

Migrasi Model Data Relasional Ke Model Data Realtime Database Firebase Untuk Aplikasi Monitoring Wisatawan

by I Ketut Gede Sudiarta

Submission date: 29-May-2023 01:01AM (UTC-0400)

Submission ID: 2104296617

File name: 20_JurnalNasional.pdf (244.8K)

Word count: 1814

Character count: 11453

MIGRASI MODEL DATA RELASIONAL KE MODEL DATA REALTIME DATABASE FIREBASE UNTUK APLIKASI MONITORING WISATAWAN

INE Indrayana¹⁾, IKG Sudiarta²⁾, IW Suasnawa³⁾

^{1,2,3)}Jurusan Teknik Elektro, Politeknik Negeri Bali,
E-mail: eddyindrayana@pnb.ac.id

Abstrak - Aplikasi Monitoring Wisatawan berbasis mobile membutuhkan akses data yang cepat untuk mendapatkan akses data yang optimal. Aplikasi Monitoring Wisatawan ini akan mencatat setiap posisi GPS perangkat yang terhubung ke dalam suatu group wisata dalam aplikasi ini. . Ada beberapa model data yang dapat digunakan untuk menyimpan data untuk aplikasi mobile diantaranya model data relational dan model data document . Model Data relasional seperti MySQL tidak bersifat realtime, sedangkan model data document seperti Firebase bersifat realtime database. Aplikasi Monitoring Wisatawan ini lebih sesuai menggunakan database firebase karena hubungan antar data tidak terlalu rumit dan membutuhkan akses data yang cepat. Penelitian ini bertujuan untuk mengetahui tahapan migrasi dari database MySQL ke Realtime Database Firebase untuk aplikasi monitoring wisatawan. Migrasi dari MySQL ke Firebase pada aplikasi monitoring wisatawan melibatkan beberapa data utama yaitu data guide, data wisatawan dan data *chatting* antar anggota group wisatawan. Aplikasi ini menggunakan google map untuk mempermudah visualisasi posisi wisatawan dalam satu group wisata.

Kata Kunci: *GPS monitoring objek, Mobile application, Monitoring group wisatawan, Realtime Database Firebase, Firebase untuk aplikasi monitoring wisatawan*

1. PENDAHULUAN

Aplikasi android memungkinkan banyak pengguna untuk mengakses data dari aplikasi tanpa terbatas pada ruang, ruang dan waktu [6] Banyak aplikasi android yang memanfaatkan Global Positioning System (GPS) sebagai data utama dalam membangun aplikasi. Diantaranya untuk aplikasi mengetahui posisi objek dalam global map, aplikasi family tracker, aplikasi untuk asisten navigasi, atau monitoring pergerakan suatu objek. Global Positioning System (GPS) merupakan salah satu alat yang berguna untuk merencanakan, mengelola dan memantau aktifitas rekreasi di alam bebas [2]. GPS merupakan alat yang relative murah dan sangat efektif untuk mengumpulkan data pergerakan wisatawan [9, 10]. Data GPS dapat juga digunakan untuk menganalisa objek wisata mana saja yang sering dikunjungi [8]. Semua aplikasi tersebut membutuhkan koordinat latitude dan longitude dari suatu objek. Koordinat latitude dan longitude ini akan memberikan posisi suatu objek di bumi ini. Koordinat ini merupakan data utama yang nantinya akan diolah lagi oleh suatu aplikasi untuk menghasilkan informasi yang bermanfaat dan memudahkan manusia untuk melakukan aktifitas tertentu. Server database akan membutuhkan performa yang tinggi karena banyaknya data yang dikirim dan diterima dari dan menuju server. Database realtime firebase merupakan salah satu database yang memiliki performa yang tinggi dalam mengolah data yang

besar [1] dibandingkan database relational seperti MySQL.

Aplikasi android dapat memanfaatkan google map untuk memvisualisasikan posisi koordinat latitude dan longitude dari suatu objek di bumi. Untuk mengaktifkan google map pada aplikasi android, ada beberapa langkah yang harus dilalui. Pertama, nama project aplikasi android harus didaftarkan ke google untuk mendapatkan google map api key. Key ini akan ditempatkan pada code project android untuk mendapatkan ijin menggunakan map dari google. Visualisasi dengan map dan menampilkan gambar tagging pada map akan memudahkan pengguna melihat posisi global dari suatu objek.

Aplikasi monitoring wisatawan ini menggunakan data koordinat latitude, longitude dan menyimpan sejumlah pesan untuk aktifitas chat group. Data koordinat ini harus terupdate setiap kali wisatawan melakukan perpindahan tempat sehingga dibutuhkan membutuhkan database yang responsif dan dapat menyimpan data yang besar. Disamping itu performa aplikasi dapat dipertahankan secara realtime. Database realtime firebase memiliki beberapa keunggulan diantaranya:

1. Database bersifat realtime database
Firebase menggunakan sinkronisasi data, semua perangkat yang terhubung akan menerima update dalam waktu milidetik.

Firestore adalah NoSQL database dan data disimpan dalam struktur pohon JSON.

2. Tetap responsif meski dalam keadaan *offline*
Aplikasi Firebase tetap responsif meskipun sedang offline atau aplikasi tidak sedang terhubung ke jaringan internet, karena SDK Firebase Realtime menyimpan data ke disk. Setelah Jaringan terhubung SDK Firebase otomatis akan melakukan update data ke server, tanpa harus ada coding untuk melakukan sinkronisasi.
3. Menyediakan banyak library
Realtime Database menyediakan library client yang dapat berintegrasi dengan Android, iOS, JavaScript, Java, Objective-C, swift dan Node.js. Firebase menyediakan API yang memungkinkan aplikasi melakukan sinkronisasi antar client serta disimpan di *cloud server*

Sedangkan MySQL adalah database manajemen system (DBMS) yang mempunyai struktur penyimpanan data dalam bentuk table relational. Data akan disimpan dalam kolom dan baris. Normalisasi tabel menyebabkan komputasi query yang tinggi pada saat memanggil data yang melibatkan banyak tabel.

Untuk melakukan migrasi ke database firebase dari database relasional, kita mendesain ulang skema data agar sesuai dengan struktur pohon. Struktur pohon yang dipakai adalah struktur data pohon JSON.

2. METODE PENELITIAN

Firebase Realtime Database adalah basis data yang disimpan di cloud dan dukungan multiplatform seperti Android, iOS, dan Web [4,7]. Data dalam firebase akan disimpan dalam struktur JSON (Java Script Object Notation) [1]. Firebase Realtime Database menggunakan sinkronisasi data setiap kali data berubah, semua perangkat yang terhubung akan menerima update dalam waktu milidetik sebagai ganti permintaan HTTP biasa. Basis data firebase akan secara otomatis menyinkronkan aplikasi klien yang terhubung dengannya. Aplikasi multiplatform yang menggunakan Android, iOS, dan JavaScript SDK akan menerima pembaruan data terbaru secara otomatis saat aplikasi terhubung ke server firebase. Beberapa hal yang perlu dipertimbangkan dalam membangun struktur basis data firebase:

1. Hindari data bertingkat
Walaupun Realtime Database Firebase menyediakan sampai 32 tingkat hirarki data, namun demikian firebase akan mengambil semua node dalam suatu lokasi di database pada saat proses baca

data [3]. Sehingga waktu akses data akan bertambah jika kita membuat banyak tingkat hirarki dalam struktur databasenya.

2. Meratakan struktur data pohon JSON
Struktur pohon JSON yang rata dan disesuaikan dengan kebutuhan antar muka yang akan ditampilkan dalam aplikasi akan meningkatkan waktu akses
3. Bangun data sesuai skala.

Langkah Langkah Migrasi dari Database Relasional ke Database Firebase

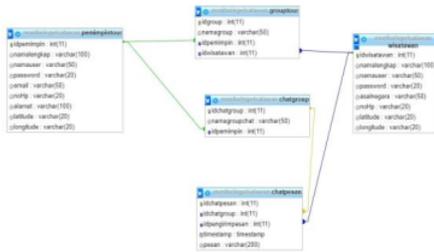
1. Identifikasi table-table yang terlibat dalam database relational
2. Identifikasi penyajian data dalam antar muka user (*user interface*) dalam membuat struktur pohon JSON dari tabel tabel tersebut
3. Denormalisasi struktur tabel untuk membentuk struktur pohon JSON sehingga tidak terlalu banyak hirarki pohon yang terjadi dan buat sedapat mungkin simpul akar yang terbentuk, dapat terpanggil seluruhnya.

Dalam membangun struktur baru pohon JSON pada database firebase, harus diingat bagaimana data tersebut dipanggil dan disajikan oleh aplikasi. firebase tidak mengenal kunci asing yang dapat kita temukan seperti pada database MySQL (database relasional). Kunci asing (*foreign key*) dipakai untuk menghubungkan antara dua tabel atau lebih pada database relasional. Sedangkan pada database Firebase tidak ditemukan kunci asing ini sehingga memungkinkan terjadinya data ganda pada saat dua sub pohon dihubungkan.

3. HASIL DAN PEMBAHASAN

3.1 Data yang terlibat dalam system

Tabel yang terlibat dalam aplikasi monitoring wisatawan adalah pemimpin rombongan wisatawan(pemimpin tour), wisatawan, group tour ,chat group dan pesan yang dikirim(pesan). Seorang pemimpin tour dapat menangani beberapa wisatawan, sedangkan seorang wisatawan hanya menjadi anggota dari satu group tour saja. Seorang pemimpin group tour dapat menjadi pemimpin group tour lebih dari satu pada saat yang berbeda. Seorang pemimpin group dapat membuat satu atau lebih chat group pada saat yang berbeda. Wisatawan boleh melakukan chat di beberapa chat group pada saat yang berbeda. Wisatawan dapat mengirimkan pesan lebih dari sekali dalam satu group chat. Visualisasi hubungan antar tabel dapat dilihat pada gambar 2 berikut



Gambar 1 Relasi tabel monitoring wisatawan

Salah satu pertimbangan dalam membuat struktur realtime database firebase adalah pengembang memperhatikan data yang akan ditampilkan pada aplikasi yang akan dibangun. Pada aplikasi monitoring wisatawan ini akan menampilkan daftar wisatawan dari satu pemimpin tour, daftar pemimpin tour, daftar chatting pesan yang ditampilkan dan posisi semua wisatawan untuk satu pemimpin tour yang ditampilkan dengan google map. Sedangkan untuk tampilan posisi semua wisatawan untuk satu pemimpin tour yang ditampilkan dengan google map tidak tergantung dengan struktur database firebase karena sifatnya hanya menampilkan visualisasi *icon tagging* posisi wisatawan saja. Dari rancangan tampilan ini kita dapat melihat ada 2 master root data yaitu data pemimpin tour sebagai sentral data yang memiliki beberapa wisatawan yang ditangani dan group chat. Rancangan tampilan pada android dapat dilihat berikut:

Nama Pemimpin Tour
* Wisatawan 1
* Wisatawan 2
* Wisatawan 3
* Wisatawan 4
* Wisatawan 5
* Wisatawan dst

Gambar 2 Rancangan Tampilan daftar wisatawan dari satu pemimpin tour

Pemimpin Tour 1
Pemimpin Tour 2
Pemimpin Tour 3
Pemimpin Tour 4
Pemimpin Tour dst

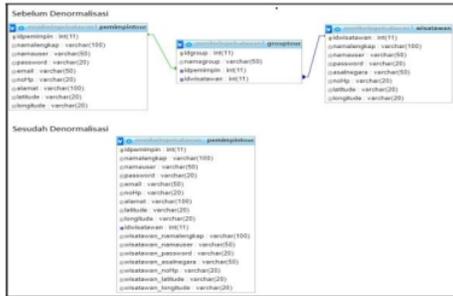
Gambar 3 Rancangan tampilan daftar pemimpin tour

Wisatawan 1
Pesan 1
Balasan Pesan dari Pemimpin Tour 1
Pesan 2
Balasan Pesan dari Pemimpin Tour 1
.....dst

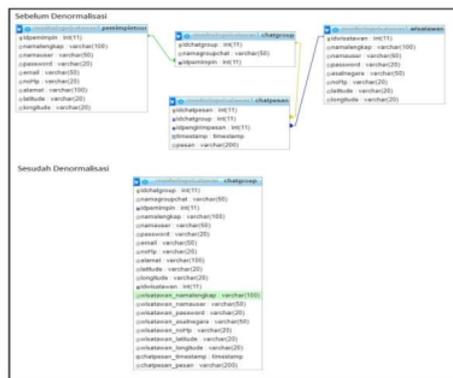
Gambar 4 Rancangan tampilan chat room antara pemimpin tour dengan wisatawan

3.2 Proses Denormalisasi

Pada gambar 1 di atas kita lihat ada beberapa relasi tabel. Pertama kita perhatikan relasi tabel pemandu, groupon, dan wisatawan. Proses denormalisasi untuk relasi tabel ini dapat dilihat pada gambar berikut:



Gambar 5. Hasil denormalisasi tabel pimpintour, grouptour dan wisatawan



Gambar 6. Hasil denormalisasi tabel chatgroup, wisatawan, chatpesan dan pimpinan tour

Implementasi pada database realtime firebase:



Gambar 7. Implementasi pada database firebase

Efek dari denormalisasi adalah munculnya data duplikasi dan konsistensi data kurang terjaga. Untuk menanggulangi masalah tersebut firebase menyediakan metode multipath update, yang akan dibahas kemudian.

4. SIMPULAN

Simpulan dari penelitian ini, bahwa proses migrasi dari database relasional (sebagai contoh database MySQL) ke database realtime firebase untuk aplikasi monitoring wisatawan dapat dilakukan dengan proses denormalisasi. Tabel-tabel pimpintour, wisatawan, grouptour, chatgroup dan chatpesan akan menjadi 2 root data besar pada database realtime firebase. Dua root data tersebut adalah pimpintour dan chatgroup merupakan hasil proses denormalisasi dengan memperhatikan antar muka aplikasi penyajian data.

REFERENSI

- [1] Almoatassem, O., Husain S.H, Parthipan, D., Mahmoud, Q.H., (2017). A Cloud-based Service for Real-Time Performance Evaluation of NoSQL Databases, *Cornell University Library*
- [2] Beeco, J. A., Hallo, J. C., & Brownlee, M. T. J. (2014). GPS visitor tracking and recreation suitability mapping: Tools for understanding and managing visitor use, *Landscape and Urban Planning*, 127, 136–145.
- [3] Google Developers (2018), Dokumentasi Firebase, download dari <https://firebase.google.com/docs/database/android/structure-data>.
- [4] Haena Khan (2014). NoSQL: A Database for Cloud Computing . *International Journal of Computer Science and Network*. Volume 3(6).
- [5] Indrayana, I N.E., Wirasyanti, D.P., Sudiarta, I K.G.,(2017), Heuristic Query Optimization for query multiple table and multiple clausa on mobile finance application, *IOP Conf. Series: Journal of Physics: Conf. Series* 953 (2017) 012049.
- [6] Khedkar, S., Thube, S. (2017). Realtime Databases for Applications. *International Research Journal of Engineering and Technology*. Volume 4(6)
- [7] Reinau, K.H., Henrik H. , Weber M., 2014. The SMS-GPS-Trip method : A new method for collecting trip information in travel behavior research. *Telecommunication Policy*, <http://dx.doi.org/10.1016/j.telpol.2014.05.006>
- [8] Silviya Karpilo, Tarmo Virtanen, Susanna Lehvarita, 2017. Smartphone GPS tracking—Inexpensive and efficient data collection on recreational movement , *Landscape and Urban Planning* 157, 608-617.
- [9] Savchuk, V., Pasichnyk, V., 2015. Modern Tendency In The Use Of GPS Technology In Tourism Industry, *Econtechmod An International Quarterly Journal*, Vol.4, No.3, pg 65-72.
- [10] Srivastava, N., Shree, U., Chauhan, N.R, Tiwari D.K. (2017). Firebase Cloud Messaging (Android), *International Journal of Innovative Research in Science, Engineering and Technology*. Volume 6(9).

Migrasi Model Data Relasional Ke Model Data Realtime Database Firebase Untuk Aplikasi Monitoring Wisatawan

ORIGINALITY REPORT

6%

SIMILARITY INDEX

4%

INTERNET SOURCES

0%

PUBLICATIONS

1%

STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

3%

★ sinta3.ristekdikti.go.id

Internet Source

Exclude quotes On

Exclude matches Off

Exclude bibliography On