



---

## Simulasi Pengiriman Notifikasi Dalam Pagar Virtual Geografis Berbasis Android

*Simulation of Sending Notifications in an Android-Based Virtual Geographic Fence*

**Aristhia Nadianti**

Fakultas Teknologi Informasi Universitas Tarumanagara Jakarta, Indonesia

\*Email: aristhianad@gmail.com

\*Correspondence: Aristhia Nadianti

---

DOI:

10.36418/comserva.v2i10.573

### ABSTRAK

#### Histori Artikel

Diajukan : 24-01-2023

Diterima : 13-02-2023

Diterbitkan : 25-02-2023

Semakin majunya teknologi, banyak hal yang mudah didapatkan secara otomatis dan tepat waktu hanya dengan penggunaan smartphone. Beberapa hal yang dengan mudahnya didapatkan secara otomatis seperti informasi, pengingat, hingga pesan dari orang lain. Dalam sistem operasi Android, hal tersebut dinamakan notifikasi. Notifikasi ditampilkan di luar UI aplikasi karena fungsinya sebagai pengingat. Pemicu dari munculnya notifikasi dapat dikarenakan waktu dan pesan yang diberikan dari pengguna lain. Namun terdapat salah satu pemicu munculnya notifikasi di dalam perangkat adalah berdasarkan keberadaan lokasi pengguna. Agar dapat memberikan notifikasi berdasarkan keberadaan lokasi pengguna, terdapat teknologi yang disebut geofence atau suatu pagar virtual geografis. Geofence merupakan suatu pagar virtual yang membatasi wilayah atau area secara geografis dengan radius tertentu. Dalam memberikan notifikasi dengan menggunakan pagar virtual, pengguna yang memasuki atau meninggalkan wilayah yang sudah dibatasi secara otomatis akan mendapatkan notifikasi dan pengguna dapat melihat keberadaannya di dalam maps yang ada di aplikasi. Simulasi dari Aplikasi ini berbasis Android. Pembuatan dari aplikasi ini menggunakan software AndroidStudio versi 2021.2.1 Patch 2 dengan bahasa pemrograman Kotlin dan dihubungkan ke Firebase Realtime Database untuk menyimpan data lokasi. Tampilan maps di dalam aplikasi menggunakan Google API.

**Kata kunci:** Android; AndroidStudio; Firebase Realtime Database; Geofencing; Notifikasi.

### ABSTRACT

*As technology advances, many things are easy to get automatically and on time with just the use of a smartphone. Some things that can easily be obtained automatically such as information, reminders, and messages from other people. In the Android operating system, this is called a notification. Notifications are shown outside the app UI because they serve as reminders. Triggers for the appearance of notifications can be due to the time and messages given from other users. However, one of the triggers for the appearance of notifications on the device is based on the presence of the user's location. In order to be able to provide notifications based on the whereabouts of the user's location, there is a technology called geofence or a virtual geographic fence. Geofence is a virtual fence that limits geographical areas or areas with a certain radius. In providing notifications using virtual fences, users who enter or leave restricted areas will automatically receive notifications and users can see their whereabouts on the maps in the application. The simulation of this application is based on Android. The development of this application uses the AndroidStudio version 2021.2.1 Patch 2 software with the Kotlin programming language and is connected to the*

*Firestore Realtime Database to store location data. Display maps in the application using Google API.*

**Keywords:** *Androids; Android Studio; Firestore Realtime Database; geofencing; Notifications.*

---

## **PENDAHULUAN**

Setiap tahunnya dunia terus berkembang termasuk teknologi. Banyaknya manfaat dari kemajuan teknologi membuat perubahan perilaku dan kebiasaan. Salah satu kemajuan pemanfaatan dari teknologi yaitu penggunaan smartphone Android. Pengguna Android dapat berasal dari berbagai latar belakang usia dari anak hingga orang tua. Kecanggihan yang dimiliki Android memiliki banyak fitur yang salah satunya adalah Global Positioning System (GPS) (Abulude et al., 2015).

Keberadaan fitur GPS memberikan banyak manfaat salah satunya dengan memberikan notifikasi secara otomatis berdasarkan lokasi keberadaan pengguna. Notifikasi dapat ditampilkan di luar UI aplikasi sebagai pengingat. Teknologi informasi yang digunakan dalam meningkatkan hal-hal tersebut adalah geofence.

Geofence merupakan pagar virtual yang membatasi suatu wilayah atau area secara geografis dengan radius tertentu. Sistem geofence dapat mengidentifikasi posisi pengguna secara otomatis dan memberikan informasi melalui notifikasi dengan membaca koordinat GPS dari suatu perangkat bergerak. Keberadaan teknologi dapat membantu pengguna dalam mengetahui jarak terdekat dari lokasi yang dituju atau sedang diberikan pagar virtual. Geofence ditampilkan di dalam maps yang ada pada smartphone Android (Priono & Setiawan, 2017). Tampilan dari maps untuk menampilkan geofence adalah dengan menggunakan Google API. Koordinat yang didapatkan GPS sebagai centroid dari geofence nantinya akan dikirim ke server dan disimpan ke dalam Firestore Real-Time Database (Asnal et al., 2022).

Penggunaan geofence dalam menampilkan keberadaan pengguna dan memberikan informasi melalui push notification berdasarkan koordinat GPS sebagai centroid, diharapkan dapat mempermudah pengguna dalam menentukan titik terdekat dari area yang dituju (Firly, 2018).

## **METODE**

### **Populasi dan Sample**

Populasi adalah sekelompok orang atau objek yang ingin dideskripsikan atau digeneralisasikan (Vogt & Johnson, 2011). Populasi tidak hanya berhubungan dengan manusia tetapi berbagai macam objek dapat dijadikan sebagai populasi. Sehingga dalam penelitian ini populasi yang digunakan adalah teori-teori yang membahas tentang pagar virtual yang didapatkan dari berbagai studi kasus (Sugiyono, 2016).

Sampel merupakan bagian dari populasi. Sampel ditunjukkan untuk menyelidiki sifat-sifat tertentu dari populasi. Sample dari populasi penelitian ini adalah teori kode program pagar virtual untuk memberikan notifikasi.

### **Data Penelitian**

#### **1. Jenis dan Sumber Data**

Dalam melakukan penelitian ini, jenis data yang diperlukan adalah teori-teori seputaran pagar virtual geografis atau geofence yang nantinya digunakan untuk melakukan pengiriman notifikasi.

Dalam melakukan penelitian ini, sumber data yang digunakan oleh adalah sumber data sekunder. Sumber data sekunder dari penelitian ini didapatkan berbagai studi kasus dan artikel yang membahas tentang teori-teori geofence (Nguyen & Krumm, 2022). Dalam penelitian juga memerlukan titik koordinat 2 untuk menghitung jarak batasan lokasi. Sumber data titik koordinat 2 didapatkan melalui aplikasi Google maps.

## 2. Teknik Pengumpulan Data

Dalam mendapatkan data-data yang diperlukan untuk penelitian perlu dilakukan teknik pengumpulan data. Teknik yang dilakukan dalam pengumpulan data adalah dengan mempelajari studi kasus.

### Variabel Penelitian

Variabel adalah subjek atau karakteristik dari penelitian yang akan diuji dan bervariasi. Variabel penelitian merupakan atribut dari suatu objek yang memiliki variasi tertentu yang sudah ditetapkan oleh peneliti untuk dipelajari sehingga dapat diperoleh informasi tersebut dan dapat ditarik kesimpulannya (Sugiyono, 2007). Hubungan antara setiap variabel maka macam-macam variabel dapat dibedakan menjadi dua:

#### 1) Variabel Independen (Bebas)

Variabel Independen merupakan variabel yang mempengaruhi atau penyebab munculnya variabel dependen. Variabel bebas dari penelitian simulasi ini adalah pagar virtual geografis atau geofence.

#### 2) Variabel Dependen (Terikat)

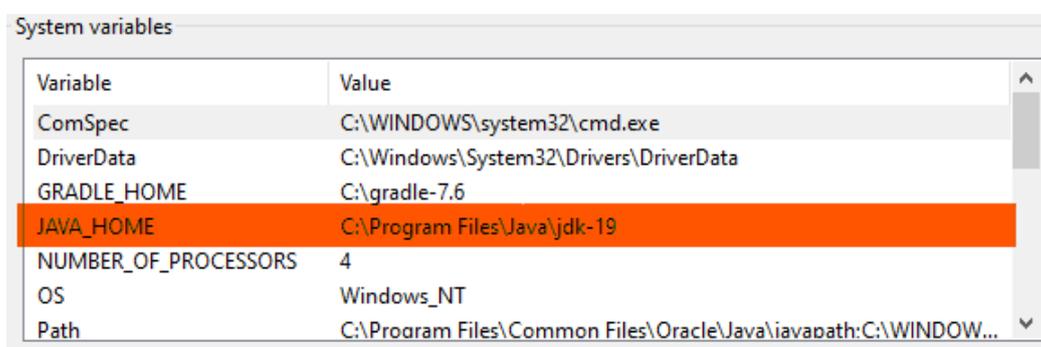
Variabel Dependen muncul karena adanya variabel Independen atau bisa disebut juga sebagai akibat dari variabel Independen. Dalam SEM (Structural Equation Modeling) Variabel dependen juga disebut dengan variabel endogen. Variabel terikat yang muncul akibat adanya variabel independen adalah pengiriman notifikasi.

### Metode dan Teknik Analisa Data

Dalam melakukan simulasi penerapan pagar virtual pada aplikasi Android, berikut hal-hal yang diperlukan untuk mengembangkan aplikasi.

#### 1. Persyaratan Sistem

Dalam melakukan simulasi terhadap pagar virtual dalam aplikasi Android memerlukan instalasi beberapa software seperti Android Studio dan Java Development Kit (JDK) (Hariadi, 2013). JDK harus diinstal terlebih dahulu ke dalam system variables yang ada pada Windows Environment seperti gambar 1.



Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
GRADLE_HOME	C:\gradle-7.6
<b>JAVA_HOME</b>	<b>C:\Program Files\Java\jdk-19</b>
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\WINDOW...

Gambar 1. JDK di dalam *environment*

Agar aplikasi dapat dijalankan, di dalam Android Studio juga perlu untuk menginstal Android Virtual Device (AVD) ataupun terhubung dengan perangkat Android yang ada menggunakan koneksi Wifi.

## 2. Google Play Service dan Akses Izin Lokasi

Aplikasi Android pada simulasi ini menggunakan salah satu fitur yang dimiliki Android yaitu Google Play Service. Dengan menambahkan Google Play Service, aplikasi simulasi ini dapat menjalankan Maps SDK yang dimiliki oleh Google serta dapat menghubungkan proyek dengan Firebase Database (Sali & Amali, 2019).

Google Play Service APIs dapat dijalankan di Android 4.4 (API level 19) atau lebih tinggi. Terdapat beberapa dependencies yang harus ditambahkan ke dalam file build.gradle aplikasi yang ada pada proyek agar Google Play Service bisa dijalankan (Syaputri et al., 2022). Berdasarkan gambar 2 dependency tersebut bertujuan agar dapat menampilkan google maps di dalam aplikasi.



Gambar 2. Dependency Maps SDK



Gambar 3. Dependency Akses Lokasi

Menurut (Google, n.d.) dalam mengakses lokasi pengguna, untuk melindungi privasi dari pengguna, aplikasi perlu meminta izin akses lokasi dengan menambahkan kode akses izin ke dalam file AndroidManifest.xml seperti `ACCESS_FINE_LOCATION`, `ACCESS_COARSE_LOCATION` dan `ACCESS_BACKGROUND_LOCATION`. Contoh implementasi kode tersebut dapat dilihat di gambar 4.

```
AndroidManifest.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     package="com.example.geofenceapps">
5
6
7     <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
8     <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
9     <uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION"/>
10
```

Gambar 1. Kode izin akses lokasi

Menghubungkan firebase database ke aplikasi perlu ditambahkan Firebase SDK ke dalam build.gradle aplikasi seperti gambar 3.5 menurut (Developers, n.d.).

```
implementation platform('com.google.firebase:firebase-bom:31.1.1')
implementation 'com.google.firebase:firebase-analytics-ktx'
implementation 'com.google.firebase:firebase-database-ktx'
implementation 'com.google.firebase:firebase-database:20.1.0'
```

Gambar 5. Firebase SDK

### 3. Menentukan radius pagar virtual

Pagar virtual atau geofence memiliki berbagai macam bentuk salah satunya berbentuk lingkaran. Menurut (Asnal et al., 2022) Agar dapat mengetahui radius dari pagar virtual sebagai data yang valid, terdapat perhitungan bernama Algoritma Equirectangular. Perhitungan ini memerlukan dua titik koordinat yaitu koordinat 1 sebagai titik terluar dan koordinat 2 sebagai centroid.

$$\begin{aligned} X &= \text{rad}(\text{long}2 - \text{long}1) * \cos(\text{rad}(\text{lat}1 + \text{lat}2)/2) \\ &= \text{rad}(106.883260 - 106.883598) * \cos(\text{rad}(-6.112745 + (-6.113518))/2) \\ &= -0.000338 * 0.9943193251 \\ &= -0.00033607 \end{aligned}$$

$$\begin{aligned} Y &= \text{rad}(\text{lat}1 - \text{lat}2) \\ &= \text{rad}(-6.112745 - (-6.113518)) \\ &= 0.00001348 \end{aligned}$$

$$\begin{aligned} Z &= \sqrt{X^2 + Y^2} * R \\ &= \sqrt{0.00000011 + 0.00000000018171} * 6371000 \\ &= 2.11 \text{ km} = \mathbf{211 \text{ m}} \end{aligned}$$

Keterangan:

rad =  $\pi(\text{phi}) : 180 = 0.01744444$

long1 dan lat1 = titik koordinat terluar

long2 dan lat2 = titik koordinat centroid

R = Radius bumi (R= 6371)

Z = jarak atau radius

### 4. Membuat Kode Geofence

Pembuatan simulasi pagar virtual pada aplikasi Android adalah dengan menggunakan bahasa pemrograman Kotlin. Agar dapat menggunakan teknologi pagar virtual atau geofence, menurut website <https://developer.android.com/>, ada beberapa kode yang perlu ditambahkan, sebagai berikut (Firly, 2018).

#### a. Mengakses API Location

Agar dapat mengakses lokasi API, di dalam metode onCreate tambahkan klien geofence yang sebelumnya sudah dibuat variable, sesuai dengan yang dijelaskan di dalam website <https://developer.android.com/>. Gambar 3.6 merupakan implementasi dari klien geofence di dalam proyek (Priono & Setiawan, 2017).

```

private lateinit var geofencingClient: GeofencingClient

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)

    setContentView(R.layout.activity_maps)
    val mapFragment = supportFragmentManager
        .findFragmentById(R.id.map) as SupportMapFragment
    mapFragment.getMapAsync( callback: this)

    fusedLocationClient = LocationServices.getFusedLocationProviderClient( activity: this)
    geofencingClient = LocationServices.getGeofencingClient( activity: this)
}
    
```

Gambar 6. Kode GeofenceClient

**HASIL DAN PEMBAHASAN**

**Sarana Simulasi**

Simulasi dilakukan untuk melakukan pembuktian apakah notifikasi dapat dikirimkan di dalam suatu pagar virtual geografis berdasarkan transisi pengguna. Agar simulasi dapat dijalankan, simulasi dilakukan dengan terlebih dahulu membuat aplikasi Android. Pembuatan aplikasi didasarkan atas kode-kode program yang didapatkan saat melakukan pengumpulan data (Sarosa, 2021). Software yang digunakan untuk membuat aplikasi android dapat dilihat pada table 1:

**Table 1. Software perancangan aplikasi**

Nama <i>Software</i>	Keterangan
<b>Android Studio</b>	Versi Chipmunk   2021.2.1 Patch 2

Adapun software yang digunakan untuk penyimpana data lokasi pagar virtual pada table 2, sebagai berikut:

**Table 2. Software Database**

Nama <i>Software</i>	Keterangan
<b>Firestore</b>	Produk yang digunakan: - Realtime Database

Simulasi dilakukan ketika perangkat Android memiliki koneksi internet serta GPS pada perangkat diaktifkan. Table 4.3 adalah perangkat Android yang digunakan untuk melakukan simulasi pengiriman notifikasi di dalam pagar virtual geografis:

**Table 3. Hardware Simulasi**

Nama Perangkat	Jumlah	Spesifikasi
----------------	--------	-------------

---

<b>Samsung Galaxy A02S</b>	1	Memori RAM: 4 GB Memori ROM: 64 GB OS: Android 11
<b>Samsung Galaxy A80 2017</b>	1	Memori RAM: 8 GB Memori ROM: 32 GB OS: Android 9

---

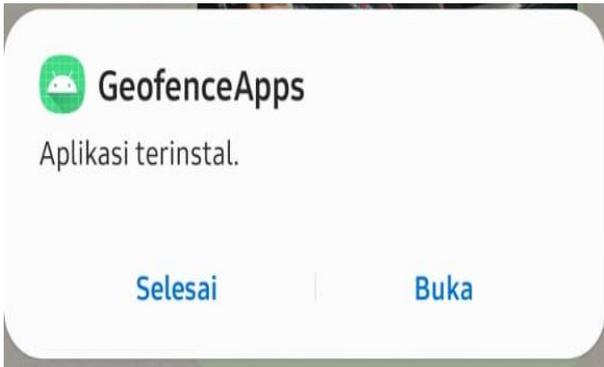
Penggunaan dua perangkat dengan operasi sistem yang berbeda agar terlihat perbedaan hasil simulasi.

**Hasil Simulasi**

Aplikasi yang sudah dirancang dapat langsung digunakan untuk melakukan simulasi pengiriman notifikasi di dalam pagar virtual geografis. Simulasi ini dilakukan menggunakan dua perangkat android dengan sistem operasi yang berbeda yaitu Android 11 dan Android 9 (Marisa, 2017). Kedua perangkat tersebut saat ingin melakukan simulasi memiliki dua hasil yang berbeda. Table ... merupakan perbedaan dari kedua operasi sistem tersebut.

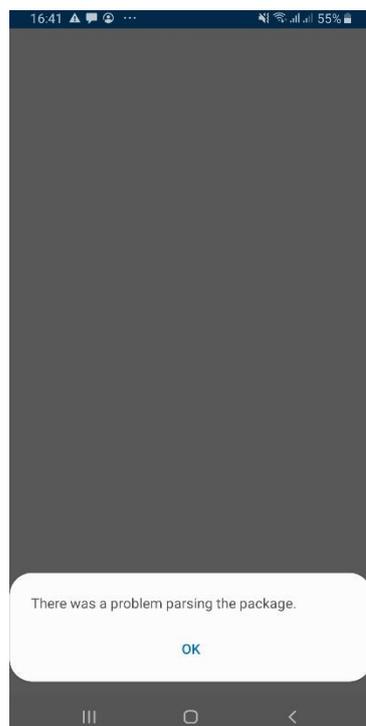
**Table 4. Hasil Instalasis Aplikasi**

---

<b>Operasi Sistem</b>	<b>Hasil</b>	<b>Keterangan</b>
Android 11		Aplikasi berhasil diinstal di Android 11

---

Android  
9



Pada Android 9, aplikasi tidak dapat diinstal karena saat perancangan aplikasi, API target yang ditentukan adalah 30. Sehingga tidak sesuai dengan API Android 9 yaitu API Level 28.

---

Proses simulasi tidak dapat dilanjutkan menggunakan perangkat Android dengan sistem operasi Android 9 sehingga hanya dapat dilanjutkan menggunakan Android 11.

### **Penyajian Hasil Penelitian**

#### **1. Perbandingan dengan Teori dan Temuan Sebelumnya**

Dalam melakukan simulasi ini terdapat beberapa perbandingan dengan teori dan penemuan sebelumnya. Pada simulasi ini, pagar virtual dapat dimanfaatkan untuk mengirimkan notifikasi ketika pengguna memasuki atau meninggalkan lokasi yang sudah diberikan pagar virtual. Sedangkan penelitian yang dilakukan Budiyanto (2020) dengan judul "Implementasi Teknologi Geofencing untuk Sistem Lokasi Dosen (Silodes) di Universitas PGRI Yogyakarta Berbasis Android", keberadaan objek yang memasuki wilayah geofence, nantinya akan ditampilkan ke dalam halaman khusus yang diberi nama "halaman notifikasi" di dalam sistem aplikasi tersebut, pengguna juga perlu untuk mengklik tombol "refresh" yang disediakan untuk mendapatkan informasi objek tersebut (Nurzam et al., 2017). Sehingga pengguna tidak mendapatkan informasi objek tersebut secara real-time melalui notifikasi yang berada di dalam panel perangkat Android (Nasution, 2018).

Menentukan lokasi pagar virtual pada simulasi ini dapat dibuat lebih dari satu lokasi atau dapat disebut dengan multiple geofence. Penelitian yang dilakukan oleh Sinaga & Setiawan, (2018) dengan judul "Implementasi Teknologi Geofencing untuk Objek Wisata Dan Pelayanan Pelanggan Berbasis Android" menghasilkan suatu sistem dimana pada aplikasi tersebut lokasi pagar virtual hanya dapat ditentukan oleh sistem dan memiliki satu lokasi geofence yaitu suatu hotel di daerah Bandung (Priono & Setiawan, 2017). Selain itu juga teknologi geofence ini dimanfaatkan untuk menemukan objek wisata yang terdekat dengan hotel tersebut dengan menentukan radius atau jarak maksimal dari pagar virtual.

#### **2. Temuan Penelitian yang diperoleh**

---

Temuan penelitian dari dilakukannya simulasi ini adalah bahwa pagar virtual geografis atau geofence dapat dengan otomatis dan real-time memberikan notifikasi kepada pengguna saat transisi pengguna, yaitu memasuki dan meninggalkan lokasi pagar virtual dapat terdeteksi oleh sistem. Namun notifikasi tidak dapat diterima jika pengguna keluar dari aplikasi karena lokasi pagar virtual secara otomatis hilang.

Dengan melakukan simulasi ini, ditemukan juga bagaimana menghitung radius untuk diterapkan ke dalam pagar virtual dengan hasil yang lebih akurat berdasarkan titik koordinat longitude dan latitude.

## **SIMPULAN**

Berdasarkan hasil dan pembahasan dari simulasi, terdapat beberapa hal yang dapat disimpulkan sebagai berikut: a) Notifikasi dapat dikirimkan berdasarkan keberadaan lokasi pengguna secara Real-time. b) Radius pagar virtual dapat dihitung menggunakan algoritma agar mendapatkan hasil yang lebih presisi. c) Pagar virtual dapat dibuat lebih dari satu di dalam satu maps yang sama.

**DAFTAR PUSTAKA**

- Abulude, F., Akinnusotu, A., & Adeyemi, A. (2015). Global Positioning System and It's Wide Applications. *Continental J. Information Technology*, 9(1).
- Asnal, H., Mirnawati, K., & Anam, M. K. (2022). Sistem Monitoring Position Lansia Menggunakan Teknologi Geofencing Berbasis Android. *SATIN-Sains Dan Teknologi Informasi*, 8(1), 61–68.
- Budiyanto, G. (2020). Implementasi Teknologi Geofencing untuk Sistem Lokasi Dosen (Silodes) di Universitas PGRI Yogyakarta Berbasis Android. *Seri Prosiding Seminar Nasional Dinamika Informatika*, 4(1).
- Firly, N. (2018). *Create Your Own Android Application*. Elex Media Komputindo.
- Hariadi, S. (2013). *Rancang bangun game edukasi Java Coding Game menggunakan metode generating tree pada Context Free Grammar*. Universitas Islam Negeri Maulana Malik Ibrahim.
- Marisa, F. (2017). *Web Programming (Client Side and Server Side)*. Deepublish.
- Nasution, A. (2018). Perancangan Aplikasi Push Notification Berbasis Android. *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, 4(2), 149–154.
- Nguyen, K., & Krumm, J. (2022). Reliable geofence activation with sparse and sporadic location measurements: Extended version. *ArXiv Preprint ArXiv:2204.00714*.
- Nurzam, F., Fajri, I. N., & Prabowo, D. (2017). Rancang bangun aplikasi media laporan aspirasi dengan firebase cloud messaging berbasis mobile. *Semasteknomedia Online*, 5(1), 4–5.
- Priono, J., & Setiawan, E. B. (2017). Implementasi geofencing dalam monitoring rute pengiriman kendaraan di sebuah perusahaan ekspedisi. *Ultimatics: Jurnal Teknik Informatika*, 9(2), 106–113.
- Sali, M. R., & Amali, L. N. (2019). Aplikasi kajian Islam dengan teknologi firebase realtime database. *Jambura Journal of Informatics*, 1(1), 1–8.
- Sarosa, S. (2021). *Analisis Data Penelitian Kualitatif*. PT Kanisius.
- Sinaga, E. E., & Setiawan, E. B. (2018). Implementasi Teknologi Geofencing Untuk Objek Wisata dan Pelayanan Pelanggan Berbasis Android. *Ilmiah Komputer Dan Informatika (KOMPUTA)*.
- Sugiyono, P. (2016). Metode Penelitian Manajemen (Pendekatan Kuantitatif, Kualitatif, Kombinasi (Mixed Methods), Penelitian Tindakan (Action Research, dan Penelitian Evaluasi). *Bandung: Alfabeta Cv*.
- Syaputri, E. R., Samsudin, S., & Ikhwan, A. (2022). Implementasi Metode Geofence Pada Aplikasi Reminder Berbasis Android. *Journal of Information System Research (JOSH)*, 3(3), 252–261.



© 2023 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY SA) license (<https://creativecommons.org/licenses/by-sa/4.0/>).