ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 2, July 2024

THE DEVELOPMENT OF A MOBILE-BASED AREA RECOMMENDATION SYSTEM USING GRID-BASED AREA SKYLINE QUERY AND GOOGLE MAPS

Annisa Annisa[1], T. Sandra Alyssa[2], Muhammad Asyhar Agmalaro[3]

[1,2,3]Department of Computer Science, IPB University, Indonesia

email: annisa@apps.ipb.ac.id[1], sandra_tengku08@apps.ipb.ac.id[2], agmalaro@apps.ipb.ac.id[3]

Abstract

Choosing a location for a business or place of residence is an essential task in our daily lives. Typically, a location is considered favorable if it is in proximity to profitable facilities that enhance its value while being distant from facilities that diminish its value. However, conducting surveys in the field to identify desirable candidate locations is not always feasible. Factors such as high costs, inclement weather, and transportation limitations can hinder survey activities. This study aims to develop a mobile-based system for location selection using the Grid-based Area Skyline (GASKY) algorithm in conjunction with Google Maps. Google Maps is widely utilized for location-based decisions and is familiar to mobile application users. GASKY is employed for its capability to recommend locations based on user-provided information regarding desired facilities near the target location and facilities to be avoided, eliminating the need to input candidate locations from survey results. The outcomes of this study include a mobile-based application that utilizes the Google Maps API to create data collection modules. This system is built using the Waterfall method, which consists of planning, analysis, design, implementation, and testing stages. The time required to obtain area recommendations depends on the size of the target area, the number of grids, and the quantity of facility types provided by the user. With each increase, the time needed to obtain area recommendations is rising significantly. Mobile-based applications utilizing GASKY offer convenience, as they can be accessed by users anytime and anywhere.

Keywords: grid-based area skyline query, GASKY, Google Maps, location selection, skyline query

Received: 20-07-2023 | **Revised:** 02-04-2024 | **Accepted:** 15-04-2024
DOI: https://doi.org/10.23887/janapati.v13i2.66155

INTRODUCTION

Location selection is a crucial aspect of daily life, particularly when it comes to business development. The choice of location significantly impacts the efficiency and effectiveness of a business [1][2][3][4]. In general, an ideal location should be in close proximity to beneficial facilities such as universities, residential areas, supermarkets, and transportation hubs, while avoiding undesirable facilities like cemeteries, garbage dumps, or construction sites.

There are currently numerous applications that offer location selection services, one of which is Google Maps [5][6][7]. Google Maps utilizes a nearby query to identify a location on a map [8][9]. However, Google Maps only considers one desirable facility when selecting a location. For instance, if a student wants to find the nearest supermarket, the nearby feature will display the nearest supermarket from the user's current location. Unfortunately, this feature cannot be utilized for location selection that takes into account multiple types of nearby facilities.

Extensive research has been conducted on site selection by utilizing locations represented as points (point selection) on a map and considering multiple types of surrounding facilities. One commonly employed method to address these challenges is the skyline query [10][11][12]. Figure 1 illustrates an example of a skyline query for hotel selection based on distance and price [13].

Hotel	Price ($)	Distance to the beach (km)
H1	20	1.0
H2	20	3.0
H3	35	0.5
H4	15	1.0
H5	10	2.0
H6	25	2.5

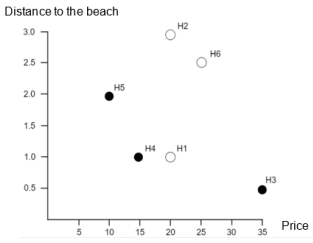

Figure 1. Skyline query example

In Fig. 1, there are 6 different hotels, namely H1, H2, H3, H4, H5, and H6. Each hotel has two dimensions: price ($) and distance (km). It can be observed that H2 and H6 are dominated by H4 and H5 because H2 and H6 have higher prices and longer distances compared to H4 and H5. H1 is dominated by H4 because it has a higher price despite having the same distance. H3 has the closest distance among all the hotels. Therefore, H3, H4, and H5 (indicated by black dots) are the skyline hotels in this case.

Generally, location selection is done by selecting from several candidate locations using the criteria of distance to the desirable facility and the undesirable facility. Next, the skyline query will select from several candidate locations those that are not dominated by others. A location is said to dominate another location if it is closer to the desirable facility and further away from the undesirable facility.

Research in [14] proposes an SSQ algorithm that utilizes Voronoi diagrams to reduce data in the skyline process, namely Voronoi-based Spatial Skyline (VS^2) for the selection of the dominant location. The research in [15] applies aggregation R-trees to select desired locations based on surrounding facilities. The research in [16] and [17] implemented a skyline-based location selection algorithm using Google Maps as a web-based application so that the benefits can be easily and quickly felt by the community.

The studies mentioned above require input in the form of location candidates as points to be selected and provide output in the form of recommendations for point locations based on desirable preferences. However, in certain situations, candidate sites to choose from may not be readily available. This could be due to users being unfamiliar with a new area or limitations that prevent surveying to identify candidate locations. In such cases, it becomes necessary to select areas or locations that do not rely on the availability of candidate sites.

The Grid-based Area Skyline Query (GASKY) algorithm was proposed in [18] and [19] to identify attractive areas based on surrounding facilities, even without the availability of candidate locations. GASKY simplifies the process of answering queries related to location selection without requiring a survey to find candidate locations. In a separate study conducted in [20], GASKY was utilized to identify attractive areas for a company's business establishment based on desirable surrounding facilities. Furthermore, in the same study, GASKY was implemented in a web-based application.

Continuing from the research conducted in [9], this study aims to implement the GASKY algorithm on Google Maps as a mobile-based location recommendation application. Mobile-based applications are more easily accessible to the public and provide convenience in terms of usability, as they can be accessed anywhere and at any time.

Enter a destination area, such as a city. Then input the desired facilities, and also input the undesired facilities into the system. Once all parameters are filled, the system will process all parameters using GASky and generate areas that match the input parameters. The development of this system utilizes the stages of the Waterfall method, considering that in the construction of this system, each stage is completed before moving on to the next stage, making the transition from one stage to the next linear and without overlapping. The requirements for this system are also stable and clear, allowing the use of the Waterfall method.

This study also includes some experiments to analyze the performance of GASKY on the developed mobile application. The remainder of this paper is organized as follows: the next section presents the methodology of our research, followed by the results and discussion. Finally, we conclude the paper and provide directions for future work.

METHOD

The research was conducted using the Waterfall method, which consists of five stages; planning, analysis, design, implementation, and testing stage. In this section, we will provide thorough explanations for each step.

In the planning step, we lay the groundwork for the project by defining the objectives and requirements of the system. The objective of building this system is to provide a location recommendation system without requiring users to input candidate locations for selection, which is typically obtained from survey results. In this system, users only need to specify which facilities they prefer to be near the recommended area and which facilities they prefer to be far from the recommended area. This system is developed using Google Maps, considering that Google Maps is a widely recognized map application among the public. The target users for this system are those who require location recommendations without needing to conduct a survey beforehand, with parameters such as the desired types of facilities near and far from the target area according to their preferences. These may include tourists, locals, travelers, food enthusiasts, or the public in general.

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 2, July 2024

First, the system should allow users to input their preferences regarding the targeted area, such as a city or village, and the types of facilities they want to be near or far from the recommended area. This includes specifying preferred facilities and dispreferred facilities. Based on the user's input, the system should generate recommendations for suitable locations that meet the specified criteria. This involves analyzing geographic data and identifying locations that align with the user's preferences. Since the system is built using Google Maps, it would involve integrating with Google Maps APIs to access geographical data, display maps, and provide location-based services. The system should have a user-friendly interface that allows users to interact with it easily. This includes providing options for inputting preferences, displaying recommended locations on the map, and presenting relevant information about each location. The system should be scalable to handle a large number of users and locations efficiently. It should also be optimized for performance to deliver fast and responsive recommendations.

Based on previous stages, in the analysis stage, we define how users will interact with the system. In this stage, we also determine what inputs are needed in the location selection process. The expected user input consists of spatial information in the form of location (latitude and longitude) considered as the query to select Points of Interest (POI). User inputs to execute this algorithm include the region's name and the desired and undesired types of facilities based on their preferences. For example, desired facilities may include universities and stations, while undesired facilities may include major competitors. The data for these POI facilities is obtained from the Google Maps API. POIs are markers on Google Maps that denote specific places or locations. The POI data employed in this study consists of spatial information, specifically the latitude and longitude coordinates that represent the POI locations.

Subsequently, spatial information in the form of latitude and longitude is retrieved from the POI specified by the user using the data collection module. The type of Google Maps API used is the Geocoding API to obtain latitude and longitude information for the facilities inputted by the user. Based on the analysis above, the feature to be developed is a recommendation search feature by filling out the provided form, consisting of the desired city region, desired types of facilities, and undesired types of facilities.

The design phase includes designing the system's interface as well as its system design. The system's interface design is carried out using Figma software. There are four system pages in the mockup created, namely the home page, dashboard, the recommended area search form which includes the target region, desired and undesired facilities, and the recommendation results page. Additionally, system design involves creating activity diagrams. Activity diagrams are created to model the processes that will be implemented in the system, from the user opening the application to the system displaying the recommended locations on the map. The recommended area search system's activity diagram can be seen in Fig. 2.

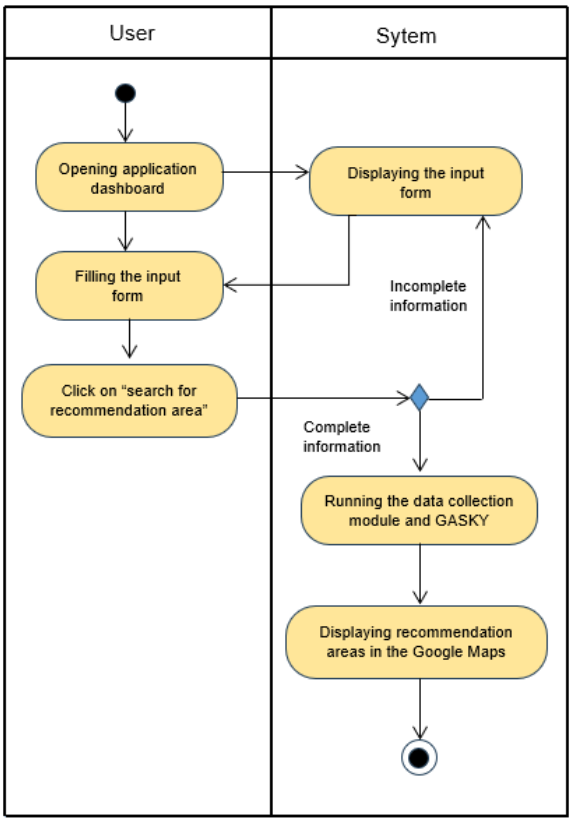

Figure 2. Activity diagram

The implementation process begins with a needs analysis to identify the features to be developed based on the activity diagram. This stage provides an initial understanding of the system to be created. Subsequently, the system design phase commences, encompassing both the interface design and system design. Following the implementation, the system undergoes testing to ensure its functionality and performance. In the forthcoming system, users will be prompted to fill out a form specifying their desired area or target location, as well as the types of facilities they desire or wish to avoid. Subsequently, the system will display the recommended areas based on the user's preferences.

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 2, July 2024

In the implementation stage, the creation of the GASKY module is also conducted, which is the implementation of the GASKY algorithm using the Python programming language. Consider a rectangular target area, denoted as A, on a map where a businessman intends to construct a new store. To simplify the process, we assume that the rectangular target area A is a square region. Initially, we divide A, into grids of size $s \times s$. Each grid is assigned a unique ID number, starting from the top-left grid $g(0,0)$ and progressing to the bottom-right grid $g(s-1, s-1)$. Furthermore, each grid is surrounded by four vertices: $v(i,j), v(i,j+1), v(i+1,j)$, and $v(i+1,j+1)$. These vertices represent the top-left (tL), top-right (tR), bottom-left (bL), and bottom-right (bR) corners, respectively, of the grid labeled as $g(i,j)$ (where $0 \leq i \leq s$ and $0 \leq j \leq s$). We call each grid a disjoint area of the target area. Fig. 3 depicts area A divided into $8\,x\,8$ grids, with the grid and vertex numbering referring to the previous explanation.

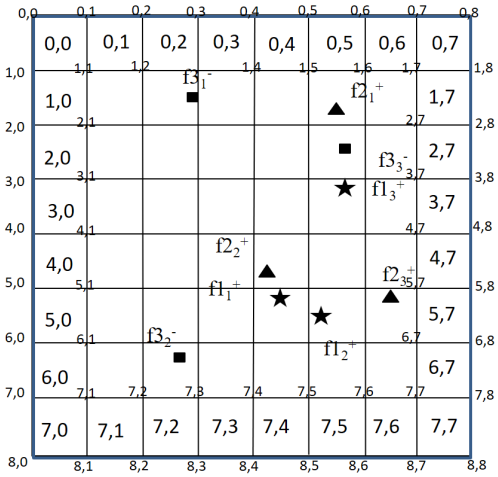

Figure 3. Disjoint areas of the targeted area [18]

Let $F = \{F1, F2, F3, \ldots, Fn\}$ represent a set of facility types, categorized into n distinct types. Each facility type can be further classified either as desirable $(+)$ or undesirable $(-)$. Moreover, each facility type consists of multiple facility objects. For instance, a desirable facility $F1^+$ is composed of two objects, $F1^+ = \{f1_1^+, f1_2^+\}$. In Fig. 3, there are three types of facilities: $F1$ and $F2$ which represent the desirable facility type, and $F3$, which represents the undesirable facility type. Each facility type consists of three objects located within the target area.

Let fk^{min} refer to the object within Fk that has the smallest or equal $dist_{min}(g, fk^{min})$ value compared to any other object in Fk. Similarly, the object fk^{max} is the object within Fk that has the largest or equal $dist_{max}(g, fk^{max})$value compared to any other object in Fk. The terms $dist_{min}(g, Fk)$ and $dist_{max}(g, Fk)$ represent the minimum and maximum Euclidean distance, respectively, from grid g to fk^{min} and fk^{max}. When considering two grids, g_i, and g_j, we say that g_i dominates g_j if, for all k $(1 \leq k \leq m)$, $dist_{max}(g, Fk)$ is smaller than $dist_{min}(g, Fk)$, we call this rule GASKY's dominance rule. The objective of GASKY is to find the collection of all grids that are not dominated by any other grid within the rectangular target area, A. The GASKY module was developed using the Python programming language based on the GASKY algorithm as shown in Fig. 4.

Algorithm 1 : Grid-based Area Skyline Algorithm (GASky Algorithm)

Input : $F_{1\ldots k}, s$
Output : Sky
1. for each $F_k \in F$ do
2. build_voronoi $V(F_k)$
3. for each vertex $v(i,j)$ from $v(0,0)$ to $v(s,s)$
4. find the closest F_k in $V(F_k)$ from $v(i,j)$
5. for each grid $g(i,j)$ from $g(0,0)$ to $g(s-1, s-1)$
6. calculate $dist_{min}(g(i,j), F_k)$ and record it into T
7. calculate $dist_{max}(g(i,j), F_k)$ and record it into T

// procedure to remove dominated grids
8. for each record in $T, t_{1\ldots p}$ do
9. if $t_i.F_{k.max} \leq t_j.F_{k.min}$ for all $F_k \in F$ do
10. remove dominated grid t_j from T

11. return Sky

Figure 4. Pseudocode of GASKY [18]

The workflow of the GASKY algorithm starts by determining the target area and then specifying the desired and undesired types of facilities. Next, create an $s\ x\ s$-sized grid, where s is a value determined by the user, and each grid has four vertices. Then, calculate the distance between each facility object and the grid. The search for the nearest facilities is done using Voronoi diagrams. These distances are then recorded in the min-max table, capturing both the minimum and maximum values for each type of facility from each grid. Next, select the non-dominated grid using the minimum and maximum distances. Subsequently, utilizing the GASKY domination rule, we can identify the skyline area grid. The domination rule states that a grid a dominates a' if the maximum value of the distance between grid *a* and its facility is smaller than the minimum value of the distance between grid *a'*

and its facility. Non-dominated grids are referred to as skyline area grids.

In the next stage, black box testing will be conducted on the developed system. In black box testing, the tester interacts with the software's user interface and provides inputs, and then observes the outputs generated by the software to ensure that the software meets the specified requirements and performs as expected from the end user's perspective. The testing process will simulate user interactions, including form completion and receiving area recommendations tailored to user preferences.

Additionally, an experiment is conducted to measure the performance of the developed GASKY algorithm in terms of the time required to locate a specific area on Google Maps. The performance of the GASKY algorithm will be highly influenced by the parameters entered by the user, namely the number of grids, the area to be searched for recommendations, and the number and types of facilities. An experiment needs to be carried out to observe how these parameters affect the performance of GASKY in the recommendation system, both in terms of running time and the number of skyline grids generated.

With a good understanding of how parameters affect system performance, users can optimize configurations or settings to achieve the desired performance. If users encounter issues with system performance, understanding the impact of parameters can help diagnose the problem. Users can determine if performance degradation is caused by changes in certain parameters and thus find appropriate solutions. Additionally, with knowledge of how parameters affect performance, users can make better decisions in selecting optimal parameters for their needs. This can lead to a better user experience and better outcomes from the system used.

Furthermore, during the experimentation phase, the system's performance in generating a skyline area is also tested by executing two scenarios. Scenario 1 examines the relationship between grid size, the number of facilities, and the resulting skyline area ratio. The resulting skyline area ratio is the number of skyline area grids generated compared to all grids generated by the GASKY algorithm. A high skyline area ratio (approaching one) means that almost all grids are skyline, indicating that the GASKY algorithm fails to provide recommendations. Such recommendation results are certainly not desirable for users.

Scenario 2 examines the relationship between grid size, number of facilities, and running time. The more grids entered by the user, the more time GASKY needs to calculate the distance between grids and facilities. Similarly, with more facilities, GASKY requires more time to calculate the distance between facilities and grids. Scenario 2 will provide users with insight to consider the number of grids and facilities before inputting them into the system because each addition of grids and facilities will result in a decrease in GASKY's performance.

Both scenarios are designed to assess GASKY's performance with various combinations of parameters entered by the user. This is crucial for users to understand how each additional parameter given to the system affects the recommendation result and time it takes for the system to generate recommendations. The experiment is conducted using two target area samples with significant differences in size: Medan City with an area of 265.1 km² and Aceh City with an area of 61.36 km².

RESULT AND DISCUSSION

The system is implemented using the JavaScript programming language and the React Native framework, while the algorithm development employs the Python programming language and the Flask framework. This integration allows for the incorporation of the GASKY algorithm and its results into the created mobile system. The Google Maps Places API is utilized to display map results from the algorithm, with data being sent in JSON format. The system's code is written in Python with the Flask framework used to develop the API endpoints.

The API collects data, processes the GASKY algorithm at the server, and returns the results to the mobile application. The mobile application visualizes the data in the form of maps, utilizing Google Maps as a supporting technology.

Fig. 5 represents the initial interface or landing page that appears upon opening the application. Users can proceed by clicking the "Cari area sekarang" (Search for recommendation area now) button. Users are required to complete four fields in the form: city area, desired facility type, undesired facility type, and grid size. The city area refers to the targeted search area. The desirable facility type corresponds to the facility type users desire to be close to the recommended area. Conversely, the undesirable facility types represent the facility types that users wish to avoid, ensuring that the recommended areas are not close to those undesired facilities.

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 2, July 2024

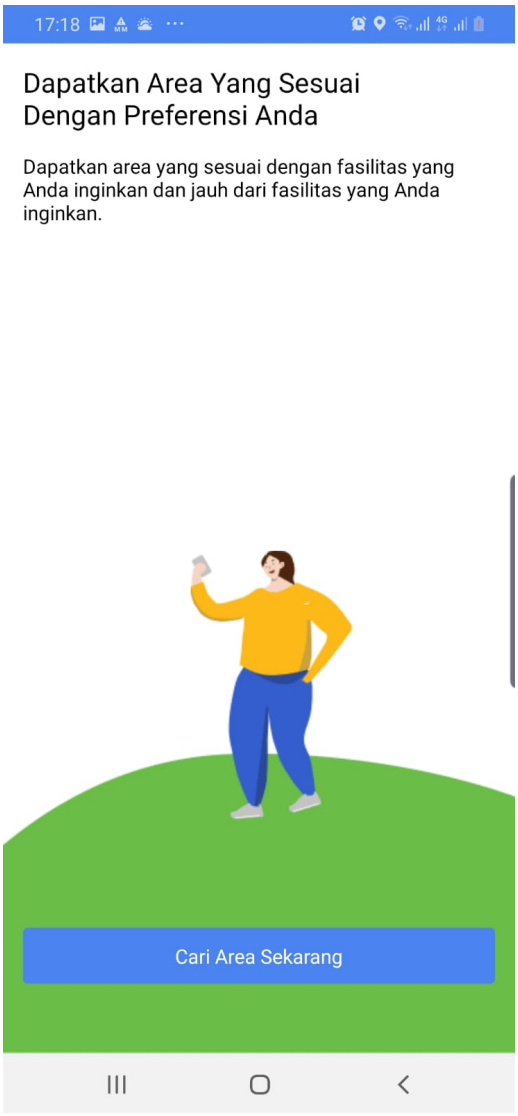

Figure 5. Landing page interface

Users can select various facilities such as hospitals, restaurants, ATMs, bicycle shops, and more. The grid size represents a perfect square integer, such as 64 (8x8) or 144 (12x12). All four fields must be filled out, and at least one desired or undesired facility type must be selected. Once all fields are completed, users can proceed by clicking the "Cari Rekomendasi Area" (Search for Area Recommendation) button. Upon completion of the process, latitude and longitude values are generated for the desired facilities, undesired facilities, and locations that are not recommended. These latitude and longitude values are then visualized using the Google Maps API. Fig. 6 illustrates an interface showcasing a form that necessitates user input.

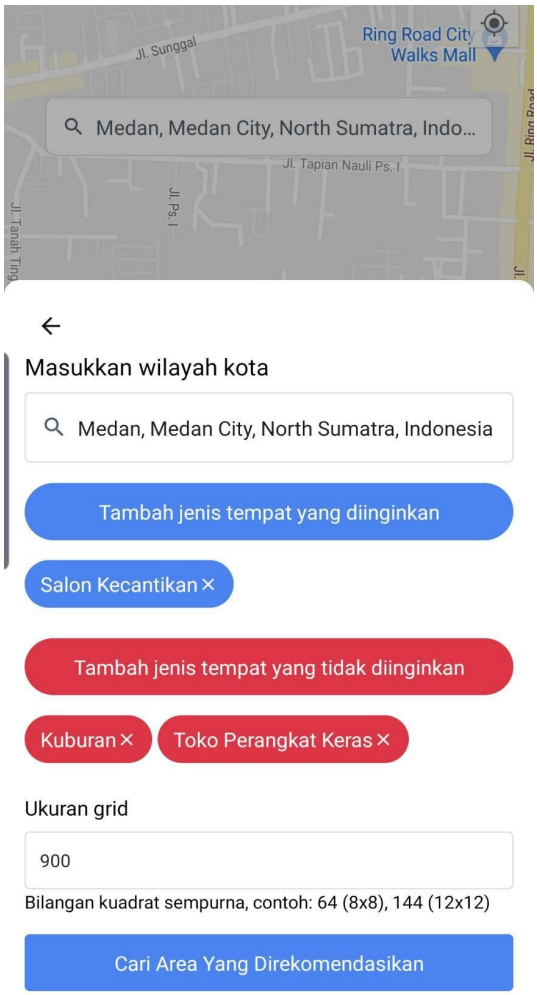

Figure 6. Dashboard page interface

Using the dashboard page interface, the system successfully enables users to input their preferences regarding the targeted area, such as city or village selection, and their desired proximity to various facilities. Users can specify preferred and dispreferred facilities, allowing for a comprehensive customization of their search criteria.

The data collection module employs the Google Maps API to retrieve information for input into the GASKY algorithm, which generates location recommendations. Two types of Google Maps API, namely the Places API and the Geocoding API, are utilized for data collection. The Geocoding API is used to obtain information based on user queries, providing longitude and latitude values. The user's query involves specifying the area name, and as an example, querying "Medan City" would yield the geometry information with latitude: 3.597031 and longitude: 98.678513. This information becomes a parameter for retrieving candidate object information around the selected area.

The Google Maps Places API is utilized to obtain point of interest (POI) information for candidate objects near the queried area, based on the facility type entered by the user. The necessary parameters include the geometric information of the queried area, the facility type, and the maximum radius. The obtained information includes the name of the POI, as well as latitude and longitude coordinates as seen in Table 1.

Table 1. Example results of the data collection module for bookstores in Medan

Place	Latitude	Longitude
Gramedia Book Store	3.5858...	98.6639...
Toko Buku Yakin	3.5946...	98.6665...
Toko Sembilan Wali	3.5857...	98.6604...
Books & Beyond	3.5846...	98.6719...

This online data collection module retrieves data from Google Maps, and its output comprises the candidate objects generated based on the user's regional query. The results are then processed using the GASKY algorithm to provide recommended areas based on the user's input.

The GASKY module is an implementation of the Grid-Based Area Skyline algorithm that processes the data generated from the data collection module. GASKY processes geographical information from the target area, selected POIs (Points of Interest) from the facilities, and the number of grids. The output of the GASKY module is recommended areas, a list of skyline grids or non-dominated grids.

The recommended areas generated by the GASKY algorithm are visualized on Google Maps. The map display includes four types of markers: blue circles, red pinpoints, blue pinpoints, and black boxes. The blue circle represents the user's current location. The red pinpoints indicate the locations of undesired facility types. The blue pinpoints mark the locations where the desired facility type is found. The black boxes represent areas that are not recommended based on user information processed by the GASKY algorithm. These black boxes signify the dominated grids generated by the processed GASKY algorithm. The recommended area is the entire area that is not covered by a black box. The grid that is not recommended is marked with a black grid, because the user does not need to pay attention to that area, considering it does not align with their preferences. The recommended grid is not marked with a black color, making it easier for the user to focus solely on the visible area. Fig. 7 presents the resulting interface, utilizing maps as a visualization tool with the assistance of the Google Maps API, specifically the Places API. The system features a user-friendly interface designed to facilitate easy interaction with users. It provides intuitive options for inputting preferences, displays recommended locations on the map, and presents relevant information about each location in a clear and accessible manner.

The testing method used for the system is black box testing. This method is sufficient to test the input and output of the system without examining the internal program structure. There are four functions of the system that are tested, including opening the dashboard page, filling out the form page, the autocomplete location section, and the recommendation area results page. The results of the black box testing of all four functions return successfully as can be seen in Table 2. After completing the black box testing on this system, it was found that the testing has successfully achieved the set goals, with the system demonstrating good performance and meeting all established functional requirements. All main functions of the system have been thoroughly tested and found to operate as expected. The testing has verified that the system meets the criteria set in the initial specifications.

To provide an overview of GASKY's performance regarding parameter combinations in the system, testing was conducted using two scenarios of parameter combinations. Fig.8 shows the results of the experiment based on scenario 1, visualized in the form of a line graph.

The lines represented by squares (□) depict the visualization results for data from Banda Aceh City, while the lines represented by diamonds (◊) depict the visualization results for Medan City. The y-axis represents the skyline area ratio to the total number of grids generated, while the x-axis represents the number of facility types inputted by the user. Fig. 8a, 8b, 8c, and 8d demonstrate the relationship between the two variables on the x-axis and y-axis for different numbers of grids: 9 (3x3), 36 (6x6), 144 (12x12), and 225 (15x15).

The experiment results indicate that as the number of grids and the area of the target city increase, the resulting skyline area ratio decreases. The decreasing ratio of skyline areas indicates that the system can provide recommendations better than when the ratio of

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 2, July 2024

skyline areas increases. An increasing number of skyline areas signifies a failure of the system to filter regions according to user preferences, marked by the nearly equal number of generated skyline areas to the total number of grids. Therefore, for wider search areas, users are encouraged to increase the number of grids so that the ratio of generated skyline areas becomes smaller.

For the same number of grids, a larger number of facility types inputted tends to result in a higher skyline area ratio. Therefore, users are expected to provide input on desired and undesired facilities in a reasonable quantity. If users input too many preferences, the system will struggle to find regions that match those preference combinations, resulting in nearly all grids becoming skyline areas, causing the system to fail its recommendation of desired regions.

Scenario 2 utilizes processing time as the y-axis and the number of facility types used as user input preferences as the x-axis. The results in Table 3 indicate that as the number of grids increases, the processing time required also increases. Additionally, a larger number of facility types inputted has an impact on the increased processing time. Figures 9a, 9b, 9c, and 9d demonstrate the relationship between the two variables on the x-axis and y-axis for different numbers of grids: 9 (3x3), 36 (6x6), 144 (12x12), and 225 (15x15). Based on the graphs in Figure 9, it is apparent that the search area size also affects GASKY's performance in this recommendation system. The larger the search area, the more time the system requires to display recommendation results on the map.

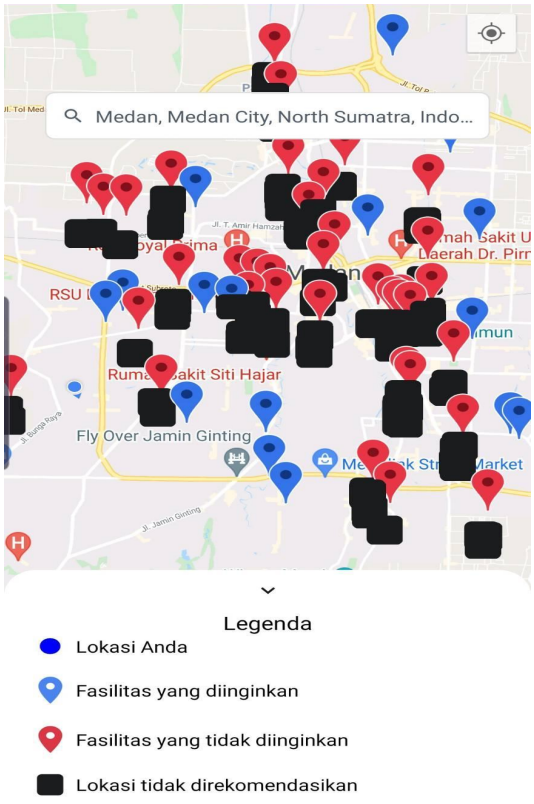

Figure 7. Result page interface

Table 2. Black box testing result

No	Function Name	Testing Conditions	Expected Results	Testing Results
1	Opening the dashboard page	If the user clicks the "search area now" button on the dashboard page	Go to the dashboard page and display the map of the user's current location and the "search area now" section	Success
2	Opening the page for filling out the recommended area search form	If the user clicks the "find area now" button on the dashboard page.	Go to the form page and display the form that needs to be filled out by the user.	Success
3	Filling out the form for recommended area search	If the user has filled out all the fields in the form, and then the user clicks on the option to search for recommended areas.	The system will display the page with the recommended area results.	Success

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 2, July 2024

If the user does not enter a city region specified on the form page, and then the user clicks on the option to search for recommended areas.	The system will remain on the form page and display a warning message saying "Please enter the city region first."	Success
If the user does not fill in at least one desired facility type or undesired facility type, and then the user clicks on the option to search for recommended areas.	The system will remain on the form page and display a warning message saying "Please enter at least one facility type."	Success

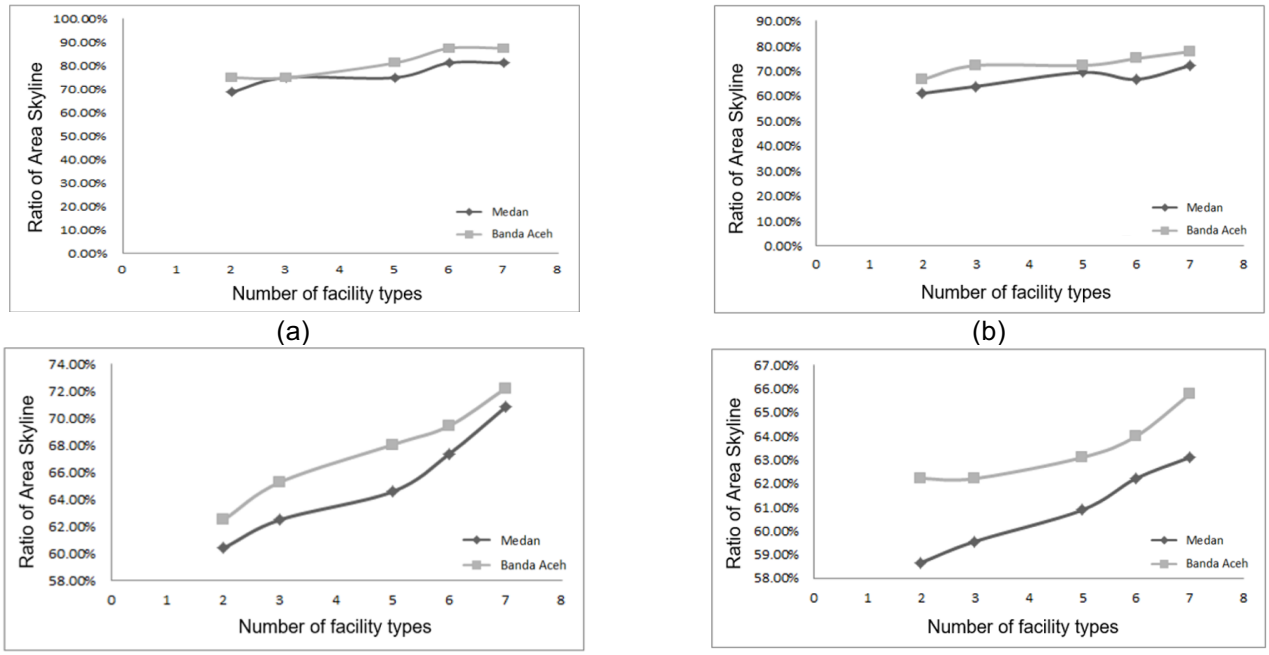

Figure 8. Scenario 1 results for the number of grids 9 (a), 36 (b), 144 (c), and 225 (d).

Table 3. GASKY running time

Number of Grids	Number of facility type	Running Time for Medan (s)	Running Time for Banda Aceh (s)
9	2	7	4
	3	7,7	3,9
	4	7,7	5
	5	8	6,2
	6	7,8	7
	7	10	8

36	2	10,5	6,5
	3	11	8,5
	4	12	10,5
	5	13	12
	6	14	13
	7	16,5	13,5
144	2	25,5	25,3
	3	32	26,7
	4	32,7	27,2
	5	32,2	28,4
	6	33	31,6
	7	36,5	33,3
255	2	38,2	35,4
	3	39,3	38,3
	4	40,7	38,5
	5	41,7	38,9
	6	45	41,3
	7	49	38

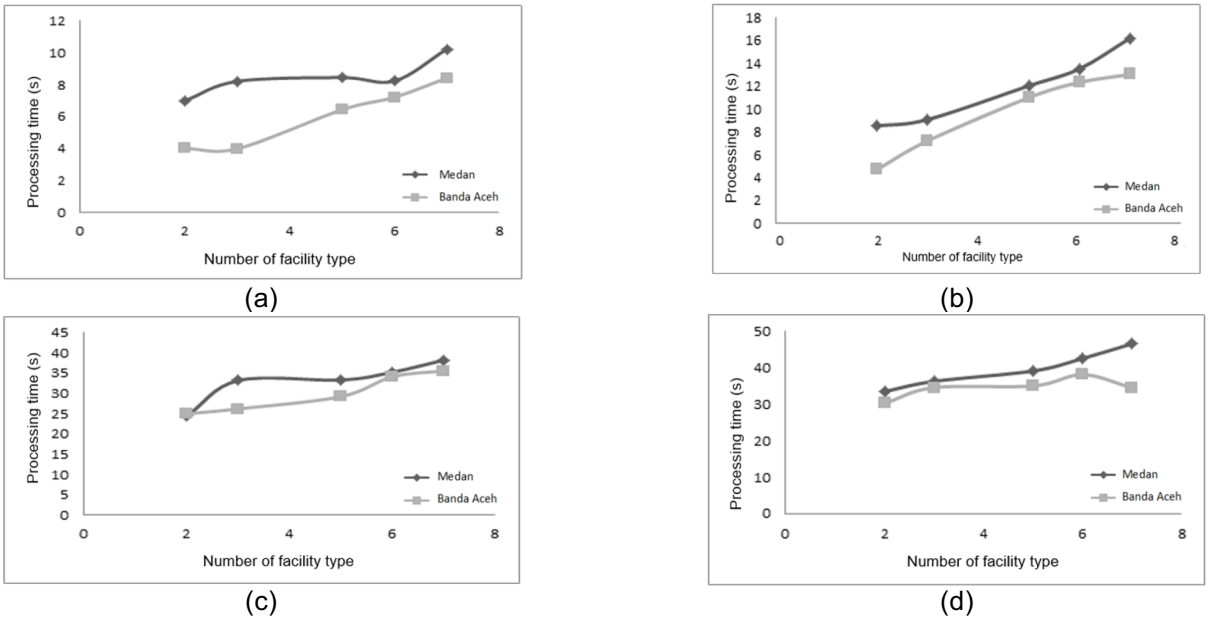

Figure 9. Scenario 2 results for the number of grids 9 (a), 36 (b), 144 (c), and 225 (d).

CONCLUSION

This research successfully developed a mobile-based area recommendation system using GASKY and Google Maps, as well as the Google Maps API, enabling the automatic retrieval of spatial data of various facility types based on user input. Following the completion of the GASKY process, the skyline results visualization is directly integrated into Google Maps within the application. In addition, the application was successfully built using the waterfall method. Following this, black box

scenarios were executed successfully in line with the established requirements. Moreover, testing was conducted to assess the impact of user-entered parameter combinations on the recommendation system's performance. This location recommendation search application is designed for mobile platforms, allowing users to conveniently access it with a reliable internet connection

Additionally, this research utilized an API, streamlining the development of functional applications without manual data input. Experimental findings reveal that processing time varies depending on the size of the target area, the number of grids, and the quantity of facility types provided by the user. As the number of facility types increases for each grid number, the time required for obtaining area recommendations significantly escalates. For further development, we will explore the use of parallel processing to accelerate the response time of the system.

REFERENCES

[1] A. K. Lumbwe, E. Nwobodo-Anyadiegwu, and C. Mbohwa, "The impact of location decision: A review," *Proc. Int. Conf. Ind. Mech. Eng. Oper. Manag.*, pp. 309–316, 2021.

[2] Y. Zhang and J. Zhang, "Catch them all: Impacts of location-based augmented reality mobile applications on local businesses," *Inf. Manag.*, vol. 58, no. 8, 2021, doi: 10.1016/j.im.2021.103550.

[3] S. Arunyanart, P. Sureeyatanapas, K. Ponhan, W. Sessomboon, and T. Niyamosoth, "International location selection for production fragmentation," *Expert Syst. Appl.*, vol. 171, 2021, doi: 10.1016/j.eswa.2021.114564.

[4] A. Arintoko, A. A. Ahmad, D. S. Gunawan, and S. Supadi, "Community-based tourism village development strategies: A case of Borobudur tourism village area, Indonesia," *Geoj. Tour. Geosites*, vol. 29, no. 2, pp. 398–413, 2020, doi: 10.30892/gtg.29202-477.

[5] B. S. Sulastio, H. Anggono, and A. D. Putra, "SISTEM INFORMASI GEOGRAFIS UNTUK MENENTUKAN LOKASI RAWAN MACET DI JAM KERJA PADA KOTA BANDARLAMPUNG PADA BERBASIS ANDROID," *J. Teknol. dan Sist. Inf.*, vol. 2, no. 1, 2021.

[6] S. Dhingra, R. B. Madda, A. H. Gandomi, R. Patan, and M. Daneshmand, "Internet of things mobile-air pollution monitoring system (IoT-Mobair)," *IEEE Internet Things J.*, vol. 6, no. 3, 2019, doi: 10.1109/JIOT.2019.2903821.

[7] T. Trasberg and J. Cheshire, "Spatial and social disparities in the decline of activities during the COVID-19 lockdown in Greater London," *Urban Stud.*, vol. 60, no. 8, pp. 1427–1447, 2023, doi: 10.1177/00420980211040409.

[8] D. Yusuf and F. N. Afandi, "APLIKASI ABSENSI BERBASIS ANDROID MENGGUNAKAN VALIDASI KORDINAT LOKASI DAN NOMOR HANDPONE GUNA MENGHINDARI PENULARAN VIRUS COVID 19," *Expert J. Manaj. Sist. Inf. dan Teknol.*, vol. 10, no. 1, 2020, doi: 10.36448/jmsit.v10i1.1492.

[9] A. Campbell, A. Both, and Q. (Chayn) Sun, "Detecting and mapping traffic signs from Google Street View images using deep learning and GIS," *Comput. Environ. Urban Syst.*, vol. 77, 2019, doi: 10.1016/j.compenvurbsys.2019.101350.

[10] S. Cuzzucoli, "Giving the Right Answer: a Brief Overview on How to Extend Ranking and Skyline Queries," pp. 1–7, 2022, [Online]. Available: http://arxiv.org/abs/2202.06430.

[11] Q. Gong, H. Cao, and P. Nagarkar, "Skyline queries constrained by multi-cost transportation networks," *Proc. - Int. Conf. Data Eng.*, vol. 2019-April, pp. 926–937, 2019, doi: 10.1109/ICDE.2019.00087.

[12] R. Amin, T. Djatna, Annisa, and I. S. Sitanggang, "Recommendation system based on skyline query: Current and future research," 2020, doi: 10.1109/ICOSICA49951.2020.9243225.

[13] A. John, S. K. Singh, M. Adimoolam, and T. A. Kumar, "Dynamic sorting and average skyline method for query processing in spatial-temporal data," *Int. J. Data Sci.*, vol. 6, no. 1, p. 1, 2021, doi: 10.1504/ijds.2021.117460.

[14] M. Sharifzadeh and C. Shahabi, "The spatial skyline queries," in *VLDB '06 Proceedings of the 32nd international conference on Very large databases*, 2006, pp. 751–762.

[15] M. S. Arefin, X. Jinhao, C. Zhiming, and Y. Morimoto, "Skyline query for selecting spatial objects by utilizing surrounding objects," *J. Comput.*, vol. 8, no. 7, pp. 1742–1749, 2013, doi: 10.4304/JCP.8.7.1742-1749.

[16] A. Annisa and S. Khairina, "Location Selection Based on Surrounding Facilities in Google Maps using Sort Filter Skyline Algorithm," *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 7, no. 2, 2021, doi: 10.23917/khif.v7i2.12939.

[17] A. Annisa and L. Angraeni, "Location Selection Query in Google Maps using Voronoi-based Spatial Skyline (VS2) Algorithm," *J. Online Inform.*, vol. 6, no. 1, p. 25, 2021, doi: 10.15575/join.v6i1.667.

[18] Annisa, A. Zaman, and Y. Morimoto, "Area skyline query for selecting good locations in a map," *J. Inf. Process.*, vol. 24, no. 6, pp. 946–955, 2016, doi: 10.2197/ipsjjip.24.946.

[19] C. Li, A. Annisa, A. Zaman, M. Qaosar, S. Ahmed, and Y. Morimoto, "MapReduce algorithm for location recommendation by using area skyline query," *Algorithms*, vol. 11, no. 12, 2018, doi: 10.3390/a11120191.

[20] T. Djatna, F. H. Putra, and Annisa, "An implementation of area skyline query to select facilities location based on user's preferred surrounding facilities," *2020 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2020*, pp. 15–20, Oct. 2020, doi: 10.1109/ICACSIS51025.2020.9263108.