when applying this technology to work in seismic domains.

Pramod Ravindra Patil and Vivek Kshirsagar [3] conducted a study on earthquake database compression using the Hadoop Hive ORC format. Big Data is data that is usually unstructured. Oladotun Omosebi, et al. [4] presented a subscription service based on FIWARE, which provides information from earthquake data that has been analyzed for disaster management scenarios and can inform users based on the severity of seismic activity from various locations around the world.

Evaldas Luksys et al. [5] conducted research focuses on proposing a tool to enable researchers to analyze and interpret large-scale datasets about earthquakes. ETEA (Enabling Tools for Earthquakes Analysis) is a point of user interaction where four four different tools are linked together for complementary and integrated approaches to analyze earthquake datasets.

Gourav Gupta et al. [6] has analyzed and visualized India seismic data from 1800 to 2014 using Big Data technology for the magnitude given with the exact geographical location, date, and amount of causality in a fraction of a second which is not possible with traditional techniques with the help of Hadoop Hive which is one of the Big techniques Data.

G. Asencio–Cortés et al. [7] made earthquake predictions in the California region using powerful computational techniques to analyze big data has emerged, allowing an analysis of large-scale data sets. This new method uses physical resources such as cloud-based architecture.

Lieu-Hen Chen et al. [8] created a 3D visualization of the earthquake's Big Data data which aims to further increase user awareness of earthquakes in Taiwan. This study uses earthquake data from the Central Weather Bureau taken from magnitudes greater than 3 SR from 1992. A. R. Barakbah et al. [9] conducted a Big Data Analysis for earthquake risk mapping system based on earthquake density projected to provinces in Indonesia.

From some of the studies above, this research deals with data retrieval from several providers with pre-processing and integration of the data in the Big Data Environment with multi-thread computation. Also, we provide earthquake analysis functions with faster computational process than single thread computation.

III. Proposed System And Originality

In this research, we propose a new system to collect, store, manage and process earthquake data in Indonesia in real-time, fast and dynamic by utilizing features in the Big Data Environment that can improve computing performance in the process of managing and analysing data earthquake in Indonesia by combining and integrating earthquake data from several providers to form a complete unity of earthquake data. An additional function is that we provide an open access with API (Application Programming Interface) embedded in this system to help access the results of earthquake data analysis such as density, probability density function and seismic association between provinces in Indonesia.

The modelling to form complete and dynamic earthquake data consists of the process of taking, pre-processing, storing data. Data is taken streaming from 1900 obtained from several providers and combined into one, namely the United State Geological Survey (USGS), the Center for European-Mediterranean Seismology (EMSC), and the Center for International Seismology (ISC). The process of data collection, merging and integration of data from three providers uses the latest method proposed by the researchers. All data combined and integrated can have data redundancy so it needs to be reduced. The data reduction process is done by calculating the closeness between the data using the Dynamic Time Warping method. Before data is stored, the location of the province of each data must be determined by calculating the proximity of the point to the Polygon GeoJson province in Indonesia.

The storage uses a database that is integrated with the Big Data Environment. The process of preprocessing and managing data is done in parallel. Methods for processing earthquake-related analyzes are embedded and implemented in this framework. Methods for analyzing earthquake data available are density calculation, Probability Density Function (PDF), and Association Rule. The results of this method can be taken by unit of time according to day, week, month and year.

The big data environment is used to facilitate and speed up the process of obtaining, preprocessing, and processing data and improving computational performance in earthquake data analysis. All processes in the Big Data Environment are carried out in parallel and the availability of a framework that is easily developed and managed.

In this study, a new system is developed to retrieve, preprocessing, store, manage and process earthquake data in Indonesia in real time by utilizing features in the Big Data Environment in which the general design of the system is explained in Fig. 1.

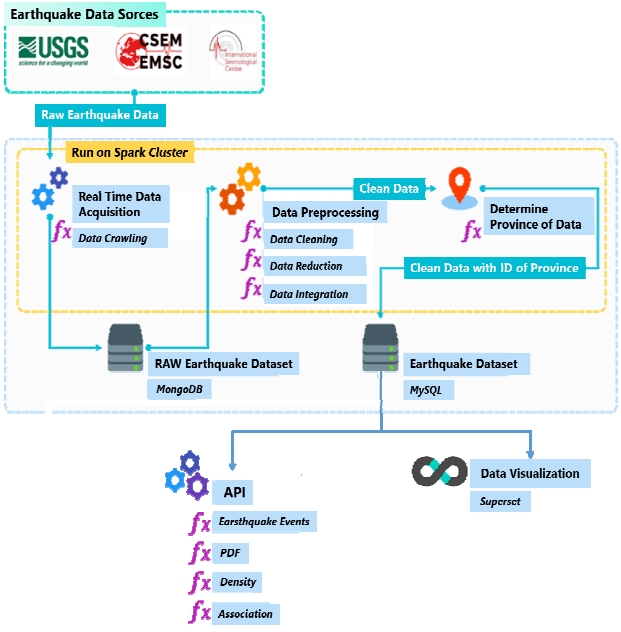

Fig. 1 Research System Design

A. Real-Time Data Acquisition

In this stage, researchers build a server used to retrieve data. The system retrieves earthquake data in Indonesia from various sources such as the USGS (United States Geological Survey), EMSC (European-Mediterranean Seismological Center), and ISC (International Seismological Center) within

a certain period. Each provider has certain rules and formats for retrieving earthquake data. Data retrieval is done by the crawling process from the data provider websites. The Crawling process utilizes the API provided by each earthquake data provider. The API helps researchers to get data in real-time every 5 minutes, where data will be directly retrieved whenever a new earthquake data occur. The result of this process is a collection of earthquake data that is still raw from various sources. The data that is entered into the raw database that integrated with Big Data.

B. Data Preprocessing

Before the data entered into a database that is tasked to storing clean data, the data will undergo a preprocessing process. The stages in this process consist of three main processes, specifically the process of data cleaning, data integration, and data reduction. The processes at this stage are:

1) Data Cleaning

Data that has been obtained in the crawling process, do the cleaning process first before going to the next process. The cleaning process is done to get rid of crawling data that is incomplete or has a missing value. Data is considered incomplete if there are empty values in one of the DateTime, latitude, longitude, depth and magnitude attributes. The cleaning process is needed because the data used in this study must be completed.

2) Data Integration

After the data collected from each provider is cleaned, the data is entered into the integration process. The data integration process is the stage to equalize the attributes that will be used. Each provider has data characteristics with various attributes. However, this study only requires five important attributes. At this stage, data from each provider is taken only five attributes, i.e DateTime, latitude, longitude, depth, and magnitude. Data cleaning results were merged into one database.

3) Data Reduction

Data from various providers that have been merged and integrated, of course, there is redundancy of data that has the same earthquake source. Each provider records an earthquake event with differents methods. This recording depends on the distance of the seismograph with the earthquake point, the method of calculating the depth of the earthquake, and others. So that an earthquake event can be recorded differently in each provider. This data reduction stage is used to detect data from each provider that both records one earthquake event and takes just one data that represents one earthquake event. In the process of data reduction, this study uses the Dynamic Programming method (the calculation method used by DTW) to calculate the proximity of data. The DateTime, latitude, longitude and magnitude attributes in the data are used to calculate the distance. These four attributes are taken because they have close distance values with other data that have the same earthquake source [10]. Dynamic time warping (DTW) is a technique for finding optimal alignment between two data sequences (time dependent) given within certain limits [11].

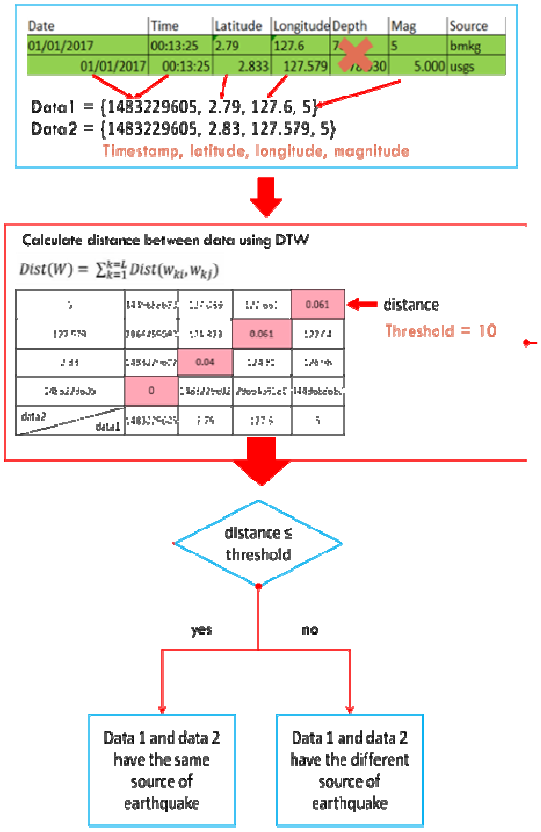

Fig. 2 Data reduction process with the DTW distance calculation method

Method to find the optimal mapping path using Forward Dynamic Programming, which can be concluded in the following 3 steps:

- Optimal Value Function: definition $D_{(i,j)}$ as the distance of DTW between $t_{(1:i)}$ and $r_{(1:j)}$, with the mapping path starting from (1,1) to (i, j).
- Recursive Formula :

$$D_{(i,j)} = |t_{(i)} - r_{(j)}| + min\begin{Bmatrix} D_{(i-1,j)} \\ D_{(i-1,j-1)} \\ D_{(i,j-1)} \end{Bmatrix} \quad (1)$$

  with initial conditions $D_{(1,1)} = |t_{(1)} - r_{(1)}|$

- Final result : $D_{(m,n)}$.

The stages described in Fig. 2 are first step, change the DateTime attribute in the form of a timestamp, secondly forming a matrix D of the two data to be compared and filling value with the data own value, then using the recursive formula, we fill all the elements of the matrix one by one, following the column-by-sequence column or row-by-row. The final answer will be available as $D_{(m,n)}$, with computational complexity $O_{(m,n)}$. If the data has a proximity ≤ threshold (10), then the two data are from the same earthquake source and vice versa.

After grouping data from the same earthquake source, the next step is to get one data that represent one earthquke source. We calculate the centroid value of each group, then calculate the proximity between the data and the centroid. The data closest to the centroid is chosen to be the data that represents the source of the earthquake.

At this stage, there are two data reduction processes. In the first process, a collection of crawling data that has been integrated will be reduced first with each other. The first reduction process produces crawling data derived from a different earthquake event.

In the second process, the reduction of crawling data is compared with the data in the database. This process is carried out in a way, data crawling results are compared with data one hour before and data one hour after the time available in the data crawling. The reduction process in the second stage produces earthquake data that represents an earthquake event stored in a database.

C. Determine Province

Earthquake data that have been obtained do not have provincial information. Preprocessed earthquake data needs to determine the location of the province before it is entered into the main database. Determination of the province of each data can be done by checking the latitude and longitude values in the data. The process utilizes GeoJSON polygon data for each province in Indonesia. Province is determined by finding the point of whether it is inside one of the polygons of a province. If the data is not contained in one of the polygons, it is sought with the closest point of the existing polygon. Polygon data is taken from GeoJSON data that has been managed by the Central Statistics Agency.

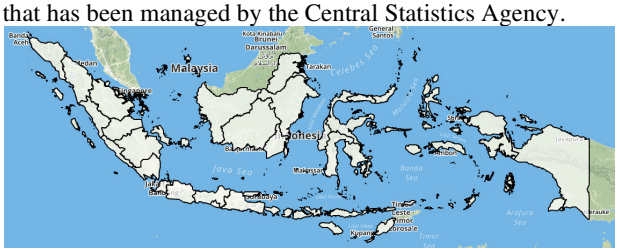

Fig. 3 GeoJSON Indonesia Administrative Boundary Level 1 data plot shown by black lines on the map

GeoJSON data taken from Indonesia Administrative Boundary Level 1 which contains the boundaries of a province in Indonesia on each island. GeoJSON data contains latitude and longitude that form a polygon along with the attributes of the polygon. In this case, the attribute taken is the name of the province. The following is GeoJSON data for the determination of provinces if they are drawn in the form of dots that are related to each other to form a multi-polygon.

D. Earthquake Dataset

Data that has gone through all the preprocessing stages is then stored in a MySQL database integrated with the Big Data Environment. The data is already unique and there is no redundancy. Earthquake data has the attributes of date, time, latitude, longitude, depth, magnitude, and id as well as the name of the province. These attributes are very important for data processing. This dataset is in the Big Data environment that automatically manages the organization of data in it. This earthquake dataset can be accessed by all earthquake data processing application platforms and can be used by other earthquake researchers.

E. Application Programming Interface (API) and Earthquake Data Analysis Function in Big Data Environment

The server provides functions that are often used to process earthquake data intended for users. The server puts the function into an API that ensures data is sent and displayed correctly. These functions consist of taking data on the server, calculating earthquake frequency, density, Probability Density Function (PDF), and Association Rule between provinces. These functions have been completed formed from previous studies, where the process of modeling and managing data statically, can be done dynamically at this time. In this research, researchers are tasked to incorporate these functions into the Big Data environment. This function is used for the computational process of analyzing earthquake data which produces up-to-date information about earthquakes.

The Density function [12] requires an Automatic Clustering algorithm that is used to automatically group earthquake data based on its location or latitude and longitude. The density calculation function is used to calculate the level of earthquake hazard in a cluster or region by using an area calculation. Probability Density Function (PDF) [13] is used to analyze the level of earthquake vulnerability in an area using mathematical calculations. The Association Rule function [14] is used to identify earthquake data associations by looking at seismic relations between provinces in Indonesia. Thus, the results of all earthquake data modeling using data streaming can provide up-to-date and better seismic information.

The Big Data modeling structure in this study was built from several technologies and frameworks, there are:

1) Framework Design

The framework design in this study explains how the flow of earthquake data processing is carried out on what framework and API are used for the processing. The server retrieves earthquake data from several providers, then pre-processing data (such as data cleaning, data reduction, and data aggregation) and determining the province at the earthquake location. So that the process does not take a long time, then all the existing nodes do work with multi-threading (multi-core tasks). To do this work, the server uses Apache Spark technology in the form of RDD API.

RDD API is used to parallel the data collection to all nodes, then parallelized to each core on the processor in each node. Cores that will be used in this process are 15 cores, so for each processor 5 cores will be used to run this process.

Afterward, data is processed in each core, then the data can be stored in a MySQL database. The reason to use MySQL as database storage is that the query process on MySQL is faster than HDFS. Besides the earthquake data is structured data so that it is suitable to use MySQL.

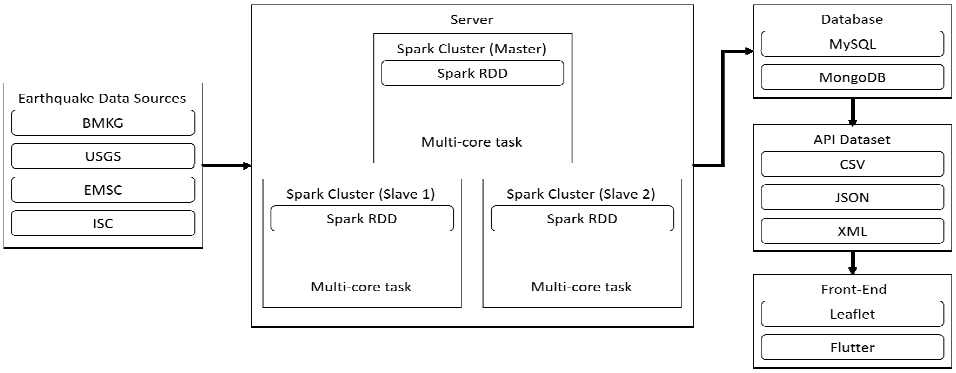

Fig. 4 Framework Design

Data in the database will be used as a dataset to be accessed by utilizing the API provided by the server. The API can be accessed by anyone and will produce different forms of data - depending on the request made. Forms of data can be CSV, JSON, or XML.

2) Server Design

The Big Data modeling structure in this study was built from several technologies and frameworks. Frameworks used are Kafka, Spark, Druid, and Superset.

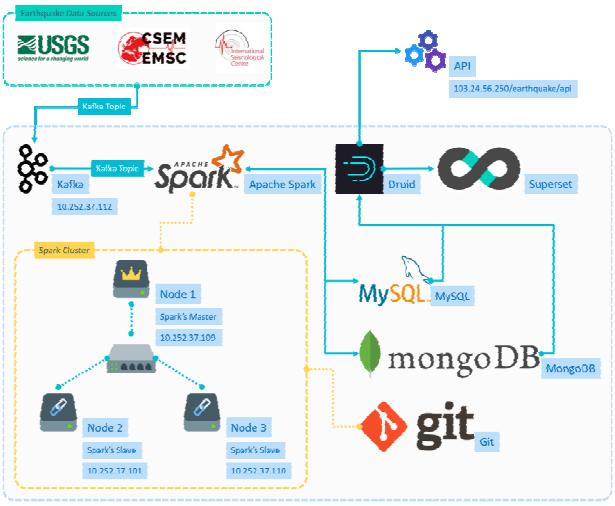

Fig. 5 Server Design

The following is a list of the frameworks and technologies that make up this system:

- Kafka. It functions to build real-time streaming data streams from several earthquake providers, i.e USGS, EMSC, and ISC. This application performs streaming and real-time data retrieval every 5 minutes by reacting to changes in data flow at the provider such as the presence of new data. Kafka is run as a cluster on single or several servers that can reach many data providers. The Kafka cluster stores stream records in a category called topic. Each note consists of a key, value, and timestamp.
- Spark. also known as Apache Spark is an open-source cluster computing framework, built for large and fast data processing. Spark is used to speed up the computing process at Hadoop. In this system, Spark is used to speed up the process of data preprocessing and seismic data analysis. Spark programming APIs used by this research are Java programming and SQL database. Besides supporting all of this workload, Spark also reduces the burden of managing maintenance tools separately.
- Druid. It provides fast analytical queries, at high concurrency, on event-driven data. Druids are used to storing, query (request), and analyze large flows. In this study, the druid platform was used for the OLAP process. When a SQL database is off, earthquake data can still be accessed using a Druid, because Druid also saves the data. Druids are optimized for sub-second queries to divide the amount of data into smaller parts, drill-down (access data at the lowest level in the hierarchy in a structured database), search, filter, and aggregate data.
- Superset. It is a web-based application that is used to visualize data. This superset is used to visualize the data contained on the server.

IV. Experiment And Analysis

The stages of implemented the steps in the proposed system design was part of the experimental process. The results of this experiment consist of the implementation of establishing communication between nodes, real-time data collection with automatically preprocessing data and determination of provinces from earthquake locations, Data Visualization, Earthquake Data Analysis API with Open Access Function, and computational performance analysis.

A. Establishment of Communication between Nodes

In this research, we used Spark and Hadoop as Big Data frameworks to process earthquake data. For Hadoop, only HDFS is used to deploy programs to all nodes. Spark and Hadoop are installed and configured on each node so that they can communicate directly and safely using SSH (Secure Shell). In the system created, the cryptosystem used on the host key in the SSH protocol uses RSA

(Rivest-Shamir-Adleman). The security of the RSA algorithm is based on the fact that factorizing large integers are known to be difficult, so communication will be safer.

For each node to communicate directly and securely, the first step that must be taken is to generate a public-key on each node. After each node generates a public-key, the public-key that has been generated at each node is added to all existing nodes. Then run the command "ssh" on each node to all nodes so that all nodes are automatically added to the list of "known_host" so that for further connections we can do it directly without entering the password first.

B. Real Time Data Collection

To provide complete earthquake data for Indonesia, the server crawls data from various sources of earthquake data providers. Earthquake data must be located in the territory of Indonesia, namely latitude -11 - 6 and longitude 95 - 142. From the criteria we obtained 3 earthquake data providers, including USGS (United States Geological Survey), EMSC (European Mediterranean Seismological Center), and ISC (International Seismological Center). The features taken are DateTime, latitude, longitude, depth, and magnitude. USGS (available from 1931), EMSC (available from 2004), and ISC (available from 1900) provide APIs or Web Services that can be accessed for earthquake data collection.

TABLE I
LIST OF API URLS FOR EACH EARTHQUAKE DATA PROVIDER

Provider	API URL
USGS	https://earthquake.usgs.gov/fdsnws/event/1/query
EMSC	http://seismicportal.eu/fdsnws/event/1/query
ISC	http://www.isc.ac.uk/fdsnws/event/1/query

The data collection process was carried out from the beginning of the year that the provider provides earthquake data. Crawlers stored the latest date of data taken at preference in the system so that data retrieval can be done efficiently. Each provider had its own initial data retrieval time because each provider provides data at different times. If there was a time when the earthquake data was last accessed in the system being run, the crawler would use that time as the date the data was retrieved. If the initial time is not found on the system that is running, the crawler would use the default time, where the time is the beginning of the year the provider provides earthquake data. The time data is stored in the database to maintain if the database is running dead, the crawler cannot retrieve data from the existing provider.

To do a query, we need to enter several parameters. Required parameters include start time, endtime, minimum latitude (minlatitude), maximum latitude (maxlatitude), minimum longitude (minlongitude), maximum longitude (maxlongitude), and sorting data (orderby). For start and end time is taken from rule crawling that has been made. For the initial and final latitude and longitude, the parameters will remain the same, namely the initial latitude -11, the final latitude 6, the initial longitude 95, and the final longitude 142. For data, sorting is done ascending so that the data is sorted from past to the latest time so that it can run crawling rule. Data access is done by accessing the API URL followed by the parameters that will be input along with the values of these parameters by using the HTTP GET method.

Each provider has its query limits for the data retrieval process and these restrictions greatly affect the data retrieval process. If the data taken exceeds the limits set by a provider, then we will not get the data we requested. Therefore, we must limit the query so that the data retrieval process runs well. The way to limit queries to crawlers made is to limit the time interval from data retrieval. The time interval taken in the crawling process is done in the range of 30 days. If the last time and the current time have a distance of more than 30 days and the collection is carried out up to the current time if it does not meet these conditions. This can prevent the limit of queries for each earthquake data provider.

TABLE II
ACTIONS ON EACH RESPONSE CODE

Response Code	Action
200 (Success)	Perform pre-processing data and enter it into the database. Date of last data collection = latest data + 1 second (update data retrieval time).
204 (No data matches the selection)	Date of last data collection = date of last query + 1 second (update data retrieval time).
400 (Parameter value out of range)	Date of last data collection = date of last query + 1 second (update preference).
409 (Response conflict)	Date of last data collection = date of last query + 1 second (update preference).
Another response code	Continue the crawling process without updating the data collection time.

Data retrieval through the internet is not free from problems. Success or failure of the request that we send can be checked through the request status. For crawlers to run better, we need to check the status of the request that has been sent along with the actions to be taken. To check, the crawler reads the response code in the form of an HTTP status code. The response code provided by the API provider could vary, but researchers have summarized the status code obtained from each process in each provider and retrieve the status code needed for the crawler to be checked in it. By checking and taking the necessary actions, crawlers can certainly run well in retrieving data at each provider. The following is the response code table and the actions taken. Crawler would retrieve data continuously without pause until the crawler reaches the condition that the last data retrieval time at each provider is less than 30 days from the current time. Crawlers active (to all providers) every 5 minutes to retrieve new data.

C. Preprocessing Data and Determination of Province from Earthquake Location

After the crawl process at each provider, earthquake data is still raw. Raw data is data obtained from each provider having a different data format and there is a missing value in the data. The data cleaning process is done by deleting data that has missing values in it. If one feature has a missing value in it, the data will be deleted automatically and not entered into the next process. Data integration

process to integrate all data so that it has a uniform data structure that is ready to be processed. The new arrangement of data attributes has a sequence of time, latitude, longitude, depth, and magnitude. Furthermore, data in kilometers will be converted to meters.

Data obtained from the same or different providers was not rule out the possibility that the data is from the same earthquake event even though some of the attributes in it have different values. To overcome this, data reduction needs to be done to eliminate data duplication. The data reduction process is done by calculating the closeness between data using DTW (Dynamic Time Wrapping) algorithm. The data will be clean data that is used and processed for the next process.

The data reduction process in the system is done twice. The first process is to get rid of data from similar crawling results. The second process is comparing each data obtained from the first process with the data in the database to ensure that the data is different from the database. For this process, the data taken from the database is data with an hour before. The process carried out in this step using parallel computation to speed up the process of comparing each data with the data contained in the database. If the results of the comparison are different, then the data is eligible to be entered into the database.

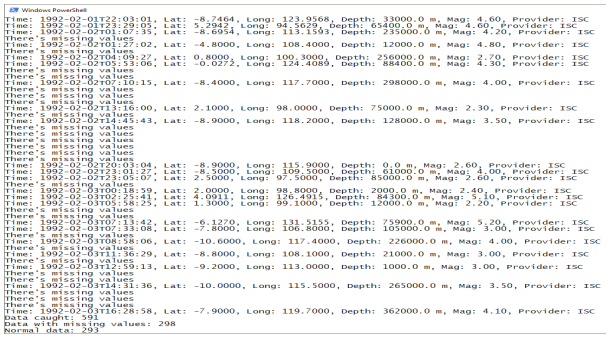

Fig. 6 Results from Real-Time Data Acquisition and Data Pre-Processing

The location of the province defined by determining whether the earthquake points (latitude and longitude) are in the provincial polygon. How to determine whether a point is in a provincial polygon using the PNPOLY (Point Inclusion in Polygon) algorithm [15].

date_time	latitude	longitude	depth	magnitude	province_name	provider_name
1935-08-03 01:10:06	4.401	96.403	25000	7.2	Aceh	USGS
1936-08-23 21:12:16	5.316	94.719	50000	7	Aceh	USGS
1936-09-19 01:01:47	3.685	97.535	20000	7.2	Aceh	USGS
1937-07-01 11:49:49	3.18	95.959	30000	6.3	Aceh	USGS
1945-07-23 03:54:56	5.159	95.916	15000	6.7	Aceh	USGS
1949-05-09 13:36:27	4.829	95.763	50000	6.5	Aceh	USGS
1952-03-08 18:37:43	2.362	97.057	35000	5.6	Aceh	USGS
1952-08-14 16:01:10	2.56	97.182	47900	5.7	Aceh	USGS
1953-11-13 16:17:11	4.05	95.971	45400	5.9	Aceh	USGS
1954-05-02 17:48:09	4.237	95.062	30000	5.9	Aceh	USGS
1955-10-21 04:32:08	4.18	95.372	35000	5.7	Aceh	USGS
1956-04-02 10:50:00	1.864	96.667	29900	6.3	Aceh	USGS
1957-01-10 04:14:52	5.901	95.138	50000	5.6	Aceh	USGS
1957-02-20 21:58:29	2.099	96.877	35000	5.8	Aceh	USGS
1957-03-11 12:09:18	2.073	97.15	35000	6.2	Aceh	USGS

Fig. 7 Earthquake Dataset in Database

This algorithm checks semi-horizontally (the value of x increases, y remains) outside of the test point and calculates how many edges are traversed. Each time the edge passes, checking changes from inside and outside. This is called the Jordan Curve theorem. If the polygon forms a planar graph with each side labeled with two polygon neighbors, then one simple way is to run a semi-infinite check from the point to touch the first edge. Then the relevant polygon is one of the edge's neighbors. Polygons and points contained in these points are very numerous (7504 polygons (1713161 points), so that if the provincial determination process is carried out in parallel it will consume a lot of resources (RAM used for 1 core can reach 1GB). Therefore, the process of determining the province is carried out with a single core and an optimized program.

D. Earthquake Data Visualization

In this research, there is a page to visualize the results of earthquake data that have gone through all the preprocessing processes and ready to use. To visualize the earthquake data, Apache Superset is used so that data can be displayed quickly [16]. In the Apache Superset, there is a dashboard that displays various information related to earthquake data that can be accessed via http://MASTER-IP:8088/.

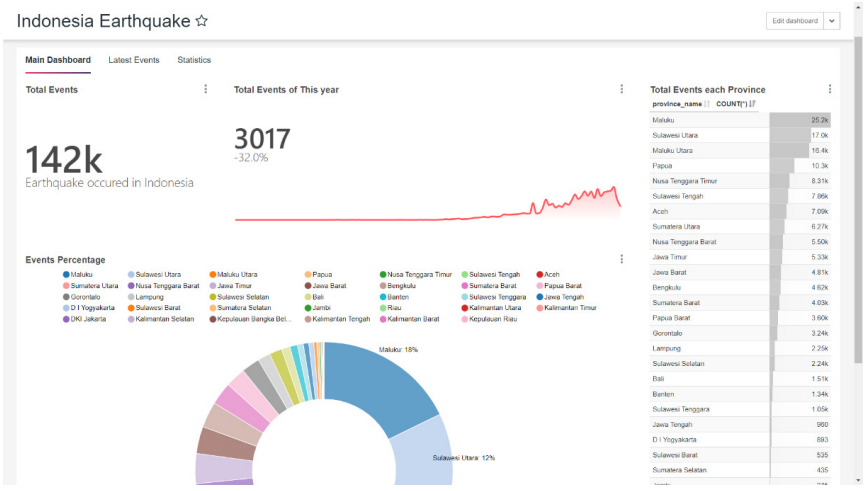

Fig. 8 Dashboard Start Page that Displays Information on Earthquake Data

In the data visualization dashboard, there are three tabs available to display various information. The tabs include Main Dashboard, Latest Event, and Statistics.

The Main Dashboard tab displays the number of earthquakes, the number of earthquakes this year, the number of earthquakes that have occurred in each province, and the percentage of the number of earthquakes that occurred in Indonesia in each province. In the Fig. 8 shows that there have been around 142 thousand earthquakes in Indonesia, and 2017 earthquakes occurred this year (2019) which experienced 32% of the number of earthquakes in the previous year, and indicated that Maluku is a province that often experiences earthquakes with a total occurrence which has recorded around 25.2 thousand earthquake events.

There are seven provinces that frequently experience earthquakes, Maluku (18% of total data), North Sulawesi (12% of total data), North Maluku (12% of total data), Papua (7% of total data), East Nusa Tenggara (6% of the total data), Central Sulawesi (6% of the total data), and Aceh (5% of the total data).

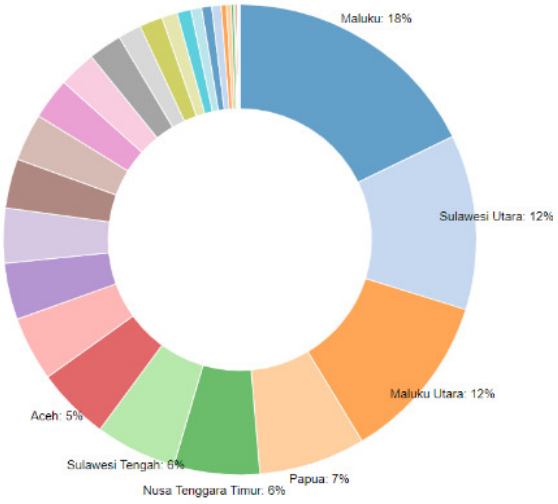

Fig. 9 Percentage of Earthquake Amounts that Happened in Indonesia

The Latest Event tab displays information about the most recent earthquake taken within the last seven days in the form of a slice diagram showing the number of earthquakes that occurred in a particular province grouped by the date of the event and a map showing the location of the earthquake as shown in the Fig. 10.

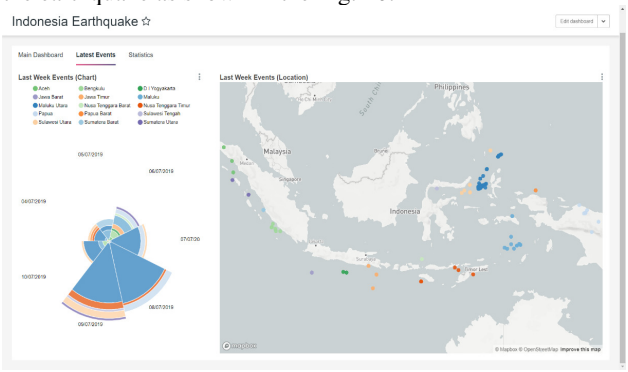

Fig. 10 Latest Events Tab that Shows Number of Earthquake Events and Earthquake Locations in the Last 7 Days

The Statistics tab displays various information in the form of graphs from earthquake data that had been obtained. One of the graphs is a graph about the frequency of the whole earthquake. The frequency graphs shown are of three types, namely earthquake frequency per year, per month and per day as shown in the Fig. 11.

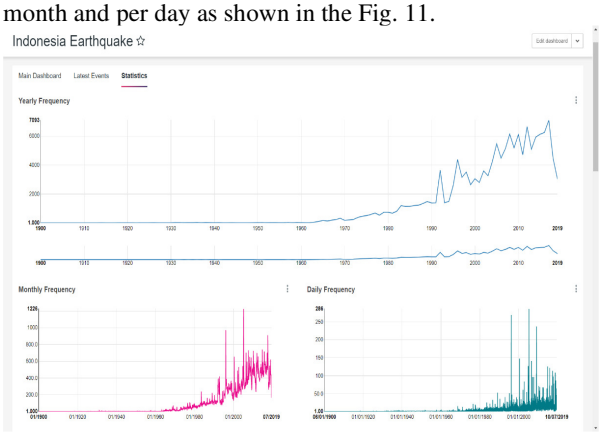

Fig. 11 Statistics Tab that Displays Frequency Per Year, Per Month, and Per Day

E. API for Earthquake Data Analysis and Modelling with Open Access Function

In this system, there are several APIs that can be accessed to retrieve or query data, to get information about earthquake data frequency, and the results of earthquake data analysis functions. To access the API in general by including a link along with additional parameters that you want to input. If the user does not specify a parameter value, then the parameter will be set to the default value. APIs that are available in this system, can be accessed via URL http://103.24.56.250/earthquake/api/[METHOD]?[PARAMETERS]. The API created can produce various data formats in the form of CSV, XML, and JSON so that users can receive the desired format.

1) Data Query

The function of this API is to retrieve earthquake data in the database on server. Users can access the API to get earthquake data via the URL http://103.24.56.250/earthquake/api/query?[PARAMETERS]. The following is a list of parameters and default values available in the earthquake data query API.

TABLE III
PARAMETERS FOR THE EARTHQUAKE DATA QUERY API

Parameter	Default value	Value
Starttime	Today	yyyy-MM-dd or yyyy-MM-ddThh:mm:ss
Endtime	Today-30 days	yyyy-MM-dd or yyyy-MM-ddThh:mm:ss
Minlatitude	Unset (no filter)	Number
Maxlatitude	Unset (no filter)	Number
Minlongitude	Unset (no filter)	Number
Maxlongitude	Unset (no filter)	Number
Mindepth	Unset (no filter)	Number
Maxdepth	Unset (no filter)	Number
Minmagnitude	Unset (no filter)	Number
Maxmagnitude	Unset (no filter)	Number
Format	Json	csv, xml, or json
Orderby	time descending	time, time-asc, magnitude, or magnitude-asc

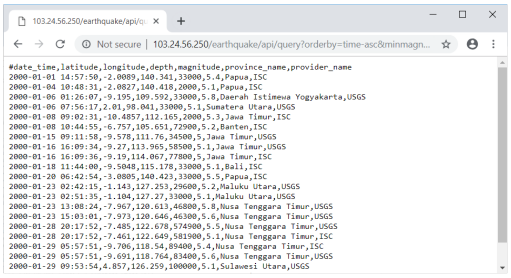

Fig. 12 Result of Data Query API with Parameters starttime=2000-01-01, endtime=2000-01-31, minmagnitude=5, orderby=time-asc, and format=csv

2) Data Frequency, Density and Probability Density Function (PDF)

Frequency, PDF, and density values are taken in periods of days, weeks, months and years. This API can be accessed through the address http://103.24.56.250/ earthquake/api/values?[PARAMETERS] to get results from earthquake data in the database. The following are parameters that can be entered along with the default values of the parameters in the API to retrieve frequency, PDF, and density values.

TABLE IV
PARAMETERS FOR EARTHQUAKE FREQUENCY, PDF, AND DENSITY VALUE APIS

Parameter	Default value	Value
Startdate	Today	yyyy-MM-dd
Enddate	Today-30 days	yyyy-MM-dd
Format	Json	csv, xml, or json

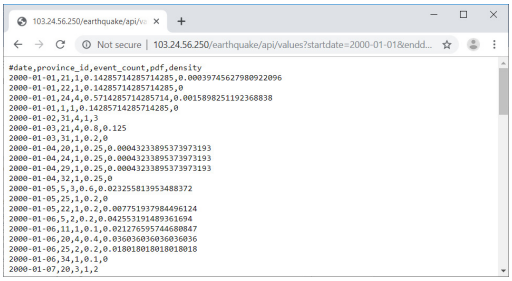

Fig. 13 Result of Data Frequency, PDF and Density API with Parameters startdate=2000-01-01, enddate=2000-01-31, and format=csv.

3) Association Rule between Province

Provincial earthquake association values are obtained from the association every day. This API can be accessed through the address http://103.24.56.250/earthquake/ api/association?[PARAMETERS] to retrieve the provincial earthquake association stored in the database. In accessing API association, it has parameters that can be input along with the default values of those parameters which are the same as the parameters in the API to get frequency, PDF, and density values.

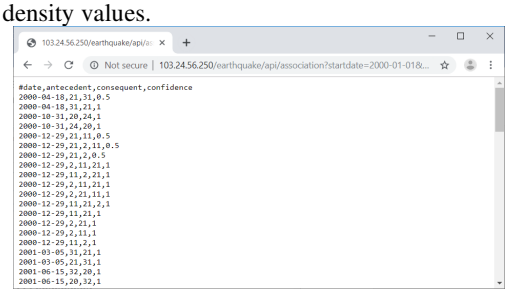

Fig. 14 Result of Earthquake Association API with Parameters startdate=2000-01-01 and format=csv

4) Open Access Function

Open Access Function is a function created with the Java programming language to access earthquake functions available on the server. The functions available in the server are PDF and Density functions. To call the function, the user must create the EarthquakeAPI class and fill the server address and file location to be processed.

TABLE V
CONSTRUCTOR AND METHOD CAN BE ACCESSED VIA EARTHQUAKE API

Constructor / Method	Function
EarthquakeAPI()	Constructs with address domain = localhost and filePath = null.
EarthquakeAPI(domain, filePath)	Constructs with given parameters.
setApiDomain(domain)	Replaces the address domain with the given domain.
setFilePath(filePath)	Replaces the file path with the given file path.
setProxy(host, port)	Set host and port proxy for network connection setting.
getPDF()	Get PDF value from given data.
getDensity()	Get Density value from given data.

Files must be in CSV format. Users can set proxy and port connections if needed in the setProxy method. Then call the functions available on the server to process the desired data. Users can access the getPDF method to get PDF values per area of data and the getDensity method to get density values per area of data. Following is a list of constructors and methods in the EarthquakeAPI class.

```
public static void main(String[] args) {
    EarthquakeAPI api = new EarthquakeAPI("103.24.56.250", "data.csv");
    //api.setProxy("proxy3.eepis-its.edu", "3128");
    String[][] result = api.getPDF();
    //String[][] result = api.getDensity();

    //PRINT RESULT
    for (String[] row : result) {
        System.out.println(Arrays.toString(row));
    }
}
```

Fig. 15 How to call PDF Function in Java

In the Fig. 15, the domain server is shown 103.24.56.250 and the data to be processed has the name data.csv. To access PDF functions on the server, the user can call the getPDF() method. The result of the PDF function in Fig. 16 is a two-dimensional array, where the array contains the name of the region and the value of the PDF (Probability Density Function).

```
[Kalimantan Utara, 0.0034443168771526698]
[Bali, 0.0057405281285878304]
[Yogyakarta, 0.008036739380022962]
[Maluku, 0.11595866819747416]
[Sulawesi Tengah, 0.04477611940298507]
[NTT, 0.06429391504018837]
[Kalimantan Tengah, 0.0011481056257175666]
[Maluku Utara, 0.1285878300803674]
[Kalimantan Selatan, 0.004592422502870264]
[Sumatera Utara, 0.03673938002296211]
[Kalimantan Timur, 0.0057405281285878304]
[Jawa Barat, 0.016073478760045924]
[Papua Barat, 0.055109070034443177]
```

Fig. 16 PDF Function Results

F. Preprocessing Computation Performance Analysis

The performance of preprocessing computation in the system is analyzed based on the time spent on a process. An analysis is done by comparing the processing time when the total cores on the server are changed. This experiment compares the process of local computing time

at the master node and distributed on the server, the effect on total of cores used in running the Distributed Preprocessing program. Where RAM memory resources are used 8 GB per executor.

The first experiment was to test the performance of Distributed Preprocessing when the computing process was done locally on a master node with 10 cores. This research experimented 3 times.

From the graph in Fig. 17, the longest processing time is at 1 core with an average time of 54.15 minutes and the fastest is using 9 cores with an average time of 9.69 minutes. As can be seen in the graph, computing time is faster when the number of cores is added. However, there are oddities when the total cores are 9 and 10. When the number of cores used increases but the performance decreases. For example, in experiment 1, core 9 had 9.68 minutes and core 10 had 9.83 minutes. Experiment 2 and Experiment 3 also experienced the same thing.

From the first experiment, it can be concluded that the performance of computing processes can be increased by adding the number of cores. However, when the core total was 9, computing time is not reduced significantly, with 10 cores had an increased computing time. This is due to one core is usually used for Spark drivers and it cannot be used for other parallel processes. Another cause is the disruption of processes other than this research. The fastest computing time is in the 3rd experiment on the number of cores 9 with 9.64 minutes.

The second experiment is testing the performance of Distributed Preprocessing when it is done on a cluster network. Testing with 3 trial scenarios. Where to use cores up to a maximum of 15 cores.

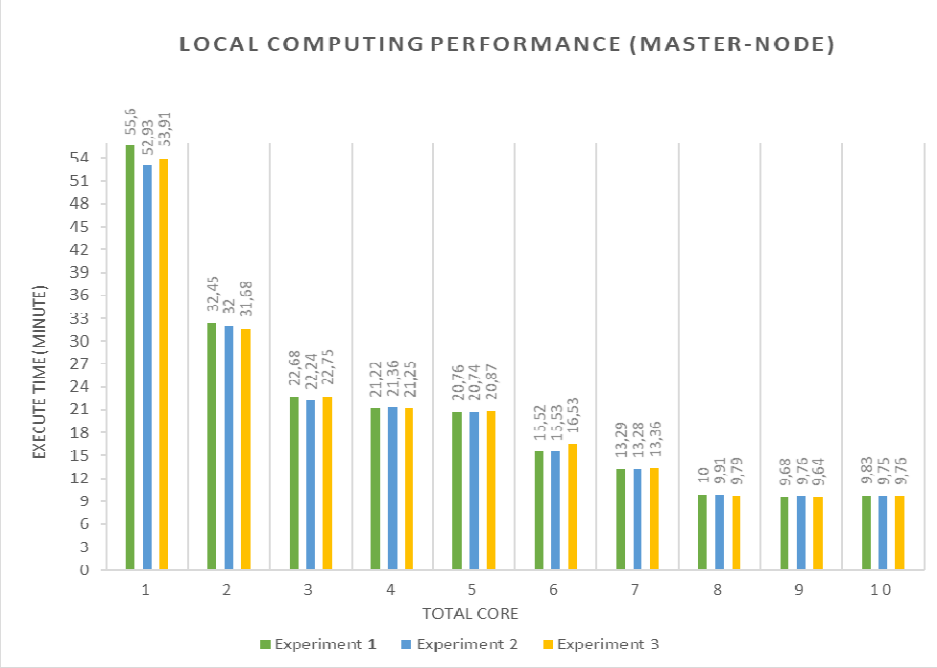

Fig. 18 Local Master Node Computation Peformance Graph

From the graph in figure 18, it can be seen that when the number of cores is added, computing time is getting faster and computing performance is getting better. In the second experiment, the number of cores added from 11 to 12 made the computational time increase. Computing performance dramatically increases when the number of core added from 1 becomes 5 cores, after that it starts experienced a slowdown performance phase. When the total cores were 12, 13 and 14, the performance time is only around 7 minutes. However, when the core number is 15, the computing time starts to descend again.

Discrepancy during the 2nd experiment where the number of cores 8 but the computing time is increasing. This is caused by the existence of another process that uses these cores at the same time as a performance analysis in this study. The reason why there can be other processes that interfere with the parallel computing process of this research is the server that is used together and competes with other research sources. The next discrepancy, when increasing cores in the number of cores 9 to the number of cores 14 experienced a slowdown in computing. The cause is the same thing as the previous oddity. The fastest computing time from the preprocessing process in experiments using a cluster node network is 4.82 minutes in the 1st experiment with a total of 15 cores.

From the two experiments conducted by comparing the performance of the master node with 10 cores and the cluster node network distributed with 15 cores, the results show that the performance of computing time on a cluster network with 15 cores is much faster than the master node. Where the cluster network computing time can reach 4.82 minutes, while the master node with the number of cores 9 only reaches 9.64 minutes. The process of analyzed the performance of computational time in preprocessing data has constraints. This constraint is caused by an error on the server that is used together with other research processes. The core is available not only for preprocessing and seismic data analysis, but there are parallel computing processes used by others.

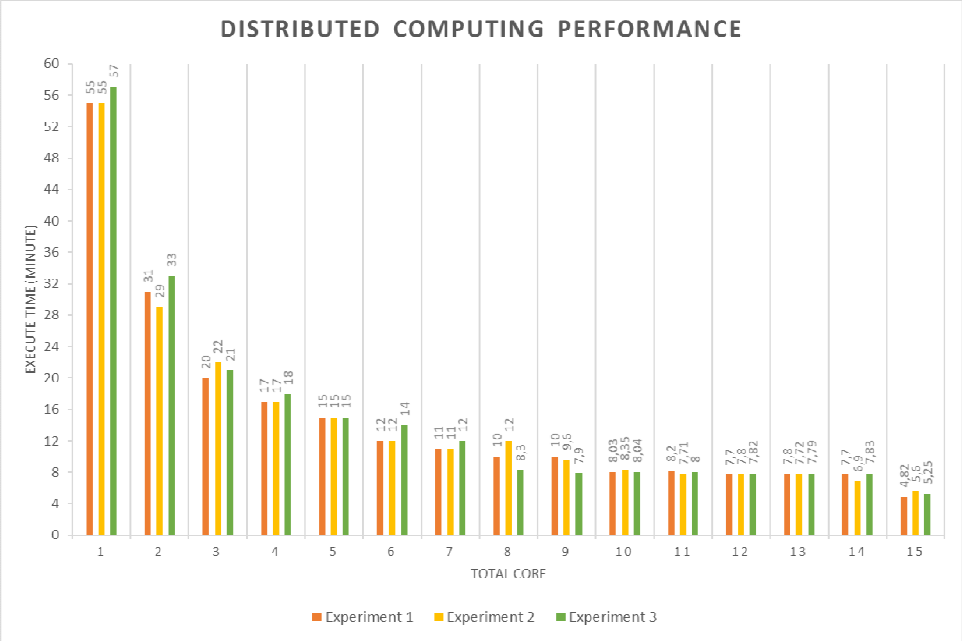

Fig. 19 Distributed Computing Peformance Graph

V. Conclusions

This paper presents a new system for collecting, storing, managing and processing earthquake data in Indonesia in real-time, fast and dynamic by utilizing features in the Big Data Environment. Data collection is done by combining and combining earthquake data from several providers to form a complete unity of earthquake data. This research conducts crawling process with its own configuration and algorithm because each provider has their own criteria for retrieving data.

The preprocessing process is in line with the process of collecting data and determining the location of the province. This process automatically cleans, integrates, and reduces new data captured. In the process of data reduction, this study uses the Dynamic Time Warping (DTW) algorithm to calculate the value of closeness between data that has the same earthquake source. The location of the province is determined by determining whether earthquake points (latitude and longitude) are in the provincial polygon using the PNPOLY (Inclusion Point in Polygon) algorithm.

API (Application Programming Interface) embedded in this system that can be accessed successfully to retrieve or request data, to get information about earthquake data frequency, and the results of earthquake data analysis functions such as density, probability density function and seismic association between provinces in Indonesia.

This system performs improvement of computational performance in the process of managing and analyzing earthquake data in Indonesia. Preprocessing earthquake data is carried out dynamically and parallel and speeds up the computational time and access time of the analysis results. This is evidenced by the computation time in the preprocessing process at the master-node for one core to spend 55.6 minutes, but when used 10 cores is reduced to 9.75 minutes. Other evidence in distributed computing when using one core takes 55 minutes, but when used 15 cores decreases to 4.82 minutes. There are some problems when the core is added, the performance time does not decrease and persists at certain seconds. This problem is caused by the server being shared with several other applications and competing for resources that both run in parallel.

References

[1] R. A. Umasugi, "Selama 2018, Gempa di Indonesia Meningkat 4.648 Kali Dibanding 2017," 29 December 2018. [Online]. Available: https://megapolitan.kompas.com/read/2018/12/29/10303711/selama-2018-gempa-di-indonesia-meningkat-4648-kali-dibanding-2017. [Accessed 15 July 2019].

[2] W. Xiuying, Z. Ling and Z. Congcong, "On Application of Big Data Mining in Earthquake Precursor Observation", *Earthquake Research in China*, Vol. 29, No. 4, pp. 452-458, 2015.

[3] P. R. Patil and V. K. Kshirsagar, "Efficient time compression earthquake database using hadoop Hive ORC format", in *International Conference on Intelligent Computing and Control Systems (ICICCS),* 2017, pp. 1361-1364.

[4] O. Omosebi, S. Sotiriadis, E. Asimakopoulou, N. Bessis, M. Trovati and R. Hill, "Designing a Subscription Service for Earthquake Big Data Analysis from Multiple Sources", in *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC),* 2015, pp. 601-604.

[5] E. Luksys, E. Asimakopoulou and N. Bessis, "Development of Tools for Data Analysis of Earthquakes", in *International Conference on Intelligent Networking and Collaborative Systems,* 2014, pp. 406-410.

[6] G. Gupta and I. Singh Gupta, "Earthquake Data Analysis and Visualization using Big Data Tool", in *International Conference on "Computing for Sustainable Global Development"*, 2016, pp. 618-621.

[7] G.Asencio–Cortés Morales–Esteban X.Shang and F.Martínez–Álvarez, "Earthquake prediction in California using regression algorithms and cloud-based big data infrastructure", *Computers and Geosciences*, Vol. 115, pp. 198-210, 2017.

[8] L.-H. Chen, H.-M. Hung, C.-Y. Chen, H.-K. Wu, Y. Takama and T. Yamaguchi, "3D Visualization of Earthquake Big Data", in *Conference on Technologies and Applications of Artificial Intelligence (TAAI),* 2017, pp. 72-75.

[9] A. R. Barakbah, T. Harsono, A. Sudarsono, R. A. Aliefyan, “Big Data Analysis for Spatio-Temporal Earthquake Risk-Mapping System in Indonesia with Automatic Clustering”, in *The 2017 International Conference on Big Data Research,* 2017, pp. 33-37.

[10] M. Muller, *Information Retrieval for Music and Motion*, Bonn, Germany: Springer, 2007, Vol. XVI, pp. 69-74.

[11] J.-S. R. Jang, Dynamic Time Warping, National Taiwan University Department of Computer Science Multimedia Information Retrieval, [Online]. Available: http://mirlab.org/jang/books/dcpr/dpDtw.asp?title=8-4%20Dynamic%20Time%20Warping, [Accessed 19 February 2019].

[12] A. E. Suliswati, A. R. Barakbah, T. Harsono and Y. Setyowati, “Earthquake density measurement using Automatic Clustering”, in *Knowledge Creation and Intelligent Computing (KCIC),* 2014, pp. 102-110.

[13] A. R. Barakbah, T. Harsono, A. Sudarsono, Pemanfaatan Klasterisasi Otomatis untuk Analisis Gempa, Surabaya: Revka Prima Media, 2019.

[14] R. Edelani, A. R. Barakbah, T. Harsono, A. Sudarsono, “Association analysis of earthquake distribution in Indonesia for spatial risk mapping”, in *The International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC),* 2017, pp. 231-238.

[15] W. Randolph Franklin, “PNPOLY - Point Inclusion in Polygon”, Rensselaer Polytechnic Institute (RPI), [Online]. Available: https://wrf.ecse.rpi.edu//Research/Short_Notes/pnpoly.html, [Accessed 14 April 2019].

[16] Apache, "Apache Superset (incubating)", Apache, [Online]. Available: https://superset.incubator.apache.org/. [Accessed 30 June 2019].