

# PERANCANGAN APLIKASI RUTE TERCEPAT PERJALANAN PAKET BERBASIS MOBILE MENGGUNAKAN METODE ROBOTIC PROCESS AUTOMATION (RPA)

## APPLICATION DESIGNING FASTEST ROUTES TRAVEL PACKAGE BASE ON MOBILE USING ROBOTIC PROCESS AUTOMATION (RPA) METHOD

Aditya Hutomo<sup>1</sup>, Anton Siswo Raharjo<sup>2</sup>, M. Husni Syahbani<sup>3</sup>

<sup>1,2,3</sup> Universitas Telkom, Bandung

uncledrew@student.telkomuniversity.ac.id<sup>1</sup>, raharjo@telkomuniversity.ac.id<sup>2</sup>,  
eksmud@telkomuniversity.ac.id<sup>3</sup>

### Abstrak

Ekspedisi pengiriman paket merupakan perusahaan yang menyediakan pelayanan jasa berupa pengiriman paket. Pengiriman paket dilakukan oleh seorang kurir yang mengandalkan pengetahuan suatu wilayah tetapi pengiriman paket yang mengandalkan pengetahuan suatu wilayah seorang kurir berdampak pada waktu pengiriman yang kurang efisien dan menyebabkan paket sampai tanpa ada seorang penerima. Dengan adanya permasalahan tersebut dirancangkan sebuah aplikasi yang dapat mengirimkan paket dengan pengurutan rute tercepat. Data alamat diambil dari *Google Maps*, maka dengan menggunakan metode *Robotic Process Automation* (RPA) yang dapat mencari jarak terpendek. Pada akhirnya aplikasi ini akan menghasilkan keluaran urutan rute alamat tercepat. Dengan demikian diharapkan aplikasi ini dapat meningkatkan pengiriman paket yang efisien. Berdasarkan hasil pengujian simulasi pencarian jarak menggunakan RPA, RPA melakukan pencarian sebanyak 100 kali percobaan, pada penelitian ini metode *Robotic Process Automation* (RPA) berhasil mencari jarak antar koordinat yang sesuai.

**Kata Kunci:** *Pengiriman paket, Robotic Process Automation (RPA), Destinasi perjalanan paket*

### Abstract

*Package delivery expedition is a company that provides services in the form of package delivery. Package delivery is carried out by a courier who relies on knowledge of a region but the delivery of a package that relies on the knowledge of a courier's area has an impact on inefficient delivery times and causes packages to arrive without a recipient. With these problems, an application is designed that can send packets with the fastest route sorting. Address data is taken from Google Maps, then using Robotic Process Automation (RPA) is a method that can find the shortest distance. In the end this application will produce the fastest address route sequence output. Thus, it is hoped that this application can improve efficient package delivery. Based on the results of the distance search simulation test using RPA, RPA searched 100 times, in this study the Robotic Process Automation (RPA) method succeeded in finding the appropriate distance between coordinates.*

**Keywords:** *Package Delivery, Robotic Process Automation (RPA), Travel Package Destination*

## 1. Pendahuluan

Indonesia memiliki beragam situs perdagangan elektronik (*Electronic Commerce / E-Commerce*) diantaranya seperti, Tokopedia, Blibli.com, dan Bukalapak. Mereka merupakan tempat perdagangan online bagi para penjual yang belum memiliki toko fisik tetapi sudah memiliki barang fisik yang akan dijual.

Para E-Commerce ini menjalin kerja sama pada pihak ke dua untuk melakukan pengiriman barang yaitu, perusahaan yang bergerak dibidang Ekspedisi seperti, JNE, J&T, dan SiCepat. Untuk itu para penjual yang telah menjual barang mereka pada salah satu E-Commerce hanya dapat mengirimkan barang melalui perusahaan ekspedisi yang telah menjalin kerja sama pada E-Commerce tersebut.

Standar pengiriman paket, pengirim paket dan penerima paket mencantumkan alamat lengkap beserta nomor telepon agar paket sampai ke penerima paket, kurir pengantar paket tentu harus memiliki pengetahuan zona wilayah tertentu. Berdasarkan observasi yang dilakukan peneliti. Pengiriman paket oleh kurir pada umumnya tidak hanya satu paket, melainkan beberapa paket yang akan sampai sesuai pada waktu estimasi kedatangan paket, umumnya seorang kurir akan mengirimkan beberapa paket tersebut ke alamat tujuan satu demi satu, terbatasnya pengetahuan kurir akan zona wilayah pada peta geografis dapat menyebabkan belum adanya pencarian rute terdekat hingga terjauh dan kurang efisien waktu kurir dalam pengantaran paket.

Perkembangan Peta digital dan GPS (Global Position System) yang dapat memantau letak keberadaan dirinya pada peta. Penggabungan dua teknologi tersebut menjadi suatu sistem yang dapat mengarahkan seseorang berpergian kesuatu tempat yang dinamakan navigasi atau pandu arah yang akan memandu perjalanan seseorang sampai ketujuan dengan menghitung kemungkinan rute tercepat [1].

Untuk mengatasi masalah tersebut dibuatlah aplikasi navigasi destinasi perjalanan paket berbasis mobile dengan metode Robotic Process Automation (RPA) yang di rancang untuk membantu kurir pengantar paket dalam mengurutkan rute terdekat hingga rute terjauh dan membuat waktu pengiriman paket lebih efisien [2].

## 2. Dasar Teori

### 2.1. *Robotic Process Automation (RPA)*

Pada proses perhitungan rute jarak terdekat penelitian ini menggunakan teknologi otomatisasi yang bernama Robotic Process Automation (RPA), merupakan sebuah software robot yang digunakan untuk melakukan task computer yang terstruktur, rutin, berulang dan akan menjadi lebih optimal pemanfaatannya jika dilakukan dalam volume yang besar. RPA dapat bekerja dalam 2 tipe mode yaitu attended mode dan unattended mode. Bot dengan mode attended akan membutuhkan manusia untuk menjalankannya, sedangkan robot dengan mode unattended dapat dijadwalkan atau dapat dijalankan dari sebuah event.

1. Attended bot artinya sebuah software robot yang memanipulasi program front-office yang sama dengan yang digunakan oleh pengguna akhir, dengan kata lain attended bot dapat disebut sebagai asisten pribadi.
2. Unattended bot artinya sebuah software bot yang digunakan untuk fungsi back-office yang memiliki dampak yang lebih luas pada alur kerja. Biasanya software ini berjalan di server organisasi dengan sedikit atau tanpa campur tangan manusia, dan dapat berjalan pada jadwal yang telah ditentukan atau secara real time, 24/7/365.

### 2.2. *Global Positioning System (GPS)*

GPS (*Global Positioning System*) adalah suatu sistem navigasi menggunakan lebih dari 24 satelit MEO (*Medium Earth Orbit* atau *Middle Earth Orbit*) [1]. yang mengelilingi bumi sehingga penerima sinyal di permukaan bumi dapat menerima sinyalnya. GPS mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan, arah, dan waktu. Satelit mengorbit pada ketinggian 12.000 mil di atas bumi dan mampu mengelilingi bumi dua kali dalam 24 jam. Satelit GPS secara kontinyu mengirimkan sinyal radio digital yang mengandung data lokasi satelit dan waktu, pada penerima yang berhubungan.

### 2.3. REST API

REST API merupakan salah satu dari desain arsitektur yang terdapat di dalam API itu sendiri. Dan cara kerja dari RESTful API yaitu REST client akan Melakukan akses pada data/resource pada REST server dimana masing-masing resource. Atau data/resource tersebut akan dibedakan oleh sebuah global ID atau URIs (Universal Resource Identifiers). Jadi, Nantinya data yang diberikan oleh REST server itu bisa berupa format text, JSON atau XML. Dan saat ini format yang paling populer dan paling banyak digunakan adalah format JSON.

Adapun metode HTTP yang secara umum dipakai dalam REST api adalah:

1. GET, berfungsi untuk membaca data/resource dari REST server
2. POST, berfungsi untuk membuat sebuah data/resource baru di REST server
3. PUT, berfungsi untuk memperbaharui data/resource di REST server
4. DELETE, berfungsi untuk menghapus data/resource dari REST server
5. OPTIONS, berfungsi untuk mendapatkan operasi yang disupport pada resource dari REST server.

### 2.4. *Google Maps API*

Peta Google API (*Application Programming Interface*) menyediakan fasilitas untuk menampilkan, memanipulasi peta beserta *feature* lainnya seperti tampilan satelit, jalan yang disediakan oleh google untuk menggunakan peta google dalam aplikasi yang dibangun.

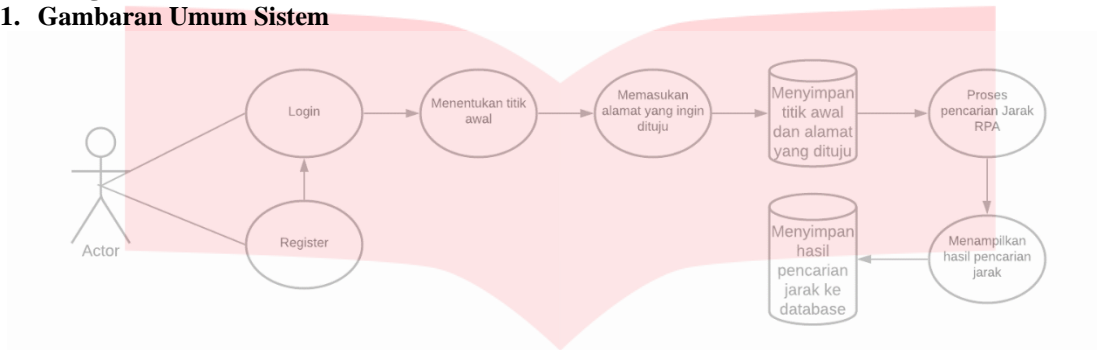
Google Maps API memungkinkan para pengembang aplikasi maupun teknologi untuk menggunakan fasilitas layanannya dan mengizinkan para pengguna layanan ini untuk memodifikasi peta dan informasi yang ada didalamnya

2.5. *Firestore Realtime*

Firestore Realtime Database, dengan SDK klien dan kemampuan real-time-nya, adalah tentang membuat pengembangan aplikasi lebih cepat dan lebih mudah. Sejak diluncurkan, telah diadopsi oleh ratusan ribu pengembang, dan seiring dengan pertumbuhan adopsi, begitu pula pola penggunaan. Pengembang mulai menggunakan Realtime Database untuk data yang lebih kompleks dan untuk membangun aplikasi yang lebih besar, mendorong batas model data JSON dan kinerja database dalam skala besar.

3. Metodologi Penelitian

3.1. Gambaran Umum Sistem



Gambar 3.1. 1 Gambaran Umum Sistem

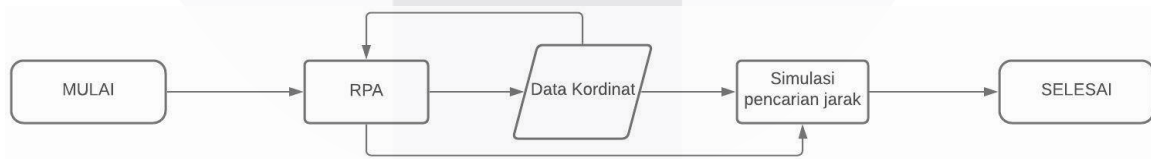
Pada aplikasi *mobile* yang akan dirancang nantinya dapat membantu para kurir pengantar paket agar dapat mengirimkan paket secara efisien dan memiliki pengurutan rute tercepat. Fitur yang akan didapatkan oleh kurir jika menggunakan aplikasi terapat pada gambar 3.1 1 serta penjelasanya.

Pada gambar 3.1 1 menjelaskan tentang bagaimana proses sistem yang dirancang agar dapat mengurutkan rute pengiriman pake. Proses tahapan sistem akan dilakukan sebagai berikut:

1. *User* akan melakukan *register* terlebih dahulu untuk mendapatkn sebuah akun, register berisi nama lengkap, email, tanggal lahir, password, dan verifikasi password.
2. Setelah memiliki akun *user* harus *login* terlebih dahulu, setelah *login* akan ditampilkan halaman utama.
3. Selanjutnya *user* akan memilih kolom “keberangkatan” untuk mencari alamat keberangkatan
4. Setelah itu *user* akan melakukan input alamat pada kolom “tujuan”.
5. Setelah semua tujuan yang akan dikirimkan dimasukan, maka sistem akan menyimpan data alamat ke database untuk proses selanjutnya
6. Selanjutnya RPA membaca database untuk mencari seluruh jarak dari titik keberangkatan sampai dengan alamat yang telah dikirim database
7. Data alamat yang telah ditemukan jaraknya kemudian disimpan dalam database untuk ditampilkan dan disimpan dalam menu utama

3.2. Perancangan Sistem

3.2.1. Pemodelan Simulasi Pencarian Jarak Pada RPA



Gambar 3.4.2 1 Pemodelan RPA

$$Kb = Kd^2$$

Keterangan:

Kb = Kordinat berangkat

Kd = Kordinat destinasi

Perhitungan:

Tabel 3.4.2 1 Kordinat RPA

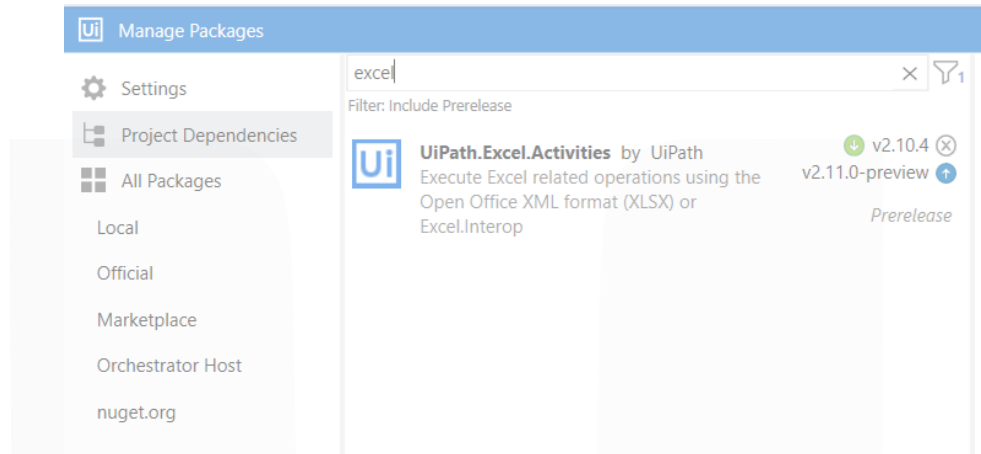
Kordinat berangkat	Kordinat destinasi
1 – 100 kordinat	100 kordinat

$100 \text{ Kb} = 100 \text{ Kd}^2 = 10.000 \text{ kordinat}$

Pada gambar diatas merupakan gambaran bagaimana cara RPA bekerja dalam melakukan proses pencarian jarak, RPA membaca sebuah data kordinat, data kordinat tersebut merupakan arsip berupa *longitude* dan *latitude*, nantinya data kordinat tersebut akan diambil satu per-satu untuk dinisialisasikan dan juga mencari jarak antar titik alamat tersebut. Berikut merupakan aktivitas yang dibuat dalam UiPath agar RPA dapat melakukan pekerjaanya:

**1. Pemilihan Aktivitas**

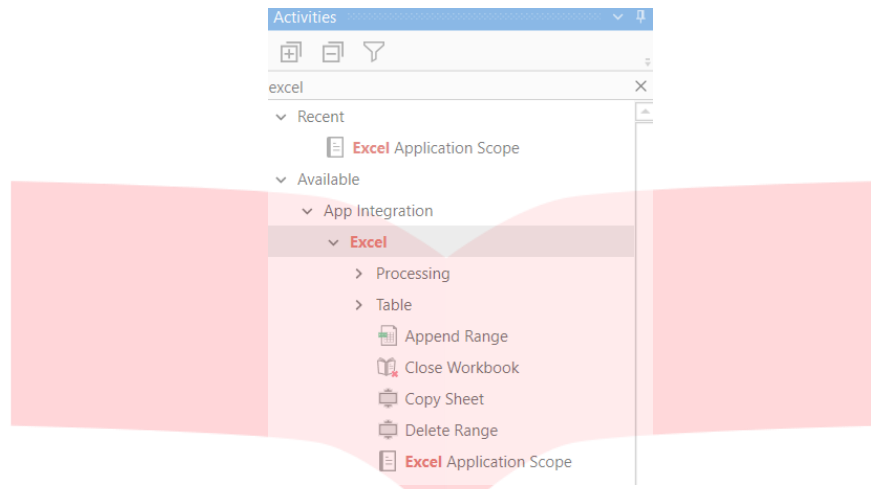
*Robotic Process Automation (RPA)* merupakan sebuah sistem otomasi yang digunakan tanpa membuat sebuah *source code* tetapi pada dasarnya RPA merubah *source code* menjadi sebuah *activity* yang sama hal nya seperti perintah yang dilakukan oleh *source code*.



Gambar 3.4.2 2 Pemilihan Aktivitas

Pada kasus kali ini aktivitas ekstensi yang dibutuhkan oleh RPA adalah *excel activities*, jadi untuk menambahkannya hanya tinggal masuk kedalam menu *manage package* pada UiPath lalu cari pada kolom pencarian dan unduh.

## 2. Kolom Aktivitas

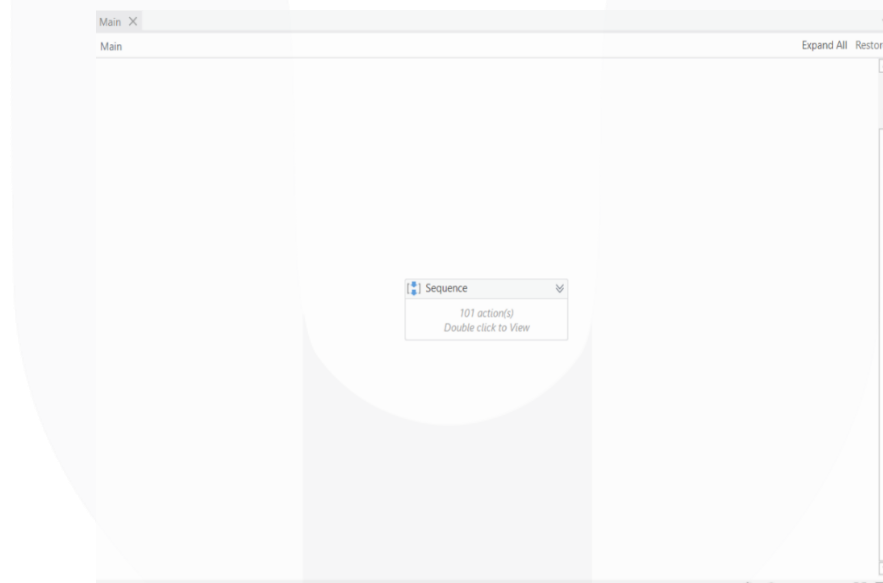


Gambar 3.4.2 3 Kolom Aktivitas

Setelah melakukan pengunduhan ekstensi aktivitas pada menu utama dibagian aktifitas jika dilakukan pencarian berupa “excel” terdapat kumpulan aktifitas yang excel butuhkan

## 3. Main Workflow

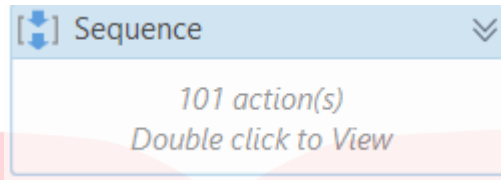
Pada gambar dibawah ini merupakan bagian alur kerja utama, merupakan bagian untuk membuat sebuah racangan proses otomasi yang dibutuhkan.



Gambar 3.4.2 4 Main Workflow

#### 4. Pembuatan Urutan Utama Aktivitas

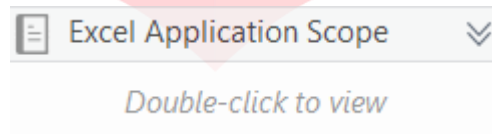
Pada tahap awal untuk membuat sebuah aktifitas pada RPA diharuskan membuat sebuah urutan (*Sequence*), urutan ini merupakan aktifitas yang akan RPA lakukan secara berurut.



Gambar 3.4.2 5 Pembuatan Urutan Utama Aktivitas

#### 5. Pemanggilan Aktifitas Lingkup Excel

Setelah membuat urutan utama selanjutnya didalam urutan tersebut dilakukan sebuah pemanggilan aktifitas untuk membaca arsip excel.

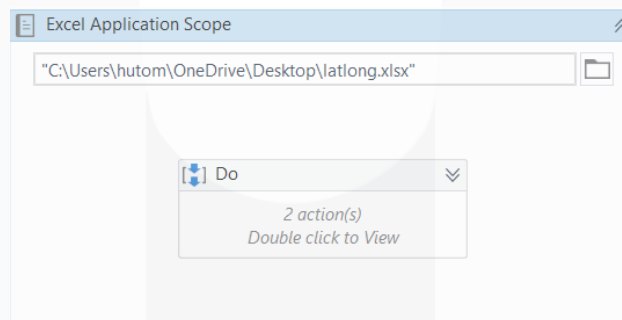


Gambar 3.4.2 6 Pembuatan Urutan Utama Aktivitas

Aktivitas tersebut adakah Excel Application Scope, aktifitas ini merupakan perintah untuk RPA agar dapat melakukan akses ke dalam lingkup excel.

#### 6. Pengisian Direktori

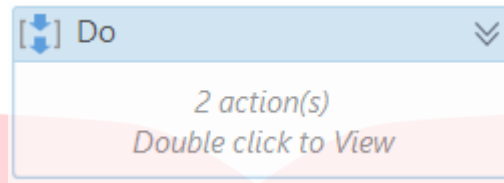
Pada aktifitas *excel Application Scope* terdapat sebuah kolom direktori agar aktifitas ini dapat mencari keberadaan arsip excel. Pada gambar dibawah ini sudah terisi sebuah direktori "C:\User\hutom\OneDrive\latlong.xlsx" yang merupakan alamat direktori arsip excel yang akan RPA cari.



Gambar 3.4.2 7 Pengisian Direktori

## 7. Aktivitas Do

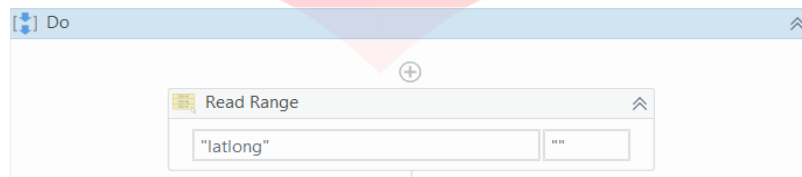
Didalam aktifitas *excel application scope* sudah terpanggil secara otomatis sebuah aktifitas *do* aktifitas ini merupakan perintah kerja selanjutnya jika direktori yang diminta oleh *excel application scope* sudah terpenuhi.



Gambar 3.4.2 8 Aktivitas Do

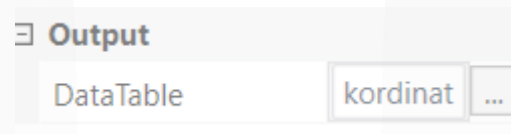
## 8. Pembacaan Halaman Pada Excel

Pada aktivitas *do* dimasukan sebuah aktivitas yang merupakan bagian dari ekstensi aktivitas excel yaitu *read range*.



Gambar 3.4.2 9 Pembacaan Halaman Pada Excel

Aktivitas pada gambar diatas berfungsi untuk membaca halaman excel pada direktori yang telah ditentukan pada aktivitas *excel application scope*. Lalu pada aktivitas *do* terdapat sebuah properti output.

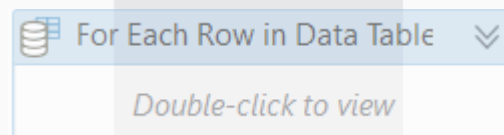


Gambar 3.4.2 10 Output Data Table

*output* ini berfungsi untuk menyimpan sementara didalam sebuah tabel data dengan variabel “kordinat” setelah dilakukannya pembacaan halaman oleh RPA.

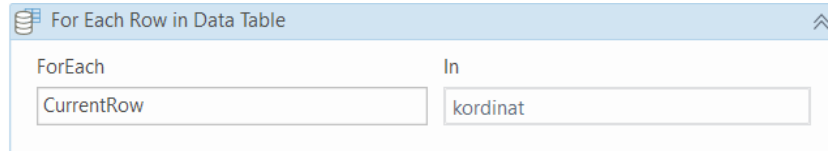
## 9. Pemanggilan For Each Row

Aktivitas ini merupakan sebuah perulangan yang akan dilakukan dengan memasukan hasil dari *output* aktivitas *read range* dengan varibel kordinat.

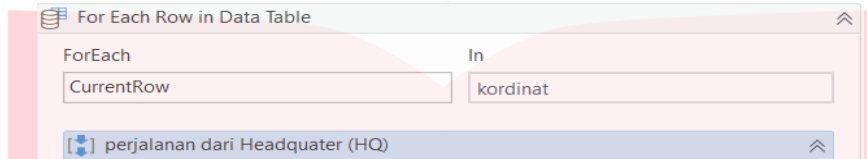


Gambar 3.4.2 11 Pemanggilan ForEach Row

Didalam aktivitas *For Each Row* terdapat sebuah kolom, kolom pertama merupakan kolom *foreach* yang merupakan nama dari kolom yang dibuat secara virtual oleh RPA, kolom kedua merupakan kolom *input*, input tersebut merupakan *output* yang dihasilkan oleh aktivitas *read range* yang berupa sebuah tabel data virtual.

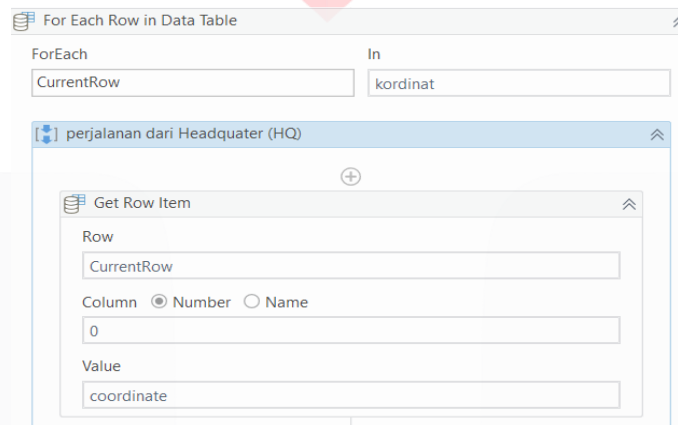


Gambar 3.4.2 12 ForEach in Data Tabel



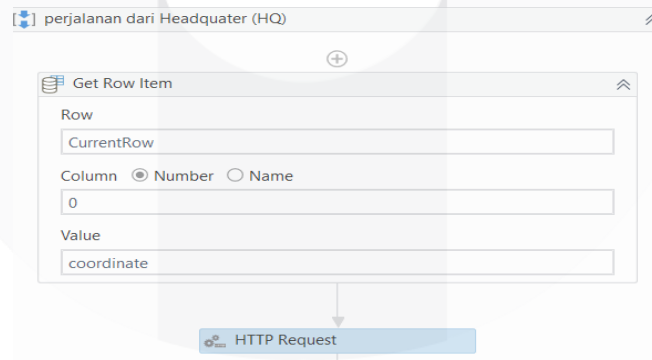
Gambar 3.4.2 13 ForEach Data Table

Lalu dalam aktivitas *foreach row* sudah secara otomatis terpanggil sebuah urutan (*sequence*) yang akan dilakukan selanjutnya oleh aktivitas *foreach row*. Pada urutan yang terdapat pada aktivitas *foreach row* dilakukan sebuah pemanggilan aktifitas *Get Row Item*.



Gambar 3.4.2 14 Get Row Item

Aktivitas ini berfungsi untuk melakukan pengambilan sebuah benda yang merupakan *longitude* dan *latitude* secara satu per-satu dan berurut berdasarkan nomor yang menghasilkan *output* dengan variabel *coordinate*.



Gambar 3.4.2 15 HTTP Request

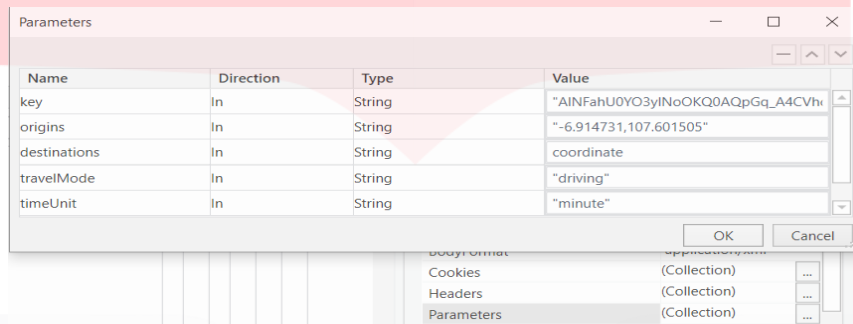
Setelah didapatkan sebuah *output* oleh aktivitas *Get row item* selanjutnya dilakukan pemanggilan aktivitas *HTTP Request*, aktivitas ini berguna untuk melakukan permintaan layanan kepada *Google Maps API*.





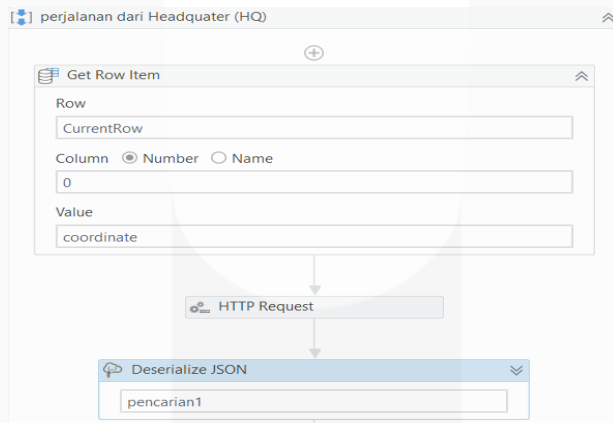
Gambar 3.4.2 16 Input

Pada aktivitas *HTTP Request* terdapat sebuah properti input yang memiliki parameter seperti pada gambar diatas, pada parameter *AcceptFormat* merupakan format Bahasa yang diterima oleh aktivitas *HTTP Request* sedangkan untuk parameter *EndPoint* merupakan situs layanan yang disediakan oleh *Google Maps*. dan pada parameter *Method* merupakan salah satu metode pada http, pada kasus ini metode yang digunakan adalah *GET* yang artinya mendapatkan infomasi yang dibutuhkan.



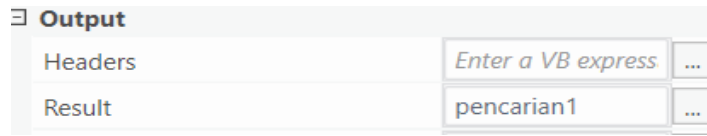
Gambar 3.4.2 17 Parameter

Lalu pada aktivitas *HTTP Request* juga terdapat sebuah properti parameter yang harus dibuat agar metode *GET* yang telah ditentukan bisa mendapatkan informasi yang dibutuhkan sesuai dengan parameter yang ditentukan.



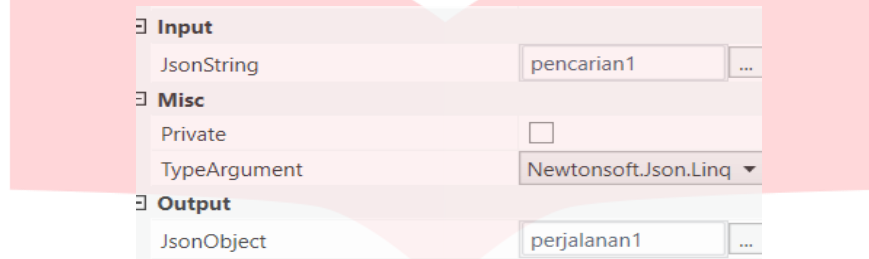
Gambar 3.4.2 18 Deserialize JSON

Setelah itu dipanggil sebuah aktivitas untuk merubah dari keluaran *HTTP Request* menjadi sebuah JSON. Setelah dilakukan pencarian jarak oleh RPA melalui layanan *Google Maps API* , google akan memilihkan jalur dengan jarak terpendek, setelah semua informasi yang dibutuhkan oleh RPA melalui aktivitas *HTTP Request* sudah sesuai , aktivitas ini membuat sebuah *outout* dengan variabel “perjalanan1”.



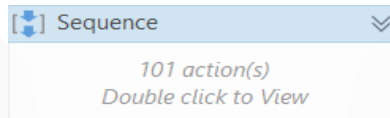
ambar 3.4.2 19 Output

Setelah *output* tersebut dibuat, selanjutnya variabel perjalanan1 dimasukkan kedalam aktivitas “*Deserialize JSON*” berupa sebuah properti input.



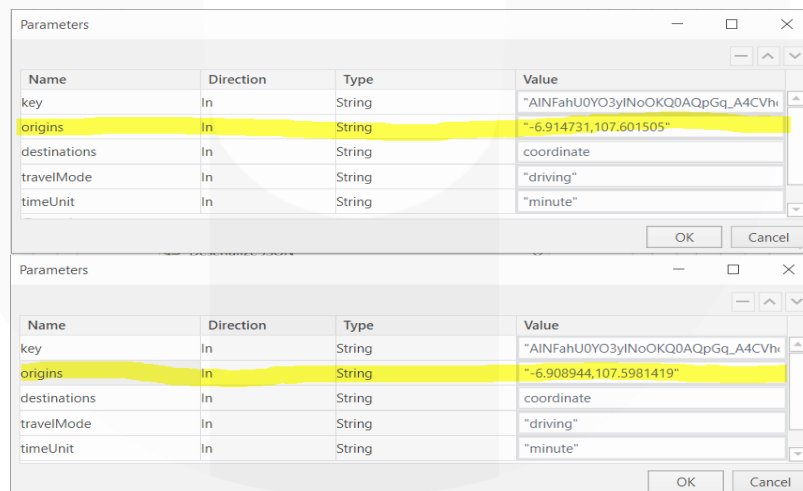
Gambar 3.4.2 20 Input

Setelah diubah menjadi JSON kemudian aktivitas ini membuat sebuah *output* dengan nama variabel yang sama yaitu “perjalanan1”.



Gambar 3.4.2 21 Sequence

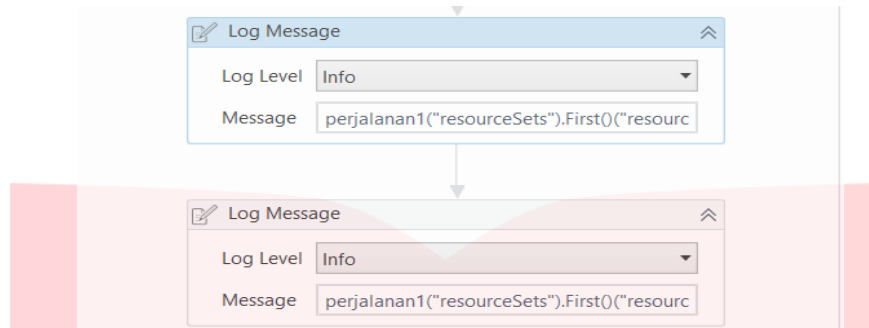
Pada urutan utama jika tampilan dikecilkan tertulis *100 action(s)* yang artinya pembuatan aktivitas diatas dilakukan secara berulang dari titik awal 1 hingga titik awal 100. Cara tersebut dilakukan agar seluruh titik awal memiliki nilai jarak agar Ketika dicari sebuah jarak terpendek dari titik awal hingga ke 100 dapat membuat sebuah rute perjalanan.



Gambar 3.4.2 22 Parameter

Yang dimaksudkan titik awal adalah pada parameter gambar diatas pada kolom *origins*, nilai dari kolom *origin* ini merupakan *longitude* dan *latitude*, aktivitas yang dibuat secara berulang selanjutnya hanya mengubah nilai *origins*.

### 10. Pemanggilan Hasil Aktivitas



Gambar 3.4.2 23 Pemanggilan Hasil

Pada tahap terakhir dibuat sebuah pemanggilan aktivitas untuk menghasilkan sebuah *output* sesuai dengan yang dibutuhkan yaitu informasi *longitude* dan *latitude*, jarak, dan waktu, Pada aktivitas “log message” dibuat sebuah fungsi pada kolom “message” yang berfungsi untuk memilih informasi yang sesuai dengan kebutuhan pada JSON.



Gambar 3.4.2 24 Fungsi

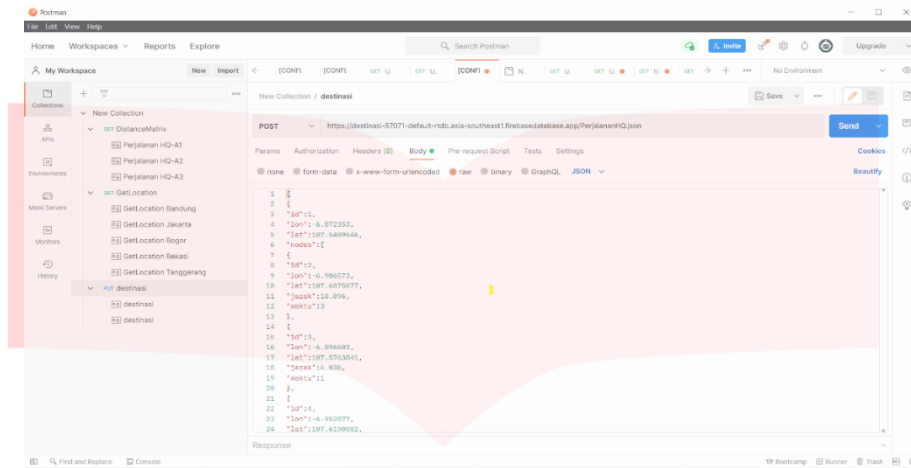


Gambar 3.4.2 25 Output JSON RPA

Didapatkan sebuah keluaran seperti gambar diatas sesuai dengan kebutuhan yang dibutuhkan.

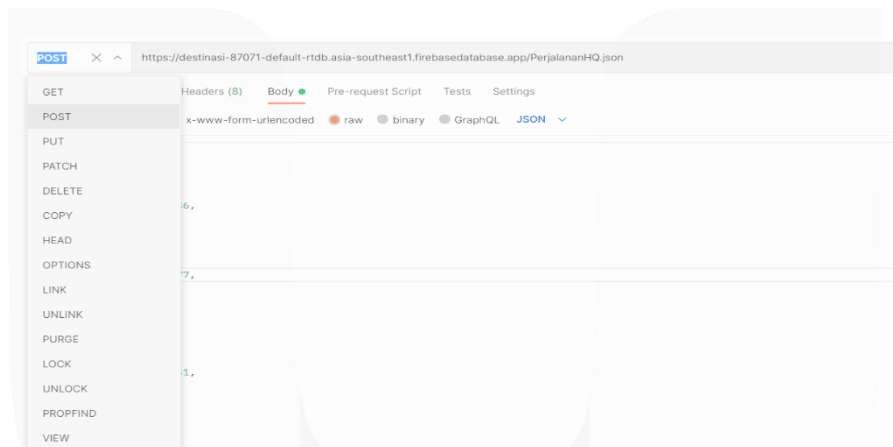
### 11. Mengunggah Data Pada Database

Setelah didapatkan sebuah keluaran JSON selanjutnya JSON ini akan dikirim ke dalam database menggunakan *POSTMAN* dengan menggunakan metode *POST* yang merupakan salah satu metode *HTTP*.



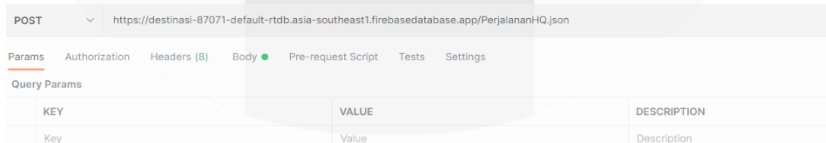
Gambar 3.4.2 26 Mengunggah Data

Gambar diatas merupakan contoh menu utama pada aplikasi *POSTMAN*, aplikasi ini berguna untuk mengunggah data JSON ke dalam database.



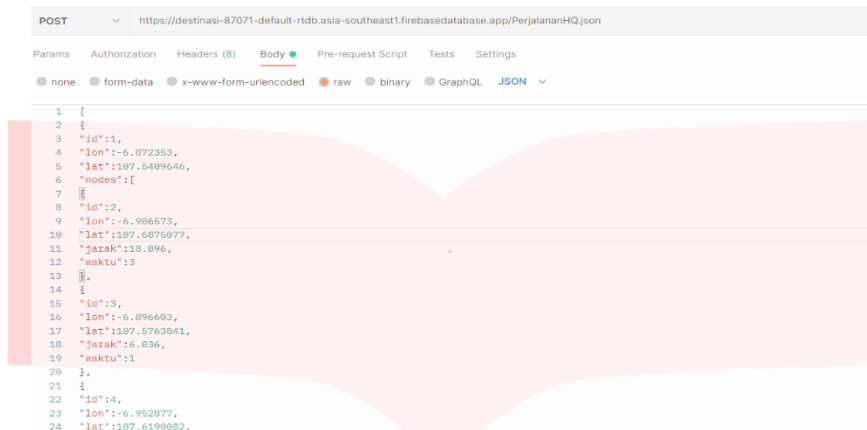
Gambar 3.4.2 27 Fitur Body

Pada gambar diatas merupakan kolom beberapa metode *HTTP* yang ingin digunakan dan pada sebelah kolom metode *HTTP*, tetapi dalam penyimpanan JSON yang telah didapat harus menggunakan metode *POST*, yang artinya mengunggah sebuah data, selanjutnya terdapat kolom *endpoint*, *endpoint* merupakan alamat URL yang akan menjadi tempat penyimpanan JSON yang ingin dikirimkan.



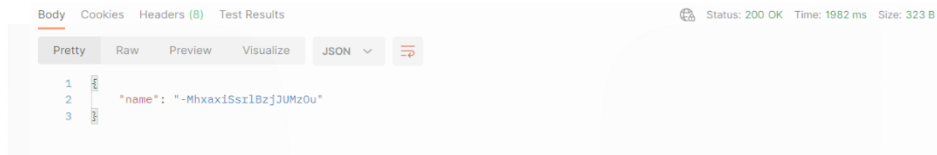
Gambar 3.4.2 28 Parameter POSTMAN

Pada gambar diatas terdapat sebuah menu *params*, menu tersebut merupakan parameter yang dapat digunakan untuk melakukan semua metode yang terdapat di dalam *HTTP* dengan menggunakan sebuah parameter.



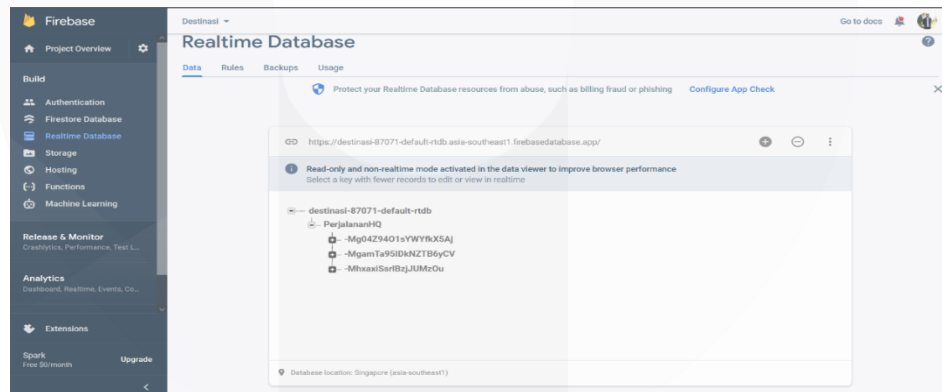
Gambar 3.4.2 29 JSON

Tetapi karena hasil JSON telah ditemukan RPA, jadi pada kasus ini hanya tinggal menyalin hasil JSON yang dilakukan oleh RPA selanjutnya dimasukkan ke dalam menu *body* pada aplikasi *POSTMAN*.



Gambar 3.4.2 30 Nama Kelas Database

Setelah melakukan pengiriman JSON ke dalam database, selanjutnya server akan merespon apakah data yang dikirimkan berhasil tersimpan dalam database atau gagal. Respon tersebut berupa angka pada kolom status, jika 200 maka data berhasil dikirim, jika 400 maka format penulisan data atau parameter didalam menu *POSTMAN* terdapat kesalahan penulisan.



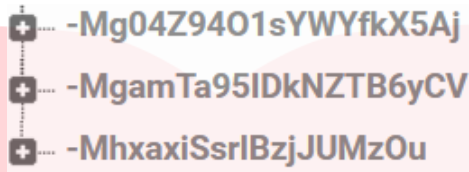
Gambar 3.4.2 31 Realtime Database

Gambar diatas merupakan database yang digunakan, database yang digunakan adalah firebase dengan menggunakan fitur *realtime database*.



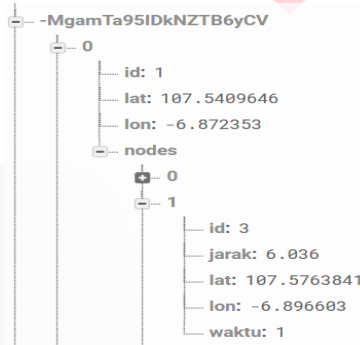
Gambar 3.4.2 32 Kelas

Pada *realtimedatabase* terdapat sebuah kelas yang telah dibuat menggunakan *endpoint* yang telah ditentukan pada aplikasi *POSTMAN*.



Gambar 3.4.2 33 Kelas Destinasi

Pada gambar merupakan nama kelas yang memiliki isi berupa *longitude*, *latitude*, jarak, dan waktu. *Longitude* dan *latitude* yang terdapat didalam kelas merupakan inisialisasi *Longitude* dan *latitude* tujuan yang telah dibandingkan dengan titik alamat awal.



Gambar 3.4.2 34 Isi Kelas

Pada gambar merupakan salah satu contoh dari isi kelas yang telah cari jaraknya antara titik awal dan semua inisialisasi titik tujuan.



Gambar 3.4.2 35 Nama Kelas Database

Gambar diatas merupakan nama kelas yang merupakan nama yang otomatis terbuat secara acak oleh *firebase*, nama kelas ini muncul setelah data yang dikirimkan kedalam *firebase* telah berhasil tersimpan.

**4. Hasil dan Pengujian**

**4.1. Waktu Pencarian Jarak RPA**

Pengujian yang dilakukan adalah mengamati waktu eksekusi pencarian jarak yang telah dilakukan oleh RPA. Berikut merupakan tabel hasil pengujian waktu pencarian jarak pada RPA:

Tabel 4.4.2 1 Hasil Pengujian RPA

No	Long/Lat	Waktu
1	-6.872353,107.5409646	55 detik
2	-6.906573,107.6875077	46 detik
3	-6.896603,107.5763841	44 detik
4	-6.952877,107.6190082	52 detik
5	-6.904364,107.5550328	56 detik
6	-6.861894,107.5749478	49 detik
7	-6.915582,107.5827456	50 detik
8	-6.907054,107.6079208	55 detik
9	-6.936230,107.6673715	54 detik
10	-6.921622,107.5771461	42 detik
...	...	...
100	-6.890579,107.5856043	55 detik
TOTAL		5.037 Detik

Pada pengujian waktu eksekusi, seluruh data akan di rata-ratakan dengan rumus sebagai berikut:

$$\text{rata - rata} = \frac{\text{Total waktu eksekusi}}{\text{banyak data}}$$

$$\text{rata - rata} = \frac{5.037}{100} = 50,37 \text{ detik}$$

Kesimpulannya dari tabel 4.4.2 1 menunjukkan bahwa dari 100 titik alamat yang dicari menghasilkan waktu pencarian yang berubah-ubah dikarenakan adanya pengaruh dari kecepatan internet oleh karena itu dibuat perhitungan waktu rata-rata dihasilkan waktu eksekusi rata-ra adalah 50,37 detik.

**5. Kesimpulan**

Setelah dilakukan perancangan, implementasi, pengujian, dan analisis pada penelitian ini, maka dapat diambil beberapa kesimpulan yaitu implementasi Implementasi metode *Robotic Process Automation* (RPA) untuk pencarian jarak pada aplikasi navigasi destinasi perjalanan paket berbasis seluler ini dapat dimplementasikan dengan baik dan menghasilkan optimasi kombinatorial yang mendekati dengan solusi optimalnya. Hasil perhitungan metode *Robotic Process Automation* (RPA) untuk pencarian jarak terhadap 100 titik alamat pengiriman ini dapat berjalan dengan baik, berdasar kan pengujian didapatkan waktu rata-rata pencarian 50,37 detik pada seluruh titik alamat pengiriman.

**Referensi:**

- [1] A. Rifai, "Sistem Informasi Pemantauan Posisi Kendaraan Dinas Unsri Menggunakan," *Jurnal Sistem Informasi (JSI)*, VOL. 5, NO. 2, pp. 603-610, Oktober 2013.
- [2] J. N. F. B. F. V. Jerome Geyer-Klingeberg, "Process Mining and Robotic Process Automation:," *Paper presented at 16th International Conference on Business Process*, 2018.
- [3] A. Y. Enty Nur Hayati, "PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA GREEDY," *Seminar Nasional IENACO*, 2014.
- [4] T. D. Shunfu Hu, "Online Map Application Development Using Google Maps API, SQL Database, and ASP.NET," *International Journal of Information and Communication Technology Research*, no. Volume 3 No. 3, 2013.
- [5] S. R. D. Josseano Amakora Koli Parera, "Perancangan Aplikasi Sistem Navigasi Objek Wisata berbasis Android pada Dinas Pariwisata Kota Makassar," *Konferensi Nasional Sistem & Informatika*, 2015.
- [6] A. A. S. S. M. A. S. L. S. M. Hendra Nugraha Lengkong, "Perancangan Penunjuk Rute Pada Kendaraan Pribadi," *E-journal Teknik Elektro dan Komputer*, 2015.
- [7] S. K. S. I. H. H. K. Yani Nurhadryani, "Pengujian Usability untuk Meningkatkan Antarmuka Aplikasi," *Jurnal Ilmu Komputer Agri-Informatika*, no. Volume 2 Nomor 2, pp. 83-93.
- [8] P. M. L. A. C. Prof. Leslie Willcock, "The IT Function and Robotic Process Automation," *The Outsourcing Unit Working Research Paper Series*, no. Paper 15/05, October 2015.
- [9] P. Saeful Hamdi, "Analisis Algoritma Dijkstra dan Algoritma Bellman-Ford Sebagai Penentuan," no. Vol. 8, 2018.
- [10] H. Ari Muzakir, "Bellman-Ford Algorithm for Completion of Route Determination: An Experimental Study," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, no. Vol. 6, p. 29~35, 2020.
- [11] J. V. M. I. B. E. S. I. A. C. Margaretha Ohlyver, "The Comparison Firebase Realtime Database and MySQL Database Performance using Wilcoxon Signed-Rank Test," *Procedia Computer Science*, no. 157, pp. 396-405, 2019.



