Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 9, No. 3, September 2023, pp. 731-745
ISSN: 2338-3070, DOI: 10.26555/jiteki.v9i3.26588

Implementation of Geographic Information System Based on Google Maps API to Map Waste Collection Point Using the Haversine Formula Method

Ni Made Ary Esta Dewi Wirastuti [1], Lino Verlin [1], Is-Haka Mkwawa [2], Khalid G. Samarah [3]
[1] Department of Electrical Engineering, Udayana University, Jalan Raya Kampus Unud, Kuta Selatan, 80361, Indonesia
[2] School of Computing and Mathematics, University of Plymouth, Plymouth, Devon PL4 8AA, United Kingdom
[3] Electrical engineering department, Mutah University, P. O. Box 81, 61710, Al-Karak, Jordan

ARTICLE INFO

Article history:

Received June 02, 2023
Revised July 31, 2023
Published August 08, 2023

Keywords:

Waste collection system;
Haversine Formula;
Geographic information system;
Google maps API;
Taxi bike driver

ABSTRACT

Human life with all its activities cannot be separated from the existence of waste but the quality of post-consumption waste is generally still low. To convert waste into a more stable form and does not pollute the environment, it is required a precise waste collection system. Waste collection system is an important part of waste management. This study is proposed a geographic information system (GIS) application based on google maps Application Programming Interface (API) to map waste collection point using Haversine formula method. The application is designed to develop a waste collection system from households to Temporary Disposal Sites (TDS), quickly and accurately. The application can be used by households and waste taxi bike drivers to communicate when the households need the taxi bike drivers to pick the waste up to the temporary collection points. A geographic information system application based on google maps API used to display the location of the taxi bike driver, TDS and the location of the waste collection (household). Haversine formula is used to get the nearest waste taxi bike driver to the location of the request for waste transportation. The result of this research is an application that can monitor and track waste collection. Using black box testing, the system has run according to the functions and scenarios designed. Based on the testing the system usability scale results, the application obtains a score of 70.125 which indicates that the application is classified as good and acceptable to users.

Corresponding Author:

Ni Made Ary Esta Dewi Wirastuti, Department of Electrical Engineering, Udayana University, Jalan Raya Kampus Unud, Kuta Selatan, 80361, Indonesia
Email: dewi.wirastuti@unud.ac.id

1. INTRODUCTION

The exponentially growing population, urbanization, and economic development have led to the rising generation of waste. The development of an effective system for waste management is considered a significant hurdle in terms of economies development [1]. Solid waste management is still a significant problem the world faces. According to a recent study, 1.3 billion tons of solid waste are generated annually [2]. Daily in homes, there will always be a lot of garbage produced. The type of garbage contents and amount varies due to family lifestyle, family type, family size, and other home daily activities, including eating and cooking behaviours. However, the quality of post-consumption waste is generally still low, especially household waste which is caused by the mixing of one type of waste with another and has not been separated into 3 (three) types of waste, namely: organic, inorganic, and hazardous waste [3]–[5]. Separation of waste at the main source is very important to improve the quality of post-consumption waste so that a synergistic sorting and collection system is needed [6], [7].

About a third to half of the solid waste generated in low- and middle- income countries are usually not collected, resulting in waste being disposed of in the streets, open spaces and drainage systems [8]. Some of the main problems that are often encountered in the management of waste transportation include the unavailability of route visualization for travel planning for waste transportation vehicles based on the most effective route, there is no estimate of the amount of waste in the collection and distribution sites. The need for good tracking, monitoring, planning and management of waste transport vehicles [9]–[11]. To optimize waste management, it is necessary to utilize digital technology and ICT (information and communication technology). Research on the optimization model of waste transportation from garbage dump (GD) to final landfill (FL) [12] to make the waste transportation can find of the top priority GD for transporting garbage to FL. Several methods of collecting and transporting waste have also been introduced [12]–[14]. However, based on the study that has been carried on, the community is still unable to monitor or track directly when the waste will be transported.

Based on the deficiencies above, it is necessary to create a system that can bridge the community in requesting the transportation and collection of waste which is carried out immediately after there is a request and the community can monitor waste collection. However, when requesting garbage pick-up, a driver or garbage pick-up officer is required who has the closest distance to the pick-up location for time efficiency when asking the public to pick up trash. No research has been found on digital-based waste collection methods, equipped with a garbage pickup monitoring and tracking system, with a method of determining the shortest distance route between garbage pickers and the community who owns and uses the garbage pickup facility.

Related to the problems above, there are several studies with the shortest distance search method that have previously been done. Research on finding the shortest route uses the Dijkstra method [15], [16], the Traveling Salesman Problem (TSP) [17]–[19] and the Haversine formula [20]–[25]. There are several studies on waste management and transportation that have been carried out by [9]–[11]. From these studies, the method most suitable is the Haversine Formula because Haversine Formula is used to search for the shortest distance by calculating between the destination location and the user.

In this research, it introduces a digital-based waste collection system, which can bridge the community in requesting waste pickup quickly and accurately, and can monitor and track waste pickup (garbage taxis). Monitoring and tracking of garbage collection uses Google Maps API technology. Merging with the Haversine formula method is used to ensure that garbage pick-up requests are served by garbage taxi drivers that have the closest distance to the pick-up location. The Haversine formula method searches for the shortest distance by calculating between the destination location and the user. This garbage collection method is implemented in the form of software or application.

2. METHODS

This research begins with the method used shown in Fig. 1 which describes the process starting from identifications of the problems and then collecting the data references. Next, we formulate the issue we need to resolve based on the data references. After complete the formulate the problem we can start to modeling the system and then develop the application. While the application development is completed we should test the system and if we find the system is does not working properly we should repeat the system modeling until the application pass the system testing process. The conclusion making process can only be done if the application has passed the system testing.

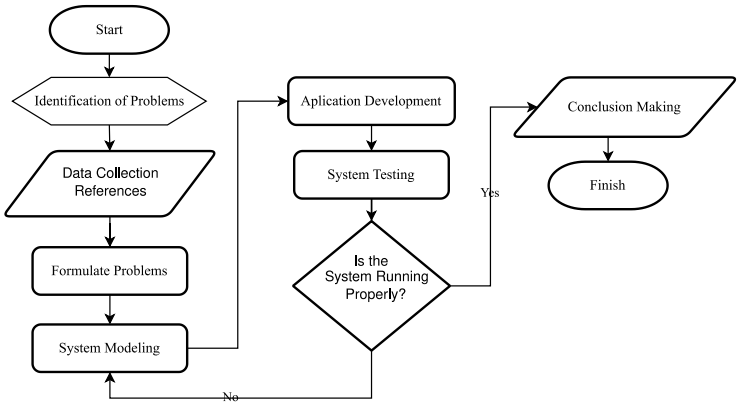

Fig. 1. Flowcharts of research stages

2.1. System Modeling

The digital-based waste collection system is an application which comes in the form of Progressive Web Application (PWA) that utilizes Google Maps API, such as the Google Maps API and the Google Directions API [26], [27]. PWAs improve traditional web apps with so-called service workers (to allow for running code in a background thread), a web app manifest (to provide metadata), off-line capabilities, and an installation-like user experience. This has evidently led to possibilities not previously available for web apps, with Progressive Web Apps performing on a par with regular native apps [28].

The Google Maps API functions to display maps on the application and to get the coordinates of the Driver, TDS location and the pick up location [23], [29]. This coordinates used to calculate the distance between Driver and Customer and or Customer and the TDS using Haversine Formula calculation in the system. The Google Direction API used to get direction between the Driver to the pick up location and the Driver to the TDS location while the Driver doing pick up order. Google Direction API serves to determine the route between two or more points. This API can draw route lines between points, get information in the form of distances and estimated arrival times [30]–[32].

This application was also used MySQL to store data. MySQL is a popular open source SQL database management system that is developed, distributed, and supported by Oracle Corporation [33]. MySQL manages a structured collection of data. A MySQL database helps you to add, access, and process the data stored in the database. MySQL stores data in separate tables. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. The SQL part of "MySQL" stands for "Structured Query Language," which is the most common standardized language used to access database [34], [35].

Based on the general system design shown in Fig. 2, the application has 4 (four) users, namely: Customer, Admin, TDS Officer and Driver, all of the users must login as access rights to use the applicatin with input the username and password. The data that has been input will be entered into the application database. After carrying out the login process, the application will display according to the user's access rights. When the Customer has successfully logged in, the application will take the Customer to the order page. When the Customer request an order then the order will be confirmed by the Driver. The Driver will go to the Customer's location to pick up garbage and send it to the TDS according to the order address. The cost of sending garbage will be received by the Driver when the Driver arrives at the TDS, this fee will be paid by the TDS Officer. The waste brought by the Driver will be weighed by the TDS Officer to determine the weight of the waste that will later be entered into the system. Then the customer will get points based on the weight and type of waste sent. Points collected by customers can be exchanged for certain items such as shopping vouchers.

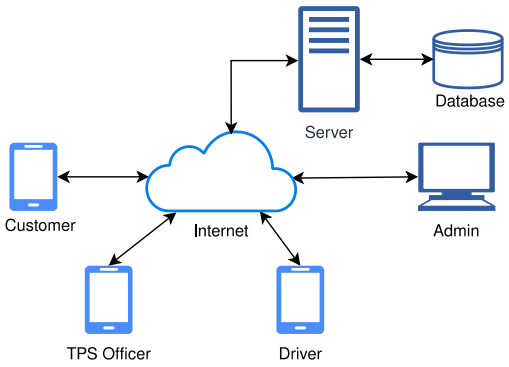

Fig. 2. General System Design

2.2. Haversine Formula Modelling

The Haversine formula will perform calculations based on the latitude and longitude of the Customer and Driver locations. Haversine Formula has its own law that is all equations are used based on the shape of a spherical earth by eliminating the factor that the earth is slightly elliptical (ellipsoidal factor). This is a special case of a general formula in spherical trigonometry which is related to the sides and angles of a spherical triangle. The calculation of the distance from one point to another on the earth surface is affected by a certain degree of curvature [36], [37]. The calculations using the Haversine formula [38]. Calculations on haversine formula use the difference or magnitude of changes in latitude (1) and longitude (2) two coordinate points in radians [21]. From the (1) and (2), we can calculate the distance between two points using the formula in (3), (4) and (5).

$$\Delta lat = lat2 - lat1 \quad (1)$$

$$\Delta long = long2 - long1 \quad (2)$$

$$a = sin2\left(\frac{\Delta lat}{2}\right) + \cos\left((lat1).\cos(lat2).sin2\left(\frac{\Delta long}{2}\right)\right) \quad (3)$$

$$c = 2.atan2\left(\sqrt{a}\,\sqrt{1-a}\right) \quad (4)$$

$$d = R.c \quad (5)$$

R is the radius of the earth is 6371 (Km), Δlat is the amount of change in latitude, Δlong is the amount of change in longitude, c is the axes calculation, d is the distance (Km) and 1° is the 0.0174532925 radians.

Pseudo-code of Haversine Formula shown on Table 1, as we can see that in the declaration has variable lon1, lon2, lat1, lat2, latitude2, latitude1, longitude1, longitude2 which are double data type. Variable status is tinyint data type whereas id_driver, get_Id, dist, distance is integer data type. Dist refers to the variable which store the longitude and latitude distance to be calculated. lon1, lon2, lat1 and lat2 are variables to store longitude and latitude limits. In lon1 and lon2 calculations, abs are used to convert the latitude value to positive if the latitude obtained is negative, while the number 111 is the value obtained from the conversion of 10 latitudes or longitude equals 111 Km. In get_Id, calculations are performed using the Haversine formula and stored in the distance variable provided that the longitude and latitude points of the destination are not outside the calculated distance limit. From then the distance must be less than dist and sorted from the smallest value with a limit of 1. Because of this limit of 1, the closest distance will be obtained, which will help in retrieving the Driver id based on the closest distance. Once Driver Id has been obtained, this will be displayed as the output, otherwise the output displays text providing a choice to the customer whether they would want to search for another driver, if the first search was not found.

Table 1. Pseudo-code in the use of the Haversine

```
Declaration:
   var lon1, lon2, lat1, lat2, latitude2, latitude1, longitude1, longitude2: double
   var status : tinyint
   var id_driver, get_Id, dist, distance: integer

Description:
   dist = x Km
   lon1 = longitude1-dist/abs(cos(radians(latitude1))*111)
   lon2 = longitude1+dist/abs(cos(radians(latitude1))*111)
   lat1 = latitude1-(dist/111)
   lat2 = latitude1+(dist/111)

   a    =  POW(SIN((latitude1 - latitude2) * pi()/180 / 2), 2) +
             COS(latitude1 * pi()/180) *  COS(latitude2* pi()/180) *
            POW(SIN((longitude1 - longitude2) * pi()/180 / 2), 2)
   c    = 2 * ASIN(SQRT( a)
   d    = 6371 * c

   get_Id = SELECT  id_driver * d as distance FROM  driver
       and  longitude2 between lon1 and lon2
       and latitude2 between lat1 and lat2
       having distance < dist ORDER BY Distance limit 1

   if get_Id != 0
       Output = get_Id
   Else
       Output = "Driver is not found, want to search again?"
   Write (Output)
```

2.3. Testing

Testing is widely recognized as an important stage of the software development lifecycle. Effective software testing can provide benefits such as bug finding, preventing regressions, and documentation [39]. The testing phase is carried out to find errors or errors that are still present in the application and the testing phase is also carried out to check the suitability of the program made with the stages of system analysis and design. In this system, testing is carried out using the black box method and usability scale.

Black box testing is a test conducted to observe the results of an application that is ready to be made to test input and output functionality, so that it can function properly without knowing the code structure and testing can be applied to unit, integration, system and acceptance testing [40], [41]. This technique is completely black-box, since it does not require the source code of the app [42]. In black box testing, there are several testing methods that can be used, namely Equivalence Partitioning, Fuzzing, Boundary Value Analysis, Cause-Effect Graph, Orthogonal Array Testing, State Transition, and All Pair Transition [40]. In this study using the equivalence partitioning test method, this technique is often used in grouped or domain black box testing when testing applications, especially when testing interfaces. Then the purpose of the test is to find out weaknesses in the system when it is tested or the input that is entered will be produced as expected so as to avoid errors or deficiencies in the application before it is used by the user.

One of the most popular standardized questionnaires mea-suring perceived usability is probably the System Usability Scale (SUS). The SUS comprises ten items, formulated as affirmative statements, for each of which the user states his/her position, by expressing his/her agreement or disagreement on a 5-point Likert scale (1 = do not agree at all; 5 = completely agree). If the user does not know what to answer for an item, he/she is requested to answer nevertheless by ticking the middle of the scale (score 3) [43]. In the reaseach research also be tested using the SUS that carried 20 tester. The tester will be asked for using the app and then participants were asked to partake in fill the SUS questionnaire, quantitatively assessing the app's subjective usability [44].

3. RESULTS AND DISCUSSION

The design and manufacture of a digital-based garbage collection application by applying the Haversine formula is based on functionality and ease of operation. So, Users can use the application effectively and efficiently. The data needed are: name, date of birth, telephone number, vehicle license plate, vehicle brand, user photo, address, latitude and longitude which are implemented into the system so that customers can request garbage transportation orders and get the nearest driver from the location of the transportation request. The process of designing and manufacturing this system is equipped with several supporting components. These components are My Structured Query Language (MySQL) and google maps API. MySQL is used to store data relating to orders, users, products and TDS locations. Meanwhile, the google maps API is used by the system to obtain the latitude and longitude used to carry out the garbage collection order and to display Driver, Customer and TDS location points.

3.1. Interface of Digital-baed Garbage Collection

The user interface of Customer can be seen on Fig. 3, on the Customer can see several main products also the number of points they have. There is also a Motor Bike button that can be used by customer to place request for waste pickup. If the Customer presses the Motor Bike button, the Customer will be redirected to the Order page which can be seen on Fig. 3(a). The customer can move the marker to change the pickup address and the destination TPS will automatically be filled in considering the closest distance to the pickup point. If the address is correct, the customer can place an order by pressing the Order button. Then the system processing to get the nearest Driver from pick up point. However, system might be failed to find the Taxi bike driver if there is no driver currently available as can be seen on Fig. 3(c), if this condition occurred the Customer can try to reorder again within few minutes. Fig. 3(d) shown the interface while system succed found the Driver and then Google Map will display the Driver's location point which is marked with a marker in the form of a motorcycle icon. If the driver's location is close to the customer's location, a notification will appear that the driver is almost there as we can see on Fig. 3(f) and Fig. 3(g).

Fig. 3. Customer user interface, (a) Home-page, (b) Order page, (c) Processing get the driver, (d) Failed get the Driver, (d) Succed got the Driver, (e) Driver moving to pick up point, (f) Driver moving to TDS

3.2. Implementation of the Haversine Formula in the Order System

The Haversine formula is implemented in the system when a customer places an order, namely a query when retrieving data from the database. The search experiment was carried out by placing an order and obtaining order data in a database that had an id of 64 which can be seen in Table 2. Latitude and longitude data on data with id 64 are used as the initial latitude and longitude in calculating distances using the Haversine formula.

Table 2. Data order

id	id_customer	id_driver	status	cost	latitude_c	lngitude_c	address	destination
60	6	9	3	5000	-8.28112	114.18461	P59M+GR	4
61	6	9	3	5000	-8.28112	114.18461	P59M+GR	4
62	6	9	3	5000	-8.28112	114.18461	P59M+GR	4
63	6	9	3	5000	-8.28112	114.18461	P59M+GR	4
64	6	NULL	1	5000	-8.30254	114.12640	M5XG+MR	1

Driver data currently active can be seen in Table 3. There are several drivers with active status. The data in Table 3 was taken before the Order experiment was carried out. The latitude and longitude data in Table 3 are used as the final latitude and longitude in the calculation of finding the shortest distance using the Haversine formula.

Table 3. Driver location prior to trial Order

id	id_driver	status	latitude	longitude	id_reject	id_order
49	9	1	-8.29358	114.17681	NULL	60
51	10	1	-8.30140	114.17861	NULL	NULL
56	18	1	-8.29691	114.18370	NULL	NULL
57	11	1	-8.29231	114.18472	NULL	NULL
58	12	1	-8.30053	114.18847	NULL	NULL
59	17	1	-8.30387	114.19321	NULL	NULL
60	19	1	-8.30145	114.19328	NULL	NULL
61	20	1	-8.30879	114.19835	NULL	NULL

When the Customer places an Order, the system will update the Driver data that has the closest distance to the Customer so that the Driver gets a notification. The results of the Driver location update after the Order made can be seen in Table 4. It can be seen that the system has changing the Driver status with id_driver 10 which means the system selects the Driver with id 10 as the Driver with the shortest distance from the location of the Order request that has been made.

Table 4. Driver location data after trial Order

id	id_driver	status	latitude	longitude	id_reject	id_order
49	9	1	-8.29358	114.17681	NULL	60
51	10	88	-8.30140	114.17861	NULL	64
56	18	1	-8.29691	114.18370	NULL	NULL
57	11	1	-8.29231	114.18474	NULL	NULL
58	12	1	-8.30053	114.18847	NULL	NULL
59	17	1	-8.30387	114.19321	NULL	NULL
60	19	1	-8.30145	114.19328	NULL	NULL
61	20	1	-8.30879	114.19835	NULL	NULL

Based on the Haversine pseudocode formula that can be seen in Table 5 , it can be seen that REarth is the radius of the earth with a length of 6371 Km. Longitude, latitude, distance and Haversine have integer type values. Read is a command to read the value of a variable, so after the longitude, radius of the earth and latitude values have been obtained, calculation of the magnitude of the change in longitude and latitude by subtracting the destination point from the starting point and then converting the results to radians. Then a calculation is performed using the Haversine formula which is stored in the Haversine variable, the distance from the starting point and destination point can be obtained by multiplying the radius of the earth by the Haversine calculation results.

Based on Table 2–Table 4 and using (1)-(5), we obtain $\Delta lat = -0.00625$, $\Delta long = 0.00221$, $a = 4.60740$, $c = 4.29297$, $d = 0.27351\ km$.

Table 5. Pseudocode Haversine Formula

```
Declaration :
   REarth = 6371.0
   longitude, latitude, Haversine, distance = int

Description:
   Read RBumi
   Read longitudes
   Read latitudes

   Radian longitude = degree to radian (longitude2 – longitude1)
   Radian latitude    = degree to radian (latitude2 – latitude 1)
   Haversine            = 2 * asin(sqrt(sin(radian latitude/2)* sin(radians
   latitude/2)) + Cos(degree to radian
                (latitude1) *Cos(degree to radian(latitude2)* Sin(degree
                longitude/2)cc)
   Distance               = REarth * Haversine
```

Using the same formulation for each Driver location, the results are as shown in Table 6. It can be seen that the shortest distance is found in the data with Id 51 which has a distance of 0.273 Km. From Table 4, we can see that the data with Id 51 or Driver with Id 10 as the Driver who gets the Order request in accordance with the results of calculating the shortest distance using the Haversine formula.

Table 6. The distance results using Haversine formula

Id	Latitude 2	Longitude 2	*a*	*c*	*d* (Km)
49	-8.29358	114.17681	6.126	1.565	0.997
51	-8.30140	114.17860	4.607	4.290	0.273
52	-8.29691	114.18369	6.379	1.597	1.017
53	-8.29231	114.18474	1.314	2.293	1.461
54	-8.30053	114.18847	1.116	2.113	1.346
55	-8.30387	114.19321	2.121	2.912	1.855
56	-8.30145	114.19328	2.135	2.922	1.861
57	-8.30879	114.19834	3.888	3.944	2.512

Several Order request trials were carried out using the active Driver location in Table 7. Order request experimental can be seen in Table 8. Table 7 and Table 8 are used to calculate the distance using the Haversine formula. In order to obtain the distance on each Order request using the Haversine formula, the results can be seen in Table 9. Then, the closest Driver is obtained for each Order Id which can be seen in Table 10. Then, a comparison between selected Driver and closest Driver, it can be concluded that out of 5 (five) trials the Order request was able to correctly select the Driver with the nearest location as shown in Table 10.

Table 7. Driver location is active

Id Driver	Latitude Driver	Longitude Driver
18	-8.71538	115.20218
19	-8.710882	115.18853
20	-8.71894	115.17746
22	-8.71453	115.18269
23	-8.70791	115.22226

Table 8. Order request trial

Id Order	Lat Order	Long Order	Selected ID Driver
68	-8.71163	115.20319	18
69	-8.71795	115.19363	19
70	-8.71609	115.19627	18
71	-8.71538	115.19627	18
72	-8.71909	115.19141	19

Table 9. Haversine formula calculation

Id Order	Distance Taxi Driver per Id (Km)				
	18	19	20	22	23
68	0.210	1.631	2.883	2.284	2.128
69	0.958	0.658	1.799	1.227	3.219
70	0.658	0.895	2.096	1.511	2.916
71	0.657	0.886	2.098	1.510	2.912
72	1.210	0.503	1.551	0.993	3.471

Table 10. Comparison of selected Driver and closest Driver

Id Order	ID Driver (Selected)	Id Driver (Closest)	Results
68	18	18	Matching
69	19	19	Matching
70	18	18	Matching
71	18	18	Matching
72	19	19	Matching

3.3. Black Box Testing

Black box testing is used to perform functionality tests on digital-based garbage collection applications using the Haversine formula method. The results of testing with the Black Box method can be seen in Table 11-Table 14.

Table 11. Testing results of general function

Cases	Scenarios	Expected results	Test results
Login	The user enters the Password and Username according to the account that has been registered to enter the system	When the Username and Password are correct, the user can access the system	Succeed
Register	Users who don't have an account enter their personal data according to the form to register in the system	Data from new users will enter the database and will automatically be registered as users with access rights as Customers	Succeed
Logout	The user is already signed in to the account but wants to log out of the current account	After the user presses the Log Out button the user exits the account that is currently being used and is redirected to the Login page	Succeed

Table 12. Testing results of admin function

Cases	Scenarios	Expected results	Test results
See Driver List	When Admin successfully logs into the system	The system displays a form and a list of Drivers	Succeed
Add Driver List	Admin fills out the form and presses the Add Action on the form on the Driver page	The system displays a form where Admin can fill in new driver data. The information system then adds the new data to the database	Succeed
Update Data Drivers	Admin chooses Update Action on one of the data from the list	The system displays a form containing the driver's old data. Admin can change the data in the form. After Admin presses Update Action, the data will be updated in the database according to the latest input	Succeed

Table 13. Testing results of customer function

Cases	Scenarios	Expected results	Test results
Displays Customer Data	The customer presses the Customer name link	The system displays customer data from the database	Succeed
Create a new freight Order	The customer presses the Order action on the Order page	The system searches for Drivers for Order requests. If it fails, a failure notification will appear and if successful, the Customer will be redirected to the delivery page	Succeed
Exchanging Points	The customer presses the Redeem action on the product detail page	If there are enough customer points to redeem, the system will reduce the number of customer points according to the amount needed, then the customer will get a unique code notification. If the customer's points are lacking, they will get a notification that the redeem failed	Succeed

Table 14. Testing results of Driver function

Cases	Scenarios	Expected results	Test results
Displays a map on the Driver page	When the Driver successfully enters the system	The system will display a map on the Driver page along with a marker which is the Driver's position	Succeed
Get Driver's current position	Driver pressing action gets current position	The marker will move if the Driver's location also changes	Succeed
Enable or Disable Driver Status	The driver presses the Power button	The Power button is blue if the Driver status is active when the Driver presses the Power button then the button will be black and the Driver status will be updated by the system to be inactive and vice versa	Succeed

3.4. Testing using System Usability Scale (SUS) Method

The System Usability Scale (SUS) is a popular measure of perceived usability. In SUS method, there are several rules in calculating scores. It's a 10-item questionnaire scored on a 101-point scale and provides a measure of a user's perception of the usability of a system. The following are the rules when calculating scores on the questionnaire [45]:

- For odd numbered question, the score for each question obtained from the user's score that reduced by 1.
- For even numbered question, the final score is obtained from the value of 5 minus the question score obtained from the user.
- The SUS score is obtained from the sum of the scores for each question which is then multiplied by 2,5.

Fig. 4 shows grades, adjectives, acceptability, and Net Promoter Score (NPS) categories associated with raw SUS scores.

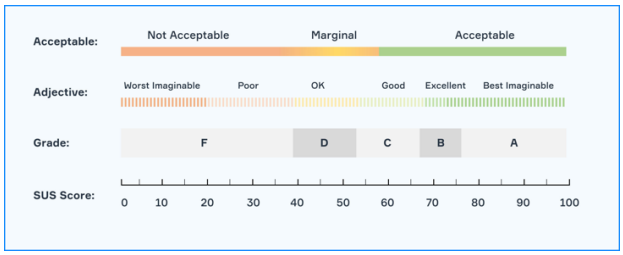

Fig. 4. SUS score

The score calculation rules apply to 1 respondent. For further calculations, the SUS score of each respondent is sought by the average score by adding up all scores and dividing by the number of respondents. The following formula calculates the SUS score use the (6) [45]:

$$\bar{x} = \frac{\sum x}{n} \quad (6)$$

Where $\bar{x}$ is average score, $\sum x$ is sum of SUS scores and n is number of respondents. The average SUS score is obtained from all respondents. The number of research respondents refers to John Broke's recommendations [39], [40], which is at least 20 people.

In this research, the SUS method was involved 20 respondents who are given 10 statements with a range of score 1-5 from strongly disagree to strongly agree. The statements of the questionnaire were as in the following :

(P1) I think I will use this feature a lot.
(P2) I feel this feature is too complicated even though it could be made simpler.
(P3) I think this feature is easy to use.
(P4) I think I need help from a technical person to be able to use this feature.
(P5) I find that there are various features that are well integrated in the system.
(P6) I think there are a lot of inconsistencies in this feature.
(P7) I think the majority of users will be able to learn this feature quickly.
(P8) I find this feature very cumbersome to use.
(P9) I'm pretty sure I can use this feature.
(P10) I have to learn many things before I can use this feature.

Table 15 shown the assessment results of 20 respondets. This result for each statement given by the respondent will be converted into a value on a scale of 0 - 4. The scores for statements 1, 3, 5, 7 and 9 will be reduced by 1 while the scores for statements 2, 4, 6,8 and 10 will be 5 minus respondent's rating. The results of the rating conversion can be seen in Table 16.

Table 15. Assessment of 20 Respondents

No	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	5	2	5	3	5	1	5	1	4	3
2	5	1	5	1	5	1	5	1	5	1
3	5	2	4	2	3	2	4	2	5	2
4	3	3	4	4	4	3	4	3	4	3
5	5	1	5	2	5	1	5	1	5	2
6	3	3	4	3	4	3	4	2	4	3
7	5	3	4	2	4	2	4	2	5	3
8	1	2	1	2	1	2	1	2	1	3
9	4	2	4	2	3	2	5	2	5	2
10	4	3	4	2	4	3	4	2	4	3
11	4	2	4	1	5	2	5	2	5	3
12	3	3	3	3	3	3	3	3	3	3
13	4	3	4	3	4	3	4	3	5	2
14	4	3	4	2	3	2	4	2	4	1
15	4	2	4	2	4	2	5	2	4	2
16	3	2	4	3	4	3	4	2	4	4
17	4	3	4	3	4	3	4	3	4	3
18	4	4	4	4	4	4	4	4	4	4
19	4	2	5	2	5	2	5	1	4	3
20	4	4	4	3	5	1	4	4	3	3

Table 16. Rating conversion of 20 respondents

No	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Total
1	4	3	4	2	4	4	4	4	3	2	34
2	4	4	4	4	4	4	4	4	4	4	40
3	4	3	3	3	2	3	3	3	4	3	31
4	2	2	3	1	3	2	3	2	3	2	23
5	4	4	4	3	4	4	4	4	4	3	38
6	2	2	3	2	3	2	3	3	3	2	25
7	4	2	3	3	3	3	3	3	4	2	30
8	0	3	0	3	0	3	0	3	0	2	14
9	3	3	3	3	2	3	4	3	4	3	31
10	3	2	3	3	3	2	3	3	3	2	27
11	3	3	3	4	4	3	4	3	4	2	33
12	2	2	2	2	2	2	2	2	2	2	20
13	3	2	3	2	3	2	3	2	4	3	27
14	3	2	3	3	2	3	3	3	3	4	29
15	3	3	3	3	3	3	4	3	3	3	31
16	2	3	3	2	3	2	3	3	3	1	25
17	3	2	3	2	3	2	3	2	3	2	25
18	3	1	3	1	3	1	3	1	3	1	20
19	3	3	4	3	4	3	4	4	3	2	33
20	3	1	3	2	4	4	3	1	2	2	25
Score Total											561
Multiplied by 2.5											1402.5
Score SUS											70.125

All scores added up and the result is 561. This total score is then multiplied by 2.5. To get the SUS score, the total score is divided by the number of respondents (i.e. 20 respondents). Then the SUS score is 70.125. Based on Fig. 3, the results of testing the system usability scale in waste collection applications are classified as margin acceptable, grade C and the adjective rating is good.

From the design and development, as well as the test results, there are some advantages and disadvantages of the digital-based garbage collection application using the Haversine formula which are:

1) The application can make requests for waste collection and get the nearest Driver,
2) The system automatically selects the nearest TDS location from garbage pickup requests.
3) The system provides feature gamification in the term of Points. The Point are obtained from how much waste is collected and a Point exchange feature for available products.

3.5. Discussion

In general, the application has been succefully allows customers to place garbage transportation orders and assigns the closest available driver based on the Haversine formula calculations. Based on the testing the application, by using black box testing, it has showed that all functions or features of the application has run accordingly to the design. The last testing of using system usability scale (SUS) showed a value of 70.125. This means the application is classified as usable properly and accepted by users.

This study created the application that make the user can request to garbage taxi to pick up immediately, and it make the buildup of garbage can be avoided. Also with the implemented of Haversine Formula to get the nearest distance in the application makes the transportation can be delivered quickly. While the previous study only provided the optimation of the routes for the garabage trucks and monitoring of the bins to get the priority pick up. Nevertheless, this study still lacking still lacking in the optimation of taxi bike routes to the pick up point or to the TDS since the main purpose of this study is only to get the nearest taxi bike. Also it would be more benefical if in the application has feature texting between the Customer and Driver while they run the order so if there is delay on the pick up is occurred, Customer can ask an information to the Driver easily.

4. CONCLUSION

Geographic Information System based on Google Maps API to map waste collection point using the Haversine formula method has been proposed and developed in term of mobile application. The application was successful to get the nearest taxi bike location, accurately. In addition, by using black box testing, it has showed that all functions or features of the application has run accordingly to the design. The last testing of using system usability scale (SUS) showed a value of 70.125. This means the application is classified as usable properly and accepted by users. The results of this study are expected to be used for further research to improve the efficiency of the waste collection management system. However, the application still has deficiencies in some features, such as no feature of texting between the Customer and Driver. Also it would be more benefical if the system can provide the optimation routes for the Driver, so the waste transportation can be delivered quickly.

REFERENCES

[1] S. Ahmad, Imran, F. Jamil, N. Iqbal, and D. Kim, "Optimal Route Recommendation for Waste Carrier Vehicles for Efficient Waste Collection: A Step Forward Towards Sustainable Cities," *IEEE Access*, vol. 8, pp. 77875–77887, 2020, https://doi.org/10.1109/ACCESS.2020.2988173.

[2] M. Saad, M. Bin Ahmad, M. Asif, M. K. Khan, T. Mahmood, and M. T. Mahmood, "Blockchain-Enabled VANET for Smart Solid Waste Management," *IEEE Access*, vol. 11, pp. 5679–5700, 2023, https://doi.org/10.1109/ACCESS.2023.3235017.

[3] E. Likotiko, S. Misaki, Y. Matsuda, and K. Yasumoto, "SGBS: A novel smart garbage bin system for understanding household garbage disposal behaviour," in *2021 Thirteenth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, pp. 1–8, 2021, https://doi.org/10.23919/ICMU50196.2021.9638956.

[4] N. C. A. Sallang, M. T. Islam, M. S. Islam, and H. Arshad, "A CNN-Based Smart Waste Management System Using TensorFlow Lite and LoRa-GPS Shield in Internet of Things Environment," *IEEE Access*, vol. 9, pp. 153560–153574, 2021, https://doi.org/10.1109/ACCESS.2021.3128314.

[5] W.-L. Mao, W.-C. Chen, C.-T. Wang, and Y.-H. Lin, "Recycling waste classification using optimized convolutional neural network," *Resour Conserv Recycl*, vol. 164, p. 105132, 2021, https://doi.org/10.1016/j.resconrec.2020.105132.

[6] W. Ma, X. Wang, and J. Yu, "A Lightweight Feature Fusion Single Shot Multibox Detector for Garbage Detection," *IEEE Access*, vol. 8, pp. 188577–188586, 2020, https://doi.org/10.1109/ACCESS.2020.3031990.

[7] B. Cao, X. Chen, Z. Lv, R. Li, and S. Fan, "Optimization of Classified Municipal Waste Collection Based on the Internet of Connected Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5364–5373, 2021, https://doi.org/10.1109/TITS.2020.2981564.

[8] C.-N. Tamakloe and E. V. Rosca, "Smart Systems and the Internet of Things (IOT) For Waste Management," in *2020 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pp. 1–6. 2020, https://doi.org/10.1109/CIVEMSA48639.2020.9132968.

[9] M. Srilatha, C. Abhinav, M. Balaram, and A. Sanjana, "Smart Monitoring and Collection of Garbage System Using Internet of Things," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pp. 335–342, 2021, https://doi.org/10.1109/ICICV50876.2021.9388438.

[10] S. V. Kumar, T. S. Kumaran, A. K. Kumar, and M. Mathapati, "Smart garbage monitoring and clearance system using internet of things," in *2017 IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, pp. 184–189, 2017, https://doi.org/10.1109/ICSTM.2017.8089148.

[11] A. Yadav, and A. Khan, "Internet of Things Based Wireless Garbage Monitoring System," in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 103–107, 2019, https://doi.org/10.1109/I-SMAC47947.2019.9032540.
[12] M. Faturrahman, "Optimization Model of Waste Transportation from Garbage Dump (GD) to Final Landfill (FL)," in *2019 7th International Conference on Cyber and IT Service Management (CITSM)*, pp. 1–4, 2019, https://doi.org/10.1109/CITSM47753.2019.8965427.
[13] M. Sarosa *et al.*, "Internet Of Things (IoT) Based Garbage Incinerator Monitoring System," in *2022 International Conference on Electrical and Information Technology (IEIT)*, pp. 146–149, 2022, https://doi.org/10.1109/IEIT56384.2022.9967906.
[14] S. Maiti, M. Vaishnav, L. Ingale, P. Suryawanshi, and S. Kumar, "Optimization of garbage collector tracking and monitoring system analysis," *CSI Transactions on ICT*, vol. 4, no. 2–4, pp. 187–192, 2016, https://doi.org/10.1007/s40012-016-0103-2.
[15] G. Xiaoxue, "Optimal Route Planning of Parking Lot Based on Dijkstra Algorithm," in *2017 International Conference on Robots & Intelligent System (ICRIS)*, pp. 221–224, 2017, https://doi.org/10.1109/ICRIS.2017.62.
[16] S. Shao, W. Guan, B. Ran, Z. He, and J. Bi, "Electric Vehicle Routing Problem with Charging Time and Variable Travel Time," *Math Probl Eng*, vol. 2017, pp. 1–13, 2017, https://doi.org/10.1155/2017/5098183.
[17] M. Niendorf and A. R. Girard, "Exact and Approximate Stability of Solutions to Traveling Salesman Problems," *IEEE Trans Cybern*, vol. 48, no. 2, pp. 583–595, 2018, https://doi.org/10.1109/TCYB.2016.2647440.
[18] L. Yu, D. Kong, X. Shao, and X. Yan, "A Path Planning and Navigation Control System Design for Driverless Electric Bus," *IEEE Access*, vol. 6, pp. 53960–53975, 2018, https://doi.org/10.1109/ACCESS.2018.2868339.
[19] M. D. A. C. Hasibuan, "Pencarian Rute Terbaik Pada Travelling Salesman Problem (TSP) Menggunakan Algoritma Genetika pada Dinas Kebersihan dan Pertamanan Kota Pekanbaru," *SATIN - Sains dan Teknologi Informasi*, vol. 1, no. 1, pp. 35–46, 2016, https://doi.org/10.33372/stn.v1i1.11.
[20] O. D. Jimoh, L. A. Ajao, O. O. Adeleke, and S. S. Kolo, "A Vehicle Tracking System Using Greedy Forwarding Algorithms for Public Transportation in Urban Arterial," *IEEE Access*, vol. 8, pp. 191706–191725, 2020, https://doi.org/10.1109/ACCESS.2020.3031488.
[21] R. Agramanisti Azdy and F. Darnis, "Use of Haversine Formula in Finding Distance Between Temporary Shelter and Waste End Processing Sites," *J Phys Conf Ser*, vol. 1500, no. 1, p. 012104, 2020, https://doi.org/10.1088/1742-6596/1500/1/012104.
[22] K. Saputra, N. Nazaruddin, D. H. Yunardi, and R. Andriyani, "Implementation of Haversine Formula on Location Based Mobile Application in Syiah Kuala University," in *2019 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, pp. 40–45, 2019, https://doi.org/10.1109/CYBERNETICSCOM.2019.8875686.
[23] Y. Dian Harja and R. Sarno, "Determine the best option for nearest medical services using Google maps API, Haversine and TOPSIS algorithm," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, pp. 814–819, 2018, https://doi.org/10.1109/ICOIACT.2018.8350709.
[24] A. Suryana, F. Reynaldi, F. Pratama, G. Ginanjar, I. Indriansyah, and D. Hasman, "Implementation of Haversine Formula on the Limitation of E-Voting Radius Based on Android," in *2018 International Conference on Computing, Engineering, and Design (ICCED)*, pp. 218–223, 2018, https://doi.org/10.1109/ICCED.2018.00050.
[25] A. Candra, D. Rachmawati, and I. N. Faradillah, "Haversine and Tabu Search in Determining the Nearest Bus Stop based on GIS," in *2021 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, pp. 176–179, 2021, https://doi.org/10.1109/DATABIA53375.2021.9650155.
[26] D. Fortunato and J. Bernardino, "Progressive web apps: An alternative to the native mobile Apps," in *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–6, 2018, https://doi.org/10.23919/CISTI.2018.8399228.
[27] P. R. Wijaya, P. V. Crisgar, M. D. F. Pakpahan, E. Y. Syamsuddin, and M. O. Hasanuddin, "Implementation of Motor Vehicle Tracking Software-as-a-Service (SaaS) Application Based on Progressive Web App," in *2021 International Symposium on Electronics and Smart Devices (ISESD)*, pp. 1–6, 2021, https://doi.org/10.1109/ISESD53023.2021.9501600.
[28] A. Biørn-Hansen, C. Rieger, T.-M. Grønli, T. A. Majchrzak, and G. Ghinea, "An empirical investigation of performance overhead in cross-platform mobile development frameworks," *Empir Softw Eng*, vol. 25, no. 4, pp. 2997–3040, 2020, https://doi.org/10.1007/s10664-020-09827-6.
[29] A. M. Luthfi, N. Karna, and R. Mayasari, "Google Maps API Implementation On IOT Platform For Tracking an Object Using GPS," in *2019 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, pp. 126–131, 2019, https://doi.org/10.1109/APWiMob48441.2019.8964139.
[30] W. G. R. M. P. S. Rathnayake, "Google Maps Based Travel Planning & Analyzing System (TPAS)," in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, pp. 1–5.2018, https://doi.org/10.1109/ICCTCT.2018.8550996.
[31] H. Nurwarsito and N. Savitri, "Development of Mobile Applications for Posyandu Administration Services Using Google Maps API Geolocation Tagging," in *2018 International Conference on Sustainable Information Engineering and Technology (SIET)*, pp. 168–173, 2018, https://doi.org/10.1109/SIET.2018.8693170.
[32] N. Xia, L. Cheng, S. Chen, X. Wei, W. Zong, and M. Li, "Accessibility based on Gravity-Radiation model and Google Maps API: A case study in Australia," *J Transp Geogr*, vol. 72, pp. 178–190, 2018, https://doi.org/10.1016/j.jtrangeo.2018.09.009.

[33] M. M. Eyada, W. Saber, M. M. El Genidy, and F. Amer, "Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments," *IEEE Access*, vol. 8, pp. 110656–110668, 2020, https://doi.org/10.1109/ACCESS.2020.3002164.
[34] B. Christudas, "MySQL," in *Practical Microservices Architectural Patterns*, pp. 877–884, 2019, https://doi.org/10.1007/978-1-4842-4501-9_27.
[35] C. Bell, *Introducing the MySQL 8 Document Store*. Berkeley, CA: Apress, 2018, https://doi.org/10.1007/978-1-4842-2725-1.
[36] P. Dauni, M. D. Firdaus, R. Asfariani, M. I. N. Saputra, A. A. Hidayat, and W. B. Zulfikar, "Implementation of Haversine formula for school location tracking," *J Phys Conf Ser*, vol. 1402, no. 7, p. 077028, 2019, https://doi.org/10.1088/1742-6596/1402/7/077028.
[37] A. Suryana, F. Reynaldi, F. Pratama, G. Ginanjar, I. Indriansyah, and D. Hasman, "Implementation of Haversine Formula on the Limitation of E-Voting Radius Based on Android," in *2018 International Conference on Computing, Engineering, and Design (ICCED)*, pp. 218–223, 2018, https://doi.org/10.1109/ICCED.2018.00050.
[38] A. Baskar, "Locating the Economic and other Performance Centres of Asia using Geodetic Coordinates, Haversine Formula and Weiszfeld's Algorithm," *IOP Conf Ser Mater Sci Eng*, vol. 912, no. 6, p. 062001, 2020, https://doi.org/10.1088/1757-899X/912/6/062001.
[39] E. Dinella, G. Ryan, T. Mytkowicz, and S. K. Lahiri, "TOGA," in *Proceedings of the 44th International Conference on Software Engineering*, pp. 2130–2141, 2022, https://doi.org/10.1145/3510003.3510141.
[40] H. T. Hidayat, Husaini, F. F. Yanuar, and A. Aprianda, "Marketplace Application Feasibility Analysis with Android-Based Black Box Testing," in *9th International Conference on Technical and Vocational Education and Training (ICTVET)* pp. 54–66, 2023, https://doi.org/10.2991/978-2-38476-050-3_7.
[41] H. Yulianton, A. Trisetyarso, W. Suparta, B. S. Abbas, and C. H. Kang, "Web Application Vulnerability Detection Using Taint Analysis and Black-box Testing," *IOP Conf Ser Mater Sci Eng*, vol. 879, no. 1, p. 012031, 2020, https://doi.org/10.1088/1757-899X/879/1/012031.
[42] D. Amalfitano, V. Riccio, P. Tramontana, and A. R. Fasolino, "Do Memories Haunt You? An Automated Black Box Testing Approach for Detecting Memory Leaks in Android Apps," *IEEE Access*, vol. 8, pp. 12217–12231, 2020, https://doi.org/10.1109/ACCESS.2020.2966522.
[43] G. Gronier and A. Baudet, "Psychometric Evaluation of the F-SUS: Creation and Validation of the French Version of the System Usability Scale," *Int J Hum Comput Interact*, vol. 37, no. 16, pp. 1571–1582, 2021, https://doi.org/10.1080/10447318.2021.1898828.
[44] R. Tennant, M. Tetui, K. Grindrod, and C. M. Burns, "Multi-Disciplinary Design and Implementation of a Mass Vaccination Clinic Mobile Application to Support Decision-Making," *IEEE J Transl Eng Health Med*, vol. 11, pp. 60–69, 2023, https://doi.org/10.1109/JTEHM.2022.3224740.
[45] Z. Sharfina and H. B. Santoso, "An Indonesian adaptation of the System Usability Scale (SUS)," in *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pp. 145–148, 2016, https://doi.org/10.1109/ICACSIS.2016.7872776.

BIOGRAPHY OF AUTHORS

Ni Made Ary Esta Dewi Wirastuti, she received the B.Eng. degree in electrical engineering from Udayana University, Bali, Indonesia, in 2000, the M.Sc. degree in mobile communication systems from University of Surrey, Guildford, United Kingdom, in 2002 and the Ph. D. degree in Telecommunication Systems from University of Bradford, West Yorkshire, United Kingdom, in 2007. From 2007 to 2009, she was a Post Doctoral Fellowship with the Mobile and Satellite Communication Research Centre (MSCRC), University of Bradford, United Kingdom worked to the VeSeL (Village e-Science for Life) project, Engineering and Physical Sciences Research Council (EPSRC) grant. Her research interest includes the development of physical layer model for the next wireless and mobile communication systems, Wireless technology. She has been a lecturer in Department of Electrical Engineering, Faculty of Engineering at Udayana University, Bali, Indonesia, since 2001. (Email: dewi.wirastuti@unud.ac.id; https://orcid.org/0000-0001-5522-8684)

Lino Verlin, she has been a student at the Department of Electrical Engineering Udayana University Indonesia. She is also an employee at Lumonata as a website developer. Her research interest includes the development of mobile programming, applying new programming languages in building a website and improving code writing. Currently, she also very interested in implementing machine learning into a website. (Email: lnoverlin@gmail.com; https://orcid.org/0009-0007-6591-8989)

Is-haka Mkwawa, he received the Ph.D. degree in computing from the University of Bradford, Bradford, U.K.,is currently a research fellow with Signal Processing and Multimedia Communications Research at Plymouth University, Plymouth, U.K. He has been working in various capacities on EU FP6 and FP7 projects since 2002 with Plymouth University, the University of Bradford, and the University College Dublin, Dublin, Ireland. He has authored several refereed publications, and coauthored the book Guide to Voice and Video Over IP: For Fixed and Mobile Networks (Springer, 2013). His current research interests include IMS media plane security for next generation of emergency communication and services, VoIP quality adaptations, mobility management in mobile and wireless networks, power saving in mobile devices, overlay networks, performance analysis and evaluation of IMS mobility management, parallel computing, and collective communication. (Email: is-haka.mkwawa@plymouth@ac.id; https://orcid.org/0000-0002-8399-0737)

Khalid G Samarah, he received the B.Sc. degree from Mutah University in Jordan in 1991, and the MSc and PHD from University of Bradford in UK in 2003 and 2007 respectively. From 17/02/2008, he worked as an assistant Professor in Mutah University until now. Khalid occupied the position of head of the electrical engineering department in Mutah University, and chief of coordinators in the military wing of Mutah University. He is interesting in radio propagation, OFDM systems in cellular communications.
(Email: kgsamarah@mutah.edu.jo; https://orcid.org/0000-0002-6288-8389)