

# **Push Notification Pada Sistem Penjualan Tiket Workshop dan Event Menggunakan *Firestore Cloud Messaging* Berbasis *Progressive Web Application***

Nirwan Jaya Suwarno, Justinus Andjarwirawan, Agustinus Noertjahyana  
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra  
Jl. Siwalankerto 121 – 131 Surabaya 60236  
Telp. (031) – 2983455, Fax. (031) – 8417658  
E-mail: nirwanj8@gmail.com, justin@petra.ac.id, agust@petra.ac.id

## **ABSTRAK**

Penggunaan teknologi *website* pada sistem penjualan tiket merupakan langkah yang tepat untuk meningkatkan proses bisnis perusahaan. Penggunaan *website* dapat memberikan dampak yang sangat baik dari sisi pembeli maupun penjual tiket yaitu dengan adanya *website* informasi akan dapat diakses secara mudah. Namun setelah berhasil melakukan proses jual beli tiket melalui *website*, penjual memerlukan waktu yang cukup banyak atau belum efektif untuk berkomunikasi dengan para pembelinya. Masalah ini terjadi karena pembeli menggunakan berbagai komunikasi, sehingga penjual perlu menggunakan berbagai aplikasi untuk menjangkau semua pembeli.

Cara berkomunikasi yang tidak efektif tersebut dapat diselesaikan dengan menyematkan fitur pada *website* yaitu *push notification*. Penjual dapat menggunakan *push notification* untuk mengirimkan pesan *broadcast* kepada para pembelinya tanpa ada satupun yang terlewat. Semua pembeli dapat dijangkau dengan mudah karena *push notification* disematkan langsung pada aplikasi sehingga pembeli tidak perlu meng-*install* aplikasi lain untuk berkomunikasi dengan penjual.

Hasil penelitian menunjukkan bahwa sistem penjualan tiket dengan *push notification* berhasil meningkatkan efektivitas dengan menurunnya waktu penggunaan aplikasi untuk mengirim pesan sebesar 38,4% dibandingkan menggunakan aplikasi eksternal.

**Kata Kunci:** *push notification, Firestore cloud messaging, penjualan tiket, Progressive web application*

## **ABSTRACT**

*The use of website technology in the ticket sales system is the right step to improve the company's business processes. The use of the website can have a great impact on the buyers and sellers of tickets, namely with the existence of a website information will be easily accessible. However, after successfully carrying out the process of buying and selling tickets through the website, the seller needs quite a lot of time or has not been effective in communicating with the buyers. This problem occurs because buyers use various communication, so the sellers need to use sundry applications to reach all buyers.*

*This ineffective way of communicating can be solved by embedding a feature on the website, namely push notifications. Seller can use push notifications to send broadcast messages to their buyers without missing anything. All buyers can be reached easily because push notifications are embedded directly in the application so that buyers do not need to install other applications to communicate with sellers.*

*The results showed that the ticket sales system with push notifications succeeded in increasing the effectiveness decrease the time of using the application to send messages by 38.4% compared to using external applications.*

**Keywords:** *push notification, Firestore cloud messaging, ticket sales, Progressive web application*

## **1. PENDAHULUAN**

*Workshop* merupakan suatu acara dimana terdapat beberapa orang dengan keahlian tertentu yang berkumpul dengan orang berprofesi serupa untuk berdiskusi dan membahas permasalahan tertentu. Kegiatan *workshop* biasanya dihadiri oleh seorang pakar atau ahli di suatu bidang sebagai mentor atau pemberi materi pada kegiatan tersebut. *Workshop* sendiri berfungsi sebagai wadah untuk bertukar informasi dan pengetahuan baru kepada pesertanya yang dilakukan melalui suatu pelatihan tertentu.

Pada penelitian sebelumnya yang berjudul *E-Event Seminar Dan Workshop Berbasis Website* [1] sudah menerapkan sistem informasi penjualan tiket *Workshop* secara *Workshop*. Pada penelitian tersebut, telah mendukung kegiatan *workshop* mulai dari pendaftaran peserta, konfirmasi pembayaran peserta, serta fasilitas tambahan seperti penyimpanan materi *workshop* dalam bentuk foto, video, maupun bentuk file lainnya. Penelitian lainnya yang berjudul *Aplikasi Pemesanan Tiket Seminar Secara Workshop Berbasis Web* di Universitas Pamulang [2] juga sudah menerapkan penjualan tiket seminar secara *online*. Pada penelitian tersebut telah mendukung kegiatan seminar yaitu pendaftaran dan pembayaran tiket.

Namun pada saat peserta sudah berhasil mendaftar pada suatu kegiatan *workshop*, peserta sulit mendapatkan informasi terbaru mengenai *workshop* tersebut. Hal tersebut dikarenakan pihak penyelenggara *workshop* memiliki kendala dalam mengirim informasi atau pesan kepada para pesertanya. Yang menjadi kendala yaitu para peserta *workshop* tidak menggunakan satu *platform* komunikasi yang sama. Rata-rata dari waktu yang diperlukan untuk menggunakan setiap aplikasi tersebut dari membuka hingga mengirim pesan *broadcast* membutuhkan waktu sekitar 112 detik (dengan asumsi bahwa pesan sudah di-*draft* terlebih dahulu lalu di *copy-paste* pada aplikasi). Oleh karena itu diperlukan suatu *platform* yang secara khusus disiapkan sebagai *platform* untuk berkomunikasi antara penyelenggara *workshop* dan peserta *workshop*.

Melihat permasalahan tersebut, penambahan *push notification* pada pada sistem penjualan tiket *workshop* merupakan solusi untuk permasalahan tersebut. *Push notification* merupakan pemberitahuan atau pesan yang dapat muncul di layar perangkat seluler pengguna. Pengiriman *push notification* akan menggunakan *firebase cloud messaging* yang merupakan sebuah layanan dari *google*. Pengiriman pesan dilakukan dengan cara klien akan berlangganan pada suatu topik (*event*). Setiap pengguna yang mendaftar pada suatu *event* atau kegiatan akan secara otomatis berlangganan untuk menerima *push notification* dari penyelenggara *event*.

## 2. LANDASAN TEORI

### 2.1 Push Notification

*Push notification* merupakan pesan yang dapat dikirim langsung ke perangkat seluler pengguna [3]. *Push notification* juga dapat diartikan sebagai pesan yang muncul di perangkat seluler [4]. *Push notification* hanya bisa diterima oleh perangkat seluler yang telah menginstall suatu aplikasi dan *memberikan izin* untuk menerima notifikasi.

*Push notification* dapat sering dijumpai pada aplikasi ramalan cuaca, aplikasi pemesanan instan hingga aplikasi pengingat. *Push notification* dibagi menjadi tiga jenis yaitu *banner notification*, *badge notification*, dan *alerts*.

### 2.2 Firebase Cloud Messaging

*Firestore* merupakan sebuah solusi yang ditawarkan Google untuk pengiriman pesan lintas *platform* yang dapat anda gunakan untuk mengirim pesan secara terpercaya tanpa biaya [5]. *FCM* mendukung pengiriman pesan dalam bentuk notifikasi. Disamping itu ada tiga sistem pengiriman pesan pada *FCM* yaitu pengiriman dari sebuah *device* ke *device* lainnya, dari *device* ke sebuah grup *device*, dan kepada *device* yang berlangganan.

*FCM* juga mendukung pengiriman pesan melalui aplikasi *client* sehingga sangat mudah untuk digunakan. Cara kerja *FCM* yaitu *FCM* menggunakan *cloud functions for firebase* sebagai server aplikasi yang digunakan untuk membuat, menargetkan dan mengirim sebuah pesan [5] lalu pesan tersebut langsung diteruskan ke aplikasi klien.

### 2.3 Progressive Web Application

*Progressive web application* atau *PWA* adalah aplikasi *web* yang menggunakan *API* dan fitur *web browser* yang muncul bersama dengan strategi peningkatan progresif tradisional untuk menghadirkan pengalaman pengguna seperti aplikasi asli ke aplikasi *web* lintas *platform* [6]. Sebuah *website* dapat dikatakan sebagai *PWA* jika memenuhi kriteria yaitu *discoverable*, *installable*, *network independent*, *progressive*, *responsive*.

*PWA* memiliki keunggulan yaitu memungkinkan untuk bisa memperkecil waktu yang dibutuhkan untuk memuat tiap halaman. *PWA* juga bagus untuk *search engine optimization* sehingga bisnis atau *website* dapat ditemukan oleh mesin pencari, tidak seperti aplikasi berbasis *android* maupun *iOS*. *PWA* juga dapat bekerja secara offline, ketika kita memiliki kapasitas internet yang terbatas, *PWA* merupakan solusinya sebab dengan *PWA* kita hanya perlu mengunduh sekali konten yang bersifat statis dan dapat membukanya berkali kali selama konten tersebut tersimpan di *cache* perangkat kita meski keadaan tanpa koneksi internet.

## 3. ANALISIS DAN DESAIN SISTEM

### 3.1 Desain Database

Database yang digunakan adalah database NoSQL yaitu *cloud firestore* dari *firebase google*. Pada pengiriman pesan *broadcast*, pesan disimpan ke dalam database *customer* dan *event organizer*. Pada database *event organizer*, data pesan yang disimpan adalah isi pesan, subject pesan, waktu pengiriman pesan, dan penerima pesan (*customer name*, *customer email*, dan status pesan) yang dapat dilihat pada Gambar 1.

```
{
  "body" : "string",
  "date" : "string",
  "subject" : "string",
  "to" : [{
    "id" : "string",
    "nama" : "string",
    "status" : "string",
  }]
}
```

Gambar 1. Database Pesan Event Organizer

Pada database *customer*, data pesan yang disimpan adalah subject pesan, isi pesan, waktu pengiriman pesan, id *event organizer*, id acara, dan status pesan seperti Gambar 2.

```
{
  "body" : "string",
  "date" : "string",
  "eoid" : "string",
  "eventid" : "string",
  "status" : "string",
  "subject" : "string",
}
```

Gambar 2. Database Pesan Customer

## 4. PENGUJIAN SISTEM

Pengujian sistem dilakukan dengan melakukan proses secara menyeluruh. Pengujian sistem dilakukan untuk mengetahui apakah sistem berjalan dengan baik atau tidak. Pengujian oleh responden dilakukan untuk mengetahui waktu penggunaan aplikasi yang akan digunakan sebagai pembanding untuk menentukan tingkat keefektifan aplikasi ini.

### 4.1 Homepage

Saat berada di halaman *homepage*, aplikasi dapat menampilkan dengan baik beberapa acara yang telah terdaftar di aplikasi. Pengguna dapat melihat keterangan atau detail dari setiap acara yang ada dengan menekan tombol *learnmore* pada acara tersebut.



Gambar 3. Halaman Homepage

## 4.2 Booking Ticket

Setelah memilih salah satu acara, pengguna akan disajikan informasi mengenai acara tersebut seperti deskripsi acara, tanggal acara, waktu acara, lokasi acara hingga harga tiket seperti Gambar 4.



Gambar 4. Halaman Detail Acara

Saat berhasil melakukan pemesanan tiket, pengguna akan dialihkan ke halaman *summary order*. Pengguna akan disajikan informasi seperti status pembayaran, jumlah tiket yang dibeli, harga tiket, kode unik pembayaran dan beberapa pilihan *payment gateway* seperti Gambar 5.



Gambar 5. Halaman Summary Order

## 4.3 Create Event

Saat pengguna akan mendaftarkan acara baru, mereka akan disajikan *form* seperti Gambar 6. Pengguna perlu memasukan beberapa data acara seperti nama acara, deskripsi acara, kategori, lokasi, kota, harga, kuota, tangga, waktu, dan gambar atau *cover* acara. Setelah melengkapi data acara, pengguna perlu menekan tombol *create* agar acara dapat tersimpan di dalam *database*.

**New Event**

Event Name:

Description:

Image:

Category:

Location:

City:

Price:  Quota:

Start Date:  Start Time:

End Date:  End Time:

Created By:

Gambar 6. Halaman Create Event

## 4.4 Withdrawl Revenue

Saat pengguna akan melakukan penarikan dana, pengguna harus menginputkan jumlah penarikan tidak lebih dari saldo yang tersedia seperti Gambar 7. Pengguna juga perlu memasukan bank dan nomor rekening tujuan penarikan dana.



Gambar 7. Halaman Withdrawl Revenue

## 4.5 Broadcast Message

Pengguna perlu memilih nama event, *subject* pesan, isi pesan dan penerima pesan seperti Gambar 8 sebelum mengirim pesan. Ketika sudah mengisi *form* tersebut, pengguna perlu menekan tombol *send* untuk mengirim pesan ke penerima.

**New Message**

Select event:

Receiver:

Subject:

Message:

Gambar 8. Halaman Broadcast Message

## 4.6 Update Ticket Status

Ketika melakukan *update ticket status*, *administrator* dapat memilih untuk merubah status menjadi *paid* atau *cancel*. Setiap baris yang tampil adalah daftar dari seluruh transaksi yang berstatus *pending payment* yang akan diubah statusnya seperti pada Gambar 9.

Search:

Customer	Event	Amount	Status	Action
Gordon	Oskara 2	275	Pending Payment	<input type="button" value="Take action"/>
Clarissa	Mahakarya DEWA19	35453	Pending Payment	<input type="button" value="Take action"/>

Gambar 9. Halaman Update Ticket Status

## 4.7 Update Withdrawl Status

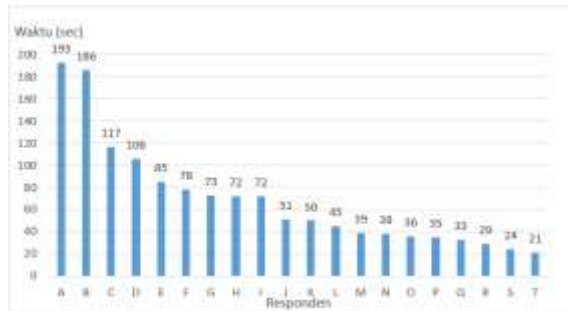
Saat berada di halaman *withdrawl*, admin akan ditampilkan daftar transaksi *withdrawl*. Admin dapat mengubah status *withdrawl* dengan cara memilih salah satu *action* yang tersedia seperti Gambar 10.

EO	Bank	Amount	Number Account	Status	Action
rnwaw@gmail.com	Bank BCA	10	312321	Pending	Pending
rnwawaya@gmail.com	OVO	5000	061236094159	Pending	Pending
rnwaw@gmail.com	Bank BCA	20000	12513125	Pending	Pending
rnwaw@gmail.com	Bank BCA	10	312321	Pending	Pending
rnwaw@gmail.com	Bank BCA	10	312321	Pending	Pending

Gambar 10. Halaman Update Withdrawl Status

## 4.8 Pengujian Oleh Responden

Responden melakukan percobaan dengan tujuan untuk mengetahui waktu yang diperlukan untuk mengirim pesan *broadcast* menggunakan aplikasi. Sebelum melakukan percobaan, responden telah diberi arahan atau langkah langkah yang perlu mereka lakukan dalam menggunakan aplikasi. Dalam melakukan percobaan, responden menggunakan perangkatnya masing masing baik melalui *mobile* atau *desktop*. Pada aplikasi, disematkan fitur *stopwatch* untuk membantu mencatat waktu yang sedang berjalan. *Stopwatch* akan dimulai ketika responden mengkonfirmasi bahwa ia telah memahami cara menggunakan dan siap untuk melakukan pengujian. *Stopwatch* yang telah berjalan akan berhenti secara otomatis Ketika responden mencapai titik tujuan yaitu ketika responden sudah berhasil mengirim sebuah pesan *broadcast*. Setelah *stopwatch* berhenti, waktu yang dihasilkan akan disimpan ke dalam *database*.



Gambar 11. Grafik Hasil Pengujian Responden

Pada Gambar 11 menunjukkan waktu yang diperlukan setiap responden untuk mengirim sebuah pesan *broadcast*. Diantara 20 responden, waktu tertinggi yaitu 193 detik oleh responden A dan waktu terendah yaitu 21 detik oleh responden T. Rata rata waktu yang dibutuhkan responden untuk mengirim sebuah pesan adalah sekitar 69,15 detik.

## 5. KESIMPULAN

Dari hasil penelitian *Push Notification* Pada Sistem Penjualan Tiket *Workshop* dan *Event* Menggunakan *Firestore Cloud Messaging* Berbasis *Progressive Web Application* serta pengujian aplikasi terhadap responden, dapat disimpulkan bahwa.

- Penyematan *push notification* sebagai media pengiriman pesan memerlukan waktu rata rata 69 detik sedangkan jika menggunakan media di luar aplikasi memerlukan waktu rata rata 112 detik. Hal tersebut berarti pengguna dapat menghemat waktu sebesar 43 detik atau 38,4% dari waktu jika menggunakan media di luar aplikasi.
- Penyematan *push notification* sebagai media komunikasi pada sistem penjualan tiket lebih efektif daripada menggunakan media komunikasi lain diluar aplikasi dilihat dari menurunnya waktu penggunaan.
- Kecepatan jaringan dan kemampuan perangkat responden yang beragam sangat mempengaruhi kinerja aplikasi yang menyebabkan batas atas dan batas bawah waktu yang diperlukan untuk mengirim pesan memiliki selisih waktu yang cukup tinggi.

Saran yang dapat diberikan untuk penelitian lebih lanjut antara lain:

- Fitur komunikasi dapat dibuat menjadi komunikasi dua arah, tidak hanya untuk mengirim pesan *broadcast* namun dapat berkomunikasi secara pribadi.
- Tampilan aplikasi dapat dibuat menjadi lebih interaktif agar menambah nilai *user experience*.
- Penambahan fitur untuk mencatat waktu saat pesan dibaca oleh pengguna akan menambah nilai aplikasi dari sisi *event organizer*.

## 6. DAFTAR PUSTAKA

- [1] A. N. Hakim and A. Nursikuwagus, "E Event Seminar dan Workshop Berbasis Web," *Elibrary Unikom*, May 2019.
- [2] A. Ikhsan, S. Ma'mun, and A. Perdananto, "Aplikasi Pemesanan Tiket Seminar Secara Online Berbasis Web di Universitas Pamulang," *Jurnal STMIK ERESHA*, vol. 15, no. 2, pp. 13–21, Oct. 2020.
- [3] Adjust, "Push Notification," *The Adjust Mobile Measurement Glossary*. [Online]. Available: <https://www.adjust.com/glossary/push-notification/>. [Accessed: 16-Nov-2020].
- [4] "Push Notifications Explained," *AIRSHIP*. [Online]. Available: <https://www.airship.com/resources/explainer/push-notifications-explained/>. [Accessed: 16-Nov-2020].
- [5] Firebase Google, *Firestore Cloud Messaging*. [Online]. Available: <https://firebase.google.com/docs/cloud-messaging?hl=id>. [Accessed: 11-Nov-2020].
- [6] MDN Contributor, "Progressive web apps (PWAs)," *MDN Web Docs*. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/Progressive\\_web\\_apps](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps). [Accessed: 09-Nov-2020].