



PENERAPAN *GRAPH DATABASE* UNTUK SISTEM REKOMENDASI KOLEKSI PERPUSTAKAAN DI ERA *BIG DATA*

Ahmad Fiqi Efendi¹; Tedy Setiadi^{1*}

¹Teknik Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan

*Korespondensi: tedy.setiadi@tif.uad.ac.id.

Disubmit : 26-03-2019
Direview : 05-04-2019
Direvisi : 20-08-2019
Diterima : 01-09-2019

ABSTRACT

Development in the library utilizes information systems in providing easy access to digital library collections. The purpose of digital library is to fulfill the need of the library users with fast and accurate data from library collections database. Digital library application used by Libraries in Higher Education is using traditional database called relational databases. Relational database have some weakness there are fix structure so it will be hard to change the structure when there volume is increasing and data variable are changing. Other than that it will need other operation called JOIN when we want to combine two variable data such ad collections and user needs. The impact is when using JOIN is the database become so slowly. This paper propose the use of Graph database as solution for those problems. Interviews and observations were used to collect data and determine input, process and output needs. Implementation is done by designing a database graph that fits the needs of the library, then executes the query in accordance with the design. In library information systems, the relationship that exists between collections and library users is crucial. Graph data base is the right alternative to describe relationships (networks) that occur between collections and users. By utilizing the relationships that are intertwined between the data, the library collection recommendation system can be explained based on these relationships.

ABSTRAK

Perpustakaan memanfaatkan sistem informasi untuk memberikan kemudahan akses koleksi perpustakaan digital. Tujuan perpustakaan digital adalah mendapatkan data yang cepat, tepat dan akurat dari database yang menyimpan koleksinya. Aplikasi perpustakaan digital di Perguruan Tinggi saat ini masih menggunakan database tradisional yaitu *relational database*. *Rasional database* ini memiliki beberapa kekurangan yaitu data yang tersimpan secara terstruktur dan tetap sehingga kesulitan manakala menangani perubahan struktur data disebabkan perkembangan *volume* dan variasi data. Kelemahan lainnya yaitu kesulitan untuk menangani data relasi variatif yang muncul antara koleksi dan pemustaka sehingga membutuhkan operasi JOIN yang rumit berakibat lambatnya operasi *database* tersebut. Dalam makalah ini akan dijelaskan penerapan *graph database* sebagai solusi yang ditawarkan untuk permasalahan yang terjadi dalam perpustakaan digital. Pada penelitian ini wawancara dan observasi digunakan untuk mengumpulkan data serta menentukan kebutuhan *input*, *proses* dan *output*. Implementasi dilakukan dengan merancang *graph database* yang sesuai dengan kebutuhan perpustakaan, kemudian eksekusi *query* sesuai dengan rancangan tersebut. Pada sistem informasi perpustakaan, relasi yang terjalin antara data koleksi dan data pemustaka sangat diperlukan. *Graph database* adalah alternatif yang tepat untuk menggambarkan relasi (jaringan) yang terjadi antara koleksi dan pemustaka. Dengan pemanfaatan relasi yang terjalin antar data maka sistem rekomendasi koleksi perpustakaan dapat dipaparkan dengan berdasarkan relasi tersebut.

Keywords: *Graph database; Library; Big data; Recommendation system.*

1. PENDAHULUAN

Perpustakaan merupakan penyedia layanan berupa koleksi buku bacaan, pusat informasi serta tempat untuk membaca dan penyimpanannya. Dampak dari perkembangan teknologi informasi yang sangat pesat, koleksi yang dibutuhkan tidak sekedar buku cetak, kebutuhan akan bahan bacaan merambah ke bentuk *e-book*, respon dari perpustakaan yaitu dengan memberikan layanan *Digital Library* (Digilib) yang dapat diakses dengan mudah melalui jaringan internet (Feret & Marcinek, 2005). “Digilib adalah suatu koleksi informasi yang dikelola berikut pelayanannya, di mana informasi disimpan dalam format digital dan dapat diakses melalui jaringan” maka Digilib

harus menyediakan serta mengelola informasi tersebut dengan sangat baik (Sarlina & Setiadi, 2017).

Untuk menyediakan serta mengelola informasi tersebut maka Digilib harus memiliki *database* yang kuat yang mampu menangani data yang besar serta dinamis. Digilib dalam implementasinya masih menggunakan RDBMS atau *database* relasional. Permasalahan yang ada saat ini adalah *database* yang digunakan kurang fleksibel dalam memenuhi kebutuhan data yang cepat berubah dan tidak terstruktur. Di samping itu, aksesnya relatif lambat karena data yang tersimpan berbentuk tabel-tabel yang sulit untuk menggambarkan relasi data sehingga memerlukan operasi JOIN untuk memanggil data yang tersimpan tersebut. Data yang tersimpan tergolong lebih besar sehingga ruang penyimpanan yang dibutuhkan menjadi lebih besar (Setialana et al., 2017).

Dalam penelitian ini, graph database merupakan solusi yang diharapkan dapat menyelesaikan permasalahan yang ada karena data yang lebih fleksibel sehingga lebih memudahkan dalam akses edit serta hapus. Oleh sebab itu, database ini sangat cocok untuk menangani data yang berskala besar, ruang penyimpanan yang dibutuhkan juga tergolong lebih kecil sehingga dapat menyimpan data dengan lebih efisien. Graph database basis data yang berkembang di era big data di mana kebutuhan pentingnya menangani data berskala besar, terutama yang dipergunakan oleh para peneliti biologi jaringan dan situs jaringan sosial, seperti Facebook dan Twitter, graph database memetakan secara langsung objek ke aplikasi dengan lebih intuitif untuk menggambarkan data set asosiatif (Jo, 2019).

Dalam kasus relasi antara koleksi dengan pemustaka, bisa dianalisis untuk mencari koleksi mana saja yang sering dipinjam, siapa saja yang sering meminjam koleksi tertentu, dan bisa memberikan rekomendasi koleksi buku bacaan kepada pengguna berbasis koneksi yang terjalin antara penggunanya. Informasi ini penting bagi pustakawan untuk pengembangan koleksi di perpustakaan (Huang et al., 2002). Dengan menggunakan graph database diharapkan dapat membantu menyelesaikan permasalahan yang ada pada sistem perpustakaan karena sistem database ini menawarkan database yang fleksibel terhadap perubahan data di samping mampu memvisualisasikan data bertipe jaringan (relasi) (Needham & Hodler, 2019).

2. METODE PENELITIAN

Penelitian ini mengkaji tentang penerapan graph database di Perpustakaan Universitas Ahmad Dahlan untuk memberikan rekomendasi koleksi buku yang sesuai dengan minat dan jurusan bagi para pengunjung perpustakaan. Untuk mendapatkan data yang akurat dalam penelitian maka terdapat subjek penelitian sebagai berikut:

- a. Anggota Perpustakaan Universitas Ahmad Dahlan
- b. Koleksi Perpustakaan Digital Universitas Ahmad Dahlan
- c. Transaksi yang terjadi di Perpustakaan Universitas Ahmad Dahlan

Dalam melakukan penelitian ini, ada beberapa metode pengumpulan data yang digunakan yaitu wawancara dan observasi. Metode pengumpulan data dengan cara wawancara langsung dengan pihak terkait, yaitu pengurus perpustakaan Universitas Ahmad Dahlan, Anggota Perpustakaan Universitas Ahmad Dahlan terkait database relasional yang sudah digunakan dan Pengunjung Perpustakaan Universitas Ahmad Dahlan terkait sistem rekomendasi yang akan dikaji. Metode Observasi dilakukan dengan mengadakan pengamatan secara langsung kepada objek penelitian untuk mendapat data transaksi, pengunjung dan koleksi buku melalui simpus.uad.ac.id.

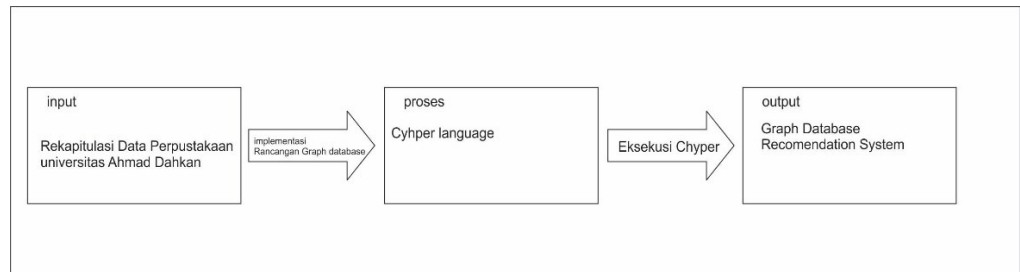
Setelah melakukan metode pengumpulan data yang diperlukan, langkah selanjutnya yaitu menganalisis kebutuhan yang diperlukan dari kebutuhan *input*, *proses*, dan *output* (Gambar 1).

- a. Kebutuhan *Input*
Kebutuhan *input* yaitu ekspor data dari *relational database* yang berupa *file spreadsheet* untuk dikonversi ke dalam sistem *graph database* pada aplikasi Neo4j.
- b. Kebutuhan *Process*

Data hasil ekspor dari *relational database* yang berupa file *spreadsheet* yang akan diubah menjadi *node* dan *edge* menggunakan formula dalam *spreadsheet* untuk dieksekusi di aplikasi *graph database*.

c. Kebutuhan *Output*

Kebutuhan *output* yaitu pemaparan data yang sudah tersimpan dalam *graph database* serta sistem rekomendasi yang merupakan hasil dari manipulasi data-data yang tersimpan dalam *graph database*.



Gambar 1. Kebutuhan Input Proses dan Output

Langkah berikutnya adalah implementasi *graph database*. Perpustakaan Digital Universitas Ahmad Dahlan atau digilib.uad.ac.id sudah memiliki *database* yang berupa *database* relasional untuk implementasi *graph database* pada sistem digilib.uad.ac.id ada beberapa tahapan yang perlu dilakukan:

1. Merancang *graph database* untuk perpustakaan digital.
2. Operasional *query* pada *graph database* dan eksekusi *query* pada *graph database*.
3. Implementasi sistem rekomendasi dengan menggunakan *people-to-people correlation*.
4. Perbandingan *query graph database* dan *relational database*.

3. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan rancangan *graph database* untuk perpustakaan digital beserta sistem rekomendasi menggunakan *People-to-People Correlation* sebagai acuannya. Berdasarkan data hasil observasi, wawancara, dan data yang diambil dari simpus.uad.ac.id diperoleh hasil berupa Rancangan Sistem *Graph Database* untuk perpustakaan, implementasinya, dan sistem rekomendasi.

Berdasarkan kebutuhan perpustakaan, data utamanya adalah koleksi dan pemustaka, sehingga *database* yang akan dibuat akan memiliki 2 *node* utama yaitu koleksi dan *user* (pemustaka).

a. *Node* Koleksi

Node koleksi memiliki beberapa atribut yaitu: judul buku, penerbit, golongan, bahasa, ISBN, *barcode*, dan kota. Dari beberapa atribut koleksi tersebut akan dipecah menjadi beberapa *node* untuk lebih memudahkan dalam klasifikasi buku. *Node* yang akan dibuat dari atribut koleksi adalah:

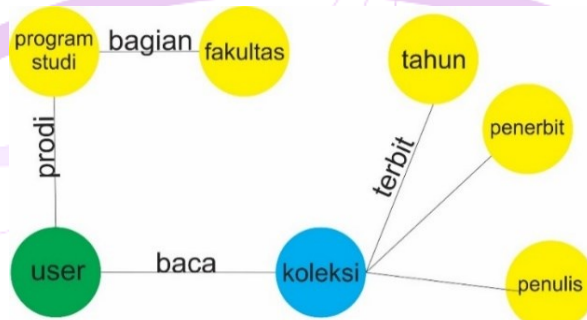
1. *Node* Koleksi
Atribut dari koleksi adalah judul buku, nomor inventaris, dan bahasa.
2. *Node* Penerbit
Atribut dari penerbit adalah nama penerbit dan kota.
3. *Node* Tahun
Atribut dari tahun adalah tahun terbit.
4. *Node* Pengarang
Atribut dari pengarang adalah nama pengarang.

b. *Node User*

Node user sendiri memiliki beberapa atribut yang akan dipecah menjadi beberapa *node*. *User* memiliki atribut nama, NIM, jurusan, dan program studi (*prodi*). Dari beberapa atribut tersebut akan dipecah menjadi beberapa *node* yaitu:

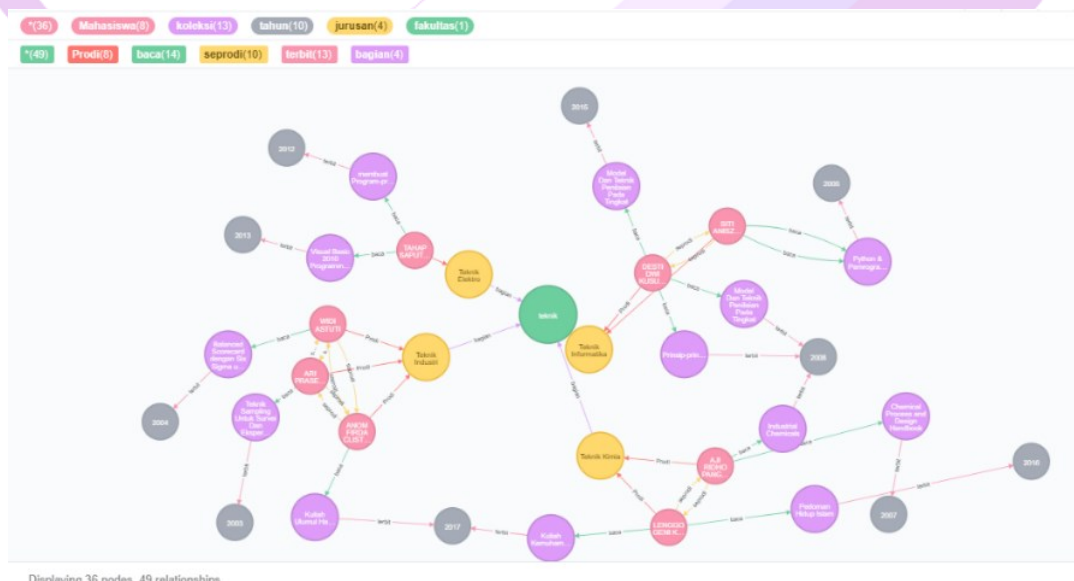
1. *Node User*
Node User akan memiliki atribut nama dan NIM.
2. *Node Jurusan*
Node Jurusan akan memiliki atribut nama jurusan.
3. *Node Prodi*
Node Prodi akan memiliki atribut nama program studi.

Data yang sudah dipecah tersebut akan dikelompokkan menjadi *node* dan *edge* untuk *input* ke DBMS Neo4j sebagai *graph database*. Data pecahan dari koleksi dan *user* yaitu koleksi, tahun, penerbit, pengarang, *user*, program studi, dan jurusan akan dijadikan *node*, untuk menghubungkan tiap *node* menggunakan *edge*.



Gambar 2. Rancangan Graph Database

Gambar 2 merupakan rancangan keseluruhan dari *graph database* yang akan dibuat untuk perpustakaan. *Node* tersebut akan saling terhubung satu dengan lainnya dengan *node user* dan koleksi sebagai *center*-nya. *Node* lain seperti tahun terbit, pengarang, jurusan, dan fakultas merupakan *node* pendukung yang dapat menghubungkan antar *user* atau antar koleksi. Penghubung antara koleksi dan *user* merupakan hasil transaksi yang terjadi. Transaksi tersebut dapat berupa peminjaman dan pembacaan.



Gambar 3. Tampilan Data Graph Database Perpustakaan

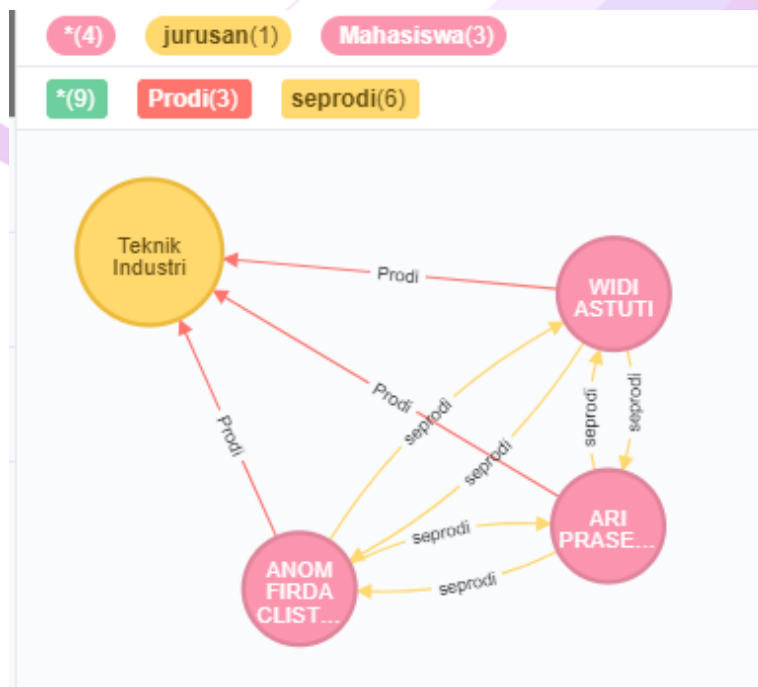
Hasil eksekusi query di Neo4J akan menampilkan seluruh data seperti yang dapat dilihat pada Gambar 3. *Node user* yang sudah ditambahkan akan menjadi *node* dengan warna merah, *node*

koleksi berwarna ungu, dan *node* jurusan serta fakultas berwarna kuning. Gambar 3 adalah hasil dari pemanggilan semua data yang sudah tersimpan pada *database*. *Node* hijau adalah *node* fakultas sedangkan *node* kuning adalah *node* jurusan, kemudian *node* merah yaitu *node* user, *node* ungu adalah *node* koleksi dan *node* abu-abu adalah *node* tahun terbit dari koleksi. Semua *node* yang sudah tersimpan dihubungkan dengan *node* lainnya menggunakan *edge* dengan *value* sesuai dengan bentuk *edge* masing-masing.



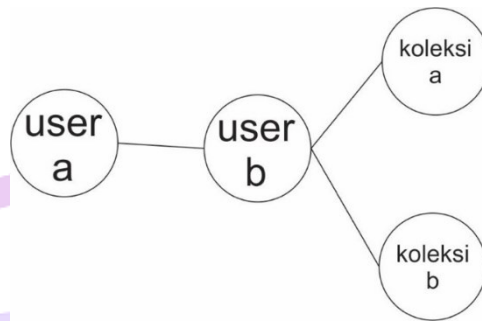
Gambar 4. Koleksi yang ada di Perpustakaan

Beberapa *query* yang bisa dilakukan antara lain menampilkan semua koleksi yang tersimpan terlihat pada Gambar 4. Selanjutnya *query* untuk menampilkan data mahasiswa dari yang terkait dengan satu jurusan terlihat pada Gambar 5. Gambar 5 menunjukkan *node* jurusan adalah *node* berwarna kuning sebagai jurusan dan merah muda sebagai *node* mahasiswa, beserta *edge* yang menghubungkan antar *node* tersebut sebagai relasi memiliki (jurusan memiliki mahasiswa).



Gambar 5. Mahasiswa dan Jurusanya

Data yang tersimpan dalam *database* ini dapat dimanipulasi untuk melakukan banyak hal. Salah satu hasil manipulasi data yang dapat dimanfaatkan oleh sistem perpustakaan adalah dengan memberikan sistem rekomendasi kepada *user*. Sistem rekomendasi yang digunakan yaitu rekomendasi berbasis *People to People Corelation*. Sistem rekomendasi jenis ini memanfaatkan koneksi antar *user* untuk dimanipulasi menjadi rekomendasi koleksi kepada *user*. Gambar 6 merupakan rancangan sistem rekomendasi untuk *user* A yaitu koleksi A dan B yang diambilkan dari koleksi yang sudah dibaca oleh *user* B. Sistem rekomendasi jenis ini bekerja dengan memanfaatkan koneksi antar *user*. Sebagai contoh, *user* A memiliki hubungan dengan *user* B maka koleksi yang sudah dibaca oleh *user* B akan dijadikan rekomendasi untuk *user* A.



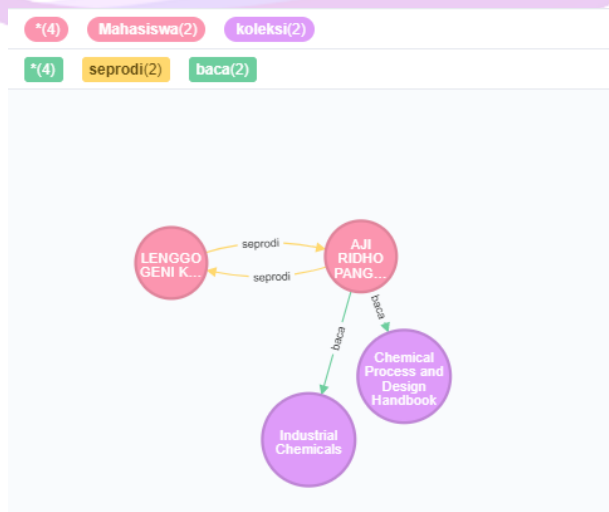
Gambar 6. Rancangan Sistem Rekomendasi Koleksi

Query untuk menampilkan rekomendasi adalah sebagai berikut (Cielen et al., 2016):

```

    ○ MATCH a=(M1600020085:Mahasiswa{name: 'Andi'})-[r:seprodi]->(b)
    ○ where (b:Mahasiswa)
    ○ WITH a, b
    ○ OPTIONAL MATCH (b)-[r:baca]->(c)
    ○ WHERE (c:koleksi)
    ○ RETURN a,b,c;
  
```

Hasil dari *query* tersebut dapat dilihat pada Gambar 7. Pada Gambar 7, *node* yang berwarna merah muda menggambarkan *user* dan ungu sebagai koleksi. Pada contoh *query* tersebut, variabel A berisi pemustaka 'Lenggo' sebagai *user* dan variabel B berisi pemustaka 'Aji'. Aji dan Lenggo mempunyai hubungan sama-sama satu program studi (prodi). Aji pernah melakukan transaksi (baca atau pinjam) koleksi 'Industrial chemicals' dan 'Chemical process and design handbook'. Hasil rekomendasi adalah koleksi-koleksi yang berelasi dengan *user* Aji akan direkomendasikan untuk diakses (dibaca atau dipinjam) oleh *user* Lenggo yaitu koleksi 'Industrial chemicals' dan 'Chemical process and design handbook'.



Gambar 7. Hasil Rekomendasi Koleksi untuk User A

4. KESIMPULAN

Berdasarkan pembahasan di atas, maka dapat disimpulkan hal-hal sebagai berikut:

- a. *Graph database* merupakan salah satu alternatif solusi sebagai *database* yang dapat digunakan untuk sistem perpustakaan.
- b. Implementasi *graph database* dilakukan dengan menekankan hubungan relasi antar *node* utama yaitu koleksi dan pemustaka.
- c. *Graph database* mampu mendukung sistem rekomendasi koleksi berdasarkan pemustaka yang telah mengaksesnya dengan baik dan mudah divisualisasikan.

DAFTAR PUSTAKA

- Cielen, D., Meysman, A. D. B., & Ali, M. (2016). *Introducing Data Science: Big data, machine learning, and more, using Python tools*. Manning Publications.
- Feret, B., & Marcinek, M. (2005). *The Future of The Academic Library and The Academic Librarian: A Delphi Study Reloaded*. *New Review of Information Networking*, 11(1), 37–63. <https://doi.org/10.1080/13614570500268381>
- Huang, Z., Chung, W., Ong, T.-H., & Chen, H. (2002). *A graph-based recommender system for digital library*. *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries*.
- Jo, T. (2019). *Text Mining: Concepts, Implementation, and Big Data Challenge*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-91815-0>
- Needham, M., & Hodler, A. E. (2019). *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*. O'Reilly Media.
- Sarlina, M., & Setiadi, T. (2017). *Pengukuran Kepuasan Pemustaka Terhadap Kualitas Pelayanan Perpustakaan Universitas Ahmad Dahlan Yogyakarta Dengan Menggunakan Metode Customer Satisfaction Index (CSI)*. *Jurnal Sarjana Teknik Informatika*, 5(2), 1–10. <http://journal.uad.ac.id/index.php/JSTIF/article/view/10843>
- Setialana, P., Adji, T., & Ardiyanto, I. (2017). *Perbandingan Performa Relational, Document-Oriented dan Graph Database Pada Struktur Data Directed Acyclic Graph*. *Jurnal Buana Informatika*, 8(2), 77–86. <https://doi.org/10.24002/jbi.v8i2.1079>

