Jurnal E-KOMTEK (Elektro-Komputer-Teknik)
Vol. 5, No. 2 (2021) pp. 203-216
https://jurnal.politeknik-kebumen.ac.id/index.php/E-KOMTEK
p-ISSN : 2580-3719 e-ISSN : 2622-3066

Design and Implementation of a Node.js Based Computer Network Monitoring Information System at the Department of Communication and Information of Kebumen Regency

Imam Taufiqurrokhman, **Candra Mecca Sufyana**
Information System, Politeknik Piksi Ganesha Bandung, Indonesia, 40274

lukman.bachtiar@gmail.com

https://doi.org/10.37339/e-komtek.v5i2.768

Published by Politeknik Dharma Patria Kebumen

Artikel Info
Submitted:
15-11-2021
Revised:
27-11-2021
Accepted:
27-11-2021
Online first :
30-12-2021

Abstract

Large and complex computer networks have a high risk of failure, so they require monitoring and mapping to maintain service availability. An information system must monitor and map computer networks so that supervision is not done manually and problem handlings can be done quicker. This study aims to design and implement a computer network monitoring-information system at the Department of Communication and Informatics of Kebumen Regency by utilizing the waterfall software development method. UML modelling language used in the design of computer network monitoring information system and implemented with the JavaScript programming language on Node.js runtime environment. This system utilizes the interval function in JavaScript and the system utility from Node.js to check the network devices. This system sends notifications of the network client status changes to registered users. Based on the test results, this system can monitor and provide notifications as well as a map of the computer network belonging to the Kebumen Regional Government, which is managed by the Department of Communication and Information Technology of Kebumen so that the process of handling computer network problems belonging to the Kebumen Regional Government can be carried out more optimally.

Keywords: Information system, Computer network, Node.js

Abstrak

Jaringan komputer yang besar dan kompleks memiliki risiko kegagalan yang tinggi, sehingga memerlukan pemantauan dan pemetaan untuk menjaga ketersediaan layanan. Suatu sistem informasi harus memantau dan memetakan jaringan komputer agar pengawasan tidak dilakukan secara manual dan penanganan masalah dapat dilakukan lebih cepat. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem informasi monitoring jaringan komputer pada Dinas Komunikasi dan Informatika Kabupaten Kebumen dengan memanfaatkan metode pengembangan perangkat lunak waterfall. Bahasa pemodelan UML digunakan dalam perancangan sistem informasi pemantauan jaringan komputer dan diimplementasikan dengan bahasa pemrograman JavaScript pada lingkungan runtime Node.js. Sistem ini menggunakan fungsi interval dalam JavaScript dan utilitas sistem dari Node.js untuk memeriksa perangkat jaringan. Sistem ini mengirimkan pemberitahuan perubahan status klien jaringan ke pengguna terdaftar. Berdasarkan hasil pengujian, sistem ini dapat memantau dan memberikan notifikasi serta peta jaringan komputer milik Pemerintah Daerah Kebumen yang dikelola oleh Dinas Komunikasi dan Informatika Kabupaten Kebumen sehingga proses penanganan jaringan komputer permasalahan yang dimiliki Pemerintah Daerah Kebumen dapat dilaksanakan dengan lebih optimal.

***Kata-kata kunci**: Sistem informasi, Jaringan computer, Node.js*

1. Introduction

The more clients on a computer network, the more complex the network administrator faces problems. Supervision is required to keep internet or intranet access available for clients. The larger and more complex a computer network, the greater the risk of failure in a network. High availability in a computer network is needed to maintain the quality of services provided to clients. The Department of Communication and Information of Kebumen, as the backbone of the SPBE implementation and technology facilitator in the Smart City program in Kebumen Regency, acts as the provider of computer networks and internet access in Government Offices, District Hospital, Public Health Center, public spaces and village offices in Kebumen Regency.

An existing study entitled "Implementation of a Network Monitoring System Using a Mikrotik Router OS at the Islamic University of Batik Surakarta" aims to monitor the entire network system continuously. It has succeeded in creating a network monitoring system. **[1]** However, this system does not have a location feature and only sends notifications in IP Addresses and network devices conditions. This system is not suitable for application in Kebumen Regency, a large territory with a complex network with locations spread across Kebumen Regency.

Based on the problem mentioned above, "Design and Implementation of a Node.js based Computer Network Monitoring Information System at the Department of Communication and Information of Kebumen Regency" was chosen as the rubric of this research. This system will have geolocation features and network maps to monitor the condition and location of computer network devices owned by the Kebumen Regional Government. With this system, it is expected that the Department of Communication and Informatics of Kebumen Regency will be able to provide more optimal internet network services so that the SPBE and Smart City programs in Kebumen Regency can do more optimally. Reports from clients are no longer required because network problems can be handled faster.

2. Method

a. Material

1) Information System

An information system combines information technology and people who use that technology to support operations and management **[2].**

2) Computer Network

A computer network is a set of computers connected simultaneously to each other to share or share resources. One resource currently widely used in a computer network is internet resources **[3]**.

3) JavaScript

JavaScript is a language used to create programs to make HTML documents displayed in a browser more interactive, not just beautiful. JavaScript provides some functionality into a web page to be a program served using a web interface **[4]**.

4) Node.js

Node is a platform that runs primarily, though not exclusively, high-performance server-side code and can easily manage many requests. Node is based on today's most popular programming language: JavaScript. Node is easy to understand yet very powerful in the hands of a developer. It is because the node is an asynchronous and event-driven programming language. In node, almost all commands are executed non-blocking, meaning the code will not stop or wait for another processor command execution. Many types of I/O, generally blocking, are executed asynchronously by nodes, such as network commands, file systems, and database commands **[5]**.

In carrying out its duties, Node.js uses v8, the JavaScript engine produced by Google. V8 itself is in charge of converting JavaScript code into bytecode. This bytecode file will be executed by Node.js **[6].**

5) MariaDB

MariaDB Server is one of the most popular open-source relational databases. It's made by the original developers of MySQL and guaranteed to stay open source. It is part of most cloud offerings and the default in most Linux distributions.

It is built upon the values of performance, stability, and openness, and MariaDB Foundation ensures contributions will be accepted on technical merit. Recent new functionality includes advanced clustering with Galera Cluster 4, compatibility features with Oracle Database and Temporal Data Tables, allowing one to query the data as it stood at any point in the past **[7].**

6) UML

UML (Unified Modeling Language) is one of the language standards widely used in the industrial world to define requirements, analyze, design, and describe architecture in object-oriented programming **[8]**.

b. Method

The waterfall method is a model that provides a sequential or sequential software flowing approach starting from analysis, design, coding, and testing **[9]**.

The stages in software development in the waterfall model are as follows:

1) Analysis

Contains analysis of the requirements, features the software made, potential users of the software, and the limitations or privileges of each user.

2) Design

Translating software development requirements from the analysis results into a process design form so that software developers can implement the design results into program code.

3) Coding

Translating the process design in program code using a programming language.

4) Testing

Tests are carried out to check the acceptability of the coding results with the design and analysis.

3. Result and Discussion

a. Analysis

UML is used in the analysis phase as a modelling language. Use case diagrams to illustrate the features needed by the users and the system limitations. Use case illustrates the required features or will be made on an application. Actor on use case diagrams illustrates potential software users and their limitations. Both actors (admins and operators) can receive email notifications and view network distribution maps, but only admins can manage data on the system. **Figure 1** shows a use case diagram containing the features of the software being developed.

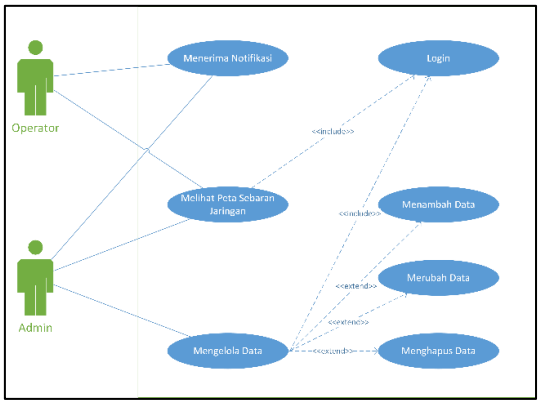

Figure 1. Use Case Diagram

b. Design

In the design phase, the author uses activity diagrams to illustrate each use case that has been planned in the analysis phase and class diagrams to illustrate the database design.

1) Login

The login process starts with the user accessing the system, and the system will display the login form if the user has not been authenticated. The user must provide credentials in a registered email and password and press the login button. The system will validate the credentials inputted by the user. The user will be redirected back to the login page if the provided credentials do not match or validation fails. The user will be redirected to the main system page if the login process is successful. The activity entry diagram is shown in **Figure 2.**

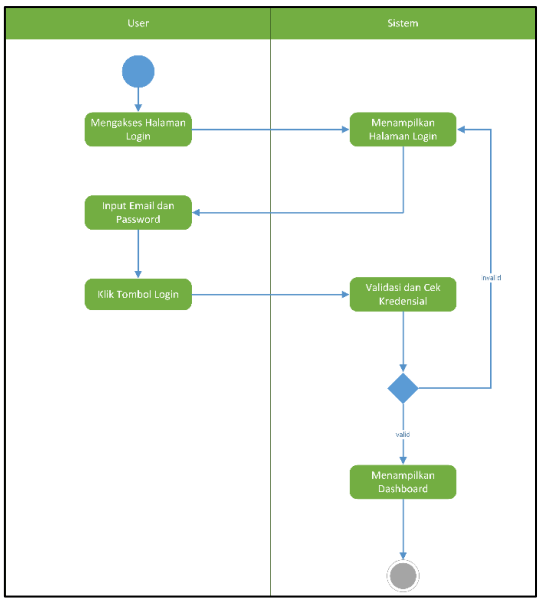

Figure 2. Activity Diagram Login

2) Accessing the Distribution Map

The process of accessing the network distribution map starts with the user accessing the system. The system will check user authentication. The user will be redirected to the main page if it is authenticated. The user will be redirected to the login page if authentication fails. The activity diagram for accessing the network map is shown in **Figure 3.**

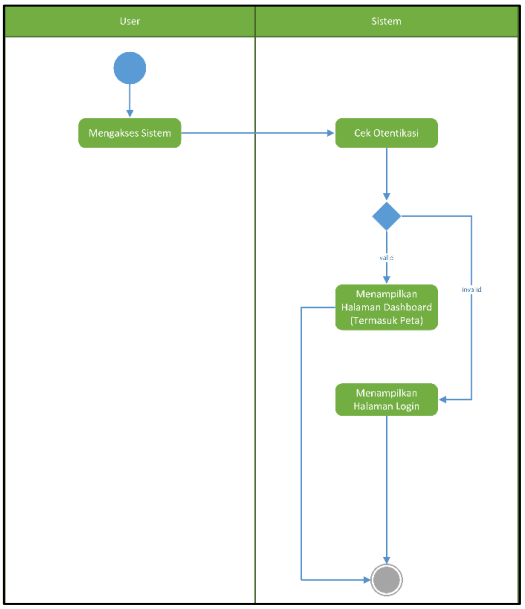

Figure 3. Activity Diagram Accessing the Distribution Map

3) Add Data

In the add data activity, the user will access a page that contains the add data form. The user will provide the data and presses the save button. The system will store the data in the database if the data is valid. Otherwise, the user will be redirected back to the add data page. The activity diagram for adding data is shown in **Figure 4.**

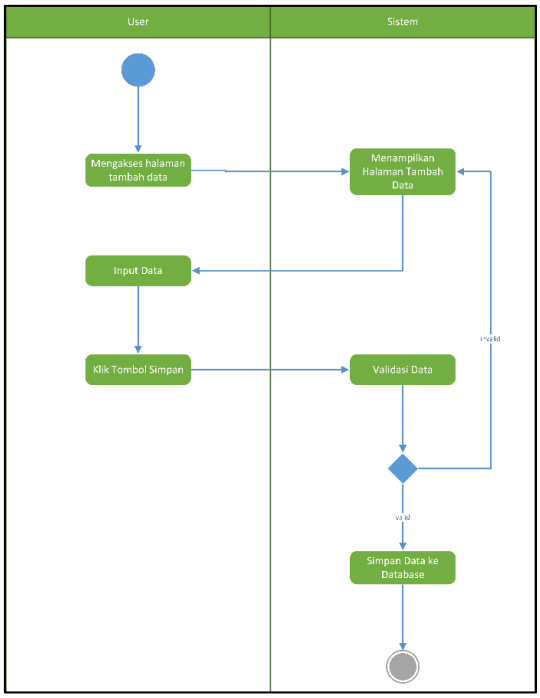

Figure 4. Activity Diagram Add Data

4) Edit Data

The user will access the edit data page in the editing data activity. Then the data edit page will be displayed by the system. Then the user inputs the data to be updated into the form provided by the system and presses the save button. The system then validates the data inputted by the user. If the provided data is valid according to the requested format, the data will be stored in the database. The user will be redirected back to the add data page if the data is not valid according to the specified format. The data editing activity diagram is shown in **Figure 5.**

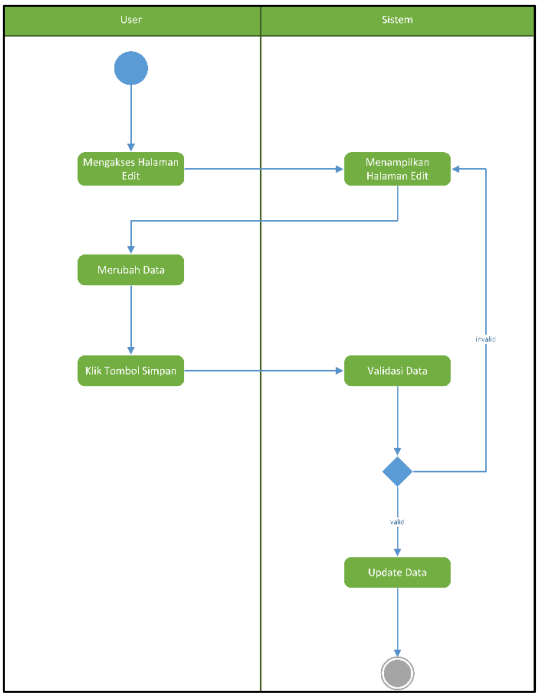

Figure 5. Activity Diagram Edit Data

5) Delete Data

In this activity, the user will access the data list page and press the delete button on the data table. The system will display a confirmation popup, and the user can choose to continue or cancel the delete process. The activity diagram for deleting data is shown in **Figure 6.**

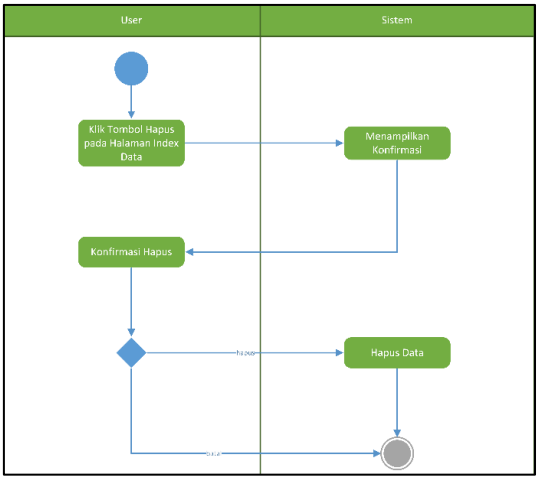

Figure 6. Activity Delete Data

6) Receiving Notification

The last use case that is translated into process design through the activity diagram is receiving a notification. The activity starts from the system side that utilizes the interval function in the JavaScript programming language that runs on

the node.js webserver to run the function continuously at certain intervals while the system is running. The system will retrieve and loop over the network device data and run a function that calls the system utility in a ping program that exists on each operating system. This function will check the connection to each network device in the device data. If there is a status change, the system will send a notification in the form of an email to registered users, both admins and operators, about changes in the device's status from online to offline and vice versa and save the status changes.

The process of sending an email runs parallel with saving the device connectivity record data. If there is no change in the status, the system will not send notifications and only save the recorded connectivity check results into the database. The activity diagram of receiving notifications is shown in **Figure 7.**

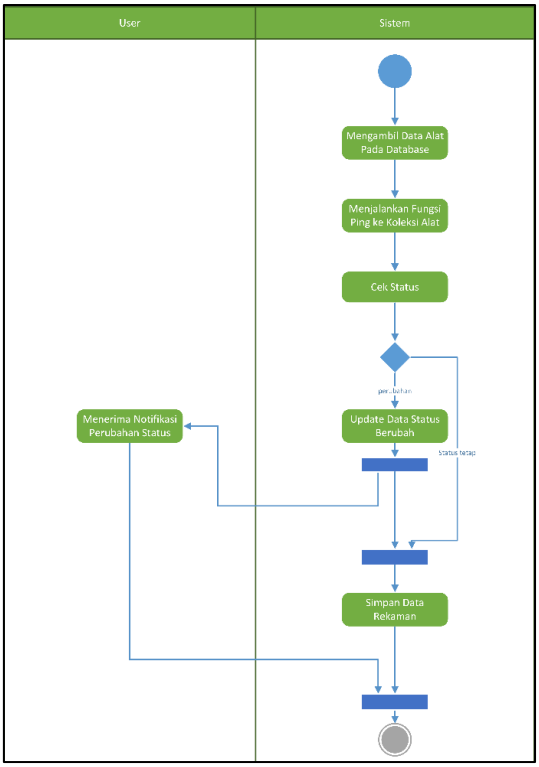

Figure 7. Receiving Notifications Activity Diagram

7) Class Diagram

The class diagram contains attributes and methods to be applied to the system to be created. This design shows the relationship between tables represented by a class with an attribute (field). In addition to displaying attributes, class diagrams also display methods or treatments applied to each object class. The class diagram is presented in **Figure 8.**

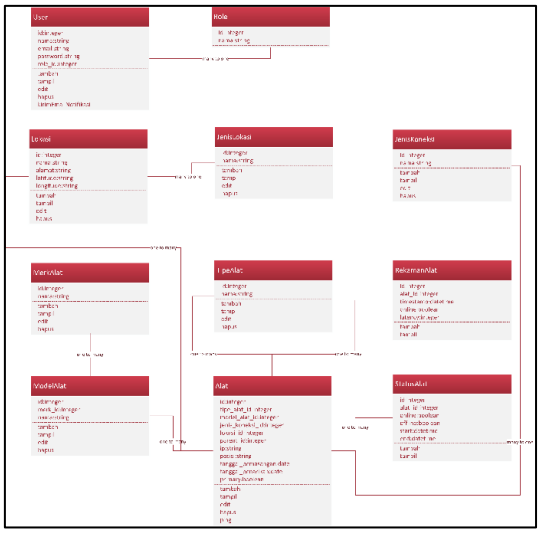

Figure 8. Class Diagram

c. Coding

The author's first thing in the coding stage was creating a database and table based on the class diagram created at the design stage. After creating databases and tables, this coding stage is continued with software development. The author uses node.js as a runtime environment on the system being built and JavaScript as a programming language to build the system.

1) Login Feature

For entering the system, the email and password registered in the system must be provided at this stage and pressing the login button. The login form is shown in **Figure 9.**

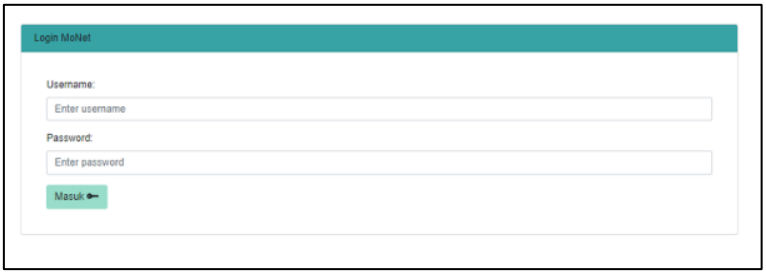

Figure 9. Login Page

2) Add Data Feature

The add data page is shown in **Figure 10.** The add data form must be filled in according to the correct format by the user, and it must press the save button to save the data.

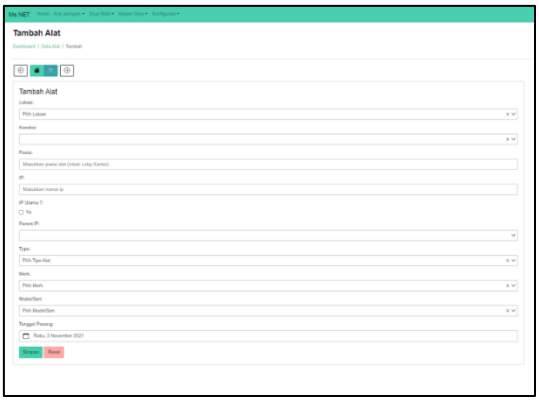

Figure 10. Add Data Page

3) Edit Data Feature

Figure 11 shows the view edit data. The user can change the data that has been saved and press the save button to update the data on this page.

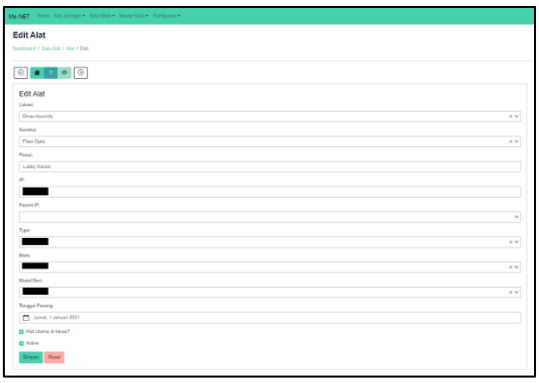

Figure 11. Edit Data Page

4) Delete Data Feature

Figure 12 displays a delete confirmation popup. The user can press the confirmation button to delete and cancel the delete action and press the "No" button.

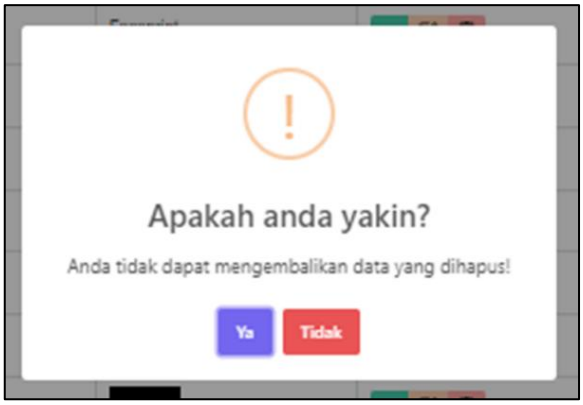

Figure 12. Delete Confirmation

5) Network Distribution Map Features

Figure 13 shows the dashboard page view. This page contains information on the number of network devices with online and offline conditions, the cumulative number of network devices downtime today in minutes and the average network latency in milliseconds.

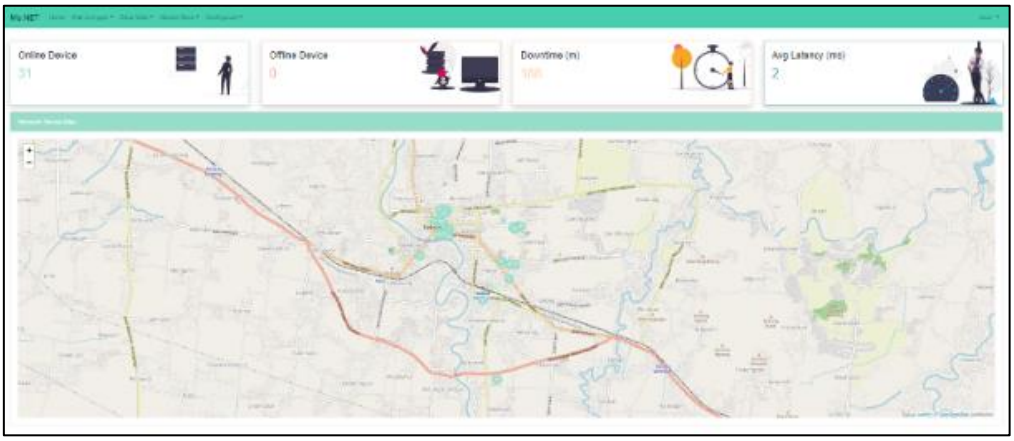

Figure 13. Dashboard Page

d. Testing

Researchers in this testing phase apply the black-box testing method. Testing black-box is a test that aims to see the acceptance of the program tasks without knowing the code used in the program **[10]**.

Table 1 shows the results of the login test with 3 test cases, and all of them produce the expected results.

Table 1. Login Test

Test Case	Expected Result	Status
Accessing Login Page	Show Login Page	Accepted
Login with invalid credentials	Login Failed	Accepted
Login with appropriate credentials	Login Success	Accepted

Table 2 shows the results of viewing the distribution map test with 2 test cases, and all of them produce the expected results.

Table 2. Viewing the Distribution Map Test

Test Case	Expected Result	Status
Accessing the Dashboard Page	Show Dashboard Page	Accepted
Accessing Dashboard Pages without authentication	Show Login Page	Accepted

Table 3 shows the results of the adding data test with 3 test cases, and all of them produce the expected results.

Table 3. Add Data Test

Test Case	Expected Result	Status
Accessing the Add Data Page	Show Add Data Page	Accepted
Adding data with an inappropriate format.	The data failed to be stored, the add data page appeared, and the notification failed to save and a list of input errors.	Accepted
Add data with the appropriate format	Data stored successfully	Accepted

Table 4 shows the results of the edit data test with 3 test cases, and all of them produce the expected results.

Table 4. Edit Data Test

Test Case	Expected Result	Status
Accessing the Edit Data Page	Show Edit Data Page	Accepted
Modify data with an inappropriate format	The data failed to save, the data edit page appeared, and the notification failed to save and a list of input errors.	Accepted
Modify data with the appropriate format	Data updated successfully	Accepted

Table 5 shows the results of the delete data test with 2 test cases, and all of them produce the expected results.

Table 5. Delete Data Test

Test Case	Expected Result	Status
Delete data, then cancel the process	Data is not deleted	Accepted
Delete data and confirm	Data deleted	Accepted

Table 6 shows the results of the test result of the receiving email notifications use case with 2 test cases, and all of them produce the expected results.

Table 6. Receiving Notification Test

Test Case	Expected Result	Status
Receive notifications when a device changes status from online to offline	Notifications in the form of emails received by registered users	Accepted
Receive notifications when a device changes status from offline to online	Notifications in the form of emails received by registered users	Accepted

4. Conclusion

The Network Monitoring Information System has been successfully built and used to map network devices at the Department of Communications and Informatics of Kebumen Regency. This system is built using the waterfall method and implemented with the JavaScript programming language in the Node.js runtime environment with MariaDB as the DBMS. The test results using the Black-Box method show that the system can be used to map computer networks and send notifications in the form of emails regarding the changes in connection status of the network devices that are connected.

References

[1] R. Rinaldo, "Implementasi Sistem Monitoring Jaringan menggunakan Mikrotik Router OS di Universitas Islam Batik Surakarta," *Emitor: Jurnal Teknik Elektro,* vol. 16, no. 2, pp. 16-22, 2016.

[2] P. A. E. Putra, I. P. Satwika and B. P. W. Nirmala, "PERANCANGAN DAN IMPLEMENTASI APLIKASI SISTEM INFORMASI PARIWISATA BERBASIS MOBILE (TRIPTO)," *Jurnal Teknologi Informasi dan Komputer,* vol. 7, no. 4, pp. 352-363, 2021.

[3] A. M. Lukman and Y. Bachtiar, "Analisis Sistem Pengelolaan, Pemeliharaan dan Keamanan Jaringan Internet Pada IT Telkom Purwokerto," *Jurnal Evolusi,* vol. 6, no. 2, pp. 49-56, 2018.

[4] A. Yani and B. Saputra, "RANCANG BANGUN SISTEM INFORMASI EVALUASI SISWA DAN KEHADIRAN GURU BERBASIS WEB (Studi Kasus di SMK Nusa Putra Kota Tangerang)," *PETIR,* vol. 11, no. 2, pp. 107-124, 2018.

[5] F. Zammetti, Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, and Docker, Pottstown: Apress, 2020.

[6] B. Raharjo, PEMROGRAMAN WEB DENGAN NODE.JS DAN JAVASCRIPT, Bandung: Informatika, 2019.

[7] M. Foundation, "MariaDB Server: The open source relational database," MariaDB, 2021. [Online]. Available: https://mariadb.org/. [Accessed 18 11 2021].

[8] D. Wira, T. Putra and R. Andriani, "Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD," *TEKNOIF,* vol. 7, no. 1, pp. 32-39, 2019.

[9] R. A. Sukamto and M. Shalahuddin, Rekayasa Perangkat Lunak, Bandung: Informatika, 2018.

[10] B. A. Priyangga, D. B. Aji, M. Syahroni, N. T. S. Aji and A. Saifudin, "Pengujian Black Box pada Aplikasi Perpustakaan Menggunakan Teknik Equivalence Partitions," *Jurnal Teknologi Sistem Informasi dan Aplikasi,* vol. 3, no. 3, 2020.