

# Rancang Bangun Aplikasi Antrian Online Rumah Sakit Berbasis Android Menggunakan Parse sebagai *Backend as a Service* (BaaS)

Kukuh Hannugroho, Radityo Anggoro, Isye Ariesanti  
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)  
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia  
*e-mail*: onngo@if.its.ac.id

**Abstrak**—Mengantri bagi sebagian orang adalah suatu aktifitas yang membosankan. Bahkan sebagian orang enggan untuk mengantri sehingga mereka keluar dari antrian sebelum mereka mendapatkan pelayanan. Bagi seorang pasien yang kondisi tubuhnya sedang tidak sehat, mengantri berjam-jam menambah ketidaknyamanan mereka. Waktu untuk mengantri akan jauh lebih efektif jika dipergunakan untuk kegiatan lainnya. Pada tugas akhir ini akan dibuat sistem antrian online pada rumah sakit dengan menggunakan Parse sebagai Backend as a Service (BaaS). Sistem antrian online akan memudahkan pasien dalam melakukan antrian pada rumah sakit. Pasien dapat mengambil antrian dari rumah dan berangkat ke rumah sakit saat gilirannya telah mendekati. BaaS merupakan model yang menjembatani pembuat aplikasi web dan mobile untuk menghubungkan aplikasi mereka dengan server. BaaS menyediakan fitur-fitur umum seperti manajemen pengguna, push notifications, dan integrasi dengan servis jaringan social. BaaS memudahkan developer dalam pengembangan aplikasi yang bersifat client-server. BaaS telah menyediakan infrastruktur utama pada server yang diperlukan oleh aplikasi secara umum. Sistem ini diharapkan dapat membantu masyarakat dalam melakukan antrian di rumah sakit. Masyarakat dapat mengambil nomor antrian melalui sistem ini dari mana saja. Mereka dapat mempergunakan waktu menunggu antrian dengan kegiatan lain yang lebih bermanfaat.

**Kata Kunci**—antrian, rumah sakit, parse, BaaS

## I. PENDAHULUAN

**M**ENGANTRI bagi sebagian orang adalah suatu aktifitas yang membosankan. Bagi seorang pasien yang kondisi tubuhnya sedang tidak sehat, mengantri dalam waktu lama menambah ketidaknyamanan mereka. Waktu yang dipergunakan untuk mengantri akan jauh lebih efektif jika dipergunakan untuk kegiatan lainnya.

Ada tiga tipe antrian yang digunakan dalam dunia nyata, yaitu antrian fisik, antrian virtual, dan antrian *mobile* [1]. Antrian fisik adalah antrian yang diwakilkan oleh pengantri itu sendiri dengan cara berdiri atau duduk berjajar. Antrian virtual merupakan perkembangan dari antrian fisik dimana pengantri diwakilkan dengan benda lain, biasanya berupa kertas bernomor. Kedua tipe antrian ini memiliki kelemahan yang sama yaitu pengantri harus datang ke lokasi dan menunggu antriannya. Untuk mengatasi hal ini maka muncullah tipe antrian yang ketiga yaitu antrian *mobile* dimana pengantri bisa

mengambil nomor antrian dari tempat lain tanpa perlu datang ke lokasi. Metode pengambilan nomor antrian dapat berupa melalui telepon dan internet. Antrian tipe ini masih jarang dijumpai di Indonesia. Padahal jumlah pengguna telepon pintar di Indonesia mengalami perkembangan yang sangat signifikan.

Sebuah system berbasis android diusulkan untuk dapat mengimplementasi antrian *online* rumah sakit. Aplikasi ini diharapkan dapat membantu pasien dalam mengambil nomor antrian dari mana saja dan memantau antrian yang ada. Pasien datang ke rumah sakit begitu nomor antrian sudah dekat sehingga pasien tidak perlu menunggu di rumah sakit terlalu lama. Aplikasi ini juga bermanfaat untuk meningkatkan kepuasan pasien terhadap pelayanan rumah sakit.

## II. TINJAUAN PUSTAKA

### A. *Backend as a Service* (BaaS)

BaaS merupakan cloud computing yang dibuat oleh suatu perusahaan untuk memudahkan developer dalam membangun dan mengoperasikan *cloud backend* untuk aplikasi mobile atau web mereka. BaaS menyediakan fitur-fitur umum yang diperlukan seperti manajemen pengguna, *push notification*, dan integrasi dengan *social networking services*. Fitur-fitur ini dapat diakses melalui *application programming interface* (API). Dengan adanya BaaS, developer tidak perlu membangun infrastruktur yang kompleks untuk sisi server aplikasi mereka. BaaS dapat membantu developer mempercepat proses pembuatan aplikasi [2].

### B. *Parse*

Parse adalah salah satu layanan BaaS yang telah diakuisisi oleh Facebook. Parse menyediakan solusi backend yang lengkap untuk aplikasi *mobile*. Dengan menggunakan parse kita tidak perlu melakukan pengkodean pada server atau memelihara server. Dalam menggunakan parse, developer harus mendaftar pada website parse dan mendefinisikan aplikasi yang dibuat. Parse akan memberikan application id dan client key yang dibutuhkan dalam menggunakan parse pada aplikasi [3]. Parse telah menyediakan komponen-komponen yang diperlukan dalam membuat aplikasi mobile antara lain penyimpanan data, manajemen pengguna, *push*

notification, dan cloud code.

C. Application Programming Interface (API)

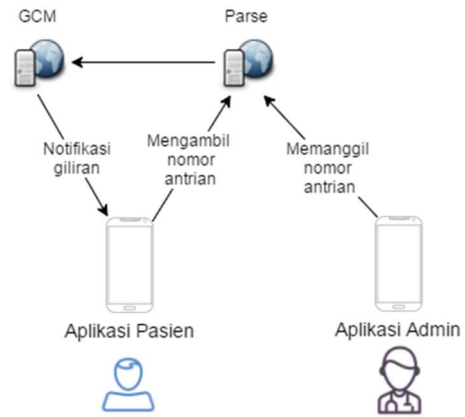
API merupakan sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu. Dalam Java, API dimasukkan ke dalam package-package yang sesuai dengan fungsinya [4].

D. Push Notification

Push notification merupakan pengiriman informasi dari suatu aplikasi server ke suatu perangkat client tanpa permintaan dari client tersebut. Hal ini berkebalikan dengan pull notification yang mengharuskan client untuk meminta informasi dari server. Keunggulan utama dari push notification dalam mobile computing adalah memungkinkan pesan untuk diterima meskipun aplikasi sedang tidak aktif [5].

E. Google Cloud Messaging (GCM)

Google Cloud Messaging (GCM) untuk Android adalah sebuah service yang mengizinkan adanya pengiriman data dari server ke pengguna Android yang compatible, dan menerima pesan dari perangkat yang memiliki koneksi yang sama dengan GCM service dengan melakukan antrian pesan dan dikirimkan ke target aplikasi Android yang sedang berjalan di perangkat target.



Gambar 1. Arsitektur Sistem

III. PERANCANGAN SISTEM

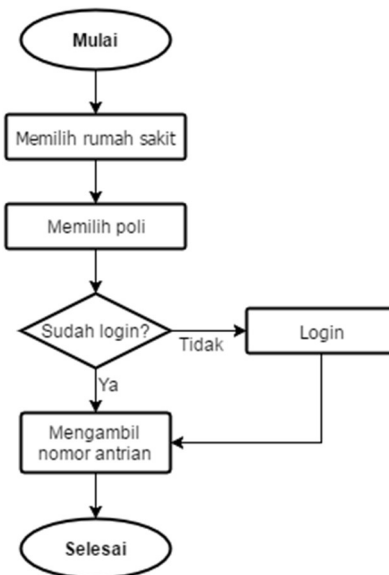
A. Arsitektur Sistem

Arsitektur sistem dibagi menjadi dua bagian, yakni pada client dan server. Modul client terdiri dari aplikasi android untuk pasien dan aplikasi android untuk admin rumah sakit. Sedangkan untuk server berupa Parse server dan Parse Cloud Code. Komunikasi antara client dan Parse server dilakukan melalui http dan push notification. Push notification untuk android memanfaatkan fasilitas Google Cloud Messaging.

Arsitektur sistem secara keseluruhan terlihat pada Gambar 1. Pasien dapat melihat nomor antrian yang sedang berjalan pada suatu rumah sakit melalui aplikasi. Selanjutnya pasien dapat mengambil nomor antrian pada poli tertentu. Pengambilan nomor antrian ini dilakukan melalui HTTP request. Pasien dapat mengetahui berapa banyak antrian saat ini hingga sampai gilirannya nanti. Jika telah mendekati gilirannya, maka sistem akan mengirimkan notifikasi melalui GCM kepada pasien tersebut. Admin rumah sakit dapat melihat berapa banyak total antrian yang ada dan nomor antrian saat itu. Selanjutnya admin dapat memanggil nomor antrian berikutnya. Pemanggilan nomor antrian juga dilakukan melalui HTTP request. Nomor antrian yang bersangkutan beserta beberapa nomor antrian selanjutnya akan mendapatkan notifikasi. Pasien dapat menunjukkan tampilan aplikasi sebagai bukti gilirannya.

B. Rancangan Proses pada Aplikasi Android untuk Pasien

Proses pada aplikasi android untuk pasien dimulai dengan memilih rumah sakit yang dikehendaki. Sistem akan memberikan daftar poli yang ada pada rumah sakit tersebut beserta status antriannya. Pasien selanjutnya memilih poli dari daftar poli ini. Jika pasien belum login ke dalam sistem, maka pasien diharuskan untuk login terlebih dahulu. Selanjutnya jika pasien belum mengambil nomor antrian pada poli tersebut, maka pasien dapat mengambil nomor antrian. Rancangan proses ini dapat dilihat pada Gambar 2.

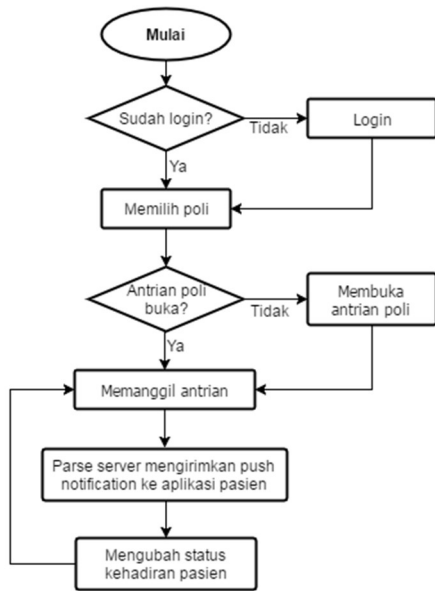


Gambar 2. Diagram Alir Proses Pengambilan Nomor Antrian oleh Pasien

C. Rancangan Proses pada Aplikasi Android untuk Admin Rumah Sakit

Proses pada aplikasi android untuk admin rumah sakit dimulai login ke dalam sistem. Sistem akan mencari rumah sakit dimana admin tersebut memiliki hak akses dan menampilkan daftar poli yang ada pada rumah sakit tersebut. Admin memilih salah satu poli. Bila antrian poli tersebut dalam keadaan tutup, maka admin dapat membuka antrian sehingga pasien dapat mengambil nomor di poli tersebut. Admin selanjutnya bisa memanggil antrian dan mengubah

status kehadiran pasien pada nomor tersebut. Bila mendekati waktu tutup poli, admin dapat mengubah status antrian menjadi *closed* sehingga pasien tidak dapat mengambil nomor antrian, sedangkan antrian yang tersisa masih dapat dipanggil. Rancangan proses ini dapat dilihat pada Gambar 3.



Gambar 3. Diagram Alir Proses Pemanggilan Nomor Antrian oleh Admin

#### IV. IMPLEMENTASI

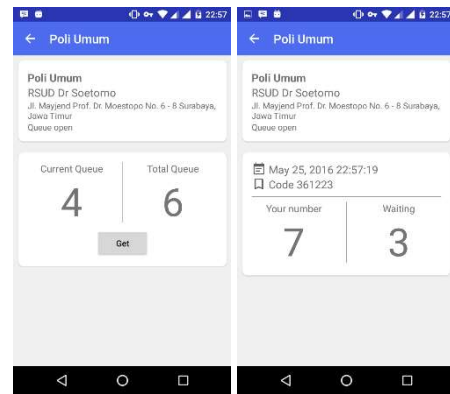
Implementasi sistem dibagi dalam 2 modul yaitu modul *server* dan modul *client*. Modul server terletak pada Parse. Pada modul server digunakan cloud code untuk melakukan komputasi yang kompleks sehingga pengaksesan ke Parse menjadi lebih cepat. Pada sistem antrian *online* ini terdapat 5 fungsi yang dilakukan di *cloud code* yaitu fungsi membuka antrian poli, fungsi menutup antrian poli, fungsi menambah antrian poli, fungsi memanggil antrian poli, dan fungsi mengubah status kehadiran nomor antrian. Modul *client* terletak pada aplikasi android. Aplikasi android yang dibuat ada 2 macam yaitu aplikasi bagi pasien dan aplikasi bagi admin rumah sakit.

#### V. UJI COBA DAN EVALUASI

Sistem diuji dalam 5 fungsionalitas utama yaitu pengambilan nomor oleh pasien, membuka antrian pada poli, pemanggilan nomor antrian disertai notifikasi di pasien, pengambilan nomor oleh admin, perubahan status kehadiran pasien, dan menutup antrian pada poli.

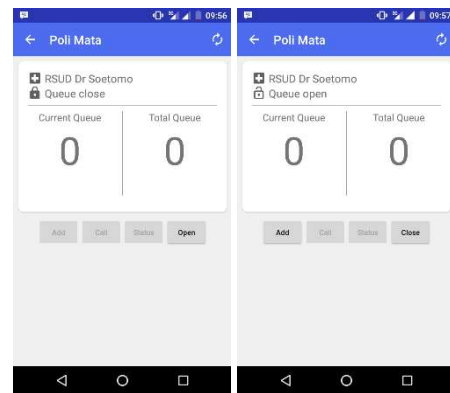
Pengujian pengambilan nomor antrian oleh pasien dilakukan dengan memilih rumah sakit dan poli. Sistem akan mengambil data antrian pada poli dan menampilkan pada halaman poli. Bila pasien belum mengambil nomor antrian maka akan ada tombol *Get* untuk mengambil nomor antrian. Dengan menekan tombol *Get* maka aplikasi akan memanggil fungsi *addQueue* di Parse Cloud Code dan mengembalikan data antrian yang telah diambil. Hasil data nomor antrian berupa nomor antrian, tanggal pengambilan, dan kode antrian

selanjutnya ditampilkan pada halaman poli. Tampilan poli setelah pengambilan nomor dapat dilihat pada Gambar 4.



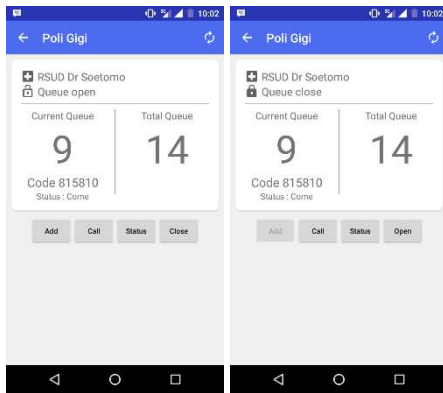
Gambar 4. Tampilan Poli Sebelum ) Pengambilan Nomor Antrian (Kondisi Awal) dan Sesudah Pengambilan Nomor Antrian (Hasil Uji)

Pengujian fungsionalitas pembuka antrian pada poli dilakukan di aplikasi admin. Admin harus sudah dalam keadaan login ke dalam aplikasi dan memiliki hak akses terhadap rumah sakit yang bersangkutan. Selanjutnya admin memilih poli dengan status *close*. Pada halaman poli, hanya tombol *Open* yang dapat ditekan karena status antrian adalah *close*. Admin menekan tombol *Open* untuk membuka antrian pada poli. Aplikasi akan memanggil fungsi *openQueue* pada Parse *Cloud Code* dan mengembalikan hasil berupa data poli dengan status *open*. Tampilan halaman poli sebelum dan sesudah pengujian dapat dilihat pada Gambar 5.



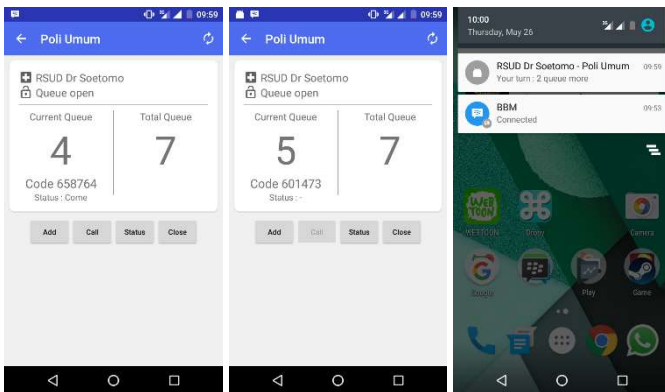
Gambar 5. Tampilan Poli dengan Status Antrian *Close* (Kondisi Awal) dan Status Antrian *Open* (Hasil Uji)

Pengujian fungsionalitas penutupan antrian pada poli dilakukan di aplikasi admin. Admin harus sudah dalam keadaan login ke dalam aplikasi dan memiliki hak akses terhadap rumah sakit yang bersangkutan. Admin memilih poli dengan status *open* dan menekan tombol *Close* untuk menutup antrian pada poli. Aplikasi akan memanggil fungsi *closeQueue* pada Parse *Cloud Code* dan mengembalikan hasil berupa data poli dengan status *close*. Pada antrian dengan status *close*, admin masih bisa melakukan panggilan nomor antrian selama masih ada nomor antrian pada poli tersebut. Tampilan halaman poli sebelum dan sesudah pengujian dapat dilihat pada Gambar 6.



Gambar 6. Tampilan Poli dengan Status Antrian *Open* (Kondisi Awal) dan Status Antrian *Close* setelah Pengujian Penutupan Antrian (Hasil Uji)

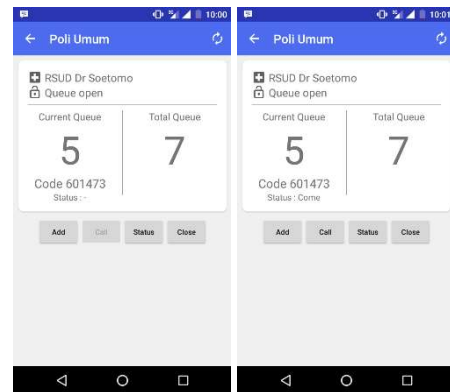
Pengujian fungsionalitas pemanggilan nomor antrian pada poli dilakukan di aplikasi admin. Admin harus sudah dalam keadaan login ke dalam aplikasi dan memiliki hak akses terhadap rumah sakit yang bersangkutan. Selanjutnya admin memilih poli yang memiliki nomor antrian yang belum terpanggil, yaitu dengan ditandai dengan nilai *totalQueue* lebih besar dari *currentQueue*. Admin menekan tombol *Call* untuk memanggil nomor antrian berikutnya, yaitu nomor antrian 5. Aplikasi akan memanggil fungsi *callQueue* pada Parse *Cloud Code* dan mengembalikan hasil berupa data poli dan data queue untuk nomor yang terpanggil. Data queue berupa nomor antrian dan kode antrian akan ditampilkan pada layar. Proses pemanggilan nomor antrian ini merupakan trigger pada Parse Server untuk mengirimkan notifikasi ke nomor antrian yang terpanggil. Fungsi *callQueue* juga melakukan proses pengiriman push notification ke nomor antrian yang terpanggil dan 4 nomor selanjutnya. Proses pengujian fungsionalitas ini dapat dilihat pada Gambar 7.



Gambar 7. Tampilan Poli yang Memiliki Nomor Antrian yang Belum Terpanggil (Kondisi Awal), Setelah Pemanggilan Nomor Antrian (Hasil Uji), dan Tampilan Notifikasi Giliran Antrian pada Pasien (Hasil Uji)

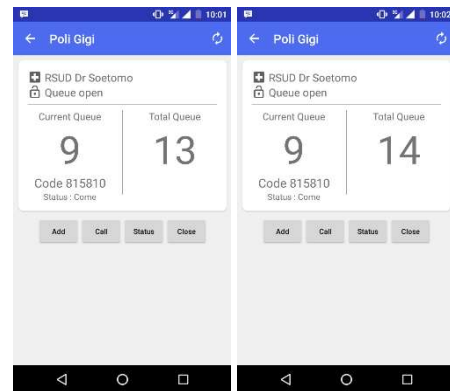
Pengujian fungsionalitas perubahan status kehadiran pasien dilakukan di aplikasi admin. Admin harus sudah dalam keadaan login ke dalam aplikasi dan memiliki hak akses terhadap rumah sakit yang bersangkutan. Selanjutnya admin memilih poli dan melakukan pemanggilan nomor antrian. Nomor antrian yang terpanggil belum memiliki status kehadiran. Dengan menekan tombol *Status*, admin diberikan 2

pilihan apakah pasien yang bersangkutan hadir atau tidak. Aplikasi akan memanggil fungsi *statusQueue* pada Parse *Cloud Code* dan mengembalikan hasil berupa data antrian sesuai status yang dipilih. Tampilan sebelum dan sesudah pengujian dapat dilihat pada Gambar 8.



Gambar 8. Tampilan Poli Setelah Pemanggilan Nomor Antrian (Kondisi Awal) dan Tampilan Poli Setelah Perubahan Status Kehadiran (Hasil Uji)

Dalam proses pengujian fungsionalitas pengambilan nomor antrian oleh admin, admin harus sudah dalam keadaan login ke dalam aplikasi dan memiliki hak akses terhadap rumah sakit yang bersangkutan. Selanjutnya admin memilih poli dengan status *open*. Dengan menekan tombol *Add* maka aplikasi akan memanggil fungsi *addQueue* di Parse *Cloud Code* dan mengembalikan data antrian yang telah diambil. Tampilan poli sebelum dan setelah pengambilan nomor dapat dilihat pada Gambar 9.



Gambar 9. Tampilan Poli Sebelum Pengambilan Nomor Antrian oleh Admin (Kondisi Awal) dan Setelah Pemanggilan Nomor Antrian oleh Admin (Hasil Uji)

## VI. KESIMPULAN/RINGKASAN

Dari hasil uji coba dan evaluasi terhadap sistem diperoleh kesimpulan antara lain sebagai berikut:

1. Sistem ini dapat memudahkan pasien dalam mengambil nomor antrian karena pasien tidak perlu datang ke rumah sakit untuk mengambil nomor antrian berdasarkan hasil uji coba fungsionalitas pengambilan nomor antrian oleh pasien.

2. Sistem ini berhasil melakukan pengambilan nomor antrian dan pemanggilan nomor antrian dengan akurat berdasarkan hasil uji coba fungsionalitas pemanggilan nomor antrian.

#### DAFTAR PUSTAKA

- [1] "Queue area - Wikipedia, the free encyclopedia," [Online]. Available: [https://en.wikipedia.org/wiki/Queue\\_area](https://en.wikipedia.org/wiki/Queue_area). [Accessed 13 September 2015].
- [2] K. Lane, "Backend as a Service (BaaS)," [Online]. Available: <http://baas.apievangelist.com/>.
- [3] "Android Developers Guide | Parse," [Online]. Available: <https://parse.com/docs/android/guide>. [Accessed 2 Juni 2016].
- [4] Y. Nugroho, March 2013. [Online]. Available: [http://kur2003.if.itb.ac.id/file/IF2281\\_Java\\_API.pdf](http://kur2003.if.itb.ac.id/file/IF2281_Java_API.pdf).
- [5] M. Rouse, March 2014. [Online]. Available: <http://searchmobilecomputing.techtarget.com/definition/push-notification>.