RESEARCH ARTICLE OPEN ACCESS

Nearby Jamu Store Locator Application Using Dijkstra Algorithm

Dwi Wulandari Sari*, Kurnia Gusti Ayu**, Yudo Devianto***
*(Faculty of Computer Science, Universitas Mercu Buana, Jakarta
Email: dwi.wulandari@mercu buana.ac.id)
** (Faculty of Computer Science, Universitas Mercu Buana, Jakarta
Email: kurnia.gusti@mercubuana.ac.id)
*** (Faculty of Computer Science, Universitas Mercu Buana, Jakarta
Email:yudo.devianto@mercubuana.ac.id)

----------**********----------

Abstract:

Jamu is a traditional medicine made from natural herbs that has been passed down as a cultural heritage throughout generations. The demand for jamu continues to increase even today, and it is now served even in star-rated hotels. Jamu gendong is a kind of jamu that is consumed by all kinds of people, from adults to children. However, jamu gendong is often only sold during certain hours, which makes it difficultfor consumers to buy it. This is an issue for those who love drinking jamu, especially for its health benefits. It is difficult to predict whether jamu gendong is in available or not. This study aims to calculate the shortest path from a nearby jamu store from a buyer, using Dijkstra algorithm with directed Graph. From testing conducted using this algorithm, a better distance optimization has been achieved. It is thus expected that this algorithm will help those who are in need of jamu under certain circumstances. The implementation of this research comes in the form of a prototype Android-based application.

Keywords — **Dijkstra, Android, Jamu, Graph**

----------**********----------

I. INTRODUCTION

This Jamu is an alternative medicinal drink that is consumed for its health benefits and has even been considered a favorite drink to some. By means of marketing, there are various types of jamu, most notable ones are jamu gendong, jamu godokan, brewed powder, pill and syrup. Jamu gendong is one of the most popular jamu because it is affordable and easy to get. However, jamu gendong does not bring the complete list of jamu as mentioned above. Because it is sold through peddling or done door-to-door, so it is not always available. This poses an issue to consumers when they are in need of jamu under certain circumstances and the jamu is not being sold.

The limitation of this study is that it conducts a simulated calculation using Dijkstra Algorithm, creates a prototype of Android-based application for locating jamu store, and provides complete information of existing jamu stores.

Several researchers have applied the Djistra Algorithm, including:

Jenni Veronika Ginting and Ertina Sabarita Barus, in their research entitled Application of Determination of Nearby Hospitals Using Dijkstra's Algorithm, discusses the application of this algorithm to find the shortest path to the nearest hospital. [4]

Mahrizal Masri, et al in his study entitled Dijkstra's Algorithm Implementation in Designing the Shortest Route Determination Application in Lake Toba and Surrounding Tourism Objects, discusses the fastest route to reach Lake Toba and its surroundings by applying this algorithm. [9]

Sunardi, et al in his research entitled Implementation of the Dijkstra Algorithm for Analysis of Transjogja Public Transport Routes Based on Android, discusses the fastest route to reach the destination of passengers without having to pass through routes that could have been avoided. [11]

In this study, the writer employs Dijkstra Algorithm because it is simple compared to other algorithms for calculating shortest path; when running time is considered, the algorithm is also more efficient. Because the determined point for locating the nearest jamu stores from the user yields positive value, this Algorithm is a more suitable choice to help user in buying jamu without having to worry about the time it is being sold.

II. THEORY AND METHOD

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

A. Shortest Path

The term shortest path here is defined as minimizing the weight of a graph[7]. In the Determining process, *Shortest Path* requires an input in the form of origin and end data. The process will then take the data from *node database* along with the distance. Afterwards, the system will check the database for the shortest path, if the given origin and end data have already been processed, a process result will be obtained from the shortest path database, if the data have not been processed, the system will save the calculation of the shortest path query in the shortest path database.[6]

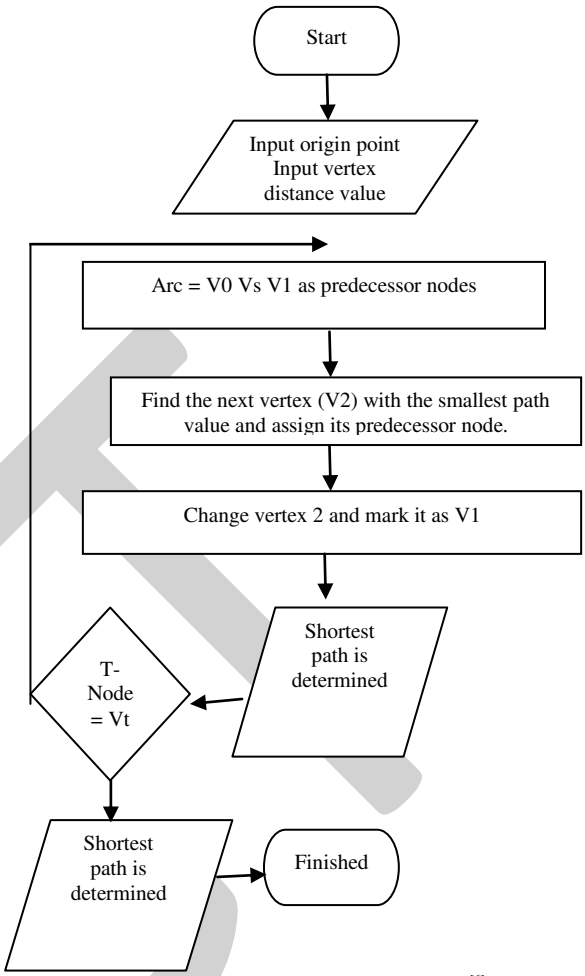


Fig 1. Flowhart *Shortest Path*[6]

B. Dijkstra Algorithm

Dijkstra Algorithm is an algorithm that determines the shortest path from one vertex to another vertex in a weighted graph. The distance between vertices is the weight of each edge in a graph. A weight must have a positive value (weight >= 0). Dijkstra Algorithm was conceived by Edger Wybe Dijkstra. The algorithm is also known as greedy algorithm, or an algorithm that solves a problem by finding the maximum value. The Dijkstra Algorithm works in finding by doing calculation between a vertex and another nearby vertex, then another nearby vertex, etc.[3]

An example of a graph to solve using Dijkstra Algorithm is as follows:

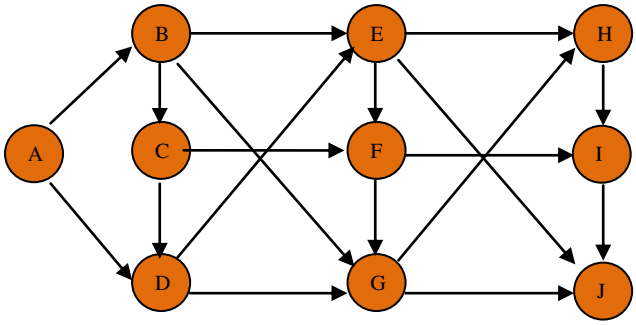


Fig 2. Graph of Dijkstra Algorithm[6]

C. Android

Android is an operation system originally developed for Linux-based mobile devices by Android Inc.,which was later acquired by Google in 2005[8]. Firebase is a backend service that is now run by Google. It serves as Google's solution for supporting mobile app developers. Firebase offers many features that allow app developers to develop mobile applications with ease[4].

D. Jamu

Jamu has long been a part of Indonesian cultural heritage, and according to Basic Health Research (Riset Kesehatan Dasar), Indonesian consumption of jamu has exceeded 50%. Jamu has been a part of traditional medicine.Traditional medicine has developed in many parts of the world and has seen increasing growth in popularity. As a nation rich in natural herbs,there are many variations of jamu made by the many local tribes of Indonesia, from Sabang to Merauke. Jamu has been passed down through generations as a medicinal drink for health benefits[2]. Research shows that 49.53% of people consume jamu for maintaining health or for medication. About 95.6% respondents who consume jamu state that they get health benefit from drinking it. A study by Riskesdas in 2010 also shown that among those who consume jamu, 55.3% consume it in liquid form (*infusum/decoct*), while the remaining 44.7% consumed it as powder or pills/tablets[6].

It is possible to identify the use of jamu as alternative to modern medicine based on the amount of jamu consumed. The data of jamu consumption as an alternative to modern medicine among people of lower-middle class economy can be seen in table 2[6].

TABLE 1
CONSUMPTION RATE OF JAMU AMONG THE LOWER-MIDDLE CLASS

Economic Level	Income	Amount	Percentage
Low	< 1 million/month	93	58%
Middle	1.5-5 million/month	40	25%
High	>5million/month	27	17%

III. RESEACH METHOD

The research methodology of this research only reaches the system designing stage. While the employed methodology is quantitative with case study, conducted specifically in Mampang district.

A. Research Stages

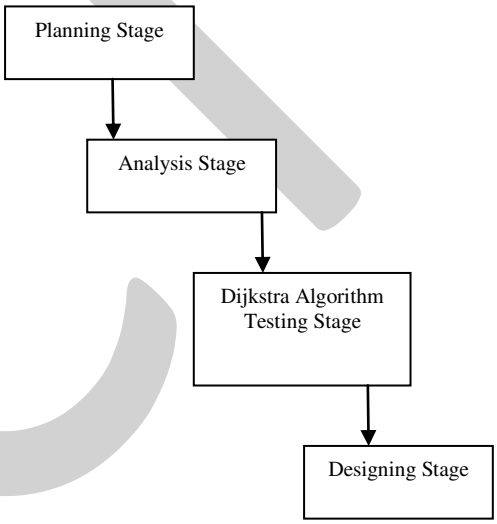


Fig 3. Stages in the research methodology

B. Data Collection Method

Data collection methods used in this study are:

1. Observation
   Observation is a method of collecting data through directly observing and getting involved as buyer to jamu stores or jamu gendong peddlers.
2. Interview
   Interview is the exchange of information between a party who asks questions and another party who serves as the source of information and provides answers or opinions to the questions. During this interview stage, the writer would interview

jamu sellers in regard to the jamu products they sell.

3. Library reviews
   The act of taking conclusion based on the data from references about Dijkstra Algorithm that are implemented in finding nearest path from a designated location.

IV. RESEACH METHOD

There are several algorithms that can be used to determine shortest path. Among them is by using dijkstra algorithm. This algorithm has been used extensively in determining the quickest route or shortest path based on certain limiting criteria[4].
For example:
V(G) = {v1, v2,…., vn}

L = Set of point ɛ V(G) that has been selected in shortest path.
D(j) = Sum of the weight of smallest path from v1 to vj.
W(I,j) = Weight of line from point v1 to vj
W*(l,j) = Sum of weight of smallest path from v1 to vj [4]

This algorithm aims to determine the shortest path using the smallest weight from a predecessor point. The following is the process of determining the shortest path to a nearby Toko Jamu using Dijkstra Algorithm. Here are the stages employed in this research:

A. ***Planning***

During this planning stage, researcher has conducted interview with owners of Jamu Stores and has collected relevant data. Because Jamu Store usually use the name of their own jamu products, researcher assigns different letters for each Jamu Store, while their location nodes are designated using the letter v. The originpoint is designated as V1. Here are the data obtained during this stage::

TABLE 2.
LIST OF JAMU STORE

Name of Jamu Store	Address	Store Location Node
Jamu Store A	Jl.Tegal Parang Selatan 1	V2
Jamu Store B	Jl. Pancoran Barat VII	V3
Jamu Store C	Jl.Mampang Prapatan VIII	V4
Jamu Store D	Jl. Mampang Prapatan iX	V5
Jamu Store E	Jl.Kemang Utara 13	V6
Jamu Store F	JL. Bangka Raya	V7
Jamu Store G	Jl. Bangka IIG no.23 A	V8
Jamu Store H	Santa Modern Place	V9
Jamu Store I	Pasar Santa Modern	V10
Jamu Store J	Jl. Warung Buncit raya	V11
Jamu Store K	Pondok Jaya	V12
Jamu Store L	Jl.Kapt. Tendean No.32	V13
Jamu Store M	Jl.Bangka II Pela Mampang	V14

Based on the above table, the locations of the Jamu Store can thus be mapped as follows:

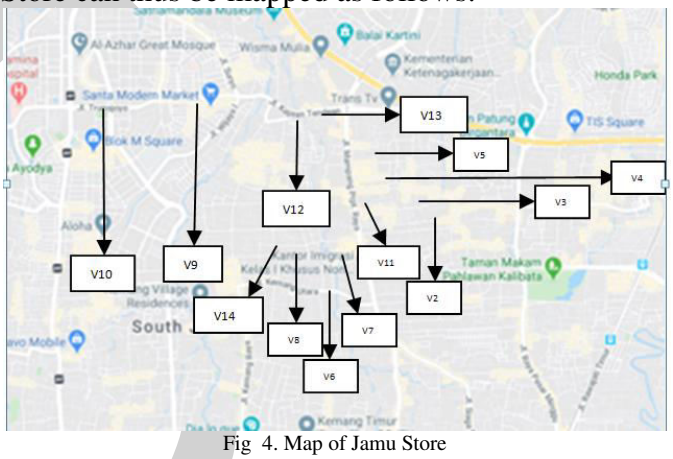


Fig 4. Map of Jamu Store

B. ***Analysis***

During analysis stage, based on the data of jamu store and location mapping, a routing using odometer (vehicle's mileage) yields the following data:

TABLE 3
DISTANCE BETWEEN JAMU STORE POINTS

Jamu Store Nodes	Distance
V1 –V2	1
V1 – V3	2
V1 – V4	3
V2 – V5	6
V2 –V7	10
V2 – V3	2
V3 – V6	5
V3 – V4	2
V4 –V5	4
V4 – V7	6
V5 – V8	6
V5 – V10	12
V5 V6	4
V6 – V9	15
V6 – V7	8
V7 – V10	15
V7 – V8	5

V8 – V11	7
V8 – V9	10
V8 – V13	13
V9 – V10	2
V9 – V12	6
V10 – V13	7
V11 – V12	4
V12 – V13	5
V11-V14	8
V12 – V14	7
V13 – V14	8

Based on this data, researcher creates the following directed graph:

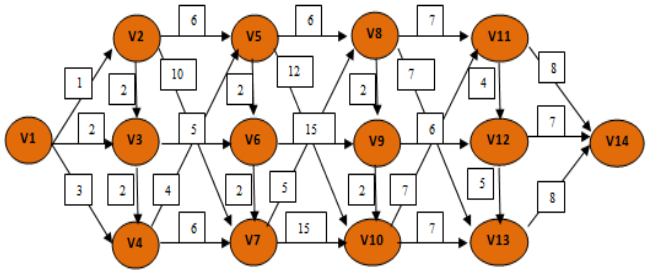


Fig 5. Directed Graph of Calculation

C. ***Testing of Dijkstra Algorithm***

The steps for determining shortest path from V1 to V14 using Dijkstra algorithm based on the above graph is as follows:

1. At first, the status of unselected node is assigned the value of "0" while selected nodes are assigned the value of "1", starting from node V1.
2. From the node that directly connects with predecessor node, which is node V1, an input of weight is made based on the weight table.
3. Predecessor (origin node) of nodes V1, V2, V3, and V4 is V1, because the distance is calculated from node V1, as such node V1 is called a predecessor node, while nodes V5, V6, V7, etc is marked "-" or no distance since there is no arc that traces back to node V1.

Table 5. Iteration 1

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
Status	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Weight	-	1	2	3	-	-	-	-	-	-	-	-	-	-
Predecessor	V	V	V	V	-	-	-	-	-	-	-	-	-	-
	1	1	1	1										

Table 6. Iteration 2

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
Status	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Weight	-	1	2	3	4	-	-	-	-	-	-	-	-	-
Predecessor	V1	V1	V2	V1	V4	-	-	-	-	-	-	-	-	-

Table 7. Iteration 3

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
Status	1	1	1	0	0	0	0	0	0	0	0	0	0	0
Weight	-	1	2	3	4	4	-	-	-	-	-	-	-	-
Predecessor	V1	V1	V2	V1	V4	V5	-	-	-	-	-	-	-	-

Table 8. Iteration 4

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
Status	1	1	1	1	0	0	0	0	0	0	0	0	0	0
Weight	-	1	2	3	4	4	6	-	-	-	-	-	-	-
Predecessor	V1	V1	V2	V1	V4	V5	V4	-	-	-	-	-	-	-

Table 9. Iteration 5

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
Status	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Weight	-	1	2	3	4	4	6	6	-	2	-	-	-	-
Predecessor	V1	V1	V2	V1	V4	V5	V4	V5	-	V9	-	-	-	-

Table 10. Iteration 6

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
Status	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Weight	-	1	2	3	4	4	6	6	10	2	-	-	-	-
Predecessor	V1	V1	V2	V1	V4	V5	V4	V5	V8	V9	-	-	-	-

Table 11. Iteration 7

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
Status	1	1	1	1	1	1	1	0	0	0	0	0	0	0
Weight	-	1	2	3	4	4	6	6	10	2	-	-	-	-
Predecessor	V1	V1	V2	V1	V4	V5	V4	V5	V8	V9	-	-	-	-

Table 12.Iteration 8

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
Status	1	1	1	1	1	1	1	1	0	0	0	0	0	0
Weight	-	1	2	3	4	4	6	6	10	2	7	-	7	-
Predecessor	V1	V1	V2	V1	V4	V5	V4	V5	V8	V9	V8	-	V8	-

Table 13. Iteration 9

Node	V	V	V	V	V	V	V	V	V	V	V	V	V	V
	1	1	1	1										

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Status	1	1	1	1	1	1	1	1	1	0	0	0	0	0
Weight	-	1	2	3	4	4	6	6	10	2	7	4	7	-
Predecessor	V1	V1	V2	V1	V4	V5	V4	V5	V8	V9	V8	V11	V8	-

Tale 13. Iteration 10

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
Status	1	1	1	1	1	1	1	1	1	1	0	0	0	0
Weight	-	1	2	3	4	4	6	6	10	2	7	4	7	-
Predecessor	V1	V1	V2	V1	V4	V5	V4	V5	V8	V9	V8	V11	V8	-

Table 14. Iteration 11

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
Status	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Weight	-	1	2	3	4	4	6	6	10	2	7	4	7	-
Predecessor	V1	V1	V2	V1	V4	V5	V4	V5	V8	V9	V8	V11	V8	-

Table 15. Iteration 12

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
Status	1	1	1	1	1	1	1	1	1	1	1	1	0	0
Weight	-	1	2	3	4	4	6	6	10	2	7	4	7	7
Predecessor	V1	V1	V2	V1	V4	V5	V4	V5	V8	V9	V8	V11	V8	V12

Table 16. Iteration 13

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
Status	1	1	1	1	1	1	1	1	1	1	1	1	1	0
Weight	-	1	2	3	4	4	6	6	10	2	7	4	7	7
Predecessor	V1	V1	V2	V1	V4	V5	V4	V5	V8	V9	V8	V11	V8	V12

Table 17. Iteration 14

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
Status	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Weight	-	1	2	3	4	4	6	6	10	2	7	4	7	7
Predecessor	V1	V1	V2	V1	V4	V5	V4	V5	V8	V9	V8	V11	V8	V12

Table 18. Result of Shortent Path with Dijkstra Algoritms

Weight	-	1	2	3	4	4	6	6	10	2	7	4	7	7
Predecessor	V1	V1	V2	V1	V4	V5	V4	V5	V8	V9	V8	V11	V8	V12

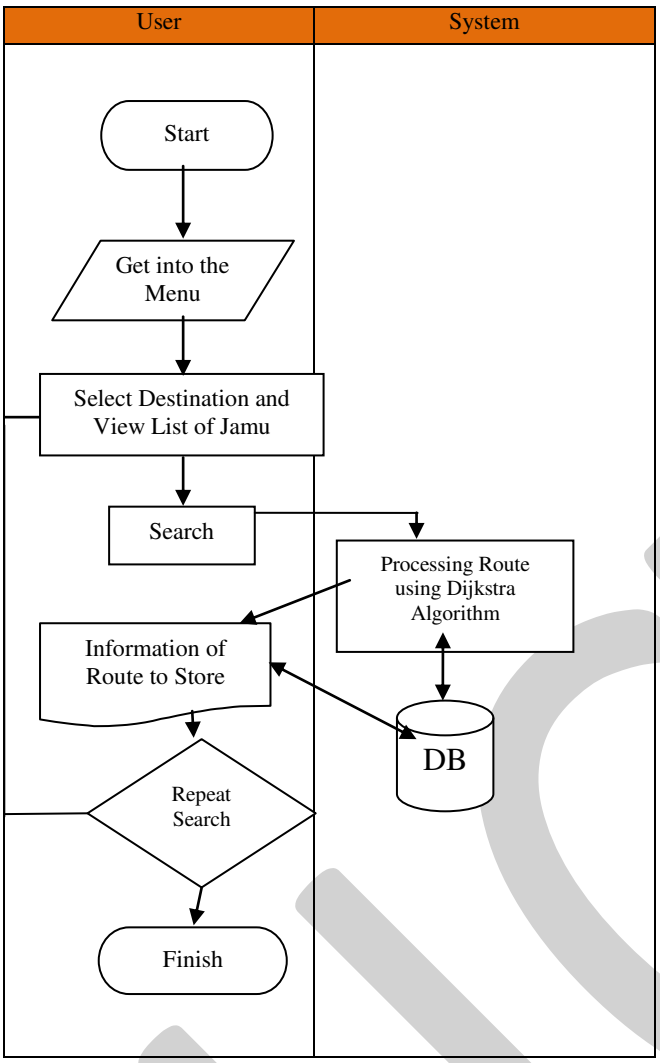
The program stops once all nodes are selected. It will then yield the shortest path from node A to every existing node. To see the selected arcs, the predecessors can be traced as follows:

:

V1 - V2 : 1
V1 – V3 : 2
V1 – V4 : 3
V1 – V4 – V5 : 4
V1 – V5 – V6 : 4
V1 – V4 – V7 : 6
V1 – V5 – V8 : 6
V1 – V5 - V8 – V9 : 10
V1 – V9 – V10 : 2
V1 – V5 - V8 – V11 : 7
V1 – V5-V8-V11 – V12 : 4
V1 – V5-V8-V11 –V12- V13 :7
V1 – V5-V8-V11 – V12 – V14 : 7

D. ***Desaign***

In the designing stage, a flowchart of the application is made.



E. ***Implementation***

Berikut In this stage, the design is implemented into a prototype Android-based app. The user menu utilizes the buyer's current location as origin/starting point:

1. Login Menu Interface
   This is the interface a user views when opening the Jamu Store Finder App. The user can choose submit if already has an account or choose register to sign up for an account.
2. Registration Menu Interface
   This is the interface for creating an account for the app. Required fields for user to fill are: Full Name, Address, Phone Nomber, Email, Username, Password. Once a registration is successful, a confirmation email will be sent for logging into the app.
3. Store Locator Interface
   This is the interface for running the nearby store locator based on user's current location at that moment. User can immediately see their current location that is used by the application as starting point for locating stores. By pressing the search menu, list of nearby jamu stores along with their corresponding distance will be displayed.
4. Jamu Store Interface
   This is the interface of a jamu store that has been selected from the main menu. User can see what products are currently available for sale. Unfortunately, the app has yet to offer ordering and delivery function, so order of purchase can only be made via WhatsApp.
5. Map Interface
   This interface will appear when user clicks on the map of any displayed jamu store. The app will then display the user's position and the position of the selected jamu store.

Fig 6. Login Menu

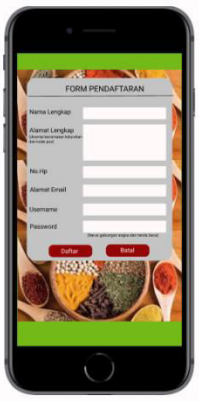


Fig 7. Registration Menu

Fig 8. Search Store Menu

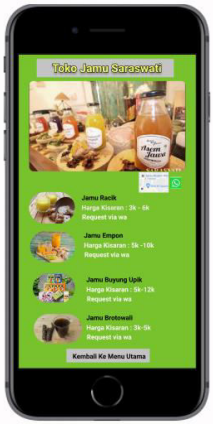


Fig 9. Jamu Store

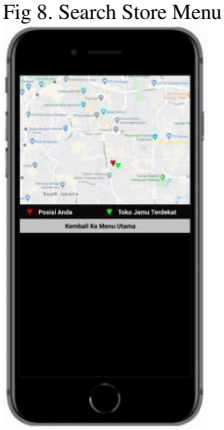


Fig 10. Map Menu

F. *ALPHA TESTING*

Testing is done using Blackbox method. Blackbox testing focuses on functional requirements of a software, with the objective of seeing if an application yields the expected/desired outputs that are in line with the function of the application.

TABLE 19

BLACKBOX TESTING

Input	Expected Results	Output Test	Test Result
Button Submit-Menu Login	Displaying jamu store search menu	Display jamu store search jamu	Ok
Registration link	Displaying registration form for new application user	Displayregistration form	Ok
Image of Jamu Store	Displaying the page of a jamu store	Display the image of selected jamu store	Ok
Map	Displaying the location of user and selected jamu store	Display the desired map	Ok
WA link	Displaying a whatsapp link that directly connects with the store's whatsapp's contact	Display WhatsApp interface	Ok
Cancel button – registration menu	Displaying login form	Display login form	Ok
Link to return to main menu	Displaying main menu visited before the current page	Display previous page	Ok

V. CONCLUSIONS

Based on the research and calculation using Dijkstra Algorithm, it can be concluded that:

1. The application is created based on the data of Jamu Stores collected from Mampang District. The data collected are among others: name, location, and products sold by each jamu store.
2. Measurement of distance is done using Odometer (mileage tracking device in vehicles).
3. By doing manual calculation using Dijkstra Algorithm, shortest path between user's location and nearest Jamu Store can be determined.

4. This jamu store locator application is an Android-based application. It consists of a login menu, store locator, map, and link to seller/vendor's WhatsApp contact, details of products sold in each store. Order and inquiries about items from the store can only be made via WhatsApp, and delivery can be made using Gosend. This application does not yet include features for purchase or delivery.

ACKNOWLEDGMENT

Our praise and gratitude goes to the LORD for all His grace and gifts so that this internal research can be successfully completed. Do not forget to thank our family *Faculty Of Computer Science, Universitas Mercu Buana,* and especially *PUSLIT Universitas Mercu Buana* for providing research funding support. The theme chosen in this internal research activity was *"NEARBY JAMU STORE LOCATOR USING DIJKSTRA ALGORITM"*. The author realizes that this report is far from perfect and therefore constructive criticism and suggestions so the authors expect to be better.

REFERENCES

[1] Anisah , N., Ecintia, D., Lukman , A., & Devianto, Y, Android-Based Boarding House Management Information System, *International Journal of Computer Techniques-Volume 5 Issue 6, Nov-Dec 2018.*

[2] Andriati , & Wahjudi., Tingkat Penerimaan Pengunaan Jamu sebagai Alternatif Obat Modern, *Journal Masyarakat, Kebudayaan dan pollitik, Vol.29, No.3*, 133-145.2016

[3] Fitroh , A., Bachri, O. S., Purnomo, A. I., & Frendianata, I., Shortest Path Finding In Geographical Information Systems Using Node Combination and Dijkstra Algorithm, *International Journal of Mechanical Engineering and technology (IJMET) Vol.9, Isuue 2*.

[4] Ginting, J. V.,. Aplikasi Penentuan Rute Rumah Sakit Terdekat Menggunakan Algoritma Dijkstra. *Jurnal Mantik Penusa, Volume 2, No.2*, Desember 2018

[5] Kesehatan, B. L., *Laporan hasil riset kesehatan dasar tahun 2010,* Jakarta: Badan Litbang Kesehatan.

[6] Khoiruddin, H. M., & Khairina , N., Pencarian Jalur Terpendek dengan Algoritma Dijkstra, *Sinkron Jurnal&Penelitian Teknik Informatika, Volume 2 Nomor 2, Oktober 2017.*

[7] Kusuma, E., Jefri, & Agung, H., Aplikasi Perhitungan Dan Visualisai Jarak Terpendek Berdasarkan Data Coordinate Dengan Algoritma Dijkstra Dalam Kasus Pengantaran Barang Di Kawasan Jabodetabek, *Jurnal SISFOKOM, Vol.8 No.1*.Maret 2019

[8] Malyana , E. ,Pemanfaat Android Dalam Perancangan Aplikasi Kumpulan Doa, *Jurnal Sains Dan Informatika, V4.11, 26 April 2018*.

[9] Masri, M., Kiswanto, A. P., & Kusuma, B. S., Implementasi Algoritma Dijkstra Dalam Perancangan Aplikasi Penentuan Rute Terpendek Pada Objek Pariwisata Danau Toba dan Sekitarnya, *SEMNASTEK UISU*.2019

[10] Mulyani, *eprints.upnjatim.ac.id.* Retrieved Novmber 19, 2019, from upnjatim.ac.id: http://eprints.upnjatim.ac.id/1710/1/file_1.pdf.

[11] Sunardi, Yudhana, A., & Kadim, A. A., Implementasi Algoritma Dijkstra Untuk Analisis Rute Transportasi Umum Transjogja Berbasis Android, *Jurnal Sistem Informasi Bisnis, 9 Mei 2019*.