

## **DESIGN OF LOST PILGRIM MONITORING APPLICATION WITH ANDROID LBS**

**Dwi Randi Ramadhani<sup>1</sup>, Bonda Sisephaputra<sup>2</sup>**

<sup>1,2</sup>*State University of Surabaya, Surabaya, Indonesia*

*[dwi.19047@mhs.unesa.ac.id](mailto:dwi.19047@mhs.unesa.ac.id), [bondasisephaputra@unesa.ac.id](mailto:bondasisephaputra@unesa.ac.id)*

### **ABSTRACT**

Muslims who are able, both physically and financially. However, the implementation of this worship often faces challenges, especially for pilgrims who get lost or separated from the group. Based on data, the number of pilgrims reported lost every year reaches thousands of cases, with one of the main contributing factors being overcrowding, language barriers, and complex environments. To overcome this problem, this research aims to design and develop an Android-based *HajjWayFinder* application by utilising the *Location Based Services* (LBS) method using the *React Native framework* and *Firebase Firestore*. The *HajjWayFinder* application is designed to assist mentors in monitoring the movement of pilgrims in real-time, tracking the location of lost pilgrims, and facilitating communication through the Panic Button feature. Using the *Extreme Programming* (XP) method approach, this application is developed with an iterative cycle to ensure functionality and user needs are met. Black-box testing showed that the application was able to run well, both in terms of main features and user interface. The results show that this application can reduce the time and effort required by the mentor to find lost pilgrims. In addition, this application helps improve the solemnity of the Hajj pilgrimage by minimising the stress and panic experienced by pilgrims. The implementation of this technology is expected to make a significant contribution in supporting the implementation of a more organised and effective Hajj.

**Keyword:** *Location Based Services, React Native, Extreme Programming, Next js, Firebase*

#### **Article Info:**

*Article history:*

*Received October 13, 2025*

*Revised January 9, 2026*

*Accepted February 20, 2026*

#### **Corresponding Author**

Dwi Randi Ramadhani

State University of Surabaya, Surabaya, Indonesia

[dwi.19047@mhs.unesa.ac.id](mailto:dwi.19047@mhs.unesa.ac.id)

### **1. INTRODUCTION**

The Hajj pilgrimage is obligatory for everyone who is capable of being physically able and healthy wal'afiat, strong energy and more than enough finances. In carrying out the pilgrimage, the pilgrims must follow all the processes of the pilgrimage regularly in order to become a mambrur pilgrimage by following all kinds of conditions and pillars so that Allah SWT will accept the pilgrimage [1].

The number of incidents of pilgrims who got lost was summarised in a website kumparan.com July 2019, where the pilgrims who got lost or lost reached 1,624 cases. On average, in a few days during the pilgrimage process as many as 25 people were separated from

their entourage. Reporting from Konteks.co.id in July 2023, the Ministry of Hajj and Umrah Report 2023, public service statistics of the Saudi Arabian Youth Association in Mecca revealed that a total of 206,582 pilgrims were reported missing during the 2023 hajj. This is becoming increasingly frequent for Indonesian pilgrims due to the increase in the number of pilgrims from year to year which continues to occur [2].

From interviews that have been conducted by researchers on one of the Hajj travel parties, that in the implementation of the Hajj pilgrimage, the travel party provides supervisors and provides direction to the pilgrims which aims to expedite the Hajj process itself, but there are always several obstacles experienced by the supervisor, namely the frequent separation of pilgrims from the group and the presence of pilgrims who leave without the knowledge of the supervisor. Pilgrims are also often separated from the group because the Holy Land is very crowded, and the average body of Indonesians is relatively smaller than pilgrims from other countries. The difficulty felt by pilgrims is in finding directions or gathering points, due to directions that use English and Arabic, as well as officers on the ground, and officers in the holy land mostly only speak Arabic.

Currently, applications made on the Android operating system are increasing, for example applications that utilise *Location Based Services* (LBS) technology. This service uses *Global Positioning Service* (GPS) technology from Google. By using LBS services, users can find out where they are [3].

One popular approach in mobile application development is using the React Native framework, especially for the Android platform. This framework allows developers to create applications with responsive and dynamic interfaces, while maximising the potential use of Android devices [4]. According to Tashildar et al. argued that the use of the React Native framework in developing the Hajj pilgrim monitoring application provides a number of significant benefits. First of all, cross-platform compatibility allows this application to run on Android and IOS devices, making it easier for users to access the application [5].

This Hajj pilgrim monitoring application can be applied in every Hajj season, it can be ensured that the implementation of the Hajj will run more solemnly and smoothly without being preoccupied with confiscating time just to look for lost pilgrims and the pilgrims concerned do not experience stress and panic because they cannot immediately return to the lodging or other worship groups.

This application will also help the Government and Hajj travel in reducing cases of pilgrims who are sick and exhausted due to getting lost. The positive impact that will be felt on the implementation of the Hajj as a whole is of course the implementation of a solemn and solemn Hajj without being preoccupied with activities that have nothing to do with the Hajj itself.

Based on the results of the description and the results of pre-interviews conducted, the purpose of this study intends to overcome the problem of the number of pilgrims who get lost. With the design of the HajjWayfinder application, it is hoped that it can help the head of the Hajj group in finding lost members of the pilgrimage team. The development of this application will use the XP (*Extreme Programming*) method and implement the LBS (*Location Based Services*) method by being developed using React Native and Firebase Firestore.

## 2. METHODS

The research method section includes the research specifications, type of research, approach method, data collection techniques, and data analysis methods used in the study. It

should be written descriptively in a single paragraph. The description of the course of research should be supported references, so the explanation can be accepted scientifically.

### A. Research Flow

The following is a research flow that will be carried out by researchers to test the performance of each *compiler* through the *rendering* process:

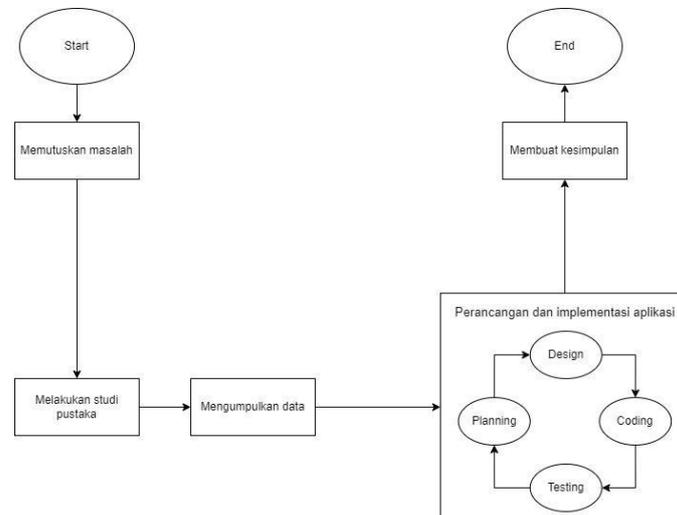


Figure 1. Flowchart of Research Flow

The following is an explanation of the flow above:

1. First, formulate the problem according to what is written in the research background.
2. Second, the research conducted a literature study to find references to the theory to be tested.
3. After that, data collection is carried out to strengthen the author in designing the *HajjWayfinder* application. Data collection from interviews with travel agent administrators.
4. Then is the design and implementation of applications using the XP (*Extreme Programming*) method which consists of several steps, namely *planning*, *design*, *coding*, *testing*.
5. Finally, a summary will be made into a conclusion as the author's final study. Problem Formulation.

Based on the research flow, this study aims to create a *HajjWayfinder application*. This is because there are still many pilgrims who get lost when performing worship. With this *HajjWayfinder* application, it is hoped that it can help the head of the pilgrimage group to find pilgrims who get lost during worship.

### B. Literature Study

Based on several previous studies, as zanam develops, many problems occur both in society and institutions. Because of this, technology is present to help overcome these problems [6]. One of them is the application, with the application will help solve various kinds of problems that occur in society or institutions. An application is a programme or software developed to support human activities. At least, applications have four main functions, namely to facilitate work, as a means of communication, entertainment, and media for updating the latest information.

### C. Data Collection

Data collection is the process of collecting relevant information or facts from various sources or respondents for the purpose of analysis, research, decision making, or other purposes

[7]. On the other hand, data collection is also the first step in the process of further processing information, the data collected can be in the form of numbers, facts, opinions, or other information relevant to a particular purpose [8]. Application difference analysis involves converting code from one programming language to another [9]. Collecting the data required for this analysis is very important to ensure that the application and web development process runs smoothly and the results are accurate. According to Tambunan, there are several types of data that need to be collected in this application analysis, namely:

a. *Source Code*

The main data required is the source code in the original programming language. This is the code that will be converted by the transpiler into the target programming language. The source code must be complete and conform to the standard syntax of the programming language used. The source code is the core of the analysis. The source code must be complete and correct according to the syntax and semantic rules of the programming language used. This code includes all instructions, control structures, variable definitions, and functions used in the program.

b. *Origin and Target Programming Language Specifications*

In-depth knowledge of the origin and target programming languages is essential. It includes syntax rules, semantics, built-in functions, and all the features present in both languages. An in-depth understanding of the home and target programming language specifications is essential. Knowledge of the built-in functions or standard libraries available in both languages is also required.

c. *Input and Output Data*

If the source code contains interactions with inputs and outputs, this data needs to be identified. This helps in ensuring that the data treats *inputs* and *outputs* correctly and maintains appropriate functionality. This includes the type of *input* received, such as input from a user or data from a file, as well as the type of output produced, such as on-screen display or file *output*. The data needs to ensure that the way inputs and outputs are handled in the target programming language is as expected.

D. *Design of the Application to be Tested*

The implementation of the application design is the main stage of developing this HajjWayfinder application because in this stage all the designs that have been made will be implemented. This stage will invest the longest time and must also pay attention to every part of the implementation process so that later the output, namely the application, runs properly and correctly in accordance with the design that has been made.

The application is designed to help pilgrims and mentors during the pilgrimage. The implementation stage is the main focus because it includes the application of the entire design that has been made. This application was built using Extreme Programming methodology, which includes four main stages: planning, designing, coding, and testing. The design process involved creating an interface using Figma and a database with Firestore, while coding used the React Native and Next.js frameworks. Testing was done using a black-box testing approach to measure the functionality of the application.

**Main Features** The application consists of two platforms, namely mobile and website. On the mobile platform, users (pilgrims and mentors) have access to features such as registration, login, real-time location, chat, notifications, and map-based navigation. Features such as Panic Button are designed to help pilgrims when they get lost. On the website platform, the admin can manage the data of pilgrims and mentors, as well as assign important locations such as rally points through the management feature.

System design uses UML diagrams such as Use Case and Activity Diagrams to describe the interaction flow. The database is designed with interconnected tables, including tables for pilgrims, mentors, chat groups, messages, and set points. The system architecture integrates users, databases, and location-based services in real-time to support effective coordination.

System testing was conducted on various devices and browsers to ensure compatibility and performance. Descriptive statistics were used to analyse the success rate of application functionality, while a t-test compared the success rate of mobile and website platforms. The test results showed that key features, such as login, navigation, and notifications, successfully functioned as expected on most devices and browsers.

The User Interface was designed for comfort and ease of use. Some of the main pages include login, homepage, location, chat, prayer times, and hajj-umrah materials. The location page allows users to see the position of other pilgrims in real-time, while the chat page supports communication between pilgrims and mentors.

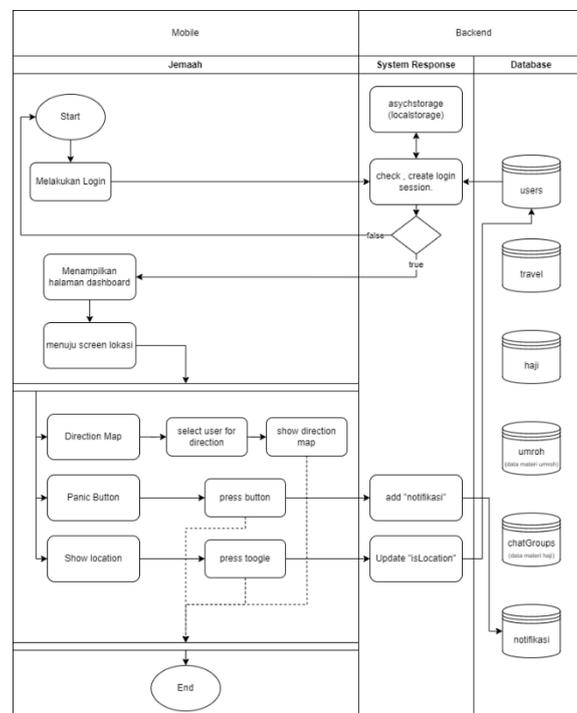


Figure 2. Application Flow Diagram

HajjWayfinder application is designed to facilitate coordination between pilgrims and mentors during Hajj, integrating location-based features in mobile platforms and *websites*. The implementation of this application follows the stages of *Extreme Programming*, from planning, designing, coding, to testing. Interface design is done with Figma, while coding utilises *React Native* and *Next.js*. In the testing phase, the app was tested on various devices and *browsers* to ensure compatibility and functionality, including features such as login, map navigation, emergency notifications, and data management. Key features of the mobile app include the *Direction Map*, *Panic Button*, and *real-time* location feature that allows users to see the position of other members. The mobile app interaction process starts from login to accessing key features such as navigation and emergency notifications. On the *website platform*, the admin has full control to manage pilgrims' data, mentors, and important gathering points, such as hotel or mosque locations.

### 3. RESULTS AND DISCUSSION

Variable comparison in this study is to test the difference between two variables [10], and in this study leads to two data coding results, namely Application and Web, which are as follows:

		Levene's Test for Equality of Variances				
		F	Sig.	t	df	Sig. (2-tailed)
Response Time Database	Equal variances assumed	13,197	0	-2,715	82	8
	Equal variances not assumed			-2,715	41,160	10

Figure 3. Result

The independent sample t test aims to compare the means of two groups that are not paired or not related to each other. The Polled Variance formula is used for the t test for equal variances:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2} \left( \frac{1}{n_1} + \frac{1}{n_2} \right)}}$$

It leads to hypothesis testing in this study, which is as follows:

H0: There is no difference in success rate between mobile and website

H1: There is a difference in the success rate between mobile and website.

#### 3.1 Analysis of Test Results

The development of the HajjWayfinder application uses the *Extreme Programming* (XP) methodology, including the design, coding, and testing stages. At the initial stage, the project structure was set up for the mobile application using *React Native* and the web application with *Next.js*. Folders such as *components*, *routes*, and *utils* supported the functionality and navigation of the application. The UI *slicing* process integrates the interface design into usable code. The implementation phase also includes *database* configuration using *Firebase Firestore*, which contains user data, chat, notifications, and hajj-umrah materials. *The Location Based Service* (LBS) method is applied for *real-time* location and navigation features on the main page of the application.

The *mobile* application is designed with features such as login, *real-time* location, navigation, and *panic button* to help pilgrims who are lost. On the *web* application side, the admin can manage the data of pilgrims, mentors, and *set points* such as the location of the mosque or hotel. The web project structure includes *app*, *database*, and *utils folders*. The main page includes a *landing page*, *dashboard*, and data management page. *Firebase Firestore* is used to store collections such as *ChatGroups* for conversation data and *Users* for user information. In the implementation of LBS, the application utilises Google Maps API to track and navigate the location between pilgrims and mentors in real-time. The following is the difference between the application and the web developed in this research:

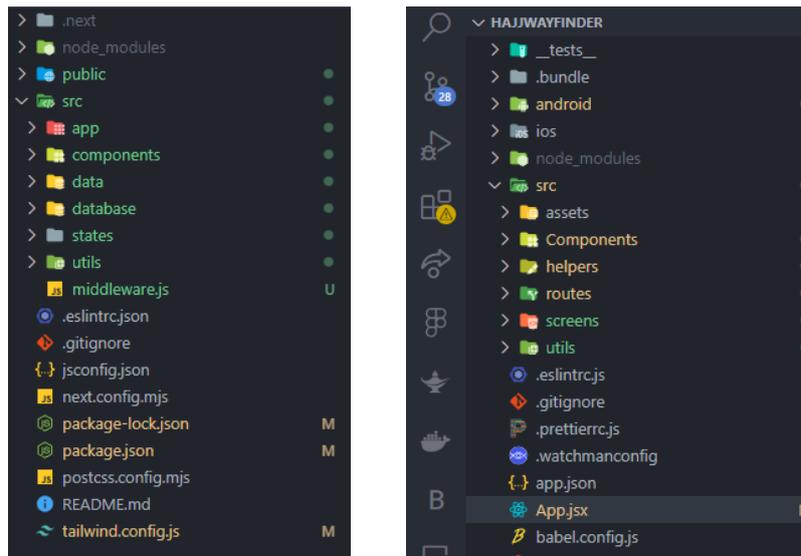


Figure 4. Project Structure

This certainly makes a difference between mobile and website, especially in the application component. Mobile application testing was conducted using the *Black Box Testing* method on five different devices, including Samsung A54 and Oppo A3S. The test results showed a high success rate in the login (96%), registration (96%), and show location (96%) features. Features such as marker monitoring and navigation showed 88%-92% success as some devices experienced connection issues. Descriptive statistical analysis shows that the app fulfils most usage scenarios with an average success rate above 90%. The following is the feature testing form of each design:

Table 1. Main Components of Maps

Component	Fungsi
<MapView/>	Serves to display interactive maps in mobile applications.
<MarkerAnimated/>	Used when you want to add animated interactions to the marker, such as moving with the user's location that changes over time (real-time tracking).
<Marker/>	Used to display an icon or image at a specific position on the map. Users can interact with the marker, such as tapping it to get more information.
<MapViewDirections/>	helps display the route of the two markers.

The web application was tested on five *browsers* (Google Chrome, Microsoft Edge, etc.), focusing on features such as login, registration, and data management. The success rate of the login feature reached 100%, while the registration and *AddUser* features showed an average success rate of 88%-96%. Other features such as *logout* and *Delete Set Point* worked well in most *browsers*. Descriptive statistics show that the web application performance is stable across all platforms, with an average success rate of 92%-100%. Although descriptively there are differences, it is necessary to prove empirically using statistical assistance using SPSS. The following are the test results of these two applications:

Table 2. *Independent Sample T Test*

	Levene's Test for Equality of Variances		t-test for Equality of Means		Sig. (2-tailed)	
	F	Sig.	t	df		
Nilai Keberhasilan	Equal variances assumed	562	456	383	78	702
	Equal variances not assumed			379	58,753	706

The table above shows the results of the Independent T Test on the success aspects of each platform. From the calculation results, it can be concluded that  $H_0$  is accepted because Sig. (2-tailed) is greater than  $\alpha$  ( $0.702 > 0.05$ ) so that  $H_0$  is accepted. This means that there is no difference in the success rate between mobile and website.

The test results show a high average success rate for HajjWayfinder mobile application features, with some important notes regarding the stability of certain functions, for example, features such as *Edit Profile* and *Change Password* have a perfect success rate on all devices. This reflects the stability of the coding logic for basic operations such as user data management and account security. This success is most likely due to the simple implementation design with no dependency on external integration or *real-time* data.

The lowest success rate on the *Marker Monitoring* feature highlights the challenges in real-time location updates on some devices. This issue may be related to several things, for example, on GPS performance, some devices may have less accurate GPS sensors or slow response. On the location update interval, the code that updates every 10 seconds may be taxing on lower-specification devices. As for the internet connection, reliance on the internet for location data synchronisation may cause lag or data loss on less stable networks.

Higher *devices* performed best with a 100% success rate in almost all features. This shows that devices with higher specifications and good hardware stability are able to handle location-based features and *real-time* data without problems. Location-based features such as *Send Notification (Panic Button)*, *Map Direction*, and *Marker Monitoring* rely on the Google Maps API. A short interruption in the API connection or interface mismatch between the API and the app can be the reason for failure on certain devices.

In the *Web Platform*, feature success rates were generally high, but challenges were encountered on certain features, for example on the performance of the ppera GX browser, where Opera GX recorded perfect success on most test scenarios. This shows that modern browsers with good JavaScript support and optimised memory management provide more stable performance. The registration feature recorded the lowest success rate, which is likely due to the complex form validation, where input data such as email, name, or password may face validation issues. Then on to browser compatibility, which leads to differences in *JavaScript* implementation in browsers such as Vivaldi or Mozilla Firefox may affect user experience.

Features such as *Add User* and *Add Set Point*, although recording high success, require multiple tests to ensure the data is stored correctly. This suggests that data management in *Firebase* Firestore may require further optimisation, especially in handling simultaneous operations.

Some of the factors that could explain the issues found are, device and browser differences, where device specifications and browser versions play an important role in application stability. Devices with low specifications tend to face challenges in running *real-time* location-based operations [11]. On the other hand, there is also the external API dependency factor, where features that utilise external APIs, such as the Google Maps API, are vulnerable to connection disruptions or changes on the API service provider side.

Gibransyah et al. explained that memory and processor management also have a direct influence. Location-based features require regular data updates, which can put a strain on devices with smaller processors or RAM [12]. Then there is strict data validation, namely data input validation that is not uniform across browsers can hinder the smoothness of registration features or other data input.

The test results show that the HajjWayfinder application overall fulfils the designed specifications. In the mobile version, *Location Based Services* (LBS)-based features such as location tracking and real-time coordinate transmission proved effective. These features greatly assist the supervisor in monitoring the pilgrims, ensuring better coordination during the pilgrimage. All basic functions, including *login*, navigation, and *panic button*, performed reliably on most devices, although some devices required further optimisation. On the *web* version, core features such as login, registration, and pilgrim or mentor data management work well across different *browsers*. The *logout*, data management, and *set point* addition features also show good compatibility, supporting the admin role in managing pilgrims and mentors efficiently. Minor constraints on the *Registration* feature can be overcome by strengthening validation or optimising the interface.

## CONCLUSION

Based on the research that has been done in the results and discussion, the conclusions obtained are a) The HajjWayFinder application was successfully designed and implemented properly. The mobile version has been designed based on Android using React Native, with key features such as login, registration, location tracking, and user profile management functioning properly. b) Location Based Services (LBS) method was successfully implemented in the application, supporting real-time location tracking feature and location delivery via "Panic Button". The application of LBS allows advisors to monitor the movement of pilgrims effectively. c) Based on the results of black-box testing and independent sample t-test analysis, the HajjWayFinder application, both mobile and web versions, shows stable performance and good compatibility. Thus, this application fulfils the research objectives to provide convenience for mentors and pilgrims in monitoring and managing activities during the Hajj pilgrimage, thereby reducing the risk of losing direction and increasing user comfort.

## REFERENCES

- [1] E. Sinduningrum, M. Permatasari, and A. R. Dzikrillah, "Rancang Bangun Aplikasi Panduan Peduli Jemaah Haji (APPJI) Berbasis Android," *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. 6, no. 1, pp. 120–127, 2021.
- [2] N. H. M. Rohman and W. Fadly, "Melek Bahasa Arab Sehari-Hari Calon Jamaah Haji dan Umroh Desa Blembem," in *PISCES: Proceeding of Integrative Science Education Seminar*, 2021, pp. 279–290.
- [3] E. Fachria, "Location-based Service (LBS) Application Design as Jeneponto District Tourist Information Media," *KnE Eng.*, pp. 252–259, 2024.
- [4] H. Järvinen, "Decentralized react native android application," 2019.
- [5] A. Tashildar, N. Shah, R. Gala, T. Giri, and P. Chavhan, "Application development using flutter," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 2, no. 8, pp. 1262–1266, 2020.
- [6] M. S. Rejeki and A. Tarmuji, "Membangun aplikasi autogenerate script ke Flowchart untuk mendukung business process Reengineering," *J. Sarj. Tek. Inform. e-ISSN*, vol. 2338, p. 5197, 2013.
- [7] E. Surahman, A. Satrio, and H. Sofyan, "Kajian Teori Dalam Penelitian," *JKTP J. Kaji. Teknol. Pendidik.*, vol. 3, no. 1, pp. 49–58, 2020, doi: 10.17977/um038v3i12019p049.

- [8] M. R. Fadli, “Memahami desain metode penelitian kualitatif,” *Humanika*, vol. 21, no. 1, pp. 33–54, 2021, doi: 10.21831/hum.v21i1.38075.
- [9] D. D. Tambunan, “Perbandingan Analisis Aplikasi Database NoSQL Redis dan SQL MySQL,” 2016, *Universitas Widyatama*.
- [10] Sugiyono, “Bab Iii Skripsi - Theresia Elvira Saputri\_4,” no. 2014, pp. 35–45, 2019.
- [11] A. D. Abdiati, S. Setiawan, and H. Supendar, “Pemilihan web browser pada mobile menggunakan metode analytical hierachy process,” *J. Infortech*, vol. 3, no. 1, pp. 26–32, 2021.
- [12] G. Gibransyah, L. Nurlani, and F. Arisandi, “Analisis Perbandingan Server Layanan Penyimpanan Data Pribadi Antara Nextcloud Dan Owncloud Menggunakan Linux Centos 7,” in *SEMNASTERA (Seminar Nasional Teknologi dan Riset Terapan)*, 2021, pp. 168–182.
- [13] Surahman, F., al Ikhsan, S. H., Satrya, F., & Kusumah, F. (2018). Seminar Nasional Teknologi Informasi Universitas Ibn Khaldun.
- [14] Sallaby, A. F., & Kanedi, I. (n.d.). Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework CodeIgniter. *Jurnal Media Infotama*.
- [15] Gunawan, R., Aulia, S., Supeno, H., Wijanarko, A., Uwiringiyimana, J. P., Mahayana, D., & Teknik, S. (2020). Adiksi Media Sosial Dan Gadget Bagi Pengguna Internet Di Indonesia. *Jurnal Techno-Socio Ekonomika*, 14(1).