

# Membangun Struktur Realtime Database Firebase Untuk Aplikasi Monitoring Pergerakan Group Wisatawan

I K G Sudiarta, I N E Indrayana, I W Suasnawa

Jurusan Teknik Elektro  
Politeknik Negeri Bali  
itutde@pnb.ac.id

## Abstrak

Penelitian ini fokus pada merancang dan membuat struktur realtime database firebase untuk aplikasi monitoring pergerakan wisatawan. Firebase merupakan salah satu cloud database yang memang didesain untuk cloud application. Database Firebase mengutamakan waktu akses data. Aplikasi ini berbasis android yang melibatkan banyak user di dalamnya. Sehingga membutuhkan struktur database yang kokoh untuk menopang aplikasi ini supaya dapat berjalan secara optimal. Data yang disimpan di sisi server adalah data latitude dan longitude dari posisi wisatawan. Data posisi latitude dan longitude ini diperoleh dari Satellite GPS melalui permintaan posisi dari masing – masing smartphone wisatawan. Aktor utama dalam aplikasi ini adalah tour leader(guide) dan anggota group wisatawan (guest). Database yang dibangun melibatkan 2 node besar yaitu node "Guide" dan node "ChatGroup" dimana sebelumnya telah melalui proses perataan struktur database. Proses pembuatan struktur model data firebase, juga perlu memperhatikan bagaimana data tersebut disajikan dalam antar muka user sehingga waktu akses data menjadi lebih optimal.

**Keywords:** *Firestore, Database NoSQL, Monitoring Pergerakan Wisatawan.*

## Abstract

*This study focuses on designing and building a realtime database of firebase databases for tourist movement monitoring applications. Firebase is a database cloud that is designed for cloud applications. The Firebase database prioritizes data access time. This application is based on android which involves many users in it. So it requires a robust database structure to support this application so that it can run optimally. Data stored on the server side is latitude and longitude data from the position of tourists. This latitude and longitude position data is obtained from Satellite GPS by requesting the position of each tourist smartphone. The main actors in this application are tour leaders (guides) and members of tourist groups (guest). The database that was built involved 2 large nodes namely "Guide" nodes and "ChatGroup" nodes which previously had gone through the database structure smoothing process. The process of creating a firebase data model structure, also needs to pay attention to how the data is presented in the user interface so that the data access time becomes more optimal*

**Keywords:** *Firestore, NoSQL Database, Tourist Movement Monitoring.*

## 1. Pendahuluan

Aplikasi Monitoring pergerakan wisatawan memerlukan database realtime firebase. Aplikasi Monitoring pergerakan wisatawan akan mencatat posisi perangkat semua user yang login ke dalam aplikasi ini. Perangkat GPS dari masing masing pengguna akan mengirimkan posisi latitude dan longitude. Aplikasi Monitoring pergerakan wisatawan ini diakses oleh banyak user dan mengirimkan posisi dari masing-masing. Waktu akses tidak akan menjadi masalah jika jumlah baris catatan dari tabel dalam basis data berkisar dari puluhan hingga ratusan rekaman,

tetapi akan menyebabkan masalah waktu akses jika *record* data telah mencapai puluhan ribu hingga jutaan *record*[1].

*Firestore Realtime Database* merupakan database yang tersimpan di cloud dan support multiplatform seperti Android, iOS dan Web[2][3]. Data pada *Firestore* akan disimpan dalam struktur JSON (Java Script Object Notation) [4]. Database *Firestore* akan melakukan sinkronisasi secara otomatis terhadap aplikasi client yang terhubung kepadanya. Aplikasi multiplatform yang menggunakan SDK Android, iOS dan JavaScript akan menerima update data terbaru secara otomatis pada saat aplikasi terhubung ke server *Firestore*.

Dalam *Firestore database*, kita dapat mengambil, mengurutkan dan memfilter data dengan query NoSQL. Database NoSQL terdiri dari empat jenis: kunci-nilai(key value), berbasis dokumen, berbasis kolom dan berbasis grafik[5][6][7]. Keunggulan dari teknologi NoSql diantaranya database yang dapat terus tersedia walau infrastruktur dari database tersebut ada yang mengalami kerusakan. Data terdistribusi secara geografis sehingga dapat diakses dimana saja, waktu respon akses data relative cepat untuk aplikasi cloud, dapat diprediksi skalanya untuk memenuhi kebutuhan data untuk keadaan sekarang maupun kedepan dan tidak membutuhkan *object relation mapping*[8].

*Firestore* memiliki SDK *real-time* dan memprioritaskan aplikasi seluler dan mendukung penyimpanan data local secara off-line. Tujuan utama dari realtime database *Firestore* adalah kinerja waktu akses data. *Realtime database* mengoptimalkan waktu akses, sehingga akses data berada dikisaran mikrodetik bahkan nanodetik, sehingga biaya akses data dapat diminimalisir.

Beberapa kemampuan *Firestore* yang ditawarkan ke pengguna dan pengembang system :

1. *Realtime database*. *Firestore* akan melakukan proses update dan sinkronisasi data, setiap ada data yang berubah. Sinkronisasi dilakukan dalam milidetik dimana proses ini sebagai ganti penggunaan permintaan HTTP biasa pada saat request data ke server.
2. Responsif walaupun saat offline. *Database Realtime Firestore* dilengkapi SDK untuk menyimpan data ke disk local. Sehingga pada saat offline pengguna tetap dapat menyimpan data ke dalam aplikasi, dan akan melakukan update data setelah jaringan terhubung ke server. Server melakukan update data ke setiap perangkat yang terhubung secara otomatis.
3. *Firestore* dapat diakses langsung dari perangkat client. *Firestore* menyediakan fasilitas untuk melakukan akses langsung melalui perangkat selular atau dapat juga melalui web browser. Aturan keamanan *Firestore* akan melakukan validasi data yang dapat diakses pada saat data dibaca dan ditulis.
4. Dapat dibuat skala ukuran database. Pada paket tertentu yang berbayar, google sebagai *owner Firestore* memberikan keleluasaan bagi pengembang untuk membagi penyimpanan menjadi beberapa database sesuai dengan kebutuhan.

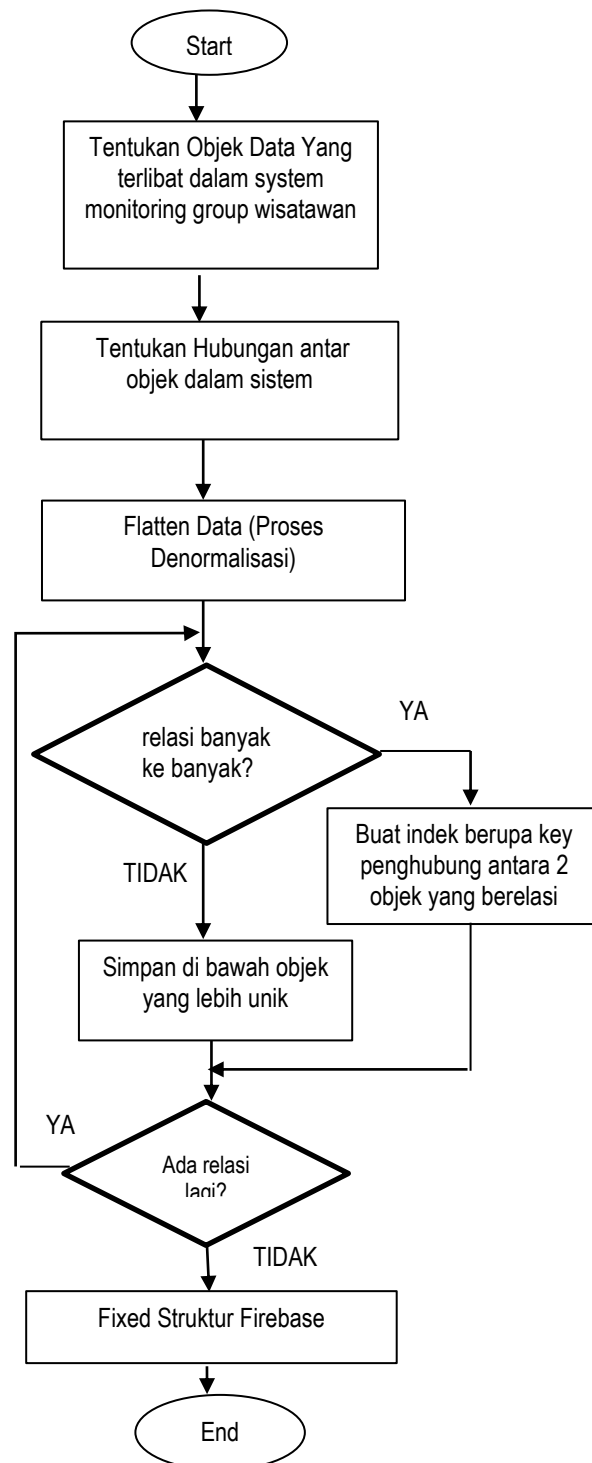
## 2. Methodologi

Semua data *Firestore Realtime Database* disimpan sebagai objek JSON. Anda dapat menganggap basis data sebagai pohon JSON yang dihosting cloud. Tidak seperti database SQL, tidak ada tabel atau record. Ketika anda menambahkan data ke pohon JSON, itu menjadi simpul dalam struktur JSON yang ada dengan kunci yang terkait[9]

Beberapa hal yang harus diperhatikan dalam membangun struktur database *Firestore* :

1. Hindari data bertingkat.  
*Firestore* memungkinkan developers membuat struktur data mencapai 32 tingkat. Pada saat aplikasi mengambil data pada suatu lokasi di database, maka *Firestore* akan mengambil semua node turunannya. Selain itu, pada saat kita memberikan akses baca tulis kepada seseorang dalam suatu node data, maka kita akan memberikan akses data ke semua data
2. Meratakan Struktur data pohon JSON.
3. Membangun data sesuai dengan skala

Tahapan proses pembentukan struktur data firebase dapat dilihat pada gambar 1.



**Gambar 1.** Proses Pembentukan Struktur Database *Firestore*

Pembentukan struktur Database *Firestore* diawali dengan menentukan objek – objek yang terlibat dalam sistem. Kemudian tentukan hubungan antar objek dalam sistem. Hubungan antar objek dalam sistem dapat sebagai hubungan satu ke satu, satu ke banyak dan hubungan banyak ke banyak. Selanjutnya dilakukan Denormalisasi untuk mendapatkan struktur database pohon JSON dengan satu node yang besar. Jika terdapat struktur pohon dengan cabang yang

bertingkat tingkat, dilakukan pengecekan antara satu node dengan node yang bersarang di dalamnya. Jika beberapa group data menjadi unik terhadap untuk objek tertentu, sarangkan group data tersebut dibawah objek yang unik tersebut. Jika satu group data memiliki hubungan banyak ke banyak maka keluarkan group data tersebut dari group data induknya, dan tempatkan group data tersebut pada root, kemudian buat indek berupa key penghubung antara grup data yang berelasi.

### 3. Hasil Dan Pembahasan

Aplikasi monitoring wisatawan ini melibatkan beberapa group data. Group Data yang terlibat dalam system diantaranya data guide dimana satu guide mewakili satu grup tour wisata. Kemudian data wisatawan yang bergabung dalam group wisata yang akan disebut sebagai guest. Guest dapat melakukan login ke salah satu group wisata dengan mengirimkan id berupa alamat email, nama dan posisi GPS perangkat mobilnya. Data GPS yaitu *latitude* dan *longitude* dikirim secara simultan ke server. Jika perangkat mobile berpindah, maka secara otomatis perangkat akan mengirimkan posisi terbaru dari perangkat mobilnya. Setiap satu group tour yang diwakili oleh satu guide dapat handle banyak guest. Satu guest dapat mengirimkan pesan ke satu guide sedangkan satu guide dapat mengirimkan pesan ke banyak guest untuk melakukan aktifitas chatting.

Guest dapat mengirim sinyal secara instan ke guide jika berada pada posisi kebingungan. Pada saat berada di lokasi wisata, seringkali para anggota tour group, terpecah-pecah menjadi group yang lebih kecil bahkan berjalan sendiri menikmati keindahan tempat wisata. Karena terlalu menikmati suasana, anggota group tour tersebut, jauh dari group induknya dan kebingungan untuk mencari tour leadernya. Pada kondisi ini, guest dapat menekan suatu tombol pada layar monitor perangkat mobile-nya, kemudian aplikasi akan mengirimkan pemberitahuan ke tour leader bahwa ada anggota group tour-nya yang kebingungan

Penyajian data pada layar monitor perangkat bergerak baik itu untuk guide dan guest, menjadi pertimbangan dalam membuat struktur database firebase. Ini karena database firebase berbasis dokumen, sehingga penyajian data ke guide maupun guest perlu dianalisa proses membaca dan menulis data ke dalam firebase. Penyajian data sangat menentukan dalam membuat struktur pohon JSON firebase. Dengan memperhatikan penyajian data proses membaca data dari database firebase dapat lebih dioptimalkan. Pembuatan struktur pohon JSON yang tidak terlalu dalam (flatten) akan mempercepat proses membaca data dari firebase.

Dari penjelasan diatas dapat dipetakan ada 3 data utama yaitu guide, guest dan data chatting. Guide terdiri dari beberapa guest. Chat terdiri dari guest dan guide karena aktifitas chat melibatkan guide dan guest untuk saling berkiriman pesan. Dalam melakukan chat pesan yang dikirim disimpan dalam Message.

Untuk lebih jelasnya digunakan data sebagai berikut. Data guide terdiri dari 2 objek dengan id sudiarta@pnb\_ac\_id dan eddy@pnb\_ac\_id. Guide sudiarta@pnb\_ac\_id mempunyai satu anggota group wisatawan(guest) yaitu michael\_xcm@gmail\_com sedangkan guide eddy@pnb\_ac\_id mempunyai 2 anggota group wisatawan(guest) yaitu chenlong@gmail\_com dan tanbingyi@gmail\_com. Anggota group dapat mengirimkan pesan ke guide sedangkan guide dapat mengirimkan pesan ke semua wisatawan yang menjadi anggota groupnya. Implementasi struktur data awal pada firebase dapat dilihat pada gambar 2.

```

{
  "Guide" : {
    "eddy@pnb_ac_id" : {
      "email" : "eddy@pnb.ac.id",
      "guests" : {
        "chenlong@gmail_com" : {
          "email" : "chenlong@gmail.com",
          "lat" : -8.63091395455793,
          "lon" : 115.21061208249706,
          "name" : "Chen Long",
          "Message" : {
            "message1" : {
              "fromGuest" : true,
              "message" : "Oh I am lost",
              "time" : "1525186045366"
            },
            "message2" : {
              "fromGuest" : false,
              "message" : "where are you?",
              "time" : "1532482150401"
            }
          }
        }
      },
      "tanbingyi@gmail_com" : {
        "email" : "tanbingyi@gmail.com",
        "Message" : {
          "message1" : {
            "fromGuest" : true,
            "message" : "Oh I am lost",
            "time" : "1525186045366"
          },
          "message2" : {
            "fromGuest" : false,
            "message" : "where are you?",
            "time" : "1532482150401"
          }
        }
      },
      "lat" : -33.868818333333334,
      "lon" : 151.209289999999998,
      "name" : "Tan Bing Yi"
    }
  },
  "lat" : -8.6307159,
  "lon" : 115.2104891,
  "name" : "Eddy Indrayana",
  "pin" : "123",
  "status" : true,
},
"sudiartha@pnb_ac_id" : {
  "email" : "itutde@pnb.ac.id",
  "guests" : {
    "michael_xcm@gmail_com" : {
      "email" : "michael_xcm@gmail.com",
      "lat" : -8.5683334,
      "lon" : 115.2948761,
      "name" : "Michael Arrow"
    }
  },
  "lat" : -8.7997022,
  "lon" : 115.1621794,
  "name" : "Ketut Sudiartha",
  "pin" : "123456",
  "status" : true
}
}
}

```

Gambar 2. Struktur Firebase Awal

```

{
  "ChatGroup" : {
    "eddy@pnb_ac_id" : {
      "Message" : {
        "message1" : {
          "fromGuest" : false,
          "message" : "Oh I am lost",
          "time" : "1525186045366",
        },
        "message2" : {
          "fromGuest" : true,
          "message" : "where are you?",
          "time" : "1532482150401"
        }
      }
    }
  },
  "Guide" : {
    "eddy@pnb_ac_id" : {
      "email" : "eddy@pnb.ac.id",
      "guests" : {
        "chenlong@gmail_com" : {
          "email" : "chenlong@gmail.com",
          "lat" : -8.63091395455793,
          "lon" : 115.21061208249706,
          "name" : "Chen Long",
        },
        "tanbingyi@gmail_com" : {
          "email" : "tanbingyi@gmail.com",
          "lat" : -33.868818333333334,
          "lon" : 151.209289999999998,
          "name" : "Tan Bing Yi"
        }
      },
      "lat" : -8.6307159,
      "lon" : 115.2104891,
      "name" : "Eddy Indrayana",
      "pin" : "123",
      "status" : true,
    },
    "sudiarta@pnb_ac_id" : {
      "email" : "itutde@pnb.ac.id",
      "guests" : {
        "michael_xcm@gmail_com" : {
          "email" : "michael_xcm@gmail.com",
          "lat" : -8.5683334,
          "lon" : 115.2948761,
          "name" : "Michael Arrow"
        }
      },
      "lat" : -8.7997022,
      "lon" : 115.1621794,
      "name" : "Ketut Sudiarta",
      "pin" : "123456",
      "status" : true
    }
  }
}

```

**Gambar 3.** Struktur Firebase Setelah Diratakan

Langkah berikutnya adalah meratakan struktur data firebase. Perhatikan gambar 2 diatas, bahwa Message diduplikasi pada setiap anggota member dalam satu grup wisatawan. Kita dapat meratakan struktur data firebase sistem dengan membuat suatu node sebagai root yaitu ChatGroup dimana setiap anggota dalam satu group wisatawan akan mendapatkan data yang sama. Berikutnya kita lihat relasi antar guide dan guest. Satu guide memiliki relasi satu ke

banyak dengan guest. Guide bertindak sebagai objek yang unik, sehingga kita bisa menempatkan data guest dibawah data guide(tidak ada perubahan). Sehingga struktur data firebase sistem menjadi seperti gambar 3.

#### 4. Kesimpulan

Membangun struktur database firebase untuk aplikasi monitoring wisatawan membutuhkan data guide, guest, chat dan Message untuk melakukan komunikasi antar guide dan guest. Database firebase mempunyai struktur data model document. Untuk retrieve data menggunakan pengurutan dan filtering. Untuk mendapatkan struktur model data firebase yang optimal perlu memperhatikan bagaimana data disimpan dan dipanggil lagi supaya proses yang melibatkan data tersebut menjadi lebih mudah. Pada aplikasi ini data data guide menjadi node JSON dan data guest menjadi sub node dari guide. Kemudian kita lakukan perataan struktur pohon JSON untuk data ChatGroup. Data ChatGroup menjadi node baru dan data guide menjadi sub node nya. Sedangkan di dalam sub node guide ada data Message.

#### Daftar Pustaka

- [1] Indrayana, I.N.E, Wirasyanti, N.M., Sudiarta, I G.K., "Heuristic Query Optimization for Query Multiple Table and Multiple Clause On Mobile Finance Application" . *IOP Conference Series: Journal Of Physics: Conf. Series* 953(2017)012049, 2017.
- [2] Khedkar, S., Thube, S., "Realtime Databases for Applications", *International Research Journal of Engineering and Technology*. Volume 4(6), 2017.
- [3] Haena Khan, "NoSQL: A Database for Cloud Computing" , *International Journal of Computer Science and Network*, Volume 3(6), 2014.
- [4] Srivastava, N., Shree, U., Chauhan, N.R, Tiwari D.K., "Firebase Cloud Messaging (Android)", *International Journal of Innovative Research in Science, Engineering and Technology*. Volume 6(9), 2017.
- [5] Almootassem, O., Husain, S.H, Parthipan, D., Mahmoud, Q.H., "A Cloud-based Service for Real-Time Performance Evaluation of NoSQL Databases". <https://arxiv.org/abs/1705.08317>, 2017.
- [6] Lahudkar, P., Sawale, S., Deshmane, V, Bharambe , K., "NoSQL Database – Google's Firebase: A Review", *International Journal of Innovative Research in Science, Engineering and Technology*, Volume 7(3), 2018.
- [7] Grolinger, K., Higashino, W.A., Tiwari, A., Capretz , M.A.M., "Data Management in Cloud Environments: NoSQL and New SQL Data Stores", *International Journal of Innovative Research in Science, Engineering and Technology*, doi :10.1186/2192-113X-2-22, 2013.
- [8] Soenaryo, S.S, Yulia, Palit, H.N., "Pembuatan Aplikasi Salesforce Automation System PadaToko IV Berbasis Android", *Jurnal Infra Universitas Petra*. Volume 5(1), 2017.
- [9] Google Developers, "Membuat Struktur Data Firebase", <https://firebase.google.com/docs/database/android/structure-data?hl=id>. unduh 7 mei 2018.