

Implementasi Sistem Analytics pada Produk menggunakan OLAP berbasis MongoDB

I Kadek Krisna Dwi Payana^{a1}, Luh Arida Ayu Rahning Putri.^{a2}

^aInformatics Department, Udayana University
Jalan Raya Kampus Unud, Jimbaran, Bali, 80361, Indonesia
¹ikadekkrisnadwipayana@gmail.com
²rahningputri@unud.ac.id

Abstract

Technology right now can knowing user what they want or find out. From what user likes, what user want and what recently they find. That kind of technology using analytics for knowing what user often see or recently visited. Users visited some product that will be tracked and saved that data for analytics. That kind of system need effective database and lower traffic databases. Using Online Analytical Processing (OLAP) can lower traffic for database other than database for business purpose or often called Online Transactional Database (OLTP). Splitting traffic for transaction request and analytics request and make more structured database. Main focus for database Online Analytical Processing (OLAP) for databases analytics and Online Transactional Processing (OLTP) for business purpose kind like product listed and main data. This system using golang (google language) for development backend cause this language are tested very fast rather than other language and easy to maintain.

Keywords: Data Warehouse, OLAP, OLTP, MongoDB, SQL

1. Pendahuluan

Teknologi saat ini dapat mengetahui apa saja produk favorit user maupun apa saja yang akan user beli atau sedang diminati. hal tersebut dikarenakan program pada aplikasi melakukan pencatatan apa saja yang user buka/akses dan mencatat hal tersebut pada database. Namun, terdapat permasalahan dimana data yang digunakan untuk transaksi biasa seperti melakukan list produk, pencatatan transaksi, detail produk dan lain sebagainya jika disatukan dengan data yang akan dilakukan analisis menjadikan *traffic* pada database tersebut menjadinya lebih berat dikarenakan adanya *traffic* untuk analisis juga [1]. Maka dari itu adanya Database *Online Analytical Processing (OLAP)*.

Online Analytical Processing (OLAP) merupakan suatu database yang bertujuan hanya untuk Proses Analisis saja. Penggunaan OLAP ini sangatlah cocok untuk suatu web service yang melakukan *transactional* biasa seperti penyimpanan data produk, transaksi pada aplikasi dan juga melakukan analisa. Peran OLAP yakni sebagai tempat untuk menyimpan data yang digunakan untuk melakukan analisis data [2]. Data analisis ini dapat digunakan dalam melakukan analisa apa saja hal yang disukai oleh pengguna, apa saja hal yang ingin dibeli oleh pengguna dan lain sebagainya.

Penggunaan *Online Analytical Processing (OLAP)* ini memiliki kelebihan dimana memperingan database utama/ OLTP (*transactional Database*) dalam *traffic* karena memisahkan antara database untuk penggunaan sehari hari dan juga kebutuhan analisa. Jadi penggunaan OLAP untuk program yang melakukan transaksi seperti biasa sekaligus melakukan analisa pada user sangat direkomendasikan menggunakan OLAP ini.[2]

2. Metodologi Penelitian

2.1. Gambaran Umum Aplikasi

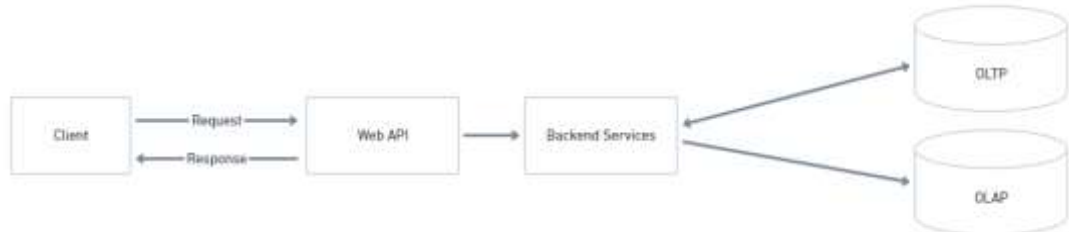
Implementasi *Analytical Database* menggunakan *Online Analytical Processing (OLAP)* ini memiliki tujuan untuk mencatat produk apa saja yang dikunjungi oleh user. Dimana

hal ini termasuk dalam *collection data* yang bertujuan untuk mengoleksi data yang nantinya dapat digunakan sebagai pengolahan data (*Big Data*).

Alur yang ditempuh pada aplikasi ini adalah pertama user melakukan kunjungan kepada salah satu produk yang ada pada Database OLTP. Lalu *Backend Services* menerima dan mengembalikan response serta mengirim catatan pengunjungan dan menyimpannya pada *OLAP* database.

2.2. Desain Aplikasi

Aplikasi yang dibuat menggunakan bahasa pemrograman golang. Flowchart program yang dibuat adalah seperti gambar 1 di bawah ini:



Gambar 1. High Level Architecture OLAP

Front-end dapat berkomunikasi dengan Back-end melalui request dan menerima response yang biasa disebut dengan *REST API*, aplikasi yang menggunakan *REST API* pada web services disebut sebagai *RESTful* [3].

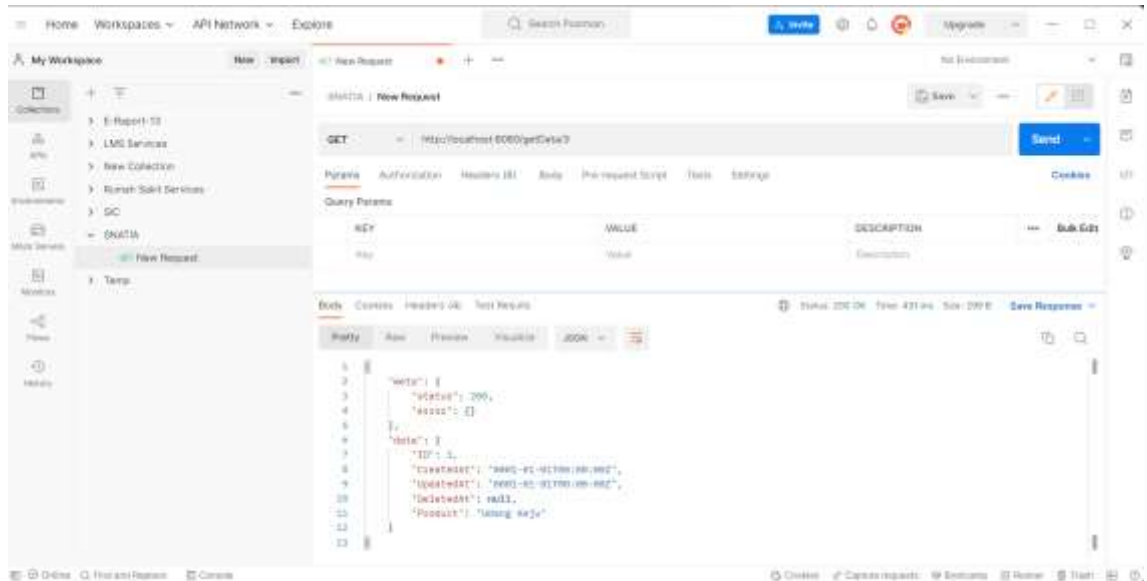
Contoh salah satu alur yang melakukan *data collection* pada *Backend Service* di *RESTful API* ini:

1. Front-end / Android (Client) melakukan request pada API dengan menampilkan detail suatu produk
2. Web API menerima alamat (request) dari Client dan mencocokkan kepada Backend Service untuk alamat yang dituju
3. Backend Services menjalankan program sesuai alamat yang di request oleh client
4. Backend Services mengembalikan response dan juga melakukan pencatatan pada *OLAP* untuk *data collection* apabila terdapat koleksi data pada service tersebut
5. Client mendapatkan response balik dari Web API

3. Hasil dan Pembahasan

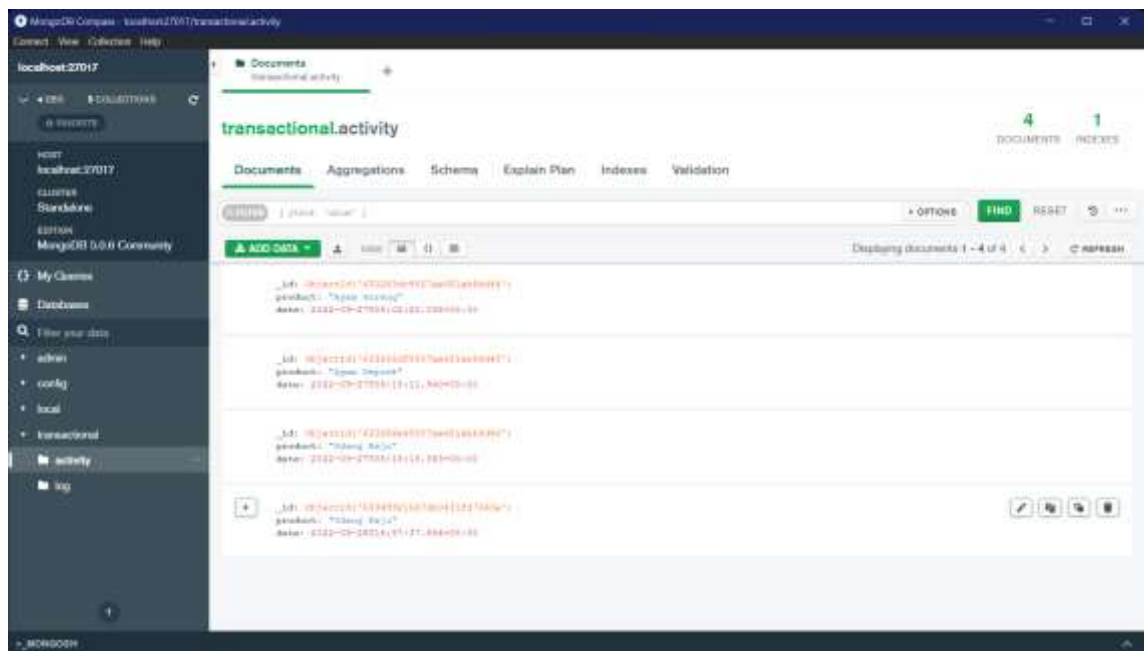
3.1. Tampilan Hasil

berikut merupakan Backend Services yang jika user menuju ke suatu produk (user melakukan click atau kunjungan) dimana nantinya akan tercatat di database OLAP seperti pada gambar 2.



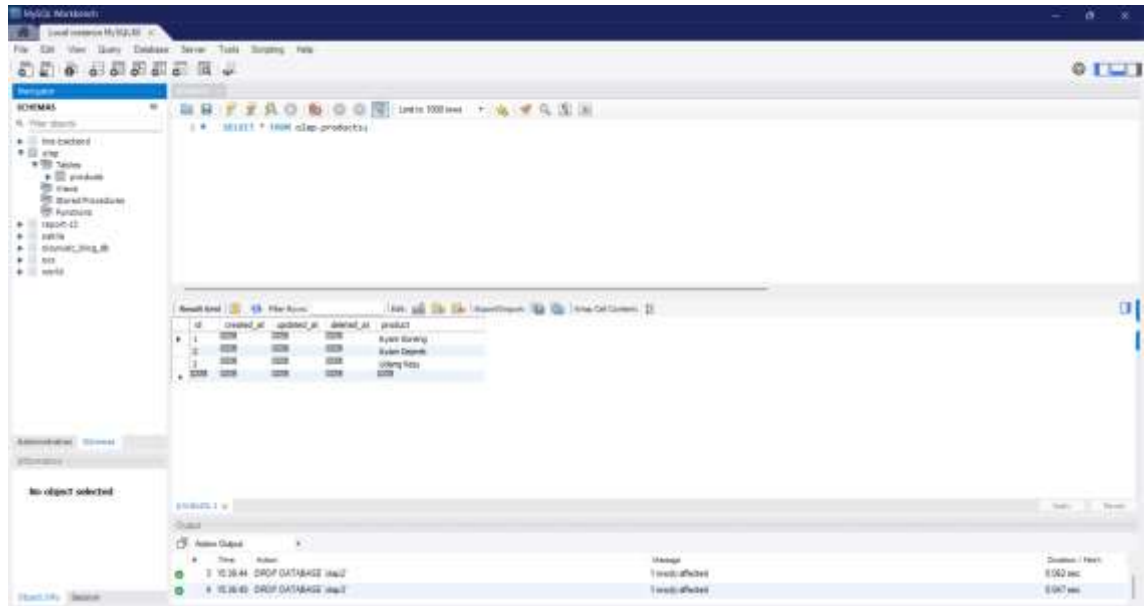
Gambar 2. User melakukan kunjungan detail produk dan mendapatkan response data

Gambar 2 merupakan contoh user melakukan request kepada API dengan alamat API *getData* dan parameter *id 3* Jadi user disini dengan kondisi memiliki minat kepada suatu produk yang ada pada database *OLTP*. lalu Backend Service mencatat user yang mengunjungi produk tersebut sebagai *Data Collection* pada database *OLAP* yang dimana berfungsi sebagai database untuk analisis.



Gambar 3. Data tercatat pada OLAP

Pada Gambar 3 terlihat bahwa produk yang sudah dikunjungi oleh user tercatat pada database. dimana data tersebut menyimpan ID, Produk dan tanggal kunjungan.



Gambar 4. Database OLTP

Pada Gambar 4 merupakan database OLTP untuk *transactional database* untuk menyimpan list produk yang ada pada aplikasi. dimana hasil pemisahan antara OLTP dan OLAP berhasil dilakukan.

3.2. Pengujian Aplikasi

Metode pengujian sistem menggunakan metode *blackbox*, dengan Hasil pengujian dapat dilihat pada Tabel 1.:

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	User melakukan request pada API	User Request dengan API yang tidak ada analisa ke OLAP	User mendapatkan response sesuai request	Sesuai Dengan Harapan	Valid
2	User Melakukan Request pada API	User Request dengan API yang ada analisa ke OLAP	User mendapatkan response sesuai request	Sesuai Dengan Harapan	Valid
3	Database Administrator (DBA) melakukan checking pada OLAP	DBA membuka database OLAP	DBA dapat melihat data yang dicatat pada OLAP	Sesuai Dengan Harapan	Valid
4	Tech dapat mematikan analisis OLAP	Backend Engineer mengubah analisis menjadi false	Tidak ada pencatatan analisis pada OLAP	Sesuai Dengan Harapan	Valid

5	Tech dapat mengaktifkan OLAP	Backend Engineer mengubah analisis menjadi true	Mengaktifkan kembali analisis OLAP	Sesuai Dengan Harapan	Valid
---	------------------------------	---	------------------------------------	-----------------------	-------

Table 1. Hasil Pengujian Aplikasi

Pada pengujian di Tabel 1. dapat kita lihat bahwa user dapat melakukan request terhadap REST API, hal tersebut dibuktikan pada Gambar 2. Pengujian lainnya juga dilakukan dengan berhubungan dengan databases, dimana dapat kita lihat database bisa diakses oleh seseorang yang memiliki role *Database Administrator* (DBA) dengan menampilkan database seperti pada Gambar 3 dan 4 dan data tersebut tidaklah tersimpan pada program.

4. Kesimpulan

Pada penelitian ini dapat disimpulkan pada sistem ini berhasil melakukan pencatatan data produk yang dikunjungi serta menyimpan data tersebut pada database OLAP. Dimana pada penelitian ini berhasil menggunakan database NoSQL yakni menggunakan *MongoDB* dan Database OLAP juga dapat diakses dengan mudah oleh *Database Administrator* (DBA) seperti pada pengujian. User juga dapat menambahkan analisa dengan menghit API pada Rest API. namun untuk saran pada penelitian selanjutnya adalah dapat dikembangkan lagi dengan menambahkan cube lainnya.

Daftar Pustaka

- [1] Kovacic, I., Schuetz, C.G., Neumayr, B. and Schrefl, M., 2022. OLAP Patterns: A pattern-based approach to multidimensional data analysis. *Data & Knowledge Engineering*, 138, p.101948.
- [2] Kleppmann, M., 2017. *Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems*. " O'Reilly Media, Inc."
- [3] Masse, M., 2011. *REST API design rulebook: designing consistent RESTful web service interfaces*. " O'Reilly Media, Inc."
- [4] Chaudhuri, S. and Dayal, U., 1997. An overview of data warehousing and OLAP technology. *ACM Sigmod record*, 26(1), pp.65-74.
- [5] Dehdouh, K., Boussaid, O. and Bentayeb, F., 2020. Big data warehouse: Building columnar nosql OLAP cubes. *International Journal of Decision Support System Technology (IJDSST)*, 12(1), pp.1-24.

Halaman ini sengaja dibiarkan kosong