The Development and Performance Evaluation of Cloud Web-Based Parking Billing System

Amin Suharjono[a], Kurnianingsih[b], Bambang Supriyo[c] and Roni Apriantoro[d]
Department of Electrical Engineering, Politeknik Negeri Semarang, Prof. Sudarto Street, Semarang, Indonesia

Keywords: Cloud, Parking Payment System, Cashless, Billing, Website.

Abstract: Along with the evolution of technology, parking billing systems are evolving and changing untransparent traditional system in reporting revenues to the government. Therefore, it's necessary to have a parking payment system that transparent and supports cashless which integrated to cloud service of parking income data from several parking locations. In this research, authors created a web-based parking billing user interface system integrated with Midtrans payment gateway to support cash-less system and integrated cloud service monitoring system for parking revenue in several locations. Delay testing in sending data from parking entry access terminal to database resulted in an average of 58.55 s. Test for web page load time at a bandwidth of 30.34 Mbps is the longest, 3.19 s on dashboard page of incoming parking data, while the fastest is 0.755 s on Ticket ID panel info object on search page. The longest website page load time test with a bandwidth of 27.91 Mbps is 4.80 s on dashboard page of incoming parking data, while the fastest is 0.814 s on successful payment page. At a bandwidth of 0.71 Mbps, test for the longest load time for web pages is 13.85 s on dashboard page for incoming parking data, while the fastest is 2.41 s for ticket ID panel info object on search page.

1 INTRODUCTION

Initially, money used as a medium of exchange, was distinguished into certain entities, such as coins and banknotes. However, along with the development of financial technology (fintech), this has given rise to new innovations in the implementation of electronic payment transactions, in order to maximize the use of cashless payment instruments, so that later a cashless society will be created. Developments and innovations in the banking system have led to the use of money as a commodity that is not concretely shaped (intangible money). This is related to the development of information and communication technology that can improve the efficiency of the payment system and reduce the time and costs required to make transactions using checks / cheque (Usman, 2017).

Electronic money payment system (e-payment) is a cashless payment instrument method stored in a server-based (e-wallet) or chip-based (e-money), with the aim of making it easier for users, when making payment transactions or transfers. E-money is a digital payment instrument that uses electronic media with chip entities. Generally, the e-Money entity is a card (debit/credit). Currently, there are several e-Money chips in the Indonesian market, namely Flazz BCA, E-Money Mandiri, Brizzi BRI and others. Meanwhile, an e-Wallet or electronic wallet is a digital payment instrument that uses server-based electronic media. In general, an e-Wallet entity is an application based on a server and in the process of using it requires a connection first with the provider. Currently, there are several e-Wallets in the Indonesian market, namely Telkomsel T-Cash, XL Tunai, CIMB Niaga Mobile Account, BBM Money Permata Bank, DOKU, and so on (Mulyana & Wijaya, 2018).

Parking is the immobile state of a vehicle that is temporary. Inside the vehicle parking, each user will be exposed to a fee/tax. The parking tax itself is in 8th place in providing income to regional taxes in

[a] https://orcid.org/0000-0003-0351-5686
[b] https://orcid.org/0000-0001-7339-7449
[c] https://orcid.org/0000-0002-5699-7829
[d] https://orcid.org/0000-0003-0471-9656

Suharjono, A., Kurnianingsih, ., Supriyo, B. and Apriantoro, R.
The Development and Performance Evaluation of Cloud Web-Based Parking Billing System.
DOI: 10.5220/0011766800003575
In *Proceedings of the 5th International Conference on Applied Science and Technology on Engineering Science (iCAST-ES 2022)*, pages 324-330
ISBN: 978-989-758-619-4; ISSN: 2975-8246

Indonesia. This is a special concern for local governments in Central Java to increase local revenue (Priyanti et al., 2019). Based on Semarang City Regional Regulation Number 11 of 2011 concerning Parking Tax, the tax rate is set at 25% of the basis for taxation, as referred to in article 5. The accuracy and suitability of the parking revenue report in each area will be very necessary. Non-transparent management of parking revenue can lead to leakage of parking tax revenue so that the target of parking tax revenue is not achieved and local income does not increase. The leakage of parking taxes is caused by taxation policies, low awareness of taxpayers, weak of supervision and low of honesty. (Hani & Djasuli, 2015). Thus, the accuracy and suitability of parking revenue reports in each area will be very necessary. Many potential parking areas have not been managed effectively and have not been maximized in providing revenue to the city government due to non-transparent management.

To minimize these leaks, a website-based parking billing system was created utilizing Cloud infrastructure integrated with APIs, where the system monitors the suitability of parking revenue reports in each parking area. The website in this system is used as a service for non-cash-based parking billing payments with e-wallets, in order to make it easier for users. In non-cash payments, there is a QR-Code that contains information on the destination data and the type of payment medium. In Indonesia, the QR-Code is made in a standard known as QRIS. To be able to connect with various types of cashless payment methods, a third-party application is needed. In this case, the author uses Midtrans as a payment gateway. The purpose and contribution of this paper is to develop a cloud-based parking system for transparency in reporting revenue, and to increase user convenience in transactions.

2 LITERATURE REVIEW

In related research, a parking lot reservation system has been created, where users will get information about the nearest parking area around them and all other possible parking area information in the city. If the slots in a particular parking area are full, the user can move to another parking area close to the user's location based on the direction of the system. The system also has an automatic billing feature, which helps users to pay in cashless mode. When the user arrives at the parking location, RFID authentication as an ID will be performed, and then the gate will open. This information will be updated by the on-premises unit located in each parking lot then sent to the cloud-based server in real time. The status of the parking lot will be detected by IR sensors and sent to the on-premises unit in real time. Once the car is parked, billing will start based on the parking time. The parking fee will be notified to the user. The status of all parking spaces will be updated in the cloud in real-time (Hainalkar & Vanjale, 2017).

On the other hand, a subscription parking system has been developed, where each parking lot has an RFID reader at the entrance and exit. The system uses Raspberry pi as a server for databases and access points where all components are connected. In addition, Arduino and ESP8266 are also used combined to read RFID-based card IDs. The data is then sent to the company's desktop application using the ESP8266 which is connected to Raspberry via an Access Point. On this system, the website and android application are used by customers to register, choose services for top up, check transaction history, and the amount of money balance they have. In this system, the calculation of parking fees is carried out by reducing the time when the RFID reader reads the driver's id when exiting and entering the parking access gate, then multiplied by the parking fee / minute where the total cost will be deducted from the customer's account (Bazzi et al., 2018).

In another study, an android-based parking lot search system was developed using the RESTFul API. The Scrum methodology was used in the development of such systems. Implementation of RESTFul API on android applications in order to facilitate parking search in Jakarta. The RESTFul API implementation has several functions as follows. (1) Displays all parking information in Jakarta, (2) Displays parking information based on parking space categories, (3) Displays the quota for motorbikes, cars, and buses or trucks in the parking lot, (4) Addition of favourite parking spots, (5) search for a parking space, (6) directions to the parking lot (Dzulqarnain et al., 2019).

Furthermore, another type of parking system is the use of Web Service as a data controller, Database for data storage and three types of applications. The Web Service is used for the user interface for manipulation to the database. Web Service plays a role in validating the requested data and sending the right response to the client according to the validation results. The client is divided into Web Client (Web Application), Mobile Client (Android Application) and Admin Client (Desktop Application). Web and Mobile clients are used by users to order and pay bills. The Admin Client is used by the administrator to add a new parking location in the system (Anghelescu & Stefan, 2017).

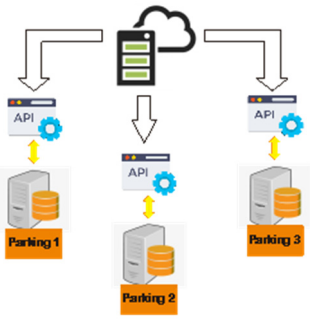

Figure 1: Block diagram of overall system.

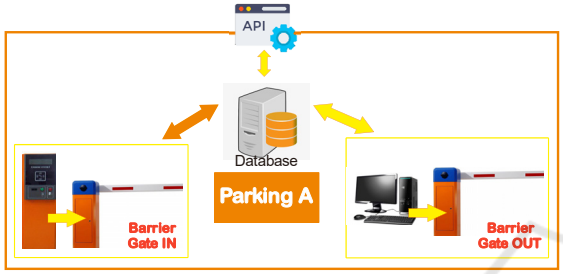

Figure 2: Design of terminal access system in each parking area.

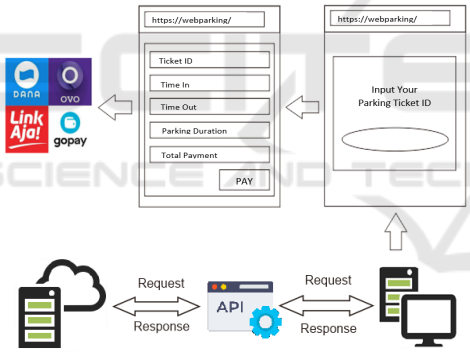

Figure 3: User interface design and its business process.

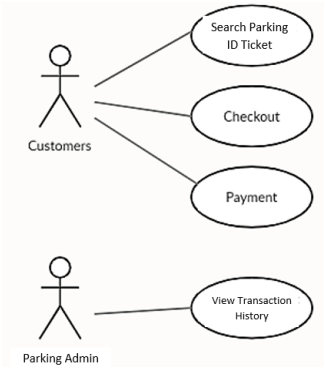

Figure 4: Use case diagram for parking billing system.

There is also a Smart Parking System that uses various technologies, such as RFID, WSN, NFC, Cloud, and cellular. The system is able to collect in real time both environmental parameters and information about the status of the parking lot as well as provide directions to the driver to the nearest empty parking lot by using a software application. The system utilizes NFC-based e-wallets to allow users to pay parking fees. Software applications installed on the cloud platform, are able to manage alert events, such as incorrect use of reserved parking spaces or reserved parking times have run out (Mainetti et al., 2015).

In order to develop an automated parking system related to previous research, the authors made a simplification of the hardware control system installed in the parking gate using the ESP32 as a controller for energy saving schemes, but still maintaining the reliability of the system. In addition, on the server-side development is carried out on the use of APIs to bridge the client. The server provides billing calculation data, check-in and check-out processes, generating QR Codes for cashless payments. While the client only sends input data, it receives and executes the output data generated by the server. In this paper, server is installed on cloud infrastructure to improve server availability and reliability.

3 RESEARCH METHOD

3.1 System Design

Figure 1 shows a prototype system that consists of three parking areas. In each parking area, there is parking vehicle data stored in the database belonging to the parking management company. The Cloud Service system requests data at each parking location but depends on the menu, provided by the API system in each parking area (REST API). The Cloud Service system contains revenue data for each parking area that can be accessed by the Government and Parking Management Entrepreneurs through the website.

The access terminal system in each parking area consists of an entrance gate, exit gate, and database. It can be seen in Figure 2. When pressing the entry button at the entrance gate, it will print a parking ticket, then the Parking Ticket ID data and Entry Date / Time are stored in the database and the gate then will be open. The data will be called later, when a Parking ID lookup has been performed. Furthermore, the user will be able to make payments.

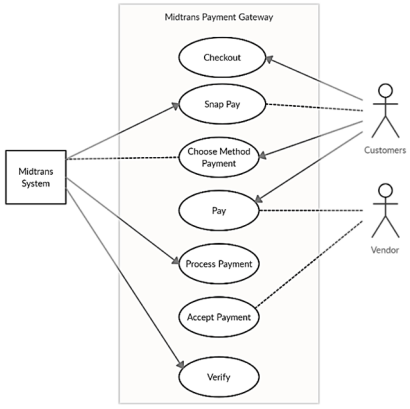

Figure 5: Use case diagram for payment gateway system.

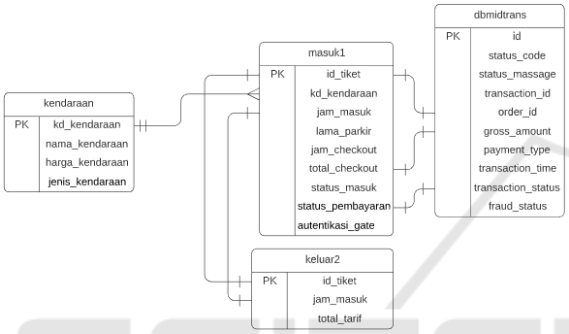

Figure 6: ERD parking billing system.

Figure 3 shows an overview design of the Web-Based Parking Billing User Interface System. When the user will pay the parking fee via the web, and his Parking Ticket ID has been found, the time data when the vehicle enters the parking lot stored in the database will be called. The data contains the Parking Ticket ID and the time of the vehicle when entering the parking lot. Then the data will be displayed along with the data on exit time, Parking Duration and Rates. The deeplink method is used for parking payments using e-wallets, where when selecting the pay menu, it will immediately switch to the e-wallet application. After payment is made, the entire data will be stored in the database, to be accessed by the Cloud Service. An API is implemented, so that the REST Server can connect to the Cloud Server, with the Web API / Web Service used is REST.

3.2 System Modelling

At the modelling stage, the design of the architecture and data structure of the web-based parking user interface system is carried out. The goal of the modelling phase is to understand the big picture of the system.

3.2.1 Case Diagram

The case diagram is designed for the Parking Billing User Interface System to determine the role and actions of the user. The diagram of the Web-Based Parking Billing User Interface System is shown in Figure 4. Meanwhile, in Figure 5 shows the use case diagram for payment gateway system from the Midtrans. The middle part shows the activity carried out by customers and parking managers.

3.2.2 Entity Relationship Diagram (ERD)

ERD functions to prevent errors in designing the database to be used. Relationships between data tables are used to connect one table to another so that it can speed up database access. There are several tables, namely vehicles, entry1, dbmidtrans, and exit2. ERD Web-Based Parking Billing User Interface System is shown in Figure 6.

3.2.3 System Testing

At the testing stage, data collection is carried out by running the system. System testing is carried out to ensure whether the system runs and functions properly according to design. Tests are also carried out to obtain data that can be analyzed. In the work test of the Web-Based Parking Billing User Interface System, four tests were carried out, namely BlackBox testing, web load time testing, delay testing, and notification testing.

3.2.4 Blackbox Testing

Blackbox Testing is a software test that focuses on the functional requirements of the software. Testers can set input conditions that will fully fulfill all functional requirements for a program.

3.2.5 Load Time Testing

Web page load time testing is done by checking the speed of the internet network used. The Speedtest application checks the network speed, where the network speed results for downloading the application are used as a reference for testing the load time of the web page. Web page load time testing is done with three different internet provider. Testing the system is done with the inspect network on the web page to find out the time it takes to access or load the web interface. The result of the load time test is a comparison of the time taken by three internet provider to access or load a web interface.

3.2.6 System Delay Testing

Delay testing is done to evaluate the delay time of a packet from one point to another which becomes its destination due to the transmission process. In the parking billing system, the delay calculation is obtained from the delta time (database time and RTC time). When the entry button at the parking access terminal is pressed, the Real Time Clock (RTC) sends the entry time, then the data will be stored and captured into the database. Delay testing is done to evaluate the time of sending data and the time of receiving data that is delayed. The delay-test that is obtained every time will be taken as a whole average.

3.2.7 Notification Delay Testing

Notification testing is done to get transaction data when a ticket ID payment has been made. Notification testing is done by filling in the Notification URL on the Midtrans portal in the configuration section. The notification URL must lead to the handling notification coding in the parking billing system. The time interval for sending notifications can be seen from the difference between the transaction time and success notification was sent. The delivery time interval is the result of the notification delay test.

4 RESULT AND DISCUSSION

4.1 Web Interface Design

The web interface design consists of implementing the search page, implementing the checkout page, implementing the snap function, and implementing the dashboard interface page. Figure 7 shows the Search Page serves as the main page for performing ticket id searches by users. The Checkout page in Figure 8 serves as a page where users can see whether the data displayed is appropriate. The snap page in Figure 9 contains a snap template of the Midtrans payment gateway which contains data for the ticket id or order ID and total payment which is transferred to the Gojek application using the deeplink method when making payments. Figure 10 shows the Dashboard page where the admin or parking manager can view parking data. The dashboard page display results in the form of a page that contains parking data in, parking out, and parking payments.

4.2 Test Result

In this section, the authors have conducted four parameter testing, (i) BlackBox Testing, (ii) Web Page Load Testing, (iii) System Delay Testing, and (iv) Notification Delay Testing.

Based on BlackBox testing, the data form the experiment explains that the results of the web-based parking billing user interface system testing as a whole get results following what is expected. It can be seen in Table I. Meanwhile, the web page load time testing was conducted to find out the time it takes for a web page to download and display all content in a browser window after the user makes a request or clicks a link in seconds. The data of measurements can be seen in Table II.

Figure 7: Search page view.

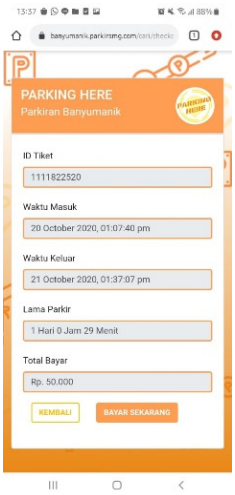

Figure 8: Checkout page view.

Table 1: BlackBox testing result.

No	Description	Actor	Expectation	Results
1	Testing the Search function when entering the correct parking ticket id	User	The system can display the search ticket id and checkout function	As expectation
2	Testing the Search function when entering a paid parking ticket id	User	The system can display a description that the id has been paid for	As expectation
3	Testing the Checkout function on the searched ticket id	User	The system displays the ticket id data, time of entry, time out, parking time and total payment	As expectation
4	Testing the Pay function using the Midtrans deeplink method	User	The system can redirect to snap Midtrans and can make transactions to Go-jek	As expectation
5	Testing the dashboard display of parking entry data	Admin	The system can display data on vehicles that enter parking	As expectation

Table 2: Page load time resting result.

No	Bandwidth (Mbps)	Role	Object	Time (s)
1	30.34	User	Search page	2.8
2			Info Panel Parking ID	0.75
3			Checkout page	1.69
4			Snap page	1.66
5			Payment successful page	1.17
6		Admin	Parking entry data dashboard	3.19
7			Payment data dashboard	2.35
8			Parking out data dashboard	1.43

Figure 9: Snap page view.

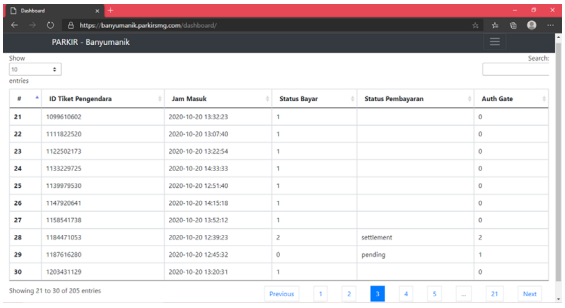

Figure 10: Dashboard page view.

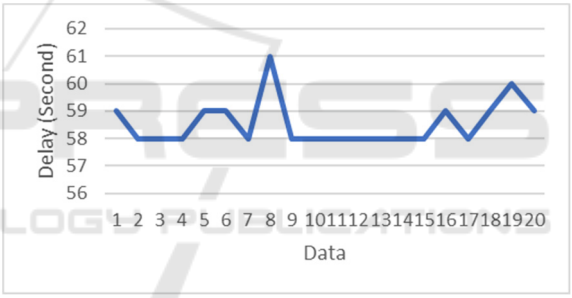

Figure 11: Parking gate delay response.

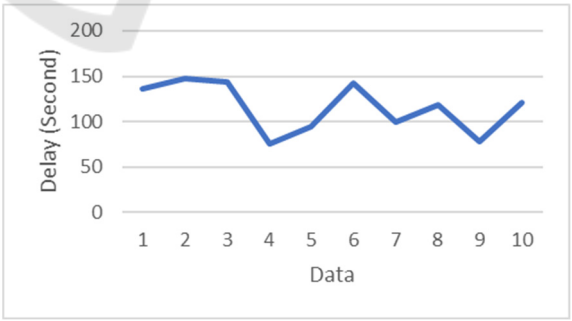

Figure 12: Payment notification delay response.

After that, the experiments continued to measure the system delay and notification delay. The system delay test is done by pressing the entry button with 20 iterations on the parking access terminal. The measurement of delay is done by calculate the difference time of data in the database with the time

of data in the parking gate. This test shows that the parking billing system delay when sending data to the database has an average delay of 58.55 s. The parking system delay graph is shown in Figure 11. Whereas, in the notification delay testing, the parking billing system delay when sending data to the database has an average delay of 115.8 s. The sending process takes a long time because it depends on several factors such as the speed of the Internet network. The parking system delay graph can be seen in Figure 12.

5 CONCLUSIONS

The authors have attempted to develop and conducted evaluation on cloud web-based parking billing system. The developed system can run well and in accordance with planning. However, the response delay for parking gate is still relatively slow. This is allegedly due to the performance of the HTTP protocol used for server and client node communication. For further development, the authors will implement the MQTT protocol for server and client node communication. This aims to reduce the delay response at the parking gate. In addition, the monolith server architecture will be upgraded to a microservice. To improve user convenience, barcode-based user IDs will be replaced with QR-codes in future research.

REFERENCES

Anghelescu, P., & Stefan, I. A. (2017). Parking management and control system. *Proceedings of the 9th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2017*, *2017-January*, 1–4. https://doi.org/10.1109/ECAI.2017.8166395

Bazzi, A., Ghandour, H., Chebbani, A., Ghareeb, M., & Abdul-Nabi, S. (2018). RFID based Paid Parking System. *International Conference on Current Trends in Computer, Electrical, Electronics and Communication, CTCEEC 2017*, 1238–1241. https://doi.org/10.1109/CTCEEC.2017.8454984

Dzulqarnain, M. A., Aziz, M. L., Rachman, M. F., & Atmadja, A. R. (2019). Aplikasi Pencarian Parkir Jakarta Berbasis Android Menggunakan Restful Api. *INTERNAL (Information System Journal)*, *1*(2), 1–13. https://doi.org/10.32627/internal.v1i2.51

Hainalkar, G. N., & Vanjale, M. S. (2017). Smart parking system with pre & post reservation, billing and traffic app. *Proceedings of the 2017 International Conference on Intelligent Computing and Control Systems, ICICCS 2017*, *2018-January*, 500–505. https://doi.org/10.1109/ICCONS.2017.8250772

Hani, U., & Djasuli, M. (2015). Menguak Fenomena Lorong-Lorong Gelap Kebocoran Pajak Dan Retribusi Parkir. *Jurnal Pamator*, *8*(2), 109–124, ISSN 1829-7935.

Mainetti, L., Patrono, L., Stefanizzi, M. L., & Vergallo, R. (2015). A Smart Parking System based on IoT protocols and emerging enabling technologies. *IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings*, 764–769. https://doi.org/10.1109/WF-IoT.2015.7389150

Mulyana, A., & Wijaya, H. (2018). Perancangan E-Payment System pada E-Wallet Menggunakan Kode QR Berbasis Android. *Komputika : Jurnal Sistem Komputer*, *7*(2), 63–69. https://doi.org/10.34010/komputika.v7i2.1511

Priyanti, T., Noviani Hanum, A., & Wibowo, E. (2019). Analisis Pengaruh Penerimaan Pajak Hotel, Pajak Reklame Dan Pajak Parkir Terhadap Pendapatan Asli Daerah (Pad) (Studi Empiris Pada Kabupaten/Kota Di Provinsi Jawa Tengah Tahun 2009-2016). *Maksimum*, *8*(3), 36. https://doi.org/10.26714/mki.8.3.2018.36-47

Usman, R. (2017). KARAKTERISTIK UANG ELEKTRONIK DALAM SISTEM PEMBAYARAN. *Yuridika*, *32*(1), 134–166. https://doi.org/10.20473/ydk.v32i1.4431