

MIGRASI TABLE BASIS DATA RELASIONAL KE COLLECTION BASIS DATA NOSQL DENGAN TEKNIK TRANSFORMASI DATA

Muhammad Nur Yasir Utomo¹⁾, Rini Nur²⁾

^{1,2)} Dosen Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang, Makassar

ABSTRACT

Current technological developments have encouraged the production of data which is not only large in volume but also varies in terms of type. This development demands a storage media with flexibility and great performance. Relational databases that have been widely used during the last few decades are no longer relevant to use. As a solution, the NoSQL database comes with the ability to store data with large volumes and with various forms of types. NoSQL made the need for migration from relational databases increased. However, the task of migrating a relational database to a NoSQL database is not an easy task. This study proposes an early stage database migration model that can move the table schema and its data to a collection in NoSQL database. The model was built by using the SHOW COLUMN query to get a list of columns in a relational database table and SELECT to get the data in the table. In the end, data transformation conducted by formed the columns and the data into Key-Value format to be stored in the MongoDB NoSQL database. Based on experiment that have been conducted using table with 200 data row and evaluation using Black Box Testing, the proposed model is proven to work well in migrating MySQL relational database table to NoSQL MongoDB collection.

Keywords: Database Migration, Relational Database, NoSQL, MySQL, MongoDB

1. PENDAHULUAN

Perkembangan teknologi utamanya internet saat ini sangat cepat sehingga terjadi peningkatan jumlah data yang sangat besar [1]. Volume data yang dihasilkan setiap detiknya sangat besar dengan bentuk yang variatif (terstruktur, tidak terstruktur atau semi-terstruktur) [2]. Hal ini menuntut media penyimpanan yang fleksibel dan performansi baik. *Database* relasional yang telah menjadi media penyimpanan dan mengelola data yang banyak digunakan dalam beberapa dekade terakhir [3] sudah tidak lagi relevan digunakan dalam menyimpan data dengan volume besar dan variatif. Untuk mengatasi masalah ini, *database* NoSQL diperkenalkan. *Database* NoSQL menghadirkan pendekatan baru untuk menyimpan dan memproses data dalam jumlah besar [4].

Database NoSQL saat ini telah menjadi alternatif populer untuk mengatasi keterbatasan pada *database* relasional karena *database* NoSQL memiliki mekanisme bawaan untuk memproses dan menganalisis data dalam jumlah besar. Selain itu, *database* NoSQL juga memiliki kemampuan untuk menyimpan data dalam berbagai format [5]. Kebebasan format data yang ditawarkan oleh NoSQL memungkinkan penanganan variasi penyimpanan data yang lebih mudah. Kemampuan NoSQL inilah yang membuat kebutuhan migrasi dari *database* relasional ke NoSQL menjadi sangat diperlukan [6], [7]. Namun demikian, migrasi *database* relasional ke *database* NoSQL merupakan sebuah tugas menantang karena NoSQL tidak menyediakan dukungan yang memadai untuk kueri SQL [8].

Beberapa penelitian telah dilakukan untuk menjawab tantangan penyediaan *tools* migrasi *database*, salahnya dilakukan oleh Solanke pada tahun 2017 [9], penelitian menggunakan data *adapter* yang diposisikan sebagai *middleware* antar aplikasi dan *database*. Data *adapter* ini dapat menerjemahkan permintaan data untuk SQL dan NoSQL. Hasil penelitian ini bekerja sangat baik untuk infrastruktur *database* Hybrid, namun tidak menyelesaikan masalah mengenai penanganan data *existing* pada SQL *database* yang ingin di migrasikan ke NoSQL.

Penelitian lanjutan kemudian dilakukan oleh El Alami pada tahun 2017 [10]. Penelitian ini memanfaatkan *metadata* dari *database* SQL untuk membuat migrasi secara otomatis. Hasil pembacaan *metadata* menjadi dasar transformasi data melalui model *rule* yang didefinisikan oleh peneliti. Kekurangan hasil penelitian ini adalah pada seleksi informasi *metadata* yang berakibat pada lambatnya proses migrasi. Pemanfaat *metadata* dalam melakukan proses migrasi data pada *database* SQL ke NoSQL juga dilakukn oleh dilakukan oleh Ramzan pada tahun 2019 [11] dengan menambahkan pendekatan data *cleansing*. Penelitian ini mengembangkan model migrasi dengan dua tahap yaitu tahap data transformasi yang memanfaatkan *metadata* dan tahap data *cleansing*. Penelitian ini mengklaim bahwa model migrasi yang diajukan berhasil melakukan

¹ Korespondensi penulis: Muhammad Nur Yasir Utomo, Telp 081340179194, yasirutomo@poliupg.ac.id

migrasi dan memperbaiki kualitas data. Namun demikian, perubahan isi data pada tahap data *cleansing* membuat data yang dimigrasikan bukan lagi data asli sehingga ada perbedaan pada isi data dalam *database* relasional dan hasil migrasi pada NoSQL.

Berdasarkan penelitian sebelumnya yang telah dipaparkan, perkembangan penelitian tentang migrasi data SQL ke NoSQL saat ini terus berkembang [12]. Namun demikian, penelitian-penelitian yang ada masih meninggalkan celah pengembangan terkait model migrasi yang dapat memindahkan data dengan cepat dan akurat tanpa mengubah isi data yang asli. Penelitian tahap awal ini mencoba menyelesaikan masalah tersebut dengan mengembangkan model perangkat lunak untuk migrasi *table* dari *database* relasional/SQL ke bentuk *collection database* NoSQL MongoDB menggunakan Python.

Penelitian ini membangun model migrasi dengan teknik transformasi data yang memanfaatkan kueri *Administrative Statement* `SHOW COLUMNS` dan kueri *Data Definition Language* (DDL) `SELECT` pada *database* relasional/SQL untuk mendapatkan skema *table* dan data yang ingin dipindahkan ke *collection* NoSQL MongoDB. Diharapkan hasil penelitian ini dapat menjadi dasar pengembangan dalam membuat *tools* migrasi dari *database* relasional ke NoSQL.

Paper ini disusun sebagai berikut. Bagian 2 menjelaskan metode penelitian yang diajukan, termasuk terkait *database* uji coba yang digunakan, langkah-langkah bagaimana migrasi *database* dilakukan, dan evaluasinya. Bagian 3 menjelaskan hasil dan analisis eksperimen penelitian. Akhirnya, bagian 4 menyimpulkan seluruh penelitian ini.

2. METODE PENELITIAN

Penelitian ini dilakukan dengan tiga tahap yaitu pengumpulan data uji coba/eksperimen, pembuatan model migrasi *table database* relasional ke *collection database* NoSQL MongoDB dan evaluasi hasil penelitian. Tiap tahap tersebut dapat dijabarkan sebagai berikut:

2.1 Pengumpulan Data Uji coba dan Setup Eksperimen

Data uji coba yang digunakan pada penelitian ini merupakan *database* MySQL dari website IDalamat.com. IDalamat.com merupakan salah website informasi alamat terbesar di Indonesia dengan lebih dari 250.000 data alamat [13]. IDalamat.com sendiri memiliki *database* yang mencakup banyak *table* seperti diantaranya *table* alamat, *table* user, *table* komentar dan lainnya.

nama_lokasi	deskripsi_lokasi	alamat_lokasi	kode_pos	no_telepon	website	email	tanggal
Politeknik Negeri Ujung Pandang (PNU)	<p>Politeknik merupakan lembaga pendidikan tinggi...	Jalan Perintis Kemerdekaan KM.10 Tamalanrea, Makas...	90245	0411-585386	http://poliupg.ac.id	pnup@poliupg.ac.id	2015-02-16
Universitas Hasanuddin Makassar (UNHAS)	Universitas Hasanuddin (UNHAS) merupakan salah sat...	Jl. Perintis Kemerdekaan Km.10 Makassar, 90245, Ko...	90245	0411-586200	http://www.unhas.ac.id/	humas@unhas.ac.id	2015-02-16
Universitas Muslim Indonesia (UMI) Makassar	<p>Universitas Muslim Indonesia (UMI) adalah unive...	Menara UMI Lt. 09, Urip Sumohardjo KM.05 Makassar,...	90231	0411-455666	http://www.umi.ac.id		2017-01-19
Universitas Negeri Makassar (UNM)	Universitas Negeri Makassar, disingkat UNM, adalah...	Kampus Gunungsari Baru Jl. A.P. Pettarani Makassar...	90222	0411-869854	http://www.unm.ac.id	info@unm.ac.id	2015-02-16

Gambar 1. Kolom-kolom dari *Table* Data Alamat Pada Website IDalamat.com

Pada penelitian ini, salah satu *table* pada *database* IDalamat.com digunakan sebagai *table* uji coba untuk dipindahkan dari *database* relasional MySQL ke bentuk *collection* NoSQL MongoDB. *Table* yang digunakan tersebut adalah *table* alamat yang memiliki 24 kolom data seperti terlihat pada Gambar 1. Untuk keperluan uji coba, jumlah baris data pada *table* ini juga diabatasi menjadi 200 data. Untuk setup uji coba, penelitian ini dilakukan dengan menggunakan perangkat PC dengan spesifikasi sistem operasi Windows, RAM 8 GB dan *processor* Inter Core i5.

2.2 Rancangan Model Migrasi

Model migrasi *database* relasional ke NoSQL yang diajukan pada penelitian ini dibangun dengan teknik transformasi data yang memanfaatkan dua jenis kueri yang ada pada *database* relasional SQL, kedua jenis kueri tersebut adalah *Administrative Statement* SHOW COLUMNS dan kueri *Data Definition Language* (DDL) SELECT. Kueri SHOW COLUMNS dimanfaatkan untuk mendapatkan informasi skema sebuah *table database* relasional dan SELECT dimanfaatkan untuk mengambil data yang ada pada *table*. Hasil dari kedua kueri tersebut dapat dimanfaatkan untuk membentuk format *key-value* data yang dapat diterima oleh *database* NoSQL. Secara rinci, model yang diajukan diperlihatkan oleh Pseudocode Gambar 2 berikut:

Algorithm 1: MySQL to NoSQL Migration Process

```

Input : SQL (MySQL) database and table name
Output: NoSQL (MongoDB) database and collection
1 conMySQL = MySQL connection to [database name]
2 if SQL database and table exist then
3   columnList = SHOW COLUMNS FROM [table name]
4   dataRows = SELECT * FROM [table name]
5   result = []
6   for x,i in dataRows do
7     document = {}
8     for y,z in columnList do
9       document[z] = dataRows[x][y]
10    end
11    result.append(document)
12  end
13  conMongoDB = Make connection to [database name]
14  collection = conMongoDB make collection [table name]
15  collection.insert_many(result)
16  end process
17 else
18   database or table not exist
19  end process
20 end

```

Gambar 2. Alur Proses Migrasi Model yang Dikembangkan

Gambar 2 diatas memperlihatkan alur proses migrasi *database* relasional ke NoSQL yang diajukan. Alur proses model migrasi yang diajukan tersebut dapat dijelaskan sebagai berikut:

1. Melakukan koneksi dengan database relasional (MySQL)
Proses ini bertujuan untuk menghubungkan model yang dibangun dengan bahasa Python dengan objek *database* relasional yang akan dipindahkan ke *database* NoSQL.
2. Melakukan pengecekan *database* dan *table* target tersedia dalam MySQL
Proses ini bertujuan untuk memastikan target *table* yang akan di pindahkan ke NoSQL benar-benar ada dalam *database*. Jika *database* dan target *table* ada, maka proses dilanjutkan, jika tidak ada maka proses langsung dihentikan.
3. Mengambil daftar kolom pada *table* target
Informasi terkait kolom apa saja pada *table* target dilakukan dengan memanfaatkan perintah SHOW COLUMN. Perintah ini memberikan hasil daftar kolom yang ada. Hasil dari proses ini akan digunakan sebagai *Key* pada format *database* NoSQL.
4. Mengambil data pada *table* target
Semua data pada *table* target diambil mengguakan perintah SELECT. Hasil dari proses pengambilan data ini akan digunakan serbagai *Value* saat dipindahkan ke *database* NoSQL.
5. Melakukan transformasi data dengan *pairing* pembentukan *Key-Value*
Key-value merupakan format data yang dapat diterima oleh database NoSQL, oleh sebab itu data dari *database* relasional harus diubah menjadi format *key-value*. Perubahan format ini dilakukan dengan melakukan perulangan untuk tiap data yang diperoleh dari langkah 4, kemudian untuk tiap data kolom

dipasangkan dengan nama kolom yang sesuai berdasarkan urutan datanya. Ilustrasi proses ini ditunjukkan oleh Tabel 1 berikut:

Tabel 1. Ilustrasi Hasil Proses Pembentukan Format Data *Key-Value*

<i>Input: Raw Data</i>			<i>Output: Key-Value Pair</i>			
Daftar kolom: <table border="1"> <tr> <td>nama_lokasi</td> <td>alamat_lokasi</td> <td>no_telepon</td> </tr> </table>			nama_lokasi	alamat_lokasi	no_telepon	<pre>{ "nama_lokasi" : "Politeknik Negeri Ujung Pandang", "alamat_lokasi" : "Jalan Perintis Kemerdekaan KM.10 Tamalanrea , Makassar 90245", "no_telepon" : "+62 (411) 585365" }</pre>
nama_lokasi	alamat_lokasi	no_telepon				
Row data: <table border="1"> <tr> <td>Politeknik Negeri Ujung Pandang</td> <td>Jalan Perintis Kemerdekaan KM.10 Tamalanrea , Makassar 90245</td> <td>+62 (411) 585365</td> </tr> </table>			Politeknik Negeri Ujung Pandang	Jalan Perintis Kemerdekaan KM.10 Tamalanrea , Makassar 90245	+62 (411) 585365	
Politeknik Negeri Ujung Pandang	Jalan Perintis Kemerdekaan KM.10 Tamalanrea , Makassar 90245	+62 (411) 585365				

6. Memasukkan data ke *database* NoSQL MongoDB

Tahap terakhir adalah proses memasukkan data yang telah berbentuk *key-value* ke dalam *database* NoSQL MongoDB. Proses ini dilakuakn dengan memanfaatkan perintah `insert_many` yang tersedia pada MongoDB agar proses insert bisa dilakukan sekaligus untuk semua data yang akan dimigrasikan.

2.3 Evaluasi Hasil Migrasi Database

Evaluasi hasil migrasi *database* dari model yang diajukan dilakukan dengan menggunakan metode Black Box Testing. Black Box Testing adalah metode dalam bidang *software testing* untuk menguji fungsionalitas dari sebuah aplikasi [14]. Black Box Testing dilakukan dengan membuat daftar uji fungsionalitas aplikasi, tiap daftar uji akan memiliki status hasil tes apakah berhasil atau tidak [15]. Adapun pada penelitian ini, beberapa hal yang akan diuji ialah *table* berhasil dimigrasikan menjadi *collection*, jumlah kolom hasil migrasi, jumlah data yang dimigrasikan dan kesesuaian *pair key-value* hasil migrasi.

3. HASIL DAN PEMBAHASAN

Model migrasi *table database* relasional ke NoSQL pada penelitian ini dilakukan dengan menggunakan Black Box Testing. Hasil dari pengujian yang dilakukan diperlihatkan oleh Tabel 2 dibawah ini:

Tabel 2. Hasil Evaluasi Migrasi *Table Database* Relasional ke *Collection* NoSQL

No	<i>Input</i>	Hasil yang diharapkan	<i>Output</i>	Status
1	Nama <i>database</i>	<i>Database</i> baru pada NoSQL	<i>Database</i> baru terbuat pada NoSQL	OK
2	Nama <i>table</i>	<i>Collection</i> baru pada NoSQL	<i>Collection</i> baru terbuat pada NoSQL	OK
3	Daftar <i>field</i> /kolom <i>table</i>	<i>Key</i> pada <i>collection</i> NoSQL sesuai dengan nama kolom dari <i>table database</i> relasional	Nama <i>key</i> dan jumlahnya pada NoSQL sesuai dengan nama dan jumlah kolom di <i>database</i> relasional	OK
4	Jumlah <i>row</i> data sebanyak 200 data	Data sebanyak 200 baris berhasil dimigrasikan	200 data pada <i>collection</i> NoSQL	OK
5	Urutan kolom dan data	Hasil pembuatan format data <i>Key-Value</i> sesuai dengan urutannya	Urutan pasangan <i>Key</i> dan <i>Value</i> sudah sesuai	OK

Berdasarkan hasil pada Tabel 2 diatas terlihat bahwa model migrasi *database* yang diajukan dapat bekerja dengan baik. Dari lima kriteria, *output* dari model yang diajukan sesuai dengan hasil yang diharapkan. Gambar 3 memperlihatkan hasil migrasi secara fisik dari *database* relasional ke *database* NoSQL.

nama_lokasi	deskripsi_lokasi	alamat_lokasi	kode_pos	no_telepon
Politeknik Negeri Ujung Pandang (PNUP)	<p> Politeknik merupakan lembaga pendidikan tingg...	Jalan Perintis Kemerdekaan KM.10 Tamalanrea, Makas...	90245	0411-585386
Universitas Hasanuddin (UNHAS) Makassar (UNHAS)	Universitas Hasanuddin (UNHAS) merupakan salah sat...	Jl. Perintis Kemerdekaan Km.10 Makassar, Ko...	90245	0411-586200

Key	Value	Type
(1) ObjectId("5f928cea7c...")	{ 25 fields }	Object
_id	ObjectId("5f928cea7cb5a03ba9e48308")	ObjectId
nama_lokasi	Politeknik Negeri Ujung Pandang (PNUP)	String
deskripsi_lokasi	<p> Politeknik merupakan lembaga pendit...	String
alamat_lokasi	Jalan Perintis Kemerdekaan KM.10 Tamala...	String
kode_pos	90245	String
no_telepon	0411-585386	String
email	pnup@poliupg.ac.id	String

(a) *Table* pada *Database* Relasional MySQL (b) *Collection* pada *Database* NoSQL MongoDB

Gambar 3. Hasil Migrasi *Database* Relasional ke NoSQL

Berdasarkan Gambar 5 diatas, terlihat bahwa kolom dan data pada *database* relasional sudah berhasil dibentuk menjadi format data *key-value pair* dengan urutan yang benar dan telah berhasil disimpan pada *database* NoSQL MongoDB.

4. KESIMPULAN

Sebuah model migrasi *table database* relasional ke *collection database* NoSQL diajukan pada penelitian ini. Model yang diajukan memanfaatkan kueri `SHOW COLUMN` untuk mendapatkan daftar kolom pada *table database* relasional dan `SELECT` untuk mendapatkan data yang ada pada *table*. Pada akhirnya kolom dan data yang diperoleh di transformasi ke format data *Key-Value* yang kemudian dimasukkan ke *database* NoSQL. Berdasarkan ujicoba menggunakan *table* dengan 200 *row* data dan evaluasi dengan metode Black Box Testing yang telah dilakukan, model yang diajukan pada penelitian ini terbukti dalam melakukan migrasi *table database* relasional ke *collection database* NoSQL.

Penelitian lanjutan dapat dilakukan dengan mengembangkan model agar dapat melakukan migrasi semua *table* yang ada pada *database* secara otomatis.

5. DAFTAR PUSTAKA

- [1] S. Hamouda and Z. Zainol, "Document-Oriented Data Schema for Relational Database Migration to NoSQL," in Proceedings - 2017 International Conference on Big Data Innovations and Applications, Innovate-Data 2017, pp. 43-50, 2018.
- [2] M. N. Y. Utomo, T. B. Adji, and I. Ardiyanto, "Geolocation prediction in social media data using text analysis: A review," International Conference on Information and Communications Technology, ICOIACT 2018, pp. 84-89, 2018.
- [3] M. N. Y. Utomo, A. E. Permanasari, E. Tungadi, and I. Syamsuddin, "Determining single tuition fee of higher education in Indonesia: A comparative analysis of data mining classification algorithms," in Proceedings of 2017 4th International Conference on New Media Studies, CONMEDIA 2017, pp. 113-117, 2017.
- [4] E. Tungadi, I. Thalib, and M. N. Y. Utomo, "Machine Learning Penentuan Penerima Beasiswa Peningkatan Prestasi Akademik (PPA) Menggunakan Metode Jaringan Saraf Tiruan (JST)," Seminar Nasional Teknik Elektro dan Informatika 2018, pp. 391-396, 2018.
- [5] Rosdiana, E. Tungadi, Z. Saharuna, and M. N. Y. Utomo, "Analisis Sentimen pada Twitter terhadap Pelayanan Pemerintah Kota Makassar," Seminar Nasional Teknik Elektro dan Informatika, pp. 87-93, 2019.
- [6] S. Ghule and R. Vadali, "Transformation of SQL system to NoSQL system and performing data analytics using SVM," Proceedings - International Conference on Trends in Electronics and Informatics, ICEI 2017, pp. 883-887, 2018.
- [7] M. N. Y. Utomo, T. B. Adji, and I. Ardiyanto, "Prediksi Geolokasi Berbasis Teks untuk Data Media Sosial Berbahasa Indonesia Menggunakan Named Entity Extraction," Universitas Gadjah Mada, 2018.
- [8] K. Ma, B. Yang, and A. Abraham, "Asynchronous Data Translation Framework for Converting Relational Tables to Document Stores," International Journal of Computers and Applications, vol. 38, no. 1, pp. 19-28, 2016.
- [9] G. B. Solanke and K. Rajeswari, "SQL to NoSQL transformation system using data adapter and analytics," Proceedings - 2017 IEEE International Conference on Technological Innovations in Communication,

- Control and Automation, *TICCA 2017*, pp. 59-63, 2018.
- [10] A. El Alami and M. Bahaj, "Migration of a relational databases to NoSQL: The way forward," *International Conference on Multimedia Computing and Systems*, pp. 18-23, 2017.
- [11] S. Ramzan, I. S. Bajwa, B. Ramzan, and W. Anwar, "Intelligent Data Engineering for Migration to NoSQL Based Secure Environments," *IEEE Access*, vol. 7, pp. 69042-69057, 2019.
- [12] Mohamed Hanine, A. Bendarag, and O. Boutkhom, "Data Migration Methodology from Relational to NoSQL Databases," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 9, no. 12, pp. 2566-2570, 2015.
- [13] M. N. Y. Utomo, A. Bastian, and A. Winursito, "Improving Speed Performance of Select Random Query in SQL Database," *INTEK Jurnal Penelitian*, vol. 7, no. 1, pp. 26-31, 2020.
- [14] M. S. Aliero, I. Ghani, K. N. Qureshi, and M. F. Rohani, "An algorithm for detecting SQL injection vulnerability using black-box testing," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 1, pp. 249-266, 2020.
- [15] S. Roohullah Jan, S. Tauhid Ullah Shah, Z. Ullah Johar, Y. Shah, and F. Khan, "An Innovative Approach to Investigate Various Software Testing Techniques and Strategies," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 2, no. 2, pp. 682-689, 2016.

6. UCAPAN TERIMA KASIH

Ucapan terima kasih kami sampaikan kepada IDalamat.com dan Adresseek.com karena telah mendukung penelitian ini dalam hal penyediaan data ujicoba. Kami juga menyampaikan terima kasih kepada Unit Penelitian dan Pengabdian kepada Masyarakat (UPPM) Politeknik Negeri Ujung Pandang yang telah mendukung pendanaan penelitian ini.