**Design and implementation of graft database**

**on MSME information system using neo4j**

**Muhammad Sholeh1, Rr. Yuliana Rachmawati 2, Erma Susanti3**

Department of Informatics, Faculty of Industrial Technology,

Institut Sains & Teknologi AKPRIND Yogyakarta, Yogyakarta, Indonesia

E-mail: 1 muhash@akprind.ac.id,  2 yuliana@akprind.ac.id, 3erma@akprind.ac.id

**Abstract.** This study aims to implement a database of information systems for small and medium enterprises (MSME) using database graphs. In the conducted application development, the used database mostly uses the relational database. With the development of large amounts of data, the tendency of ever increasing data volume has an impact on the increasingly large table size and the number of merging data on the query (JOIN) has an impact on the time of accessing the query. This data speed is a problem in the relational database, and one solution is to use the noSQL database. In this research, the implementation of noSQL used a graph database and used the Neo4j application. The research methodology and database development carried out in this study is by collecting and analyzing, conceptual, logical, physical level databases on data related to the needs of MSME information systems such as MSME locations, MSME actors and MSME products. The data is then carried out conceptual level database design and proceed with logical level database design. This research resulted in a graft database that stored the MSME data. The results of the node from the graph database implementation were the nodes of city, sub-district MSME, and product and connector of the sub-district\_entrance node which connecting the MSME location (sub-village) with the sub-district, the node connector of industrial\_typed\_entrance which connecting the MSME actors with the industry types

1. **Introduction**

As one of the economic actors, MSMEs must improve and be ready to face this era of disruption. The development of Information and Communication Technology (ICT) provides new opportunities that can overcome some MSME problems. Although it is not a major problem, ICT opportunities are enormous; and based on the conditions in the field, it shows that the adoption of ICT by the MSME sector is still not optimal compared to large companies. The utilization of ICT can also be used to carry out promotions, to introduce the MSME locations/centres in a location, as well as other benefits such as dialogue, discussion, and online consultation with consumers online, so that consumers can be proactively and interactively involved in the products designing, developing, marketing, and sales. The above efforts can be done by building an MSME information system application. One information system that can be used is the website-based application that is on [www.umkmbantul.com](http://www.umkmbantul.com).

In developing information systems applications, databases play a very important role. In this database, all data needed in the information system is processed. Understanding the Database according to Sumathi.[1], A database is a well-organized collection of data that are related in a meaningful way, which can be accessed in different logical orders. Database systems are systems in which the interpretation and storage of information are of primary importance. The database should contain all the data needed by the organization as a result, a huge volume of data, the need for long-term storage of the data, and access of the data by a large number of users generally characterize database systems.

Another understanding of the database according to Singh [2], a database is defined as a collection of logically related data sore together that designed to meet the information needs of an organization. It is basically an electronic, filling cabinet, which contain computerized data files. It can contain one data file (a very small database) or large number of data files (a large database) depending on organization needs. A database is organized in such a way that a computer program can quickly select desired pieces of data.

In developing the [www.umkmbantul.com](http://www.umkmbantul.com) website application, the database is built using the relational database and using MYSQL. This MSME database will grow along with the growth of the processed data. Thus, the database design must be considered in the application development, in terms of the processes of storage and data retrieval. Along with the data growth, the use of relational database has been unable to meet the development of the very large data. One alternative is to use the noSQL database.

According to Sasaki [3], the development of noSQL includes the growing volume of data resulting in an increasingly heavy relational database. The tendency of the ever-increasing volume of data has an impact on the increasingly large table size, and the number of merging data on the query (JOIN) has an impact on the time of accessing the query. The data speed relates to the data changes speed and the data model changes. This data speed is a problem in relational databases because relational databases are not designed to handle data changes too often. Data diversity is based on the fact that data can be incomplete or rare, connected or not, and structured or unstructured. This diversity is not anticipated in the development of relational databases.

One type of database included in noSQL is the graph database. According to Kadir [4], graph database is a type of database that uses graph structure to store data. This database is intended to overcome the weakness of relational database, especially to avoid time-consuming merging operation. The data storage model uses the principle of Graph Theory, where the data entry is represented as Node, and also as Edge that connects each Node. Implementation of this graph database can use Cayle, Filament, GraphDB, and other applications. One of the applications that can be used to create the graph database

is Neo4.

Neo4j is a database that can be used to create graph database made by the Neo4j company. Neo4j is the database that meets the requirements of atomicity, consistency, isolation, durability (ACID).. Some research exploring the graft database using Neo4j include [5], [6],[6], research comparing graft database with relational base [6], [7]. [8], [9]

The results of research and experiments that have been carried out in this study produce a data query process that reads data in the noSQL database. The resulting response shows the results of the response time is very fast compared to the query for the make-update-delete process. Data update requests in the noSQL database have the longest response time when compared to processing data requests in the process of reading and deleting data. . Whereas the query process for deleting data in the noSQL database has a faster response time than the data creation query. The use of the noSQL database is one way that can be used in handling storage process issues. The database that is stored in the document is the NoSQL database type in use today. This study aims to measure the response time of queries conducted on the noSQL database stored in the document. The use of this nosql database can overcome the rapid development of the internet and cloud computing which certainly encourages the availability of databases to be able to store and process large data effectively, as well as demanding high performance when reading and writing.[10].

The research conducted by Percuku [11], explained that one of the weaknesses of the database is in terms of response time that can be quite long and has an impact on performance especially if applied to very large data and another weakness is the difficulty of the database to develop according to business needs. To overcome this shortcoming, the database implementation that store large data can utilize new technologies such as NoSQL data storage. The research that has been carried out aims and tries to improve the process by modelling and processing data using the Neo4j database.

The research conducted by Hong [12], Explores the implementation of the graft database with Neo4j about the film star database. By focusing on film data, Neo4j-based analysis is conducted in this paper. Firstly, Neo4j and Cypher Query Language are introduced. Then Neo4j is applied to analyses the associations among key objects in film data which are directors, actors etc. Neo4j database is good at dealing with complex and multi-connection data, using Neo4j database to store and manage film data makes it convenient for film data analysis.

Taking into account the development of large data and the inconsistency of a data, of course, will raise problems in the process of data storage. The use of relational databases is not suitable for unstructured data processing. The use of social media such as Twitter requires data storage that can store large data and does not contain data stored. One solution to the problem of unstructured data is to use the nosql database and one of them is a graphical database with neo4j. In research conducted by developing a mini blog prototype using python applications. [13]

1. **Research Method**

The purpose of database design is to meet the information that contains the needs of users. The database is a collection of data that are interconnected with one another, and certain software are used to manipulate it. The database is one important component in the information system because it contains data to provide information for its users. Figure 1, the research stages, especially in making the conducted database.



Figure 1 Stages of Making the Database

In this research the method used in database design includes four stages, namely:

1. Data Collecting and Analysis

Data collecting and analysis aims to collect and analyse the data or information which are needed in the system.

1. Design of Conceptual-Level Database

Design of Conceptual-Level Database aims to check the users’ needs, the limitations, and the relationships.

1. Design of Logical-Level Database

Design of Logical-Level Database aims to map the conceptual design into the database model that will be used.

1. Design of Physical-Level Database

Design of Physical-Level Database aims to implement the results of the conceptual-level design and the logical-level design to get the database design that will be used.

1. **Discussion and Results** 
   1. **Relational Database Implementation**

On the relational basis, the database implementation process was carried out by conducting the normalization process. The database normalization aimed to eliminate and reduce the data redundancy, ensuring the data dependencies as well as ensuring the data was stored in the right table. In the normalization process, all designed tables were analysed, whether the tables still had deviations or not, especially in the processes of adding data, deleting data, and repairing data. The design result from the MSME database is on the Figure 2.

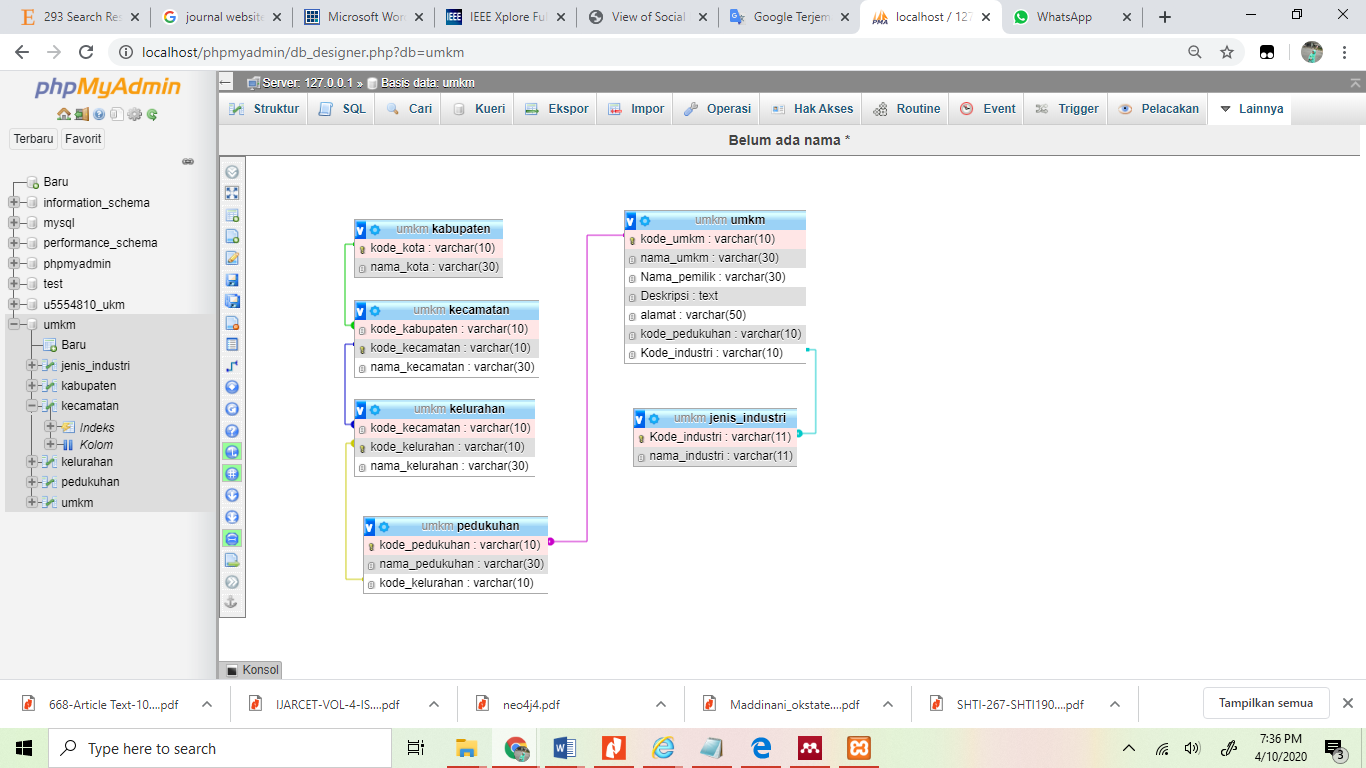


Figure 2 Data Relation in Relational Database

The relationship between tables is the main key in the database implementation. The information needed in the process of displaying data is the result of queries from interrelated tables. Figure 2 is the result of relations between tables in the database. The relation will relate all the existing tables. Relationships are carried out by relating the table of regency\_address (alamat\_kabupaten) , table of district\_address (kecamatan), table of sub-district\_address (kelurahan), table of MSME (UMKM) , and table of industry\_type (jenis\_industri).

The results of the database design are implemented using MySQL, a number of table creation processes include:

* The making of the table of regency:

CREATE TABLE `alamat\_kab` (

`kab\_id` int(11) NOT NULL AUTO\_INCREMENT,

`kab\_nama` varchar(100) DEFAULT '',

PRIMARY KEY (`kab \_id`)

) ENGINE=MyISAM AUTO\_INCREMENT=10 DEFAULT CHARSET=latin1 ROW\_FORMAT=DYNAMIC;

* The making of the table of MSME

CREATE TABLE `alamat\_ukm` (

`alamat\_umkm\_id` int(11) NOT NULL AUTO\_INCREMENT,

`umkm\_id` int(11) NOT NULL,

`kab\_id` int(11) NOT NULL,

`kec\_id` int(11) NOT NULL,

`kel\_id` int(11) NOT NULL,

`dsn\_id` int(11) NOT NULL,

`rw\_umkm\_id` int(11) DEFAULT NULL,

`rt\_umkm\_id` int(11) DEFAULT NULL,

`alamat\_umkm\_detail` varchar(200) NOT NULL,

`jenis\_umkm` varchar(10) DEFAULT 'cabang',

PRIMARY KEY (`alamat\_umkm\_id`) USING BTREE

) ENGINE=MyISAM AUTO\_INCREMENT=15 DEFAULT CHARSET=latin1;

* 1. **Graph Database Implementation**

The first step in implementing a graph database is to form nodes. Nodes that will be formed are the nodes that contain cities, districts and sub-districts, MSME and industry\_type. Figure 3 is the design of the making and accessing of graph database. This design shows the MSME actors as well as the types of industry and locations of the MSME actors.



Figure 3 Design of Graph and Relationship between MSME and Regional Positions

The design of the graph database Figure 3 will be made of nodes containing Cities, Sub-districts, MSME and industry\_type data, and 3 connectors, namely SUB-DISTRICT\_ENTRANCE, DISTRICT\_ENTRANCE, and INDUSTRY\_TYPE\_ENTRANCE

* 1. **The Making of the Node of City**

From the picture graph design ... nodes are made. The nodes made first are the nodes of city, the nodes of sub-district

$ create (:kota {id:1, nama:'Bantul'})

$ create (:kota {id:1, nama:'Yogyakarta'})

$ create (:kota {id:1, nama:'Sleman'})

$ create (:kota {id:1, nama:'Kulon Progo'})

$ create (:kota {id:1, nama:'Gunung Kidul'})

The results of the query process are shown in Figure 4.

$ MATCH (n:kota) RETURN n LIMIT 25

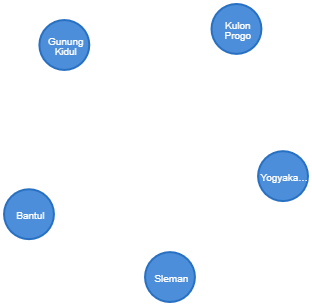


Figure 4 Nodes of city which are formed

* 1. **The Making of the Node of Sub-district**

$ create (:kelurahan {id:'kl1', nama:'Tamantirto})

$ create (:kelurahan {id:'kl2', nama:'Bangunjiwo'})

$ create (:kelurahan {id:'kl3', nama:'Ngestiharjo'})

$ create (:kelurahan {id:'kl4', nama:’Tirtonirmolo '})

$ create (:kelurahan {id:'kl5', nama:'Pendowoharjo'})

$ create (:kelurahan {id:'kl7', nama:'Bangunharjo'})

$ create (:kelurahan {id:'kl7', nama:'Panggungharjo'})

$ create (:kelurahan {id:'kl6', nama:'Timbulharjo'})

The result of the query process are shown in Figure 5.

$ MATCH (n:kelurahan) RETURN n LIMIT 25

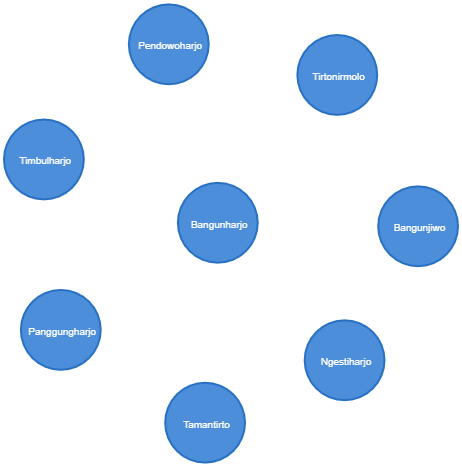


Figure 5 Nodes of sub-district which are formed

* 1. **Making the Relationship of Kasihan District**

Making a relationship of a sub-district, including the districts within a city, for example, making the relationship of the Sub-districts of Tamantirto, Ngestiharjo, Tirtonirmolo, and Bangunjiwo, all of them are within the Kasihan District which is within the Bantul Regency.

$ MATCH(k:kota), (c:kelurahan)

where k.nama='Bantul' AND c.nama='Tamantirto'

create (c) -[:MASUK\_KECAMATAN{kecamatan: ['Kasihan']}]->(k)

$ MATCH(k:kota), (c:kelurahan)

where k.nama='Bantul' AND c.nama='Tirtonirmolo'

create (c) -[:MASUK\_KECAMATAN{kecamatan: ['Kasihan']}]->(k)

$ MATCH(k:kota), (c:kelurahan)

where k.nama='Bantul' AND c.nama='Ngestiharjo'

create (c) -[:MASUK\_KECAMATAN{kecamatan: ['Kasihan']}]->(k)

$ MATCH(k:kota), (c:kelurahan)

where k.nama='Bantul' AND c.nama='Bangunjiwo'

create (c) -[:MASUK\_KECAMATAN{kecamatan: ['Kasihan']}]->(k)

The results of the query process are shown in Figure 6.

$ MATCH p=()-[r:MASUK\_KECAMATAN]->() RETURN p LIMIT 25

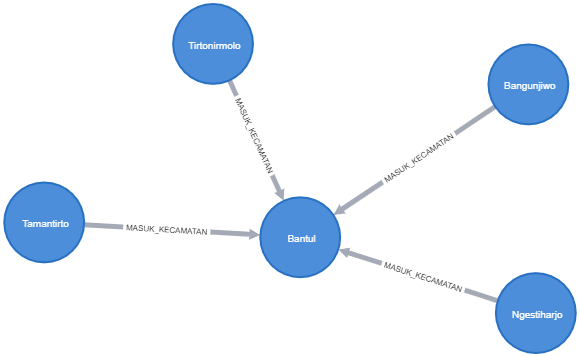


Figure 6 Connecting among the nodes of sub-district and the nodes of city

* 1. **Making the Node of MSME**

This node is used to store the data related to the data of MSME, and the process of the node is:

$ create (:umkm { pemilik:'Agung', nama\_ukm:'CV. AKSIS JOGJA', deskripsi:'AKSIS JAYA Bengkel adalah bisnis di bidang Bengkel Mobil. Alamat Jl Sukolilo 263 '})

$ create (:umkm { pemilik:'Kirana', nama\_ukm:'Jahit Baju', deskripsi:'menerima jahitan khusus wanita'})

$ create (:umkm { pemilik:'Bambang', nama\_ukm:'CV. SERBA MAKANAN', deskripsi:'Kepuasan pelanggan merupakan prioritas’})

$ create (:umkm { pemilik:'Partono', nama\_ukm:'CV MODIV YOGYA', deskripsi:' bengkel mobil'})

The results of the query process are shown in Figure 7.

$ MATCH (n:umkm) RETURN n LIMIT 25

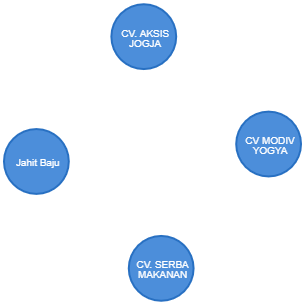


Figure 7 Nodes of MSME which are formed

* 1. **Making the Node of Industry Type**

This node is used to store the data related to the industry types of the MSME actors, and the process of the node is:

$ create (:Jenis\_industri {id:'JI1',nama:'Fashion’})

$ create (:Jenis\_industri {id:'JI2',nama:'Otomotif’})

$ create (:Jenis\_industri {id:'JI3',nama:'Kuliner’})

$ create (:Jenis\_industri {id:'JI4',nama:'Pendidikn'})

The results of the query process are shown in Figure 8.

$ MATCH (n:Jenis\_industri) RETURN n LIMIT 25

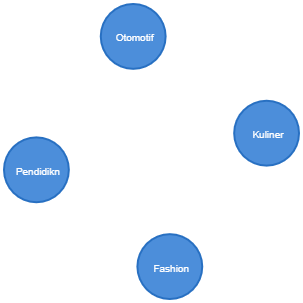


Figure 8 Nodes of Industry Type which are formed

* 1. **Making the Relationship between MSME and Industry Types**

The relationship between MSME and the types of industry aims to determine the types of business of the MSME actors. For example, the making of the relationship between an MSME, the *CV. AKSIS JOGJA*, which has a car repair shop business, and is included in the automotive industry.

$ MATCH(u:umkm), (j:Jenis\_industri)

where u.nama\_ukm='CV. AKSIS JOGJA' AND j.nama='Otomotif'

create (u) -[:MASUK\_JENIS\_INDUSTRI{jenis:['bengkel mobil']}]->(j)

The results of the query process are shown in Figure 9.

$ MATCH p=()-[r:MASUK\_JENIS\_INDUSTRI]->() RETURN p LIMIT 25

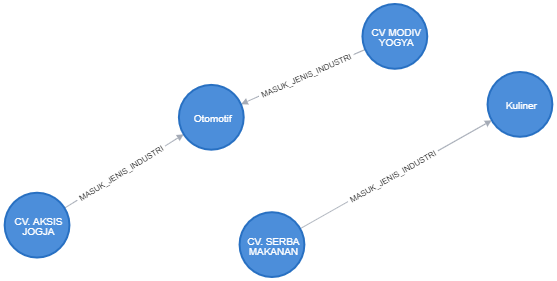


Figure 9 Connecting between Nodes of MSME and Node of Industry Types

* 1. **Making the Relationship between MSME and the Locations of Sub-district Region**

Another relationship which is made is the locations of sub-village of the MSME actors in which sub-districts they are located. For example, an MSME called the *CV. AKSIS JOGJA* is built, and located in Gentak Sub-village which is in Tamantirto Sub-district.

$ MATCH(u:umkm), (kl:kelurahan)

where u.nama\_ukm='CV. AKSIS JOGJA' AND kl.nama='Tamantirto'

create (u) -[:MASUK\_KELURAHAN{kelurahan: ['Jetis']}]->(kl)

The results of the query process are shown in Figure 10.

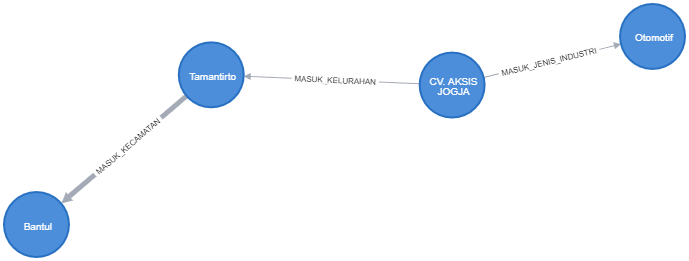


Figure 10 –Connecting between Nodes of MSME and Nodes of Sub-district

From the process of making nodes and links/connectors, all nodes in the management of MSME data are interrelated, and the relationships between nodes have been seen. Several tests that can be done include:

* Looking at the overall linkages of existed nodes

$ match (k:kota) -[c:MASUK\_KECAMATAN] - (l:kelurahan) -[m:MASUK\_KELURAHAN] - (u:umkm)-[j:MASUK\_JENIS\_INDUSTRI]- (ji:Jenis\_industri)

return k,c,l,m,u,j,ji

The results of the query process are shown in Figure 11.

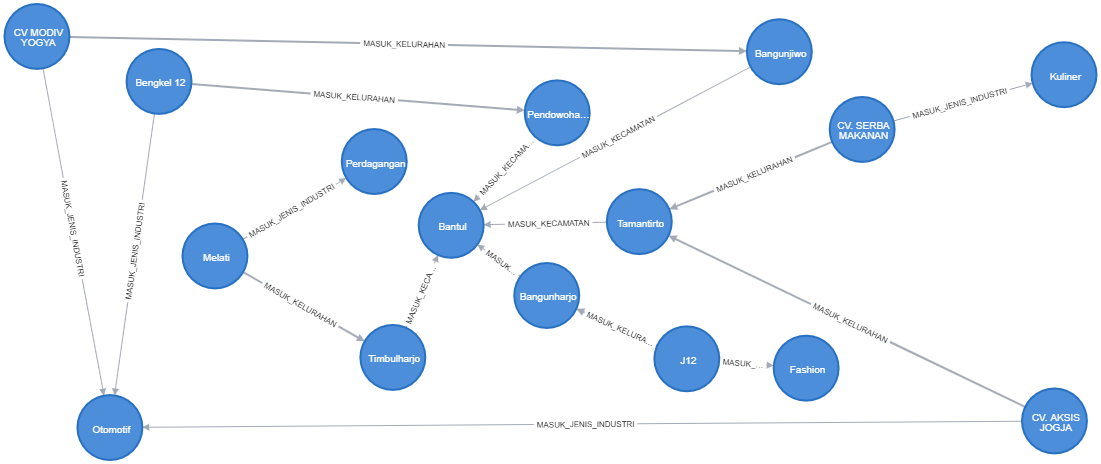


Figure 11 Connecting between all nodes

* Looking at the overall nodes linkages in the text form
* $ match (k:kota) -[c:MASUK\_KECAMATAN] - (l:kelurahan) -

[m:MASUK\_KELURAHAN] - (u:umkm)-[j:MASUK\_JENIS\_INDUSTRI]-

(ji:Jenis\_industri)

Return k.nama,c.kecamatan,l.nama,m.kelurahan,u.nama\_ukm,j.jenis,ji.nama

The results of the query process, besides producing nodes in the form of graphs, can also be displayed in the form of texts, and can be exported to the forms of CSV and JSON. Figure 12 displays the query results in the text form, Figure 13 is the export to the CSV form, and Figure 14 is the export to the JSON form.

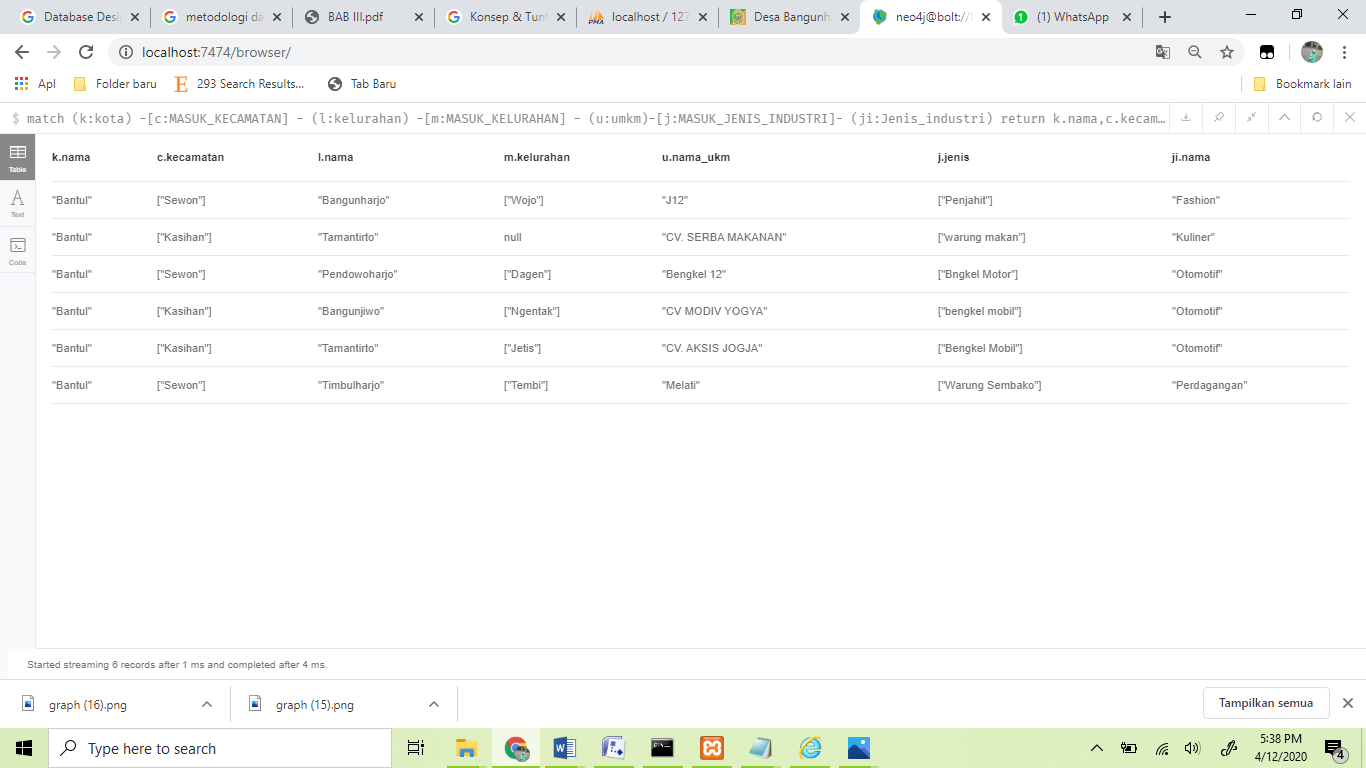


Figure 12 Query results in text form

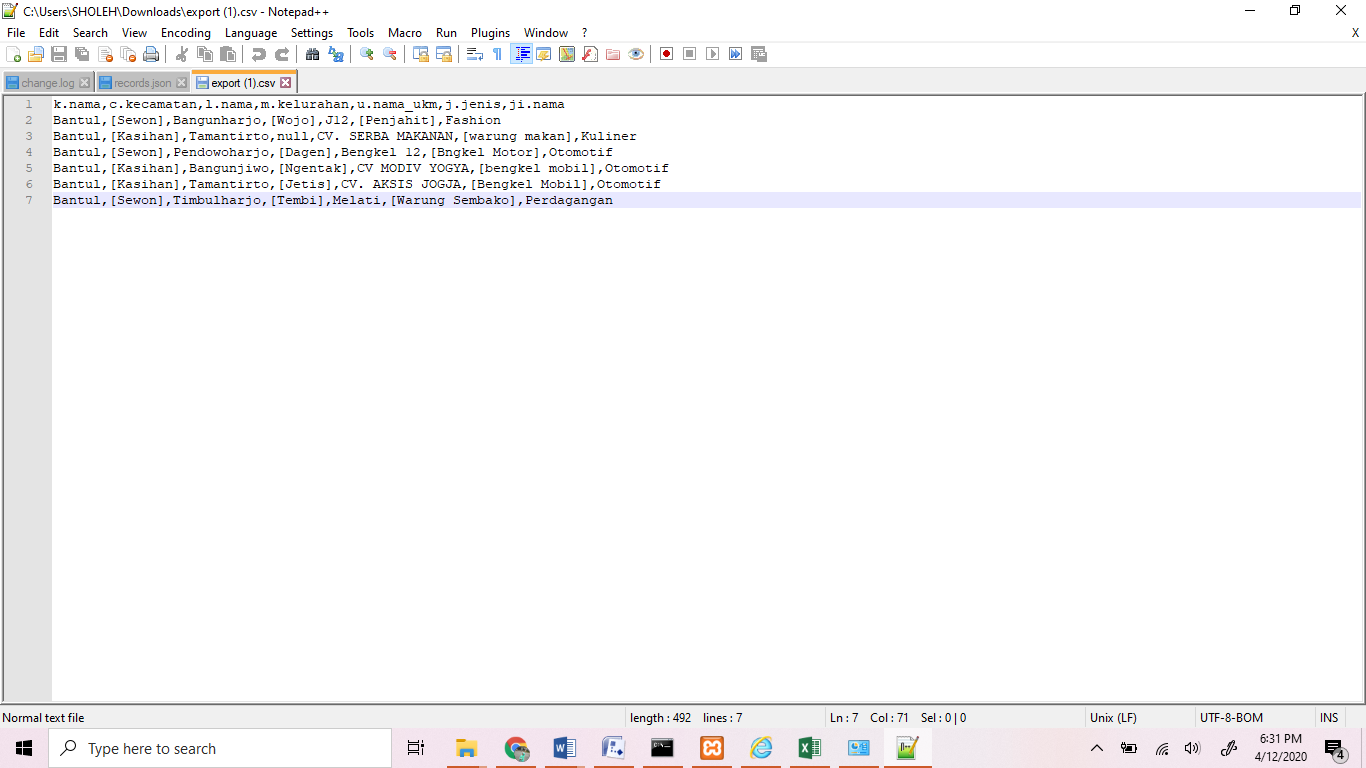


Figure 13 Export results to CSV form

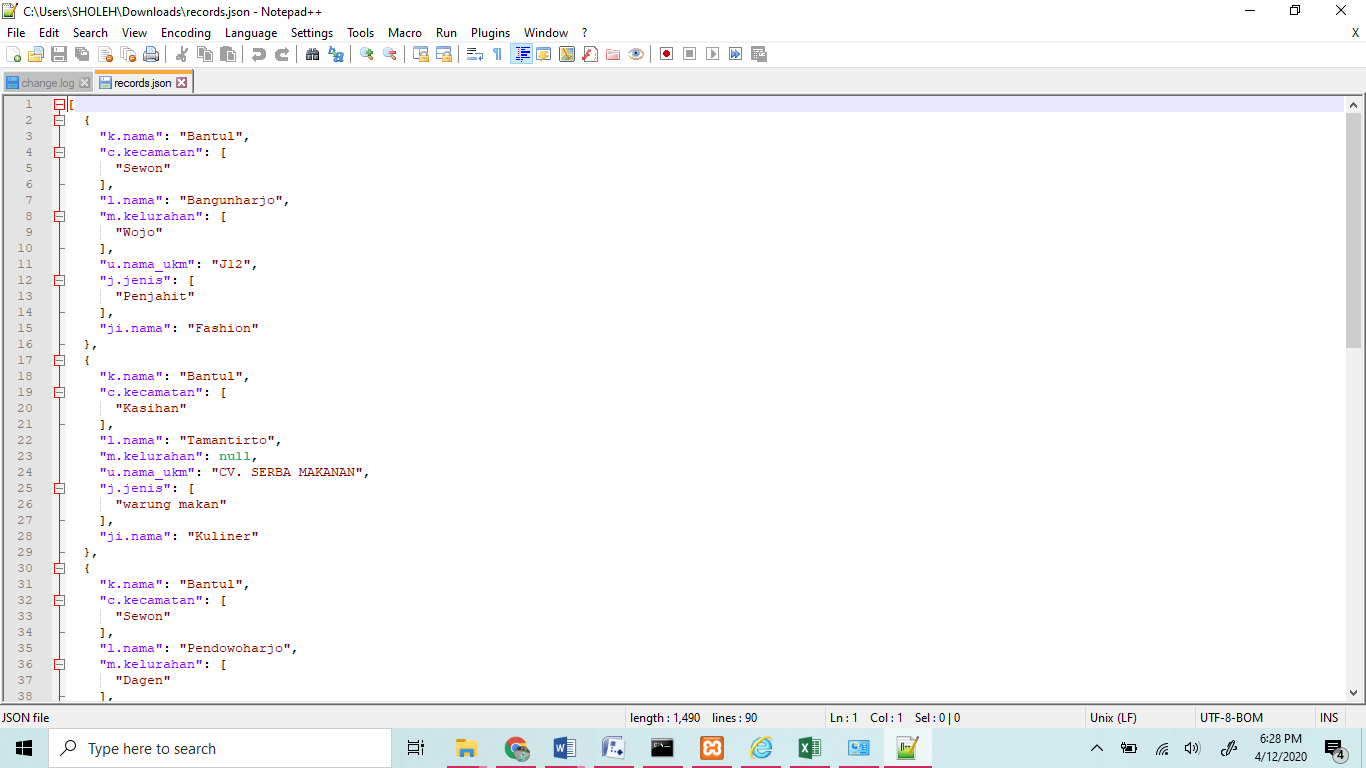


Figure 14 Export results to JSON form

* Looking at the MSME actors in certain sub-districts, for example, the MSME actors in Tamantirto Sub-district.

$ match (k:kota) -[c:MASUK\_KECAMATAN] - (l:kelurahan) –

[m:MASUK\_KELURAHAN] - (u:umkm)-[j:MASUK\_JENIS\_INDUSTRI]-

(ji:Jenis\_industri)

where l.nama='Tamantirto'

return k.nama,c.kecamatan,l.nama,m.kelurahan,u.nama\_ukm,j.jenis,ji.nama

The results of the query process are shown in Figure 15.

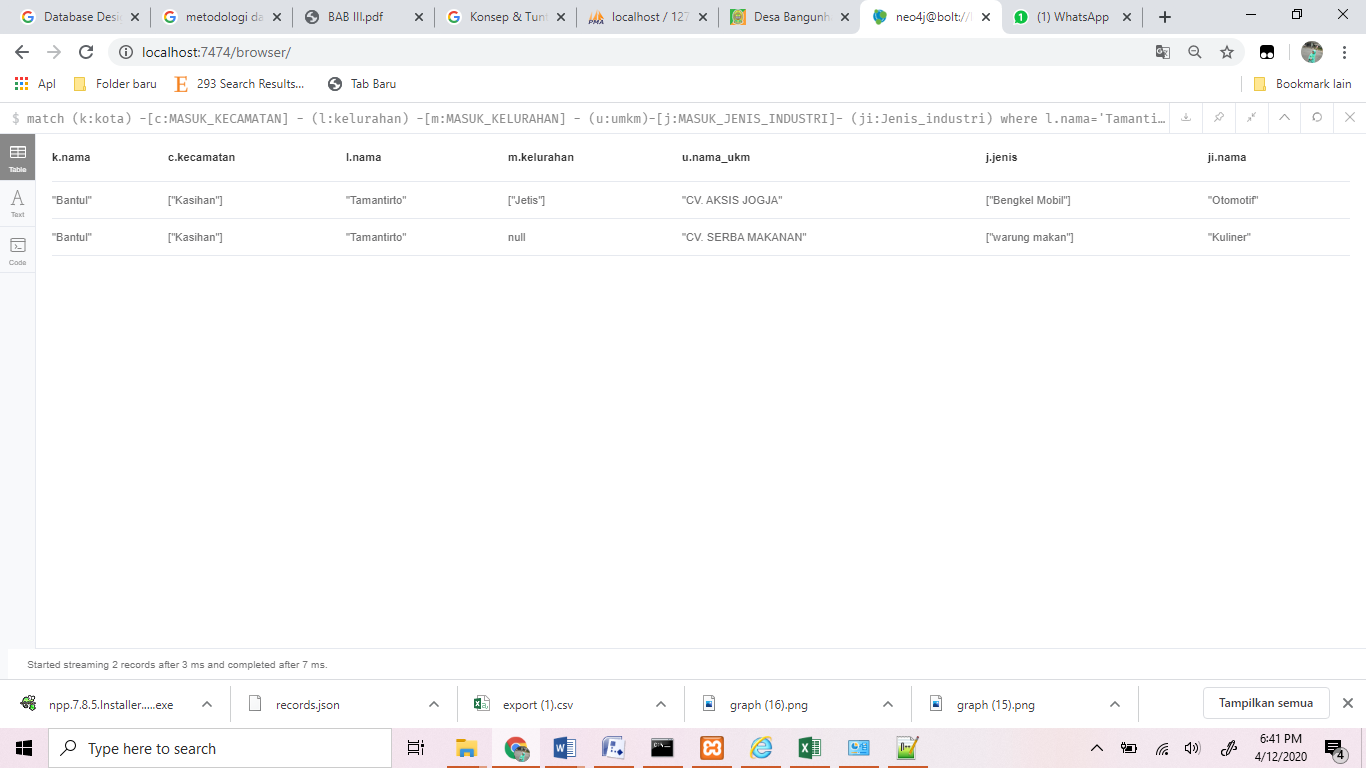


Figure 15 Query Results in Criterion of Tamantirto Sub-district

* Looking at the MSME actors in certain Districts, for example, the MSME actors in Kasihan District.

$ match (k:kota) -[c:MASUK\_KECAMATAN] - (l:kelurahan) –

[m:MASUK\_KELURAHAN] - (u:umkm)-[j:MASUK\_JENIS\_INDUSTRI]-

(ji:Jenis\_industri)

where c.kecamatan=["Kasihan"]

return k.nama,c.kecamatan,l.nama,m.kelurahan,u.nama\_ukm,j.jenis,ji.nama

The results of the query process are shown in Figure 16.

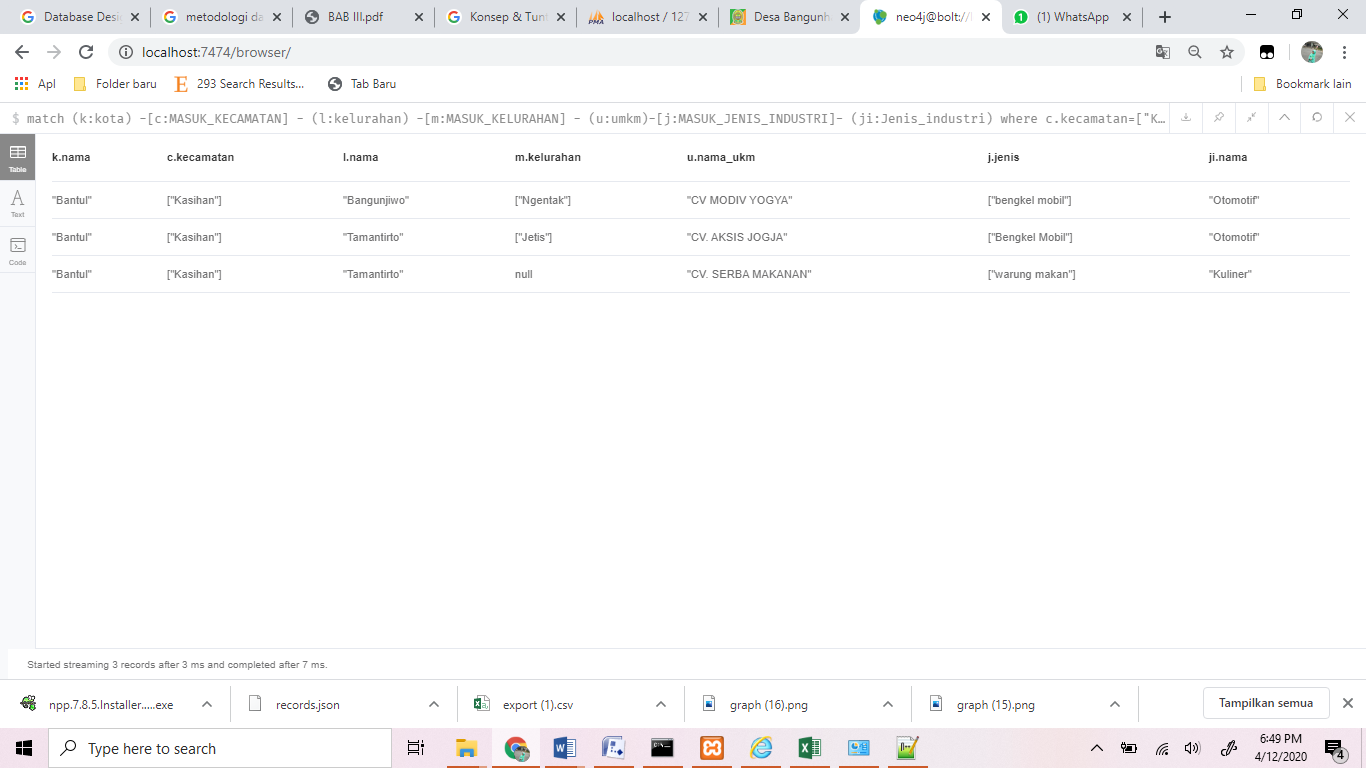


Figure 16 Query Results in Criterion of Kasihan District

* Looking at the MSME actors working in the field of automotive industry

$ match (k:kota) -[c:MASUK\_KECAMATAN] - (l:kelurahan) -

[m:MASUK\_KELURAHAN] - (u:umkm)-[j:MASUK\_JENIS\_INDUSTRI]-

(ji:Jenis\_industri)

where ji.nama='Otomotif'

return k.nama,c.kecamatan,l.nama,m.kelurahan,u.nama\_ukm,j.jenis,ji.nama

The results of the query process are shown in Figure 17.

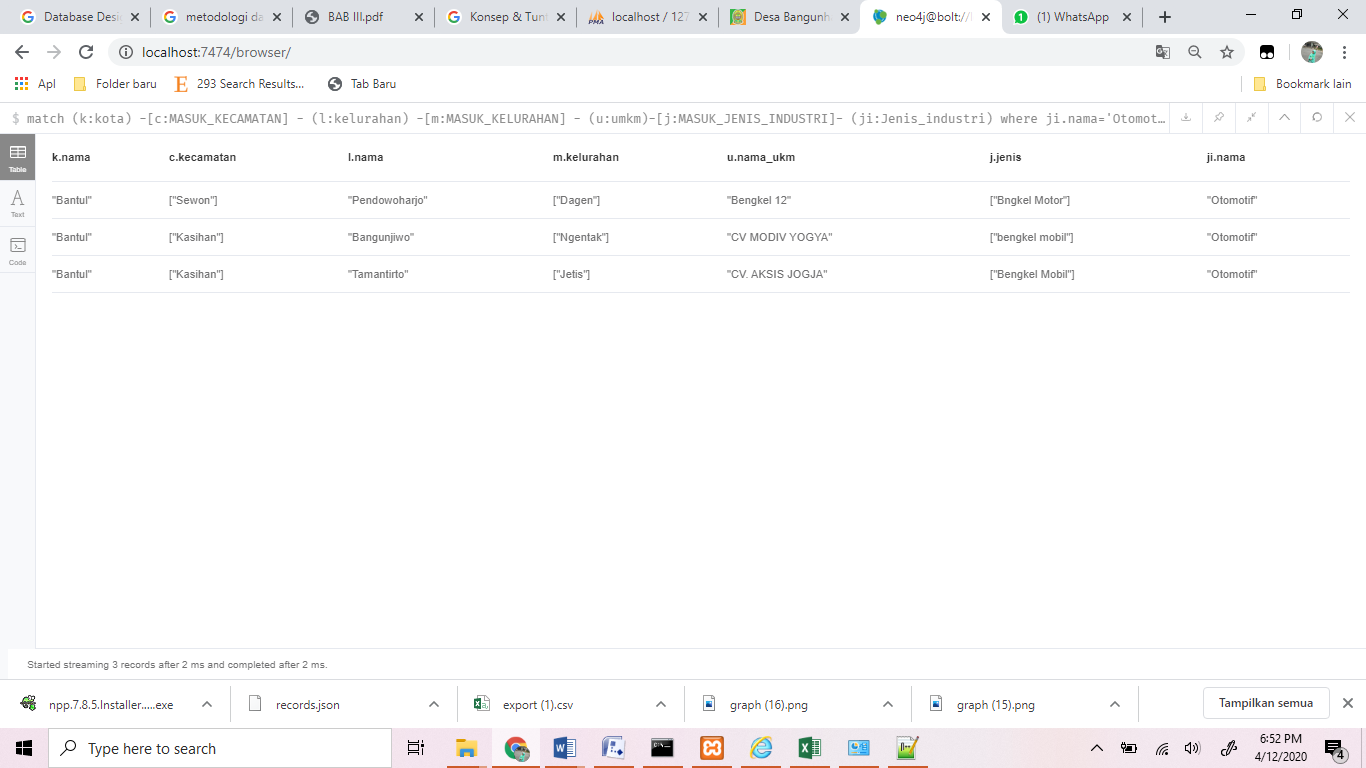


Figure 17 Query Results in Criterion of MSME Actors Working in the Field of Automotive Industry

1. **CONCLUSION**

The graph database can be an alternative to overcome the weaknesses of the relational database. The large data growth and the access speed require a reliable database. In this research, the data processed for the MSME information is implemented in the form of graph database. The results of the database can display various information needed in the data processing. The nodes which are made still contain the MSME data which can be needed in the MSME information system. Nodes that are made up consist of nodes containing location data and nodes containing MSME data.. All nodes that are formed can be interconnected by the relationship made between the nodes. The process of displaying the data, both in the form of graphs and text, can be displayed with the query syntax.

**Acknowledgment**

The author would like to thank the Directorate of Research and Community Service, the Directorate General of Research and Development for providing funding support for research activities in the scheme of superior applied research in universities in the 2020 and thank to research team.

**References**

[1] S. E. S. Sumathi, *Fundamentals ofRelational Database Management Systems*. Springer-Verlag Berlin Heidelberg, 2007.

[2] S. K. Singh, *Database Systems Consepts, Design and Applicatiom*. New Delhi: Dorling Kindersley, 2009.

[3] B. M. Sasaki, J. Chao, and R. Howard, “Graph Databases For Beginners.”

[4] A. Kadir, *Dasar Basis Data Graft & NEo4j*. Yogyakarta: Penerbit Andi, 2020.

[5] S. Patil, G. Vaswani, and A. Bhatia, “Graph Databases- An Overview,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 1, pp. 657–660, 2014.

[6] S. A. T. Mpinda, L. G. Maschietto, M. T. S. Prado, and M. X. Ribeiro, “Graph database application using Neo4j: Railroad planner simulation,” *ICEIS 2015 - 17th Int. Conf. Enterp. Inf. Syst. Proc.*, vol. 1, no. 04, pp. 399–403, 2015, doi: 10.5220/0005469003990403.

[7] H. R. Vyawahare, P. P. Karde, and V. M. Thakare, “An efficient graph database model,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 88, no. 10, pp. 1292–1295, 2019, doi: 10.35940/ijitee.I7805.0881019.

[8] S. Medhi and H. Baruah, “Relational database and graph database: A comparative analysis,” *J. Process Manag. New Technol.*, vol. 5, no. 2, pp. 1–9, 2017, doi: 10.5937/jouproman5-13553.

[9] H. R. Vyawahare, P. P. Karde, and V. M. Thakare, “Improving performance using relational and graph database,” *Int. J. Recent Technol. Eng.*, vol. 8, no. 2, pp. 1722–1726, 2019, doi: 10.35940/ijrte.B1862.078219.

[10] R. Gunawan, “Seri Sains dan Teknologi Pengukuran Query Respon Time pada Nosql Database Berbasis Document Stored P-ISSN 2477-3891 E-ISSN 2615-4765,” *J. Siliwangi*, vol. 4, no. 2, pp. 100–103, 2018.

[11] A. Perçuku, D. Minkovska, and L. Stoyanova, “Modeling and Processing Big Data of Power Transmission Grid Substation Using Neo4j,” *Procedia Comput. Sci.*, vol. 113, pp. 9–16, 2017, doi: 10.1016/j.procs.2017.08.276.

[12] H. Lu, Z. Hong, and M. Shi, “Analysis of film data based on Neo4j,” *Proc. - 16th IEEE/ACIS Int. Conf. Comput. Inf. Sci. ICIS 2017*, pp. 675–677, 2017, doi: 10.1109/ICIS.2017.7960078.

[13] E. Zhang, J. Fiaidhi, S. Mohammed, O. Rd, T. Bay, and O. N. Pb, “Social Recommendation using Graph Database Neo4j ： Mini Blog , Twitter Social Network Graph Case Study,” *Int. J. Futur. Gener. Commun. Netw.*, vol. 10, no. 2, pp. 9–20, 2017.