

# Desain Aplikasi Antrian Berbasis Android

Yosafat Setia Budi<sup>1</sup>, Henry Novianus Palit<sup>2</sup>, Yulia<sup>3</sup>

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) - 8417658

E-mail: yosafatsetiabudi@gmail.com, hnpalit@petra.ac.id, yulia@petra.ac.id,

## ABSTRAK

Menunggu adalah suatu hal yang membosankan bagi kebanyakan orang. Apalagi, jika seseorang menunggu dalam sebuah antrian yang sangat panjang. Hal tersebut memberikan efek jera karena mereka tidak mengetahui berapa lama lagi mereka harus menunggu hingga mendapatkan pelayanan.

Oleh karena itu, dibuatlah suatu aplikasi sistem antrian berbasis Android yang dapat memberikan informasi perkiraan waktu tunggu bagi Customer yang sedang berada dalam sebuah antrian, sehingga mereka dapat mengetahui estimasi waktu mereka akan dilayani. Aplikasi Android pada skripsi ini memiliki fitur menampilkan informasi antrian saat ini yang dibutuhkan oleh Customer, seperti nomor antrian milik Customer dan perkiraan waktu tunggu, nomor antrian yang dilayani saat ini, serta jumlah orang yang menunggu sebelum Customer tersebut. Informasi tiket Customer dikirimkan pada aplikasi Android dengan fitur *Google Cloud Messaging (push notification)*.

Dari hasil pengujian, sistem ini dapat menghasilkan perkiraan waktu tunggu sekitar 65,51% tanpa toleransi *over time* dan mencapai 67,95% dengan toleransi  $\pm 2$  menit. Hasil tersebut didapatkan dengan menggunakan histori data waktu kedatangan dan pelayanan yang dimiliki oleh sebuah instansi atau organisasi.

**Kata Kunci:** Sistem Antrian Berbasis Android, Perkiraan Waktu Tunggu, *Push Notification*, *Queueefficient*.

## ABSTRACT

*Waiting is a boring thing for most people. Moreover, if a person waiting in a very long queue. It provides a deterrent effect because they do not know how much longer they must to wait to get a service.*

*Therefore, they invented a queueing system Android-based application that can provide information on the estimated waiting time for the Customer who is in a queue, so that they can know time estimation when they will be served. Android app on this thesis has a feature displays the current queue information required by the Customer and the current estimation waiting time, the queue number being served now, as well as the number of people waiting before the Customer. Customer ticket information delivered to the Android application with a feature of Google Cloud Messaging (push notification).*

*From the test results, the system can generate the estimated waiting time of about 65.51% without tolerance over time and reached 67.95% with a tolerance of  $\pm 2$  minutes. The results obtained using historical data held by an agency or organization.*

**Keywords:** Android-based Queueing System, Waiting Time Estimation, *Push Notification*, *Queueefficient*.

## 1. PENDAHULUAN

Saat ini, banyak dijumpai sistem antrian yang digunakan oleh berbagai instansi atau perusahaan untuk melayani para *customer* mereka. Tidak sedikit instansi atau perusahaan yang menggunakan teknologi digital/komputerisasi dalam mengimplementasikan sistem antrian untuk memberi kemudahan dan kenyamanan kepada para *customer* mereka. Bahkan, sekarang sudah mulai jarang ditemui penggunaan sistem antrian manual (tanpa teknologi digital/komputerisasi) di kota-kota besar. Namun, sistem antrian yang mayoritas digunakan tersebut belum sepenuhnya efisien dan memberikan kepuasan bagi para *customer*. Hal ini disebabkan karena seorang *customer* tidak mengetahui berapa lama lagi ia harus bersabar untuk menunggu giliran pada sebuah antrian yang panjang. Belum lagi bila seorang *customer* memiliki keperluan lain di tempat yang berbeda, sehingga ia harus meninggalkan tempat tersebut karena antrian yang masih panjang. Jika nomornya sudah dipanggil dan ia tidak berada di tempat, maka orang tersebut wajib mengambil nomor antrian lagi dan harus menunggu lagi sampai nomor barunya dipanggil. Ia tidak pernah mengetahui informasi perkiraan waktu kapan nomornya dipanggil.

Meresponi masalah tersebut dan dengan memanfaatkan teknologi terkini, yaitu alat berupa *smartphone* yang umumnya dimiliki masyarakat, maka ditemukan sebuah ide yang dapat menjadi solusi bagi masalah tersebut sehingga dapat meningkatkan produktivitas bagi masyarakat. Dengan membuat sebuah aplikasi *smartphone* berupa sistem antrian untuk memberikan informasi mengenai posisi antrian yang berjalan secara *real-time* dan informasi perkiraan waktu tunggu yang akan dikirimkan melalui fitur *push notification* bagi para *customer* yang sedang berada dalam sebuah antrian. Sistem ini diharapkan dapat membantu kedua belah pihak, baik *server* (instansi atau perusahaan) maupun *customer* dalam menjalankan kegiatan rutin yang ada.

## 2. TINJAUAN PUSTAKA

### 2.1 Teori Probabilitas

Probabilitas adalah suatu nilai yang digunakan untuk mengukur tingkat terjadinya suatu kejadian yang acak. Kata probabilitas itu sendiri sering disebut dengan peluang atau kemungkinan. Probabilitas secara umum merupakan peluang bahwa sesuatu akan terjadi. Probabilitas biasanya dinyatakan dengan bilangan desimal (seperti 0,50, 0,20 atau 0,89) atau bilangan pecahan seperti 5/100, 20/100, 75/100. Nilai dari probabilitas berkisar antara 0 sampai dengan 1. Jika semakin dekat nilai probabilitas ke nilai 0, maka semakin kecil juga kemungkinan suatu kejadian akan terjadi. Jika semakin dekat nilai probabilitas ke nilai 1, maka semakin besar peluang suatu kejadian akan terjadi [12].

Terdapat dua prosedur penting untuk menentukan probabilitas dari suatu kejadian [10].

#### 1. Metode Klasik

Jika suatu kejadian dapat terjadi dalam  $h$  cara yang berbeda dari total  $n$  cara yang mungkin, maka probabilitas dari kejadian tersebut adalah  $h/n$ .

Contoh: Probabilitas munculnya kepala ketika sebuah koin dilemparkan sekali. Koin mempunyai dua kemungkinan sama besar, yaitu kepala dan ekor (dengan asumsi koin ideal, koin tidak menggelinding atau berdiri tegak). Maka probabilitas dari munculnya kepala adalah satu cara dari dua cara tersebut adalah  $1/2$ .

## 2. Metode Frekuensi

Jika setelah suatu eksperimen diulang  $n$  kali, dimana  $n$  sangat besar, terlihat bahwa suatu kejadian terjadi sebanyak  $h$  kali, maka probabilitas dari kejadian tersebut adalah  $h/n$ . Ini juga disebut sebagai probabilitas empiris dari kejadian tersebut.

Contoh: Jika kita melempar sebuah koin sebanyak 1000 kali dan kepala muncul sebanyak 532 kali, maka probabilitas kemunculan kepala adalah  $532/1000 = 0,532$ .

## 2.2 Distribusi Poisson

Distribusi probabilitas Poisson menjelaskan berapa kali sebuah kejadian terjadi selama interval tertentu. Interval tersebut dapat berupa waktu, jarak, luas, atau volume. Distribusi probabilitas Poisson memiliki tiga ciri-ciri [4]:

1. Variabel acaknya adalah berapa banyak sebuah kejadian terjadi selama interval yang ditentukan.
2. Probabilitas kejadian tersebut proporsional dengan ukuran interval.
3. Tidak ada pengulangan interval dan interval-interval lainnya saling bebas.

Distribusi Poisson dapat dijelaskan secara matematis dengan rumus:

$$P(x) = \frac{\mu^x e^{-\mu}}{x!}, x = 0, 1, 2, \dots \quad (1)$$

$e$  = konstanta, yaitu 2,71828 (basis dari sistem logaritmis Napier)

$\mu$  = nilai rata-rata dari kejadian (sukses) dalam suatu interval

$x$  = jumlah kejadian (sukses)

$P(x)$  = probabilitas untuk sebuah nilai  $x$  tertentu

## 2.3 Teori Antrian

Teori antrian dikembangkan dan dikembangkan oleh AK. Erlang, seorang insinyur Denmark, pada tahun 1910 [3]. Disiplin antrian tampak pada pelanggan (entah barang ataupun orang) akan dilayani berdasarkan yang lebih dahulu datang. Pelanggan dapat datang dalam jarak waktu kedatangan yang sama atau dapat pula secara random, dengan jarak waktu kedatangan yang tidak sama. Keadaan yang kedua ini lazim terjadi dan banyak dijumpai.

Sistem antrian adalah suatu himpunan pelanggan, pelayan, dan suatu aturan yang mengatur pelayanan kepada pelanggan. Sedangkan keadaan sistem menunjuk pada jumlah pelanggan yang berada dalam suatu fasilitas pelayanan, termasuk dalam antriannya. Salah satu populasi adalah jumlah pelanggan yang datang pada fasilitas pelayanan. Besarnya populasi merupakan jumlah pelanggan yang memerlukan pelayanan. Sistem antrian memiliki beberapa komponen seperti berikut [3]:

1. Populasi masukan (*input population*)  
Berapa banyak pelanggan potensial yang dapat memasuki sistem antrian, dibagi menjadi dua, yaitu:
  - Populasi input terbatas, yaitu jumlah kedatangan pelanggan potensial terbatas.
  - Populasi input tak terbatas, yaitu jumlah kedatangan pelanggan potensial tak terbatas.

## 2. Bentuk kedatangan

Bentuk kedatangan para pelanggan biasanya diperhitungkan melalui waktu antarkedatangan, yaitu waktu antara kedatangan dua pelanggan yang berurutan pada suatu fasilitas pelayanan. Bentuk ini dapat bergantung pada jumlah pelanggan yang berada dalam sistem ataupun tidak bergantung pada keadaan sistem tersebut. Bila bentuk kedatangan ini tidak disebut secara khusus, maka dianggap bahwa pelanggan tiba satu per satu (*constant arrival distribution*). Asumsinya ialah kedatangan pelanggan mengikuti suatu proses dengan distribusi probabilitas tertentu. Distribusi probabilitas yang paling sering digunakan adalah distribusi Poisson, dimana kedatangan bersifat bebas, tidak terpengaruh oleh kedatangan sebelum ataupun sesudahnya. Asumsi distribusi Poisson menunjukkan bahwa kedatangan pelanggan sifatnya acak (*arrival pattern random*) dan mempunyai rata-rata kedatangan sebesar  $\lambda$  ( $\lambda$ ). Dengan demikian, terdapat dua pola kedatangan (*arrival pattern*) yaitu : (1) menggambarkan tingkat kedatangan per unit waktu, atau (2) menggambarkan jumlah kedatangan dalam periode waktu tertentu secara berturut-turut dalam waktu yang berbeda.

## 3. Bentuk pelayanan

Bentuk pelayanan ditentukan oleh waktu pelayanan, yaitu waktu yang dibutuhkan untuk melayani pelanggan pada fasilitas pelayanan. Besaran ini dapat bergantung pada jumlah pelanggan yang telah berada di dalam fasilitas pelayanan ataupun tidak bergantung pada keadaan tersebut.

Pada suatu fasilitas pelayanan, pelanggan akan masuk dalam suatu tempat pelayanan dan menerima pelayanan secara tuntas oleh *server* (fasilitas pelayanan). Bila tidak disebutkan secara khusus pada bentuk pelayanan ini, maka dianggap bahwa satu pelayanan dapat melayani secara tuntas satu pelanggan. Bentuk pelayanan dapat konstan dari waktu ke waktu. Rata-rata pelayanan (*mean serving rate*) diberi simbol  $\mu$  ( $\mu$ ) merupakan jumlah pelanggan yang dapat dilayani dalam satuan waktu, sedangkan waktu rata-rata yang dipergunakan untuk melayani setiap pelanggan diberi simbol  $1/\mu$  unit (satuan). Jadi  $1/\mu$  adalah waktu rata-rata yang dibutuhkan untuk suatu pelayanan.

## 4. Fasilitas pelayanan

Fasilitas pelayanan berkaitan erat dengan baris antrian yang akan dibentuk. Desain fasilitas pelayanan ini dapat dibagi dalam tiga bentuk, yaitu:

- a. Bentuk seri, dalam satu garis lurus ataupun garis melingkar.
- b. Bentuk paralel, dalam beberapa garis lurus antara satu dengan yang lain paralel.
- c. Bentuk *network station*, yang dapat didesain secara seri dengan pelayanan lebih dari satu pada setiap stasiun. Bentuk ini dapat juga dilakukan secara paralel dengan stasiun yang berbeda-beda.

Fasilitas pelayanan dikelompokkan berdasarkan jumlah yang tersedia, yaitu:

- a. Fasilitas pelayanan tunggal (*single-channel/server*), hanya satu fasilitas pelayanan.
- b. Fasilitas pelayanan majemuk (*multiple-channel/server*), terdapat beberapa fasilitas pelayanan.

## 5. Distribusi pelayanan

Distribusi pelayanan dapat ditetapkan berdasarkan salah satu dari dua cara berikut:

- a. Berapa banyak pelanggan yang dapat dilayani per satuan waktu, atau

b. Berapa lama pelanggan dapat dilayani.

Proses pelayanan pada umumnya menggunakan distribusi peluang tertentu, seperti distribusi eksponensial negatif ataupun menggunakan parameter distribusi Poisson.

#### 6. Disiplin antrian

Disiplin antrian adalah aturan dimana para pelanggan dilayani atau disiplin pelayanan (*service discipline*) yang memuat urutan (*order*) para pelanggan menerima layanan. Aturan pelayanan menurut urutan kedatangan ini dapat didasarkan pada:

- FIFO (First In First Out)** atau **FCFS (First Come First Served)**, dimana pelanggan yang akan dilayani terlebih dahulu adalah pelanggan yang datang lebih dahulu. Contohnya pada antrian di loket penjualan karcis kereta api.
- LIFO (Last In First Out)** atau **LCFS (Last Come First Served)**, dimana pelanggan yang terakhir datang adalah pelanggan yang dilayani pertama/lebih dahulu. Contohnya pada sistem bongkar muat barang dalam truk, dimana barang yang terakhir masuk justru akan keluar terlebih dahulu.
- SIRO (Service In Random Order)**, dimana pelayanan dilakukan secara acak. Sering juga dikenal dengan **RSS (Random Selection for Service)**. Contohnya pada arisan, dimana pelayanan dilakukan berdasarkan undian (*random*).
- Pelayanan berdasarkan prioritas, dimana pelayanan didasarkan pada prioritas khusus. Misalnya, dalam suatu pesta, tamu-tamu yang dikategorikan VIP akan dilayani lebih dahulu.

#### 7. Kapasitas sistem pelayanan

Kapasitas sistem pelayanan adalah jumlah maksimum pelanggan, mencakup yang sedang dilayani dan yang berada dalam antrian, yang dapat ditampung oleh fasilitas pelayanan pada saat yang sama. Sebuah sistem yang tidak membatasi jumlah pelanggan di dalam fasilitas pelayanannya dikatakan memiliki kapasitas tak terhingga, sedangkan suatu sistem yang membatasi jumlah pelanggan yang ada dalam fasilitas pelayanannya dikatakan memiliki kapasitas yang terbatas.

#### 8. Karakteristik sistem lainnya

Dalam praktek sistem antrian, mungkin pelanggan tidak akan memasuki sistem antrian jika mengetahui sudah banyak pelanggan yang menunggu, sehingga pelanggan meninggalkan antrian dan tidak kembali. Ada juga pelanggan yang berpindah-pindah dari satu antrian ke antrian lainnya karena ingin cepat dilayani.

Keterangan variabel [2]:

$n$  = Jumlah pelanggan dalam sistem

$P_n$  = Probabilitas kepastian  $n$  pelanggan dalam sistem

$\lambda$  = Jumlah rata-rata pelanggan yang datang per satuan waktu

$\mu$  = Jumlah rata-rata pelanggan yang dilayani per satuan waktu

$P_0$  = Probabilitas tidak ada pelanggan dalam sistem

$P$  = Tingkat intensitas fasilitas pelanggan

$L$  = Jumlah rata-rata pelanggan yang diharapkan dalam sistem

$L_q$  = Jumlah pelanggan yang diharapkan menunggu dalam antrian

$W$  = Waktu yang diharapkan oleh pelanggan selama dalam sistem

$W_q$  = Waktu yang diharapkan pelanggan selama menunggu dalam antrian

$1/\mu$  = Waktu rata-rata pelayanan

$1/\lambda$  = Waktu rata-rata antar kedatangan

$s$  = Jumlah fasilitas pelayanan

## 2.4 Multiple Channel Model (M/M/s)

Salah satu model paling sederhana adalah model saluran tunggal (*multiple-channel model*) yang ditulis dengan notasi “sistem M/M/s”. Komponen dari sistem ini adalah [2]:

- Populasi input tak terbatas.
- Distribusi kedatangan pelanggan potensial mengikuti distribusi Poisson. Rata-rata kedatangan pelanggan per satuan waktu adalah variabel *random* suatu distribusi probabilitas Poisson. Dalam notasi (M/M/s), tanda M pertama menunjukkan rata-rata kedatangan yang mengikuti distribusi probabilitas Poisson. Sedangkan arti M kedua adalah tingkat pelayanan yang mengikuti distribusi probabilitas Poisson. Huruf (s) yang terdapat pada (M/M/s) menyatakan jumlah fasilitas pelayanan.
- Disiplin pelayanan mengikuti pedoman FCFS.
- Fasilitas pelayanan terdiri dari saluran tunggal.
- Distribusi pelayanan mengikuti distribusi Poisson. Diasumsikan bahwa  $\lambda$  lebih kecil dari  $\mu$  ( $\lambda < \mu$ ) yaitu rata-rata jumlah kedatangan pelanggan per satuan waktu lebih kecil dari rata-rata jumlah pelanggan yang dapat dilayani per satuan waktu dalam sistem.
- Kapasitas sistem diasumsikan tak terbatas.
- Tidak ada penolakan maupun pengingkaran.

Persamaan yang digunakan dalam sistem (M/M/s) adalah sebagai berikut [2]:

$\mu$  = rata-rata tingkat pelayanan untuk setiap fasilitas pelayanan

$$\rho = \frac{\lambda}{\mu s} \quad (2)$$

$$\lambda < \mu s \text{ atau } (\rho < 1) \quad (3)$$

Dalam setiap sistem antrian distribusi Poisson, akan selalu terjadi hubungan yang berkait antara  $L$ ,  $L_q$ ,  $W$  dan  $W_q$ . Hubungan tersebut dapat dinyatakan dalam persamaan berikut [2]:

$$L = \lambda W \quad (4)$$

$$L_q = \lambda W_q \quad (5)$$

$$W = W_q + \frac{1}{\mu} \quad (6)$$

jika  $\lambda \geq \mu s$ , maka jumlah pelanggan dalam sistem akan menjadi tak terbatas. Oleh karena itu, jika kita berpegang pada persamaan (2) maka persamaan yang digunakan dalam sistem (M/M/s) adalah sebagai berikut [2]:

$$P_0 = \left[ \sum_{n=0}^{s-1} \frac{(\lambda/\mu)^n}{n!} + \frac{(\lambda/\mu)^s}{s!} \frac{1}{1-(\lambda/s\mu)} \right]^{-1} \quad (7)$$

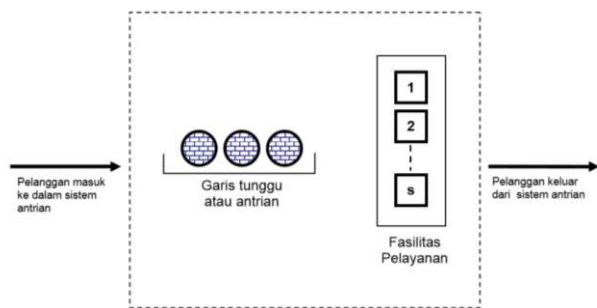
$$L_q = \frac{P_0 (\lambda/\mu)^s \rho}{s!(1-\rho)^2} \quad (8)$$

$$W_q = \frac{L_q}{\lambda} \quad (9)$$

$$W = W_q + \frac{1}{\mu} \quad (10)$$

$$L = \lambda W = L_q + \frac{\lambda}{\mu} \quad (11)$$

Gambar 1 menunjukkan struktur umum dari model antrian yang memiliki dua komponen utama yaitu : (1) Garis tunggu atau sering disebut antrian (*queue*), dan (2) Fasilitas pelayanan (*service facility*). Pelanggan atau konsumen menunggu untuk mendapatkan jasa pelayanan. Setiap pelanggan menunggu giliran untuk memasuki fasilitas pelayanan, menerima pelayanan, dan akhirnya keluar dari sistem pelayanan.



Gambar 1. Sistem Antrian Multiple Channel

## 2.5 PHP

PHP adalah bahasa pemrograman *open source* yang paling banyak digunakan untuk membangun *website*. PHP disediakan secara gratis dan memiliki komunitas yang mendukung pengembangannya.

PHP dapat digunakan dalam tiga cara utama [13]:

- *Server-side scripting*  
PHP pada awalnya dirancang untuk membuat konten web yang dinamis, dan masih paling cocok untuk tugas tersebut. Untuk menghasilkan HTML, diperlukan PHP *parser* dan *server* web yang akan digunakan untuk mengirim dokumen berkode. PHP juga telah menjadi populer untuk menghasilkan dokumen XML, grafis, animasi Flash, file PDF, dan banyak lagi.
- *Command-line scripting*  
PHP dapat menjalankan *script* dari *command-line*, seperti Perl, awk, atau Unix shell. Biasanya menggunakan *script* baris perintah untuk tugas-tugas administrasi sistem, seperti *backup* dan *log parsing*; bahkan beberapa *CRON job type script* dapat dilakukan dengan cara ini (tugas PHP nonvisual).
- *Client-side GUI applications*  
Menggunakan PHP-GTK, kita dapat menulis aplikasi *cross-platform GUI* di PHP.

## 2.6 MySQL

MySQL adalah *Relational Database Management System* (RDBMS) yang bersifat *open source*. *MySQL server* dapat mengelola banyak *database* pada waktu yang sama. Bahkan, banyak orang mungkin memiliki *database* berbeda yang dikelola oleh server tunggal MySQL. Setiap *database* terdiri dari struktur untuk menyimpan data dan data itu sendiri. MySQL merupakan *database* yang paling populer untuk digunakan dalam *website*, dikembangkan untuk menjadi cepat dan berukuran kecil, khusus untuk situs web. MySQL sangat populer untuk digunakan dengan situs web yang ditulis dalam PHP, perlu diketahui bahwa PHP dan MySQL dapat bekerjasama dengan baik [11].

## 2.7 AJAX

Awal dari Ajax seperti yang digunakan saat ini dimulai dengan rilis dari Internet Explorer 5 pada tahun 1999, yang memperkenalkan objek *ActiveX* baru, yaitu *XMLHttpRequest*. *ActiveX* adalah teknologi Microsoft untuk menandatangani *plug-in* yang menambahkan perangkat lunak tambahan untuk komputer pengguna. pengembang browser lainnya kemudian mengikuti, tapi daripada menggunakan *ActiveX*, mereka semua menerapkan fitur sebagai bagian asli dari interpreter JavaScript. Namun, bahkan sebelum itu, bentuk awal dari Ajax sudah muncul yang

menggunakan frame tersembunyi pada halaman yang berinteraksi dengan *server* di latar belakang. *Chat room* mengadopsi awal teknologi ini, menggunakannya untuk polling dan menampilkan posting pesan baru tanpa memerlukan *page reload* [5].

## 2.8 jQuery

jQuery adalah sebuah library yang dibangun dengan menggunakan JavaScript untuk mengotomasi dan menyederhanakan perintah-perintah umum. Meskipun ada banyak library lain semacamnya, namun jQuery jauh lebih populer karena kemampuannya untuk menjalankan perintah pada peramban lama. jQuery berjalan pada *browser* bersamaan dengan JavaScript biasa, terutama dipergunakan untuk animasi dan AJAX, yang cukup sulit untuk diprogramkan dengan vanilla JavaScript, namun bisa diketik dalam beberapa baris singkat dengan jQuery [9].

Dengan jQuery, kita tidak hanya mendapatkan tingkat kompatibilitas *cross-browser* yang sangat tinggi (ya, bahkan termasuk Internet Explorer), namun kita juga memiliki akses cepat dan mudah untuk manipulasi HTML dan DOM, fungsi khusus untuk berinteraksi langsung dengan CSS, kemampuan untuk *control events*, sebuah tools yang kuat untuk menciptakan efek profesional dan animasi, dan fungsi untuk melakukan komunikasi Ajax dengan *web server*. jQuery juga merupakan dasar untuk berbagai *plug-in* dan utilitas lainnya juga [6].

## 2.9 Google Cloud Messaging

Google Cloud Messaging (GCM) adalah layanan gratis yang memungkinkan *developers* untuk mengirim pesan antara server dan aplikasi klien. Ini termasuk *downstream messages* dari server ke aplikasi klien, dan *upstream messages* dari aplikasi klien ke server. Misalnya, *downstream messages* yang ringan bisa menginformasikan aplikasi klien bahwa ada data baru yang akan diambil dari *server*, seperti dalam kasus notifikasi "new email". Untuk kasus penggunaan seperti *instant messaging*, pesan GCM dapat mentransfer hingga 4KB *payload* ke aplikasi klien. Layanan GCM menangani semua aspek antrian pesan dan pengiriman ke dan dari aplikasi sasaran klien. Tiga komponen dalam interaksi GCM, yaitu GCM *Connection Servers*, *App Server*, *Client App*. Berikut proses yang terjadi ketika aplikasi server mengirim pesan ke aplikasi klien [1]:

- *App Server* mengirim sebuah pesan ke GCM *Connection Servers*.
- *Connection Servers* akan mengantrikan dan menyimpan pesan jika *device* dalam keadaan *offline*.
- Ketika *device* terhubung (*online*), server koneksi GCM mengirimkan pesan ke *device*.
- Pada *device*, aplikasi klien menerima pesan sesuai implementasi spesifik *platform*.

## 2.10 Push Notification

*Push Notification*, juga disebut *Server Push Notification*, adalah penyampaian informasi dari aplikasi perangkat lunak untuk sebuah perangkat komputasi tanpa permintaan khusus dari klien. Keuntungan penting dari *push notification* dalam komputasi *mobile* adalah bahwa teknologi tersebut tidak memerlukan aplikasi khusus yang terbuka atau aktif pada sebuah perangkat *mobile* agar pesan dapat diterima. Hal ini memungkinkan *smartphone* untuk menerima dan menampilkan media sosial atau peringatan pesan teks, bahkan ketika layar perangkat terkunci dan media sosial yang mengirimkan notifikasi dalam kondisi tertutup sekalipun [8].

Berdasarkan teori pada Bab 2,  $L_q$  adalah jumlah pelanggan yang diharapkan menunggu dalam antrian dan  $W_q$  merupakan waktu yang diharapkan pelanggan selama menunggu dalam antrian. Ada beberapa variabel yang dibutuhkan untuk mendapatkan nilai  $W_q$ , yaitu  $\lambda$  yang merupakan nilai *arrival rate*,  $\mu$  yang merupakan nilai

servicing rate, dan  $s$  yang merupakan jumlah *server/counter/CS* yang aktif melayani.

Formula 1 (Persamaan 5)

$$L_q = \lambda W_q$$

Formula 2

$$L_q = \lambda_{\text{dynamic}} W_q \quad (12)$$

$\lambda_{\text{dynamic}}$  merupakan  $\lambda$  yang didapatkan dari *Arrival Time* 1 jam terakhir.

Formula 3 (berdasarkan *counter/server/CS*)

$$\begin{aligned} \text{Waktu tunggu} &= \frac{\text{jumlah menunggu}}{s} \times \frac{1}{\mu} \times 60 \text{ menit} \\ &= \frac{\text{jumlah menunggu}}{\mu s} \times 60 \text{ menit} \end{aligned} \quad (13)$$

Berdasarkan observasi yang telah dilakukan, didapatkan fakta bahwa *Customer* menunggu dalam antrian menggunakan nomor tiket. Jumlah menunggu adalah jumlah tiket yang menunggu dilayani sebelum nomor tiket milik seorang *Customer*. Diasumsikan bahwa saat ini semua CS sedang melayani *Customer*, jadi dapat dipastikan tidak ada loket yang kosong tanpa *Customer* dan setiap CS memiliki kemampuan melayani yang sama.  $1/\mu$  merupakan waktu rata-rata pelayanan per CS per pelanggan.

Waktu tunggu didapatkan dari jumlah menunggu dibandingkan dengan jumlah *server/counter* yang aktif, kemudian dikalikan dengan kemampuan CS melayani *Customer* dalam 1 jam (60 menit).

Formula 4 (berdasarkan jumlah kedatangan)

$$\begin{aligned} \text{Waktu tunggu} &= \frac{\text{jumlah menunggu}}{L_q} \times W_q \times 60 \text{ menit} \\ &= \frac{\text{jumlah menunggu}}{\lambda} \times 60 \text{ menit} \end{aligned} \quad (14)$$

jumlah menunggu ( $q_{\text{length}}$ ) merupakan jumlah tiket yang belum dilayani di depan nomor tiket *Customer* bersangkutan.

$L_q$  merupakan rata-rata banyaknya pelanggan menunggu dalam antrian selama  $W_q$ . Jumlah menunggu dibandingkan dengan  $L_q$  kemudian dikalikan dengan  $W_q$  dan 60 menit (untuk mendapatkan hasil dalam satuan menit).

## 4. HASIL

Tabel 1 merupakan hasil simulasi pengujian formula perkiraan waktu tunggu.

**Tabel 1. Tabel Sebagian Hasil Simulasi Formula**

| Q_<br>len | R  | F1   | Diff_<br>F1 | F2   | Diff_<br>F2 | F3    | Diff_<br>F3 | F4    | Diff_<br>F4 |
|-----------|----|------|-------------|------|-------------|-------|-------------|-------|-------------|
| 8         | 18 | 8.43 | -9.57       | 0.02 | -17.98      | 11.25 | -6.75       | 12.64 | -5.36       |
| 9         | 18 | 8.43 | -9.57       | 0.03 | -17.97      | 12.66 | -5.34       | 14.22 | -3.78       |
| 9         | 18 | 8.43 | -9.57       | 0.04 | -17.96      | 12.66 | -5.34       | 14.22 | -3.78       |
| 9         | 18 | 8.43 | -9.57       | 0.06 | -17.94      | 12.66 | -5.34       | 14.22 | -3.78       |
| 10        | 18 | 8.43 | -9.57       | 0.09 | -17.91      | 14.06 | -3.94       | 15.8  | -2.2        |
| 11        | 18 | 8.43 | -9.57       | 0.12 | -17.88      | 15.47 | -2.53       | 17.38 | -0.62       |

**Tabel 1. Tabel Sebagian Hasil Simulasi Formula (Lanjutan)**

| Q_<br>len | R  | F1   | Diff_<br>F1 | F2   | Diff_<br>F2 | F3    | Diff_<br>F3 | F4    | Diff_<br>F4 |
|-----------|----|------|-------------|------|-------------|-------|-------------|-------|-------------|
| 12        | 20 | 8.43 | -11.57      | 0.21 | -19.79      | 16.88 | -3.12       | 18.96 | -1.04       |
| 13        | 27 | 8.43 | -18.57      | 0.21 | -26.79      | 18.28 | -8.72       | 20.54 | -6.46       |
| 13        | 23 | 2.62 | -20.38      | 0.1  | -22.9       | 16.46 | -6.54       | 20.54 | -2.46       |
| 7         | 18 | 2.62 | -15.38      | 0.13 | -17.87      | 8.86  | -9.14       | 11.06 | -6.94       |
| 8         | 14 | 2.62 | -11.38      | 0.17 | -13.83      | 10.13 | -3.87       | 12.64 | -1.36       |
| 8         | 15 | 2.62 | -12.38      | 0.29 | -14.71      | 10.13 | -4.87       | 12.64 | -2.36       |
| 9         | 16 | 2.62 | -13.38      | 0.29 | -15.71      | 11.39 | -4.61       | 14.22 | -1.78       |
| 9         | 16 | 2.62 | -13.38      | 0.37 | -15.63      | 11.39 | -4.61       | 14.22 | -1.78       |
| 9         | 16 | 2.62 | -13.38      | 0.46 | -15.54      | 11.39 | -4.61       | 14.22 | -1.78       |
| 9         | 19 | 2.62 | -16.38      | 0.58 | -18.42      | 11.39 | -7.61       | 14.22 | -4.78       |
| 9         | 17 | 2.62 | -14.38      | 0.72 | -16.28      | 11.39 | -5.61       | 14.22 | -2.78       |
| 10        | 14 | 2.62 | -11.38      | 1.38 | -12.62      | 12.66 | -1.34       | 15.8  | 1.8         |
| 11        | 15 | 2.62 | -12.38      | 1.38 | -13.62      | 13.92 | -1.08       | 17.38 | 2.38        |
| 12        | 17 | 2.62 | -14.38      | 1.38 | -15.62      | 15.19 | -1.81       | 18.96 | 1.96        |
| 13        | 17 | 2.62 | -14.38      | 2.12 | -14.88      | 16.46 | -0.54       | 20.54 | 3.54        |
| 14        | 17 | 2.62 | -14.38      | 2.12 | -14.88      | 17.72 | 0.72        | 22.12 | 5.12        |
| 11        | 23 | 2.62 | -20.38      | 2.64 | -20.36      | 13.92 | -9.08       | 17.38 | -5.62       |
| 8         | 19 | 2.62 | -16.38      | 0.9  | -18.1       | 10.13 | -8.87       | 12.64 | -6.36       |
| 9         | 18 | 2.62 | -15.38      | 1.12 | -16.88      | 11.39 | -6.61       | 14.22 | -3.78       |
| 10        | 23 | 2.62 | -20.38      | 1.38 | -21.62      | 12.66 | -10.34      | 15.8  | -7.2        |
| 9         | 22 | 2.62 | -19.38      | 1.71 | -20.29      | 11.39 | -10.61      | 14.22 | -7.78       |
| 9         | 21 | 2.62 | -18.38      | 2.12 | -18.88      | 11.39 | -9.61       | 14.22 | -6.78       |
| 6         | 17 | 1.05 | -15.95      | 1.29 | -15.71      | 6.9   | -10.1       | 9.48  | -7.52       |
| 7         | 18 | 1.05 | -16.95      | 1.29 | -16.71      | 8.06  | -9.94       | 11.06 | -6.94       |
| 8         | 18 | 1.05 | -16.95      | 1.05 | -16.95      | 9.21  | -8.79       | 12.64 | -5.36       |
| 9         | 19 | 1.05 | -17.95      | 0.57 | -18.43      | 10.36 | -8.64       | 14.22 | -4.78       |
| 10        | 15 | 1.05 | -13.95      | 0.46 | -14.54      | 11.51 | -3.49       | 15.8  | 0.8         |
| 10        | 18 | 1.05 | -16.95      | 0.37 | -17.63      | 11.51 | -6.49       | 15.8  | -2.2        |
| 9         | 14 | 1.05 | -12.95      | 0.46 | -13.54      | 10.36 | -3.64       | 14.22 | 0.22        |
| 10        | 16 | 1.05 | -14.95      | 0.46 | -15.54      | 11.51 | -4.49       | 15.8  | -0.2        |
| 11        | 16 | 1.05 | -14.95      | 0.46 | -15.54      | 12.66 | -3.34       | 17.38 | 1.38        |
| 12        | 15 | 1.05 | -13.95      | 0.46 | -14.54      | 13.81 | -1.19       | 18.96 | 3.96        |
| 13        | 15 | 1.05 | -13.95      | 0.46 | -14.54      | 14.96 | -0.04       | 20.54 | 5.54        |

**Tabel 1. Tabel Sebagian Hasil Simulasi Formula (Lanjutan)**

| Q_len        | R  | F1     | Diff_F1 | F2     | Diff_F2 | F3     | Diff_F3 | F4     | Diff_F4 |
|--------------|----|--------|---------|--------|---------|--------|---------|--------|---------|
| 11           | 17 | 1.05   | -15.95  | 0.46   | -16.54  | 12.66  | -4.34   | 17.38  | 0.38    |
| 10           | 16 | 1.05   | -14.95  | 0.37   | -15.63  | 11.51  | -4.49   | 15.8   | -0.2    |
| 10           | 18 | 1.05   | -16.95  | 0.46   | -17.54  | 11.51  | -6.49   | 15.8   | -2.2    |
| 9            | 14 | 1.05   | -12.95  | 0.46   | -13.54  | 10.36  | -3.64   | 14.22  | 0.22    |
| 10           | 16 | 1.05   | -14.95  | 0.46   | -15.54  | 11.51  | -4.49   | 15.8   | -0.2    |
| 11           | 16 | 1.05   | -14.95  | 0.46   | -15.54  | 12.66  | -3.34   | 17.38  | 1.38    |
| 12           | 15 | 1.05   | -13.95  | 0.46   | -14.54  | 13.81  | -1.19   | 18.96  | 3.96    |
| 13           | 15 | 1.05   | -13.95  | 0.46   | -14.54  | 14.96  | -0.04   | 20.54  | 5.54    |
| ...          |    |        |         |        |         |        |         |        |         |
| AVGDif       |    | 111.13 |         | 14.57  |         | 3.05   |         | 2.39   |         |
| Std.Dev      |    | 301.43 |         | 30.55  |         | 2.72   |         | 2.28   |         |
| +/- 2min     |    | 21.09% |         | 29.20% |         | 59.22% |         | 67.95% |         |
| No Over Time |    | 87.42% |         | 83.56% |         | 87.22% |         | 65.51% |         |

**Q\_len** (*Queue Length*): jumlah tiket yang menunggu sebelum nomor tiket *Customer* bersangkutan.

**R** (*Real Wait Time*): waktu tunggu *Customer* sebenarnya dalam antrian hingga dilayani (dalam menit).

**F1** (Formula 1): hasil estimasi berdasarkan rumus Formula 1.

**Diff\_F1** (*Difference F1*): Selisih dari F1 dikurangi R.

**F2** (Formula 2): hasil estimasi berdasarkan rumus Formula 2 (Persamaan 12).

**Diff\_F2** (*Difference F2*): Selisih dari F2 dikurangi R.

**F3** (Formula 3): hasil estimasi berdasarkan rumus Formula 3 (Persamaan 13).

**Diff\_F3** (*Difference F3*): Selisih dari F3 dikurangi R.

**F4** (Formula 4): hasil estimasi berdasarkan rumus Formula 4 (Persamaan 14).

**Diff\_F4** (*Difference F4*): Selisih dari F4 dikurangi R.

Bilangan positif pada kolom Diff merupakan estimasi waktu yang melewati/melebihi waktu tunggu sebenarnya, sedangkan bilangan negatif merupakan estimasi waktu yang berada dibawah/sebelum waktu tunggu sebenarnya. Dapat dikatakan bilangan positif menunjukkan *over time*.

Berdasarkan analisa dari pengujian, hasil F3 memiliki perkiraan *over time* yang lebih sedikit dibandingkan dengan hasil F4 (dapat dibandingkan antara kolom Diff\_F3 dengan Diff\_F4). Namun, jika dibandingkan dari besarnya selisih antara perkiraan waktu tunggu dengan waktu tunggu sebenarnya (R), ditemukan bahwa hasil F4 lebih mendekati R daripada hasil F3. Besarnya selisih dapat kita lihat pada rata-rata selisih (AVG Diff) pada Tabel 2. Rata-rata Diff\_F3 3,05 dan Diff\_F4 2,39. Dapat disimpulkan bahwa F4 memiliki selisih yang lebih kecil (Diff\_F4 lebih mendekati R) dibandingkan F3.

Ditinjau pula jika diberikan toleransi +/- 2 menit, maka hasil F4 lebih banyak sukses dibandingkan hasil F3, yaitu sebanyak 335

kali dari total 493 data. Standar deviasi juga menunjukkan hasil F4 lebih rendah dibandingkan hasil F3. Maka dari itu, penulis memilih F4 yang lebih cocok untuk digunakan, karena selain lebih mendekati waktu tunggu sebenarnya (R) secara psikologis orang mengantri akan lebih senang apabila diberikan perkiraan atau estimasi yang memiliki selisih waktu lebih sedikit, dibandingkan yang tidak *over time* namun masih harus menunggu lebih lama lagi (hasil F3).

## 5. KESIMPULAN

Setelah membuat Desain Aplikasi Antrian Berbasis Android, dapat diambil dua kesimpulan, yaitu:

- Ujicoba perhitungan perkiraan waktu tunggu dengan data yang didapat dari layanan publik menghasilkan perkiraan yang cukup baik, yaitu sekitar 65,51% tanpa toleransi *over time* dan mencapai 67,95% dengan toleransi +/- 2 menit.
- User mendapatkan informasi secara *real-time* dan notifikasi tiket pada aplikasi Android.

## 6. DAFTAR PUSTAKA

- [1] Google Developers. 2016. Google Cloud Messaging: Lifecycle Flow. URI=<https://developers.google.com/cloud-messaging/gcm#lifecycle>
- [2] Hartanto, E. 2005. *Model Antrian*. URI=[http://eko\\_hartanto.staff.gunadarma.ac.id/Downloads/files/31472/Model+Antrian.pdf](http://eko_hartanto.staff.gunadarma.ac.id/Downloads/files/31472/Model+Antrian.pdf)
- [3] Kakiay, T. J. 2004. *Dasar Teori Antrian untuk Kehidupan Nyata*. Yogyakarta: ANDI
- [4] Lind, D. A., Marchal, W. G., & Wathen, S. A. 2007. *Teknik-teknik Statistik dalam Bisnis dan Ekonomi 1 (ed.13)*. Jakarta: Salemba.
- [5] Nixon, R. 2012. *Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites*. Sebastopol, CA: O'Reilly Media, Inc.
- [6] Nixon, R. 2014. *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. Sebastopol, CA: O'Reilly Media, Inc.
- [7] Rosidi, R. I. 2004. *Membuat Sendiri SMS Gateway Berbasis Protokol SMPP*. Yogyakarta: Penerbit ANDI.
- [8] Rouse, M. *Push Notification*. URI=<http://searchmobilecomputing.techtarget.com/definition/push-notification>
- [9] Saputra, E. Y. 2014, April 8. *Mengenal Javascript, AJAX, jQuery, AngularJS, dan Node.js*. URI=<http://ekajogja.com/mengenal-javascript-ajax-jquery-angularjs-dan-node-js/>
- [10] Spiegel, M. R., Schiller, J. J., & Srinivasan, R. A. 2004. *Teori dan Soal-soal Probabilitas & Statistik (ed. 2)*. Erlangga.
- [11] Suehring, S., & Valade, J. 2013. *PHP, MySQL, JavaScript & HTML5 All-in-One For Dummies*. Hoboken, New Jersey: John Wiley & Sons.
- [12] Susanti, M. N. 2014. *Statistika Deskriptif dan Induktif*. Yogyakarta: Graha Ilmu.
- [13] Tatroe, K., MacIntyre, P., & Lerdorf, R. 2013. *Programming PHP 3rd Edition*. Sebastopol, CA: O'Reilly Media.