

## Transformasi Database *Graph* Dokumen Standar Mutu SPMI STMIK El Rahma Yogyakarta

Rachmad Sanuri<sup>1</sup>, Momon Muzakkar<sup>2</sup>, Afif Ismail<sup>3</sup>

<sup>1</sup> Program Studi Sistem Informasi, STMIK El Rahma Yogyakarta

<sup>2,3</sup> Program Studi Informatika, STMIK El Rahma Yogyakarta

e-mail: <sup>1</sup>sanuri@stmikelrahma.ac.id, <sup>2</sup>muzakkarmomon@gmail.com, <sup>3</sup>afifismail021@gmail.com

### Abstrak

Basis data relasional atau RDBMS (Relational Database Management System) telah banyak digunakan secara luas. Akan tetapi pada saat ini perkembangan data mengalami pertumbuhan yang sangat cepat, seiring kebutuhan ekstraksi informasi dan kebutuhan untuk saling terhubung pada data yang besar. Dengan meningkatnya volume data dan kebutuhan interoperabilitas antar sistem, pengaksesan database relasional melalui metode query yang melibatkan banyak operasi join yang kompleks dapat mengurangi kinerja suatu sistem. Graph Databases dapat menjadi solusi sebagai sebuah database yang dapat menyimpan koneksi antar entitas dengan menyediakan koneksi traversing dengan cepat, mudah dan efisien dalam pengaksesan data. Penelitian ini merupakan hasil proses migrasi sebuah basis data relasional dokumen Sistem Penjaminan Mutu Internal (SPMI) kedalam basis data model graph. Proses migrasi sistem database relasional dokumen penjaminan mutu model parent-child tree hierarchical kedalam database graph, bertujuan agar pengambilan data dapat menggunakan metode query data secara semantik serta membandingkan proses akses data dan kinerja diantara keduanya.

**Kata kunci**—Relational Database, Graph Database, Database Migration, NoSQL

### 1. PENDAHULUAN

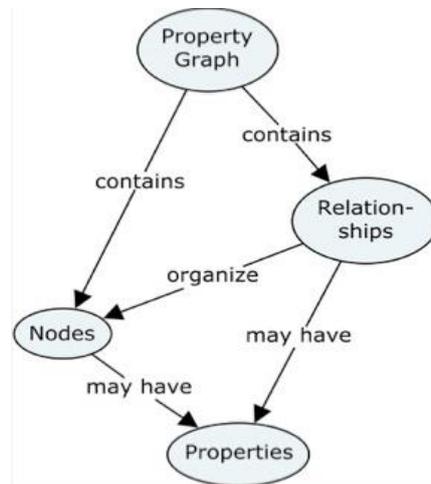
Model database relasional atau *Relational Database Management System* (RDBMS) adalah kumpulan tabel, baris dan kolom untuk mengatur dan menyimpan data. Sebuah tabel dapat ditampilkan sebagai sebuah matriks baris dan kolom, dimana setiap perpotongan baris dan kolom berisi nilai data tertentu. Inilah bentuk relasional karena semuanya baris berbagi *field* yang sama dalam tabel dan hubungan dapat dibuat di antara tabel untuk menyimpan dan mengambil data yang dipilih secara efisien. *Script SQL (Structured Query Language)* merupakan sebagai standar untuk mengakses data dari database relasional. SQL digunakan untuk menambah, memperbarui atau menghapus baris data, mengambil subset data dalam pemrosesan transaksi dan mengelola semua aspek database untuk membantu perusahaan lebih memahami dan mengeksekusi informasi sekaligus meningkatkan proses *decision-making*.

Model *Graph Database* adalah kumpulan *node* sebagai bulatan atau titik (*Vertex*), kemudian *edge* atau garis relationships antar *node*, dan *property* yang mewakili dan menyimpan data yang menggunakan struktur graph sebagaimana divisualisasikan pada gambar 1. Pada saat ini *Graph Database* semakin banyak diminati karena mempunyai kemampuan yang powerful untuk data modeling, dengan memberikan kesesuaian yang lebih dekat dengan data dunia nyata[1].

*Graph Database* menyimpan, memproses, dan mengkoneksikan query antar data secara efisien, dengan menyimpan hubungan objek data sebagai node, sedangkan hubungan atau relationships antara objek dinyatakan dengan garis (*Edge*) dalam suatu model data. Metode yang digunakan dalam pengambilan data menggunakan query data secara semantik berdasarkan pendekatan NoSQL. Pada model database relasional untuk melakukan proses komputasi terhadap suatu *relationships data* pada proses query dengan operasi *join* kadang merupakan hal yang rumit dan kompleks.

Pada *graph database* menyimpan koneksi sebagai entitas persisten. Dengan mengakses koneksi secara efisien dan memungkinkan melakukan traversing data dengan mudah tanpa *effort* yang besar terutama dalam operasi *join relation* antar data. *Graph property* berisi entitas yang terhubung sebagai node yang dapat menampung attributes (*key-value-pairs*). *Node* dapat ditandai dengan label yang mewakili peran yang barbed pada sebuah domain. Sebuah *relationship* menyediakan koneksi yang

terarah dan relevan secara semantik antara dua entitas node. Suatu relationship akan selalu mempunyai arah dari suatu tipe, node awal, dan node akhir. Kemudian sebuah relationship node dapat memiliki properti.



Gambar 1. Model Database Graph

Dalam kasus nyata suatu domain dapat memiliki banyak entitas, dengan berbagai properti, dan relationship antar entitas yang kompleks. Sedangkan pada model database relasional suatu entitas dan propertinya disimpan dalam tabel terstruktur. Kemudian *relationship*-nya disimpan melalui *derived foreign key references*. Metode pengaksesan entitas dilakukan melalui operasi *join* suatu entitas dengan entitas lain.

Sistem Penjaminan Mutu Internal atau SPMI merupakan kegiatan penting untuk melakukan penjaminan mutu perguruan tinggi. Menurut Permendikbud Ristek No.53 Tahun 2023 menjelaskan kegiatan SPMI dilakukan secara otonom yang berfungsi untuk mengendalikan dan meningkatkan penyelenggaraan pendidikan tinggi agar lebih tertata. Dalam penyelenggara SPMI setiap perguruan tinggi memiliki tugas untuk menyusun beberapa dokumen sebagai penunjang dan pendukung [2].

Penelitian ini bertujuan untuk melakukan migrasi database dokumen SPMI dari suatu model relational database *parent-child hierarchical* data dokumen ke dalam graph database, khususnya dokumen yang memuat data kebijakan, standar, audit mutu dan manual penjaminan mutu. Penelitian ini juga membandingkan proses akses data dan kinerja antara database relasional dan graph database, dengan membandingkan kegunaan relatif dari database relasional *PostgreSQL* dan database Graph *Neo4j*. Dengan penggunaan database graph memungkinkan pengambilan data menggunakan metode query data secara semantik.

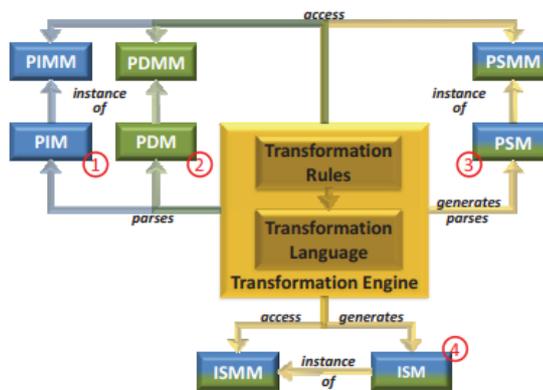
## 2. METODE PENELITIAN

Metode penelitian yang digunakan adalah pendekatan kerangka konseptual. Pendekatan ini merupakan suatu bentuk kerangka berpikir yang dapat digunakan sebagai pendekatan dalam memecahkan masalah. Biasanya kerangka penelitian ini menggunakan pendekatan ilmiah dan menunjukkan hubungan antar variabel dalam proses analisisnya. Sistematisa penelitian digunakan untuk menjelaskan tahapan-tahapan pada penelitian, bagan kerangka berpikir akan memberikan informasi detail terkait tahapan dalam pelaksanaan penelitian. Adapun gambar bagan kerangka berpikir dalam penelitian ini dapat dilihat pada gambar 2.

Dalam penelitian ini metode pengembangan sistem yang digunakan adalah *Model Driven Architecture* (MDA). Dengan menggunakan MDA dapat meningkatkan produktivitas pembangunan sistem informasi bisnis[3]. *Model Driven Architecture* adalah metodologi pengembangan perangkat lunak yang berfokus pada pembuatan dan eksplorasi domain model yang diciptakan oleh *Object Management Group* (OMG) pada 2001[4]. Tahapan dari proses MDA terlihat pada gambar 3.



Gambar 2. Bagan Kerangka Berpikir



Gambar 3. Tahapan Proses MDA

Salah satu tujuan utama MDA adalah untuk memisahkan spesifikasi fungsional dan detail dari suatu sistem untuk di implementasi pada platform tertentu. Arsitektur MDA membagi proses transformasi dalam empat tahapan, yaitu *Platform Independent Mode* (PIM), *Platform Description Model* (PDM), *Platform Specific Model* (PSM), *Implementation Specific Model* (ISM)[5]. Kemudian MDA juga membagi menjadi tiga jenis transformasi dalam sebuah siklus pengembangan perangkat lunak atau SDLC[6]. Ada tiga jenis transformasi arsitektur yang diidentifikasi tersebut yaitu *Platform Independent Model* (PIM) ke dalam PDM, model PDM ke dalam model PSM and model PSM ke dalam *Code*.

Detail dari proses dalam MDA tahap pertama adalah proses penentuan PIM, yaitu penentuan terhadap platform independen yang didukung, PIM harus independen tidak tergantung dengan vendor tertentu. Tahap kedua adalah PDM, yaitu penentuan layanan dari platform. Tahap ketiga adalah PSM, untuk mendefinisikan aturan atau role transformasi yang tepat. Tahap keempat adalah ISM yaitu mendefinisikan PSM yang mencakup detail implementasi sekaligus *runtime*-nya. Oleh karena itu ISM dapat dipahami sebagai implementasi pengkodean yang sebenarnya.

Pada saat ini ada tiga jenis pendekatan berbasis proses yang berkembang dalam transformasi database, yaitu transformasi dengan pendekatan dengan pemrograman, transformasi dengan pendekatan template dan transformasi dengan pemodelan. Pada penelitian ini digunakan metode MDA dengan dengan pendekatan template.

### 3. HASIL DAN PEMBAHASAN

Sebagai objek dari penelitian adalah aplikasi sistem informasi manajemen audit mutu internal sistem penjaminan mutu yang berbasis *Odoo Enterprise Resource Planning* (ERP) pada Lembaga

Penjaminan Mutu STMIK El Rahma Yogyakarta yang menggunakan database PostgreSQL sebagai meta model database sumber untuk menyimpan data dokumen standar mutu SPMI[7].

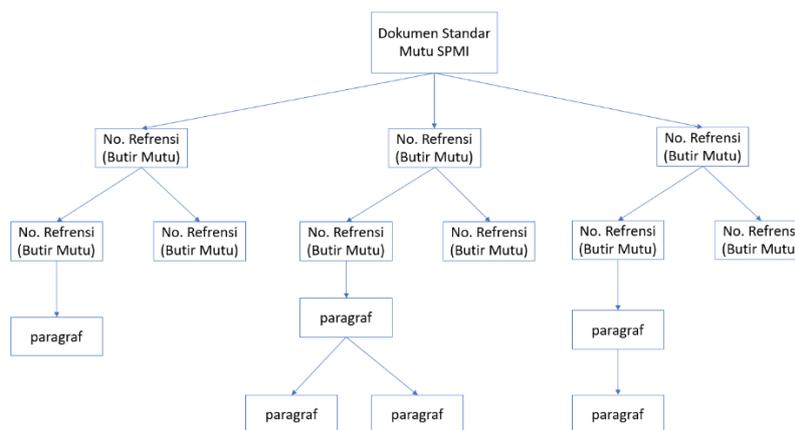
Tahap pertama dari transformasi database penelitian ini adalah penentuan PIM dengan memilih meta model database target menggunakan database Neo4J. Sistem database Neo4J adalah salah satu perangkat lunak manajemen basis data NoSQL yang independent dan populer pada saat ini. Sistem database Neo4J menyimpan dan menyajikan data dalam bentuk graph. Neo4J sangat cocok untuk menyimpan dan mengambil data yang mempunyai banyak *relationship* data yang saling berkaitan.

### 3.1. Model Data Dokumen Standar Mutu SPMI

Perguruan Tinggi sebagai penyelenggara SPMI akan memiliki tugas untuk menyusun beberapa dokumen, seperti dokumen kebijakan, standar mutu, tata cara penilaian (manual), dokumen audit mutu, hingga dokumen terkait dokumentasi formulir. Dokumen standar mutu SPMI tersebut memiliki jenis-jenis dokumen yang berbeda dan masing mempunyai hubungan kompleks antara dokumen satu dengan dokumen lainnya. Setiap dokumen akan memiliki entitas data dan tingkatan hierarki yang beda.

Tahap Analisa kebutuhan inilah merupakan tahap kedua yaitu *fase* PDM. Setiap Aturan Standar Mutu SPMI akan memiliki satu atau beberapa jenis dokumen standar mutu SPMI. Dalam sistem masing - masing akan memiliki klausa bernomor sebagai *children entities*. Kemudian setiap klausul Standar Mutu SPMI akan memiliki nomor dokumen atau ID dan memiliki paragraf sebagai entitas anak dengan nomor klausa.

Suatu paragraf klausa akan memiliki nomor dan akan mempunyai sub paragraf sebagai *child data paragraph* dalam dokumen standar mutu. Sebagaimana ditunjukkan dalam Gambar-2 tentang model data struktur sistem dokumen Standar Mutu SPMI.



**Gambar 4. Visualisasi Struktur Data Dokumen Mutu SPMI**

Selain masing-masing hubungan *parent-child*, terdapat juga hubungan silang antara berbagai jenis dokumen standar mutu SPMI. Setiap klausa akan memiliki nomor referensi yang dapat dihubungkan dengan klausa bernomor lainnya pada dokumen yang lain. Setiap klausa atau paragraf dapat disimpan sebagai item dalam sebuah model dan dapat dihubungkan ke item lainnya.

### 3.2 Roles Transformasi Database Relational Menjadi Database Graph

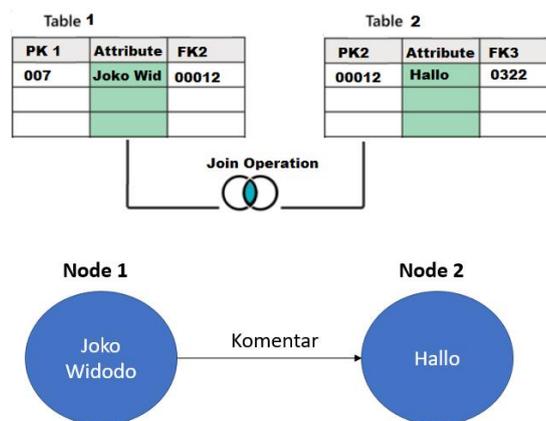
Pengembangan sistem database dokumen ini melibatkan beberapa meta-model. Pengembangan meta-model digunakan metode M2M (*Model to Model*) dan metode transformasi M2T (*Model to Text*). Sehingga dibutuhkan pengkodean untuk mentransformasikan kedalam database berorientasi graph. Selain itu dibutuhkan penerapan algoritma transformasi M2M (gambar 5) dan Pemrograman menggunakan QVT melalui *Operational Mappings language* [8], kemudian Transformasi M2T dilakukan dengan Pemrograman Aceleo.

Tahap *Platform Specific Model* atau PSM dalam proses migrasi pada system ini adalah tahapan proses implementasikan konversi sesuai dengan *roles* transformasi yang telah ditentukan. Informasi metadata database sumber digunakan untuk membuat *node label* dan *relationship type* dalam database

graph. Sedangkan data pada tabel sebagai *instance node* dan *edge* atau garis relationships antar node. Berikut prinsip-prinsip sebagai roles dalam proses transformasi.

1. Setiap tabel diwakili oleh suatu label pada *node graph database*.
2. Setiap baris dalam tabel adalah sebuah *node*.
3. Setiap kolom pada tabel menjadi properti dari *node*.
4. *Primary key* pada tabel akan dihapus akan tetapi domain dari *primary key* akan dipertahankan.
5. Menambahkan *unique constraints* sebagai *business primary keys*.
6. Menambahkan indeks untuk atribut pencarian yang sering dilakukan.
7. Mengganti *foreign keys* dengan *relationship* pada masing-masing tabel yang berhubungan
8. Data dalam tabel yang dinormalisasi dan diduplikasi dengan menempatkan di node terpisah untuk mendapatkan model yang lebih baik.

Join tabel diubah menjadi relationship, kolom pada tabel tersebut menjadi relationship properti.



Gambar 5. Roles Transformasi Model Database Relasional Ke Model Database Graph

### 3.2.1 Transformasi Rules M2M

Tahap *Implementation Specific Model (ISM)* dilakukan melalui proses transformasi model database relasional sebagai input dan sebuah model database graph sebagai output. Hal utama dalam aturan transformasi ini adalah menetapkan korespondensi antara seluruh bagian skema relasional database dan seluruh elemen database berorientasi graph. Tujuan dari penetapan roles tujuannya adalah untuk mengubah setiap tabel dan *rows*-nya menjadi sebuah *node*, dengan membuat dokumen serta *relationship* untuk setiap node. Pada gambar 6 adalah cuplikan kode untuk mentransformasi database penerapan dari transformasi *roles* M2M.

### 3.2.2 Roles Transformasi M2T

Kode *script* pada gambar 7 untuk melakukan transformasi M2T yang berfungsi membuat database berorientasi graph didalam Neo4J, dengan cuplikan code transformasi menggunakan *script Aceleo*.

## 3.3 Hasil

Metode MDA yang digunakan dalam penelitian ini adalah menggunakan pendekatan template untuk menghasilkan Database Graph. Sebagaimana dijelaskan diatas pendekatan ini memerlukan meta-model database baik itu dari sumber dan target database. Berikut adalah skema *meta-model* database sumber dan *meta-model* target database.

### 3.3.1 Relational Database Sebagai Meta-model Database Sumber

Pada gambar 8 dibawah ini memberikan ilustrasi skema dari RDBMS sebagai meta-model sumber database.

```

modeltype RDBMS uses "http://www.example.org/bdr2graph";
modeltype OD uses "http://www.example.org/oDTTarget";

transformation RDBMS2GD(in RDBModel:RDBMS, out ODModel:OD);
main() {
  RDBModel.objects()[Schema]->map
  schema2DataBase();
}
mapping Schema::schema2DataBase () :
Data_Base {

  name:= self.name;
  nodes := RDBModel.objects()[Table] -> maptableToNode();

}
mapping
Table::tableToNode):_Node{
  name:=self.name;
  document:= self.row -> map
  rowToDocument();
}

mapping Row::rowToDocument():Document{
  name:=self.id;
  self.columns ->forEach(c) {
    if c.foreign_key->size()=0 then {
      Couple1+= c.map
      columnToCouple();
    }else {
      documents+=c.foreign_key.map
      foreignKey2Document(c.value);}
    endif;
  }
}

mapping Foreign_key::foreignKey2Document(c:String):Document{
  name:=self.table.name;
  Couple1:=self.primary_key.column.map fkToCouple(c)
}

mapping Column::columnToCouple():Couple{
  Attribut:=self.name;
  valeur:=self.value;
}
mapping Column::fkToCouple(c:String):Couple{
  Attribut:=self.name;
  valeur:=c;
}
}

```

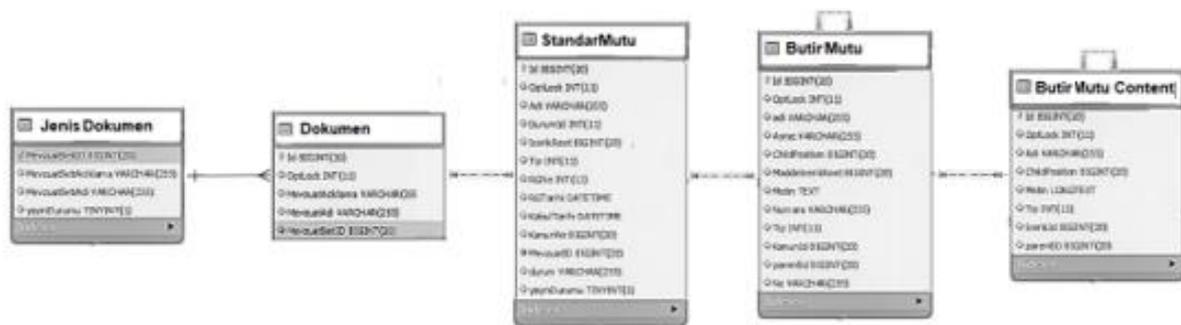
**Gambar 6. Cuplikan Kode Transformasi Rules M2M**

```

[comment encoding = UTF-8 /]
[module generate("http://www.example.org/oGDTTarget")]
[template public generateElement(aDB: Data_Base)]
[comment @main]
[for (c : Node | aDB.Node)]
  [file (aDB.name.concat('/',c.name).concat('.apoc.cypher', false, 'UTF-8')]
  [Node2Document(c)]
[/file]
[/for]
[/template]
[template public Node2Document(c:Node)]
[for (D : Document | c.document)] {
  [for (cp : Couple | D.Couple1)separator ( , )
  "cp.Attribut": "cp.valeur"]
  [/for]
  [if (D.documents->size()>0),
  [for (dc : Document | D.documents)separator ( , )
  "[dc.name]": [ ' ] ]
  [document(dc)] [ ' ] ]
  [/for]
  [/if]
}
[/for]
[/template]
[template public document(dc:Document)]
[for (cp : Couple | dc.Couple1)]
  [for (cp : Couple | dc.Couple1)separator( , )
  "cp.valeur"]
  [/for]
[/for]
[/template]

```

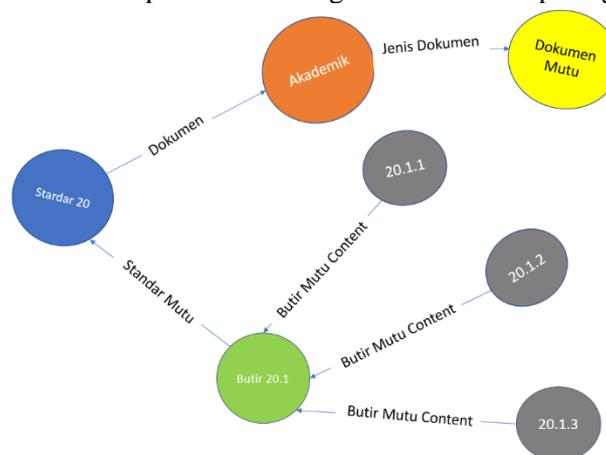
**Gambar 7. Potongan Kode Transformasi Rules M2T**



Gambar 8. Relasi Antar Tabel RDBMS Dokumen Penjaminan Mutu SPMI

### 3.3.2 Graph Database Sebagai Meta-model Document-oriented Target

Skema *meta-model* dari Graph Database target diilustrasikan pada gambar 9 berikut



Gambar 9. Model Graph Dokumen PenjaminanMutu SPMI

### 3.4. Pengujian Dengan Query

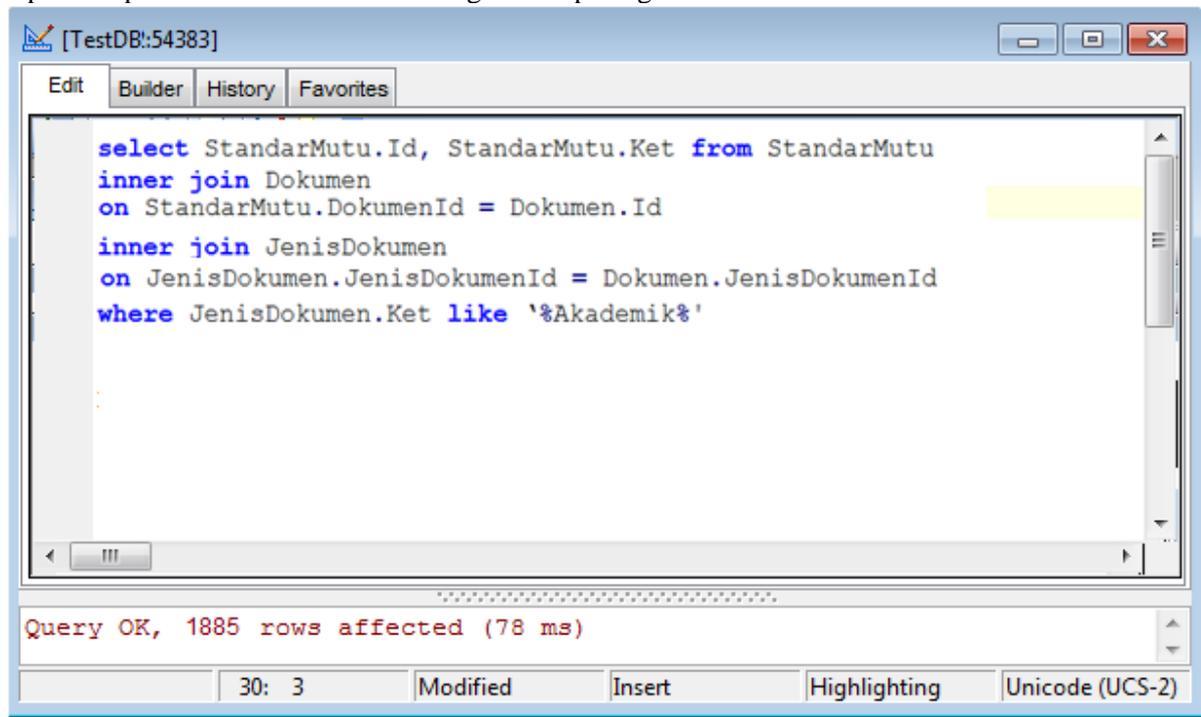
Setelah database sistem dokumen SPMI dimigrasikan selanjutnya akan dibandingkan kinerja akses diantara dua data database tersebut. Dengan cara mencari nilai data dengan pola hubungan yang sama. Akses kedalam database tersebut menggunakan metode query, pada database relasional digunakan *SQL script* sedangkan pada database graph menggunakan *Cypher script*.

Proses query pada database *PostgreSQL* yang digunakan untuk mengambil semua data pada database Dokumen Standar Mutu SPMI dari sistem induk yaitu pencarian kondisi yang mengandung data "Akademik". Seperti yang ditunjukkan pada gambar 10, diperlukan dua *operasi join* untuk mengakses data dari beberapa *table*, melalui operasi *join table* maka query ini akan menurunkan kinerja jika dikemudian hari data pada tabel - tabel tersebut semakin besar.

Pada gambar 11 merupakan implementasi akses data pada database Neo4J untuk mengambil semua data Dokumen Standar Mutu SPMI dari sistem dengan data yang dicari adalah kata "Akademik". *Script query* dengan bahasa Cypher lebih pendek, dengan hasil query 10 kali lebih cepat dibandingkan database relasional. Pada gambar gambar 11 adalah salah satu potongan *script query cypher*, selama proses pencarian data tidak membutuhkan operasi join yang rumit atau proses traversing yang menggabungkan semua data tabel untuk pencarian entitas tertentu. Selama eksekusi *query* proses yang dilakukan mengakses database dengan data "Akademik". Sesudah itu pengaksesan children node hanya melalui relasi node.

Proses query dengan Cypher lebih pendek karena graph database menyimpan data dalam model yang menyerupai objek pada dunia nyata sekaligus hubungan antar objeknya. Notasi *query Cypher* lebih mudah dipahami untuk operasi pada data yang kompleks dibandingkan dengan query dengan SQL pada database relasional. Selain itu query Cypher lebih efisien daripada SQL untuk model data *tree-hierarchy*. Proses Eksekusi query pada database graph lebih cepat daripada database relasional pada

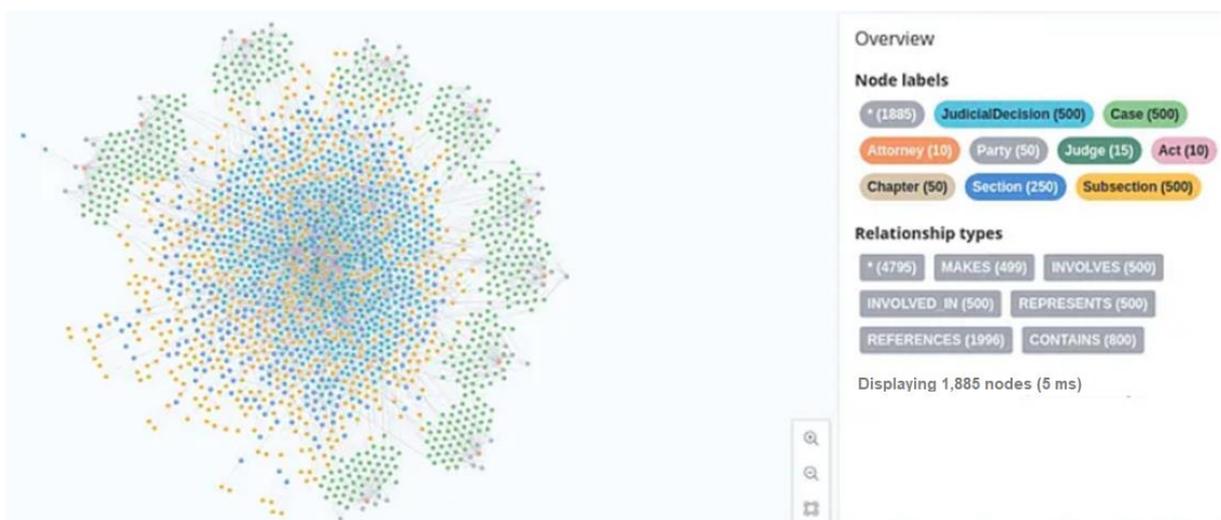
data model *tree-hierarchy* sebagaimana struktur data pada sistem dokumen standar SPMI. Percobaan akses dengan data sampel sejumlah ribuan records data sample pada database graph, dengan hasil lebih cepat dari pada database relational sebagaimana pada gambar 12.



Gambar 10. Query Pada Database PostgreSQL Dengan SQL

```
match(StandarMutu : StandarMutu)-[*]->(JenisDokumen: JenisDokumen) where
JenisDokumen. JenisDokumenKet ="Akademik" return StandarMutu;
```

Gambar 11. Query Pada Database Neo4j Dengan Cypher



Gambar 12. Hasil Query Dengan Cypher

#### 4. KESIMPULAN

Metode transformasi database dengan metode MDA dapat digunakan untuk memigrasikan model database relational ke dalam model database graph. Proses transformasi yang dikembangkan menggunakan QVT untuk mengubah relational database menjadi model graph database, sedangkan

untuk mempercepat transformasi dapat dilakukan melalui proses otomatisasi dengan melakukan pemrograman menggunakan Aceleo untuk melakukan transformasi data dari database *PostgreSQL* ke dalam Graph database *Neo4j*. Proses Eksekusi query pada database graph lebih cepat daripada database relasional pada data model *tree-hierarchy* sebagaimana database pada sistem dokumen standar SPMI yang digunakan pada penelitian ini. Untuk kedepan perlu ditingkatkan kemampuannya untuk melakukan transformasi dan akses data pada platform data yang lebih besar secara *real-time*.

#### DAFTAR PUSTAKA

- [1] *All Data International*, Kelebihan dan Kekurangan dari Penggunaan Graph Database, <https://alldataint.com/kelebihan-dan-kekurangan-dari-penggunaan-graph-database/>, diakses tanggal 25 April 2024 pukul 20.00 WIB
- [2] Tim Pengembang SPMI Direktorat Penjaminan Mutu Kemdikbud, SPMI | Sistem Penjaminan Mutu Internal, <https://spmi.kemdikbud.go.id/>, diakses tanggal 27 April 2024 pukul 14.00 WIB
- [3] Miller, J. Mukerji, "MDA *Guide Version 1.0.1*", 2013,OMG, Halaman 1-3.
- [4] Object Management Group (OMG) Team, Object Management Group Model Driven Architecture (MDA) MDA Guide rev. 2.0, <https://www.omg.org/cgi-bin/doc?ormsc/14-06-01>, diakses tanggal 28 April 2024 pukul 08.00 WIB
- [5] Bézivin J., Gerbé O. (2001), "Towards a precise definition of the OMG/MDA framework". In: ASE, halaman 274-278.
- [6] Sara Gotti, Samir Mbarki (2019). IFVM Bridge: A Model Driven IFML Execution. International Journal of Online and Biomedical Engineering (iJOE). Vol. 15 No. 4. halaman 111-126.
- [7] Rachmad Sanuri, Momon Muzakkar dan Darmansyah Nugraha, "Pengembangan Odoo Enterprise Resource Planning (ERP) Modul Audit Mutu Internal Sistem Penjaminan Mutu Dengan Metode Accelerated SAP", Jurnal FAHMA, Vol.21, No.3, September 2023, DOI: <https://doi.org/10.61805/fahma.v21i3.100>
- [8] Eclipse QVT Developer Team, Eclipse QVT Operational, <https://projects.eclipse.org/projects/modeling.mmt.qvt-oml>, diakses tanggal 28 April 2024 pukul 08.00 WIB