



## Pengembangan Aplikasi Navigasi Berbasis *Android* : Kalkulasi *Back Azimuth* dan Deteksi Deviasi Secara *Realtime*

Rafael Axel Pramudya Divangga<sup>1</sup>, Chairullah Naury<sup>2</sup>, Ari Pantjarani<sup>3</sup>

<sup>1,2,3</sup> Manajemen Informatika, Politeknik Assalaam Surakarta, Surakarta, Indonesia

<sup>1</sup>[rafael.axel01@gmail.com](mailto:rafael.axel01@gmail.com), <sup>2</sup>[ch.naury@politeknikassalaam.ac.id](mailto:ch.naury@politeknikassalaam.ac.id), <sup>3</sup>[pantjarani@politeknikassalaam.ac.id](mailto:pantjarani@politeknikassalaam.ac.id)

Article History: Received: December, 16 2024; Accepted: December, 24 2024; Published: December, 30 2024

### ABSTRACT

Navigation plays a crucial role in daily life, supporting human activities to explore the environment effectively. In the military field, navigation involves understanding terrain science, map orientation, resection, intersection, azimuth, back-azimuth, distance measurement, elevation, compass, and topographic map interpretation. These skills make navigation critical in defense and tactical operations, requiring high precision and quick mobility in dynamic combat environments. The use of traditional tools has proven inefficient in unfamiliar and complex terrains, thus necessitating a mobile and practical navigation application. The author designed an Android-based application focused on calculating back-azimuth and detecting user deviation in real-time. The methodology used is Rapid Application Development (RAD), involving planning, workshop design, and implementation stages. Through repeated collaboration with users, the application's requirements and design are refined to ensure alignment with user needs and organizational goals.

**Keywords:** navigation; back-azimuth; deviation; android.



Copyright © 2024 The Author(s)

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

### PENDAHULUAN

Navigasi merupakan cara menentukan posisi dan arah perjalanan baik di medan sebenarnya maupun di peta (Rosia et al., 2022). Navigasi, sebagai kemampuan untuk menentukan posisi dan arah, memainkan peran krusial dalam berbagai aspek, dari kehidupan sehari-hari hingga berbagai pengembangan teknologi. Dalam kehidupan sehari-hari, navigasi membantu manusia untuk menjelajah dan beroperasi di medan asing dengan lebih efektif. Dengan bantuan teknologi navigasi, kita dapat menemukan rute tercepat untuk mencapai suatu tujuan, menghindari kemacetan lalu lintas, dan menemukan tempat-tempat baru dengan mudah. Selain itu, navigasi juga berperan penting dalam berbagai bidang seperti penerbangan, pelayaran, dan eksplorasi luar angkasa, di mana akurasi dan presisi sangat penting untuk keselamatan dan keberhasilan misi.

Di militer, navigasi menjadi komponen kritis dalam operasi pertahanan dan taktis yang membutuhkan presisi dan keandalan tinggi dalam lingkungan pertempuran yang kompleks dan dinamis. Kemampuan untuk menentukan posisi sendiri dan posisi musuh secara akurat dapat menjadi faktor penentu dalam pertempuran. Penulis sendiri pernah mengikuti latihan navigasi darat di kesatuan pendidikan militer. Sebagai komandan regu, penulis mengalami kesulitan saat bernavigasi menggunakan alat konvensional yaitu kompas dan peta yang membutuhkan banyak

waktu dan memerlukan jalur garis lurus berdasarkan pembacaan derajat kompas. Hal ini khususnya sulit dilakukan di kawasan hutan dimana banyak rintangan menghalangi kemampuan untuk mempertahankan lintasan lurus sesuai dengan pembacaan kompas. Dalam kondisi medan yang sulit, seperti hutan lebat atau pegunungan, penggunaan kompas dan peta konvensional dapat menjadi sangat menantang dan menghambat mobilitas serta kecepatan operasi.

Seiring dengan perkembangan teknologi, kebutuhan akan sistem navigasi yang lebih canggih dan efisien semakin meningkat. Sistem navigasi modern yang berbasis teknologi digital dan sensor canggih telah mulai menggantikan metode konvensional. Aplikasi navigasi berbasis *Android* menawarkan solusi yang lebih praktis dan mudah digunakan, memungkinkan pengguna untuk mendapatkan informasi navigasi secara *real-time* dan melakukan kalkulasi yang diperlukan dengan cepat. Keberadaan aplikasi navigasi yang mampu melakukan kalkulasi *back azimuth* dan mendeteksi deviasi secara *real-time* sangat diperlukan untuk meningkatkan efektivitas dan efisiensi dalam bernavigasi di medan yang sulit.

Berdasarkan latar belakang yang penulis paparkan sebelumnya, rumusan masalah penelitian ini adalah bagaimana mengembangkan aplikasi navigasi yang dapat membantu dalam kalkulasi *back azimuth* serta mendeteksi adanya deviasi ketika pengguna berjalan dari titik keberangkatan ke sudut yang ingin dituju, secara efisien dan mudah digunakan (*mobile*). Penelitian ini bertujuan untuk merancang dan mengembangkan sebuah aplikasi berbasis *Android* yang mampu memberikan informasi navigasi secara akurat dan membantu pengguna dalam mempertahankan lintasan yang diinginkan, sehingga dapat meningkatkan efektivitas dan efisiensi navigasi di medan yang kompleks dan dinamis. Dengan adanya aplikasi ini, diharapkan pengguna dapat bernavigasi dengan lebih mudah dan cepat, serta mengurangi risiko tersesat atau keluar jalur.

Untuk pengembangan aplikasi ini, metode *Rapid Application Development (RAD)* digunakan. *Rapid Application Development (RAD)* merupakan salah satu metode strategi pengembangan sistem, yang mengutamakan kecepatan pengembangan melalui keterlibatan pemakai luas dalam penggunaan suatu rangkaian konstruksi, dimana rangkaian tersebut berfungsi sebagai suatu model (*prototype*) sistem yang lebih efektif. Pengembangan *RAD (Rapid Application Development)* dengan tahapan *Requirement Planning, Workshop Design, dan Implementation*, dalam pengembangan rancangan sistem informasi monitoring pajak bumi dan bangunan, menjadikan sistem jadi lebih cepat dan efisien (*Pratama, Nurdiawan, & Pramudita, 2019*). Penggunaan *RAD* memastikan bahwa aplikasi yang dikembangkan sesuai dengan kebutuhan pengguna dan dapat disesuaikan dengan cepat jika ada perubahan persyaratan. Bahasa pemrograman yang digunakan adalah *Dart*, yang dikenal karena efisiensinya dalam pengembangan aplikasi *mobile* yang cepat dan dapat diandalkan. *Dart* digunakan bersama dengan *Flutter*, sebuah *framework* yang memungkinkan pengembangan antarmuka pengguna yang responsif dan menarik secara *visual* di platform. Dengan memanfaatkan teknologi-teknologi ini, aplikasi navigasi yang dikembangkan diharapkan dapat memberikan performa yang optimal dan pengalaman pengguna yang baik.

## METODE

Metode yang penulis gunakan dalam pengembangan aplikasi navigasi berbasis *android* : kalkulasi *back azimuth* dan deteksi deviasi secara *Real-time* adalah sebagai berikut:

### A. Metode Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan melalui dua metode utama, yaitu observasi dan wawancara dengan rekan sesama siswa tentara. Metode ini dipilih untuk mendapatkan informasi yang komprehensif dan relevan mengenai kebutuhan dan tantangan yang dihadapi dalam navigasi darat.

#### 1. Observasi:

Penulis melakukan observasi langsung selama latihan navigasi darat di kesatuan pendidikan militer. Observasi ini bertujuan untuk memahami kondisi lapangan, peralatan yang digunakan, serta kesulitan yang dihadapi oleh pengguna saat bernavigasi menggunakan alat konvensional seperti kompas dan peta. Melalui observasi, penulis

dapat mengidentifikasi aspek-aspek penting yang perlu diperbaiki dan fitur yang diperlukan dalam aplikasi navigasi yang akan dikembangkan.

2. Wawancara:

Wawancara dilakukan dengan rekan sesama siswa tentara yang memiliki pengalaman dalam navigasi darat. Wawancara ini bertujuan untuk mendapatkan masukan dan pandangan dari pengguna mengenai masalah yang sering dihadapi, kebutuhan spesifik, dan harapan mereka terhadap aplikasi navigasi yang akan dikembangkan. Wawancara dilakukan secara mendalam untuk menggali informasi yang lebih detail dan memastikan bahwa aplikasi yang dikembangkan sesuai dengan kebutuhan pengguna.

B. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *Rapid Application Development (RAD)*. *RAD* merupakan metode pengembangan yang mengutamakan kecepatan pengembangan melalui keterlibatan pengguna yang luas dalam penggunaan suatu rangkaian konstruksi, di mana rangkaian tersebut berfungsi sebagai suatu model (*prototype*) sistem yang lebih efektif. Pengembangan *RAD* memiliki tahapan-tahapan sebagai berikut:

1. *Requirement Planning*

Pada tahap ini, penulis dan pengguna (siswa) bertemu untuk mengidentifikasi tujuan aplikasi serta syarat informasi yang dibutuhkan. Tahap ini melibatkan:

1.1 Identifikasi Tujuan:

Menentukan apa yang ingin dicapai oleh aplikasi, seperti kalkulasi *back azimuth* dan deteksi deviasi secara *real-time*.

1.2 Keterlibatan Pengguna:

Melibatkan pengguna dari berbagai level dalam organisasi untuk mendapatkan gambaran yang lengkap tentang kebutuhan dan masalah yang dihadapi.

1.3 Pengumpulan Informasi:

Mengumpulkan data melalui observasi dan wawancara untuk memahami kebutuhan pengguna secara mendalam dan detail.

1.4 Analisis Kebutuhan:

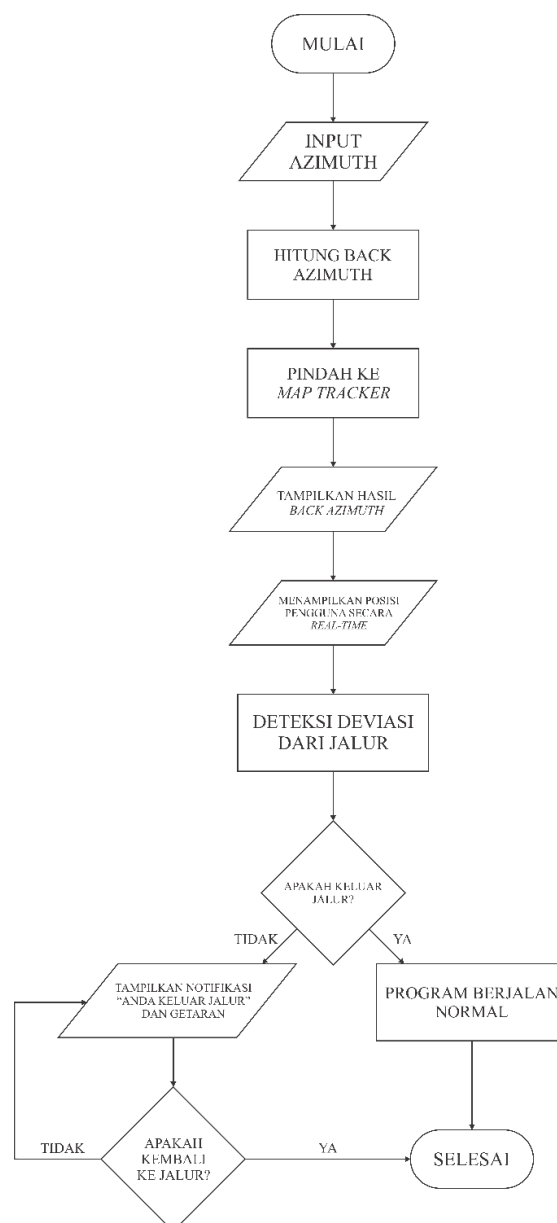
Menganalisis data yang diperoleh untuk merumuskan kebutuhan sistem yang spesifik dan realistis.

2. *Workshop Design*

Tahap ini melibatkan perancangan dan penyempurnaan *prototype* melalui serangkaian *workshop*. Kegiatan dalam tahap ini meliputi:

2.1 Pengembangan *Prototype*:

Membuat model awal (*prototype*) dari aplikasi berdasarkan kebutuhan yang telah diidentifikasi pada tahap *requirement planning*. Langkah perancangan sistem adalah dengan menentukan *flowchart*.



Gambar 1. Flowchart

## 2.2 Kolaborasi dengan Pengguna:

Mengadakan *workshop* di mana pengguna berinteraksi langsung dengan *prototype* yang telah dibuat. Pengguna memberikan umpan balik mengenai fungsi dan fitur aplikasi.

## 2.3 Analisis Umpan Balik:

Menganalisis umpan balik dari pengguna untuk mengidentifikasi kekurangan dan area yang perlu ditingkatkan.

## 2.4 Perbaikan dan Penyempurnaan:

Melakukan perbaikan dan penyempurnaan pada *prototype* berdasarkan umpan balik dari *workshop*. Proses ini berulang hingga *prototype* memenuhi kebutuhan pengguna.

## 3. Implementation

Pada tahap implementasi, aspek-aspek bisnis dan teknis yang telah disetujui selama *workshop design* diintegrasikan ke dalam sistem yang sesungguhnya. Kegiatan dalam tahap ini meliputi:

### 3.1 Coding:

Menulis kode program sesuai dengan desain yang telah disepakati. Pada penelitian ini, pengembangan aplikasi menggunakan bahasa pemrograman DART dengan platform *Android*.

### 3.2 Testing:

Menguji aplikasi untuk memastikan bahwa semua fungsi bekerja dengan baik dan sesuai dengan kebutuhan pengguna. Pengujian dilakukan dalam beberapa tahap, mulai dari *unit testing*, *integration testing*, hingga *user acceptance testing* (UAT).

### 3.3 Deployment:

Mengimplementasikan aplikasi ke lingkungan yang akan digunakan oleh pengguna. Proses ini melibatkan instalasi aplikasi pada perangkat pengguna dan memastikan bahwa aplikasi berjalan dengan lancar.

### 3.4 Training dan Support:

Memberikan pelatihan kepada pengguna tentang cara menggunakan aplikasi dan menyediakan dukungan teknis jika diperlukan.

## HASIL DAN PEMBAHASAN

### A. Hasil

Pada bagian ini, akan dijelaskan mengenai hasil dari pengembangan aplikasi navigasi berbasis *Android* yang berfokus pada kalkulasi *back azimuth* dan deteksi deviasi secara *real-time*. Proses pengembangan ini menggunakan metode *Rapid Application Development* (RAD), yang terdiri dari tiga tahap utama: *requirement planning*, *workshop design*, dan *implementation*.

#### 1. Requirement Planning

Pada tahap *requirement planning*, penulis mengumpulkan informasi melalui observasi dan wawancara dengan rekan sesama siswa. Dari hasil observasi dan wawancara, diperoleh beberapa kebutuhan utama yang harus dipenuhi oleh aplikasi, antara lain:

##### 1.1 Kalkulasi Back-Azimuth

Aplikasi mampu melakukan kalkulasi *back azimuth* secara otomatis.

##### 2.1 Deteksi Deviasi

Aplikasi harus mendeteksi deviasi dari jalur yang ditetapkan secara *real-time*.

#### 2. Workshop Design

Pada tahap *workshop design*, *prototype* awal aplikasi dikembangkan dan diuji coba oleh para pengguna. Beberapa umpan balik yang diperoleh selama *workshop design* antara lain:

##### 2.1 Input Azimuth

Pengguna dapat memasukkan nilai *azimuth* awal.

##### 2.2 Kalkulasi Back Azimuth: Aplikasi menghitung *back azimuth* berdasarkan input dari pengguna.

##### 2.3 Pelacakan Posisi Real-Time: Aplikasi melacak posisi pengguna secara *real-time*.

##### 2.4 Deteksi Deviasi: Aplikasi mendeteksi apakah pengguna keluar dari jalur yang ditentukan dan memberikan notifikasi serta getaran jika terjadi deviasi.

#### 3. Implementation

Pada tahap implementasi, aplikasi dikembangkan menggunakan bahasa pemrograman DART dan platform *Android*. Fitur utama aplikasi meliputi:

##### 3.1 Kalkulasi Back Azimuth

Pengguna dapat memasukkan *azimuth* awal, dan aplikasi akan menghitung *back azimuth* secara otomatis.

##### 3.2 Deteksi Deviasi Real-Time

Aplikasi melacak posisi pengguna secara *real-time* dan mendeteksi jika pengguna keluar jalur. Jika pengguna keluar jalur, aplikasi akan menampilkan notifikasi dan memberikan getaran.

### 3.3 Navigasi *Visual*:

Aplikasi menyediakan panduan *visual* untuk membantu pengguna tetap di jalur yang benar.

## B. Pembahasan

### 1. Analisis Hasil Pengembangan

Hasil pengembangan aplikasi menunjukkan bahwa penggunaan metode RAD dalam pengembangan aplikasi navigasi berbasis *Android* ini sangat efektif. Dengan melibatkan pengguna akhir dalam proses desain dan uji coba, penulis dapat memastikan bahwa aplikasi yang dihasilkan benar-benar sesuai dengan kebutuhan pengguna.

### 2. Keunggulan dan Kelemahan Aplikasi

#### 2.1 Keunggulan:

##### a. *User-Friendly*

Antarmuka aplikasi dirancang dengan sederhana sehingga mudah digunakan oleh siapa saja.

##### b. Akurasi Kalkulasi

Aplikasi mampu menghitung *back azimuth* dengan akurat. Menggunakan rumus:

$$\text{Back Azimuth} = \frac{(\text{azimuth} + 180^\circ)}{360^\circ}$$

##### c. Pelacakan Lokasi Pengguna

Aplikasi dapat melacak secara akurat posisi pengguna dengan memanfaatkan GPS yang ditandai dengan titik *latitude* dan *longitude*

##### d. *Real-Time Tracking*

Pelacakan posisi pengguna berjalan secara *real-time* tanpa ada keterlambatan

##### e. Notifikasi dan Getaran

Fitur notifikasi dan getaran membantu pengguna tetap berada di jalur yang benar.

#### 2.2 Kelemahan:

##### a. Ketergantungan pada Sinyal GPS

Aplikasi sangat bergantung pada kualitas sinyal GPS, yang mungkin tidak selalu baik di area yang sangat terpencil atau tertutup.

##### b. Baterai Handphone

Penggunaan aplikasi yang terus menerus dapat menguras baterai ponsel seiring waktu berjalan.

### 3. Rekomendasi Pengembangan Lebih Lanjut

Berdasarkan hasil dan pembahasan di atas, beberapa rekomendasi untuk pengembangan lebih lanjut adalah:

#### a. Integrasi dengan Peta Digital

Mengintegrasikan aplikasi dengan peta digital untuk memberikan visualisasi jalur yang lebih baik.

#### b. Peningkatan Algoritma Deteksi Deviasi

Mengoptimalkan algoritma deteksi deviasi untuk meningkatkan kecepatan dan akurasi.

#### c. Fitur Tambahan

Menambahkan fitur tambahan seperti penanda titik penting (*waypoints*) dan integrasi dengan perangkat *wearable* untuk notifikasi yang lebih efektif. Menambahkan fitur *history* derajat yang telah dimasukkan pengguna, sehingga

pengguna tetap dapat memantau dari mana pengguna berangkat.

#### 4. Tampilan Aplikasi

Berikut tampilan dari aplikasi navigasi:

##### 4.1 Tampilan *splash*

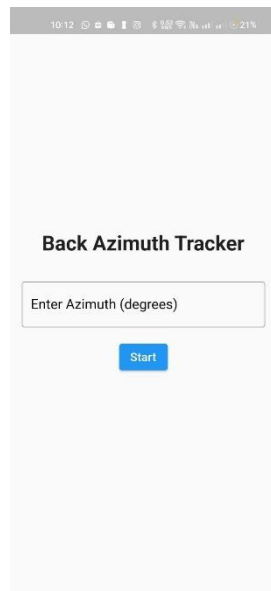
Ini adalah tampilan awal saat pengguna membuka aplikasi



Gambar 2. *Splash screen*

##### 4.2 Tampilan awal dan input derajat *azimuth*

Setelah *splash screen* pengguna akan masuk ke tampilan awal untuk memasukkan derajat *azimuth*



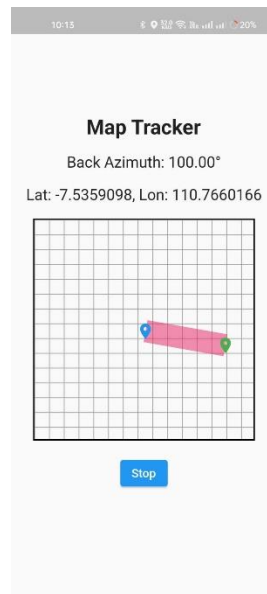
Gambar 3. *Tampilan awal*



Gambar 4. *Tampilan input azimuth*

#### 4.3 Tampilan Map Tracker

Setelah pengguna memasukkan derajat *azimuth* dan menekan tombol “start”, aplikasi akan mengkalkulasi *back-azimuth* dan melacak lokasi pengguna dengan bantuan GPS. Serta menampilkan garis lurus yang akan menjadi *track* (garis pink) pengguna.



Gambar 5. Tampilan map tracker

#### 4.4 Notifikasi jika terjadi deviasi

Jika pengguna keluar dari *track* aplikasi akan memberi notifikasi berupa getaran dan visual peringatan “anda keluar jalur”. Pengguna hanya perlu menekan tombol “OK” untuk melanjutkan navigasi dan kembali ke *track*.



Gambar 6. Notifikasi deviasi

### 5. Pengujian

Langkah terakhir adalah menguji fungsi dari sistem yang telah dibangun. Pengujian ini dilakukan menggunakan metode *black box*.



### 5.1 Metode Pengujian *Black box*

*Black box* merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak. Pengujian *black box* bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi. (Ningrum et al., 2019). Teknik pengujian *black box* yang digunakan dalam pengujian ini adalah:

#### 1. *Equivalence Partitioning*

Teknik ini membagi data uji menjadi beberapa partisi yang setara, untuk memastikan bahwa setiap partisi diperlakukan sebagai satu kesatuan. Misalnya, dalam pengujian input *azimuth*, data dibagi menjadi rentang nilai valid dan invalid. Kasus uji yang mewakili setiap rentang ini dijalankan untuk memastikan bahwa aplikasi dapat menangani semua kemungkinan input secara efektif.

#### 2. *Boundary Value Analysis*

Teknik ini berfokus pada pengujian nilai-nilai batas input. Dalam aplikasi ini, batas-batas seperti  $0^\circ$  dan  $360^\circ$  untuk *azimuth* diuji untuk memastikan bahwa aplikasi dapat menangani nilai-nilai ekstrem dengan benar. Pengujian juga dilakukan pada nilai-nilai yang sedikit di luar batas normal untuk memverifikasi respons aplikasi terhadap input yang mendekati batas.

#### 3. *Decision Table Testing*

Teknik ini digunakan untuk menguji kombinasi masukan yang berbeda dengan menggunakan tabel keputusan. Misalnya, tabel keputusan digunakan untuk memverifikasi bagaimana aplikasi berperilaku ketika *azimuth* diberikan dalam berbagai kondisi, seperti input valid, input invalid, atau kondisi ketika deviasi terdeteksi.

#### 4. *State Transition Testing*

Teknik ini digunakan untuk memastikan bahwa aplikasi berfungsi dengan benar dalam berbagai kondisi yang berbeda. Dalam konteks aplikasi navigasi, pengujian dilakukan untuk memverifikasi bahwa aplikasi dapat beralih dengan benar antara berbagai kondisi seperti mode perhitungan *back azimuth*, mode pelacakan posisi, dan mode deteksi deviasi.

### 5.2 Hasil Pengujian Sistem

Berikut adalah hasil pengujian dari beberapa fungsi utama aplikasi navigasi dengan menggunakan metode *black box*:

Tabel 1. Metode *black box*

No	Fitur yang Diuji	Skenario Pengujian	Input	Output yang Diharapkan	Hasil Pengujian
1	Input <i>Azimuth</i>	Pengguna memasukkan nilai <i>azimuth</i> awal	<i>Azimuth</i> awal ( $0^\circ - 360^\circ$ )	Aplikasi menerima nilai dan menampilkan <i>azimuth</i> awal	Berfungsi baik
2	Kalkulasi <i>Back Azimuth</i>	Aplikasi menghitung <i>back azimuth</i> berdasarkan input <i>azimuth</i> awal	<i>Azimuth</i> awal ( $0^\circ - 360^\circ$ )	Aplikasi menghitung dan menampilkan <i>back azimuth</i> yang benar	Berfungsi baik

No	Fitur yang Diuji	Skenario Pengujian	Input	Output yang Diharapkan	Hasil Pengujian
3	Pelacakan Posisi <i>Real-Time</i>	Aplikasi melacak posisi pengguna secara <i>real-time</i>	Koordinat GPS	Aplikasi menampilkan posisi pengguna pada peta secara <i>real-time</i>	Berfungsi baik
4	Deteksi Deviasi	Aplikasi mendeteksi deviasi pengguna dari jalur yang ditentukan	Koordinat GPS, Jalur yang ditentukan	Aplikasi memberikan notifikasi dan getaran jika pengguna keluar dari jalur, tidak ada notifikasi jika tetap pada jalur	Berfungsi baik
5	Notifikasi dan Getaran	Pengguna keluar dari jalur yang ditentukan	Koordinat GPS	Aplikasi memberikan notifikasi "Anda keluar jalur" dan getaran	Berfungsi baik
6	Status Aman	Pengguna tetap pada jalur yang ditentukan	Koordinat GPS	Aplikasi tetap berjalan normal tanpa memberikan notifikasi	Berfungsi baik

## KESIMPULAN

Penelitian ini berhasil dikembangkan aplikasi navigasi berbasis *Android* yang fokus pada kalkulasi *back azimuth* dan deteksi deviasi secara *real-time*. Menggunakan metode *Rapid Application Development (RAD)*, aplikasi ini dirancang untuk meningkatkan efektivitas navigasi di medan yang kompleks.

Hasil pengujian sistem dengan metode *black box* menunjukkan aplikasi berfungsi sesuai spesifikasi, meskipun ada beberapa area untuk perbaikan. Aplikasi ini terbukti efektif dalam memberikan informasi navigasi yang akurat dan mendeteksi deviasi, dengan potensi besar untuk meningkatkan navigasi di kondisi sulit. Pengembangan dan pengujian lebih lanjut diharapkan dapat meningkatkan stabilitas dan fungsionalitas aplikasi.

#### DAFTAR PUSTAKA

- Pratama, N., Nurdiawan, A., & Pramudita, H. (2019). *Rapid Application Development (RAD)* pada Pengembangan Sistem Informasi *Monitoring* Pajak Bumi dan Bangunan. *Jurnal Teknologi Informasi*, 15(2), 75-84. <https://doi.org/10.31294/p.v21i2>
- F. C. Ningrum, D. Suherman, S. Aryanti, H. A. Prasetya, dan A. Saifudin. (2019). "Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions," *Jurnal Informasi Universitas Pamulang*, 4(4), 125–130. <https://doi.org/10.32493/informatika.v4i4.3782>
- Rajeev, S., Samani, A., Panetta, K., Agaian, S. (2019). *3D Navigational Insight Using AR Technology*, 2019 *IEEE International Symposium on Technologies for Homeland Security (HST)*, 1–4. <https://doi.org/10.1109/HST47167.2019.9033006>
- Rosia, I., Derta, S., Efriyanti, L., & Okra, R. (2022). Penerapan Aplikasi Arcgis Dalam Pembuatan Peta Topografi Pada Pendidikan Navigasi Darat Mpa Jamarsingsia Iain Bukittinggi. *KOLONI*, 1(3), 862-871. <https://doi.org/10.31004/koloni.v1i3.255>