

Web and Application Program Interface (API) Design “Parmon” Modern Parking Application

Aco Wahyudi*

Software Engineering Technology,
Agriculture Polytechnic of Samarinda,
75242, Indonesia

Acoxs0707@gmail.com

*Corresponding Author

Eko Junirianto 

Software Engineering Technology,
Agriculture Polytechnic of Samarinda,
75242, Indonesia

eko@politanisamarinda.ac.id

Annafi Franz

Software Engineering Technology,
Agriculture Polytechnic of Samarinda,
75242, Indonesia

annafifranz@gmail.com

Received: 2020-10-13; Revised: 2020-10-13; Accepted: 2021-08-10; Published: 2021-09-01

Abstract—Samarinda city is one of the cities with large local revenue potential in the large parking fee sector. This is evidenced by the quite fast increasing number of users of two-wheeled and four-wheeled vehicles. But due to the increase in vehicle users in Samarinda it has not had a significant impact on the parking fee sector. This is because there is no clear record of every parking transaction. In this case the parking lot is open or roadside. The collection of parking fees in Samarinda there are still many obstacles faced by UTPD parking management, there are still many illegal parking spaces managed by irresponsible elements. One technology that is currently quite developed is the web. With the web, all data and information can be distributed effectively and efficiently and contain and display various information that can be accessed by various devices. Based on the problems and technological developments that have been presented, an idea emerged about the need to create a web-based information system to store and record transactions at any time and create an API for supporting applications. It is expected from this study that every transaction for drivers and drivers can be recorded so that the government can clearly know the income and the amount of levy tax from parking that must be paid by each parking attendant in the “Parmon” application.

Keywords—Parking, Web, API (Application Program Interface)

I. INTRODUCTION

Samarinda City is one of the cities with the potential for local revenue in the parking fee sector which is quite large. This is evidenced by the quite rapid increase in both two-wheel and four-wheeled vehicle users. According to the Samarinda Central Statistics Agency, in 2016 the number of vehicles totaled 129,281,079 vehicles and in 2017 it increased to 138 556 669 vehicles.

However, the increasing number of vehicle users in Samarinda has not had a significant impact on the parking fee sector. This is because there is no clear recording of each parking transaction. In this case in an open parking lot or a curb. Collecting parking fees in Samarinda there are still many obstacles faced by the UTPD (Regional Technical Implementation Unit) parking management,

there are still many illegal parking lots organized by irresponsible individuals so that in collecting parking fees in Samarinda city from the original income target Blood (PAD) that has been determined never reaches the target (Budiarto, 2016).

The rapid development of information technology encourages many organizations and companies to strive to be able to show competitive advantages in order to continue to survive (Ramadiani et al., 2016). The impact of the development of the world of Information and Communication Technology (ICT) at this time increased access to the internet began to change lifestyles Humans in various aspects, be it in the fields of economy, politics, and the social life of society, are now colored by the sophistication of the world of Information and Communication Technology (Sundari, 2016). One technology that is currently quite developed is the web. With the web, all data and information can be channeled effectively and efficiently and contain and display various information that can be accessed by various devices. In addition, the development of smartphone technology has also begun to make dependence on society. The dependence of the community on a smartphone cannot be separated from the conveniences offered by the development of this device. Along with the development of these smartphone devices, the development of applications that support the use of smartphones is also accompanied by the features offered. The trend of a technology that supports these applications is now starting to increase in its development, especially on smartphones with the Android operating system, from a standalone application to an application that can connect with other applications via a global network. This technology is now known as the Web Service API (Syarwani, 2019).

Based on the problems and technological developments that have been conveyed, the idea emerged that the need to create a web-based information system to store and record transactions for each parking attendant and create an API for supporting applications. The author plans to design and create a web and API using the Laravel framework.

II. LITERATURE REVIEW

A. *Study of Literature*

Some of the literature as a guide and reference in this paper:

1. Research conducted by Kurniawan & Puspita (2010) with the title Parking Service Information System Equipped with a Camera is a parking service information system application that makes it easier to run parking applications, faster leaders can monitor the results of parking reports at any time, and the presence of cameras increases parking security at parking area. With this application, it is expected to solve these problems.
2. Research conducted by Hanaf (2015) in his journal entitled Design and Building of a Web-Based Car Workshop Information System is an information system application as a tool to provide the right solution to solve problems in the PT. Surya Kencana includes transactions that are not recorded effectively and providing information about customer vehicle maintenance that cannot be accessed by customers anytime and anywhere. With this system, it is hoped that PT. Surya Kencana can solve these problems and can improve service to customers.
3. Research conducted by Sundari (2016) with the title Web-Based Public Health Center Service Information System is an information system application that has queuing activities, registration, and patient medical records. In general, the system performance in patient services running at the health center is not optimal because it is still processing patient data and medical record data still using bookkeeping or manual media. So patient services at the puskesmas are not effective and efficient, because the bookkeeping media slows down the making of reports. Implementation of the program using the PHP language with a database using MySQL. This system was created to aim to build a computerized information system, making it easier for the health center to process patient data and patient medical records into reports.
4. Research conducted by (Solihin & Kurniadi, (2017)with the title Design and Build a Parking Information System for Garut College of Technology (STTG) is a parking information system application that can make it easier to improve security in parking management by using the software development methodology used, namely the Rational Unified Process (RUP). By making a parking management information system at STTG, it is hoped that it will be able to assist in parking management and improve security and make parking at STTG more organized.
5. Research conducted by Perkasa & Setiawan (2018)with the title Community Data Web Service Development Using the REST API with Access Token is an application that provides community web service data to create and register job search cards using available community data. This

application is able to provide and facilitate many parties, such as data administrators to monitor data usage, register employees in data input, and people can register independently.

B. *Web (World Wide Web)*

WWW or the World Wide Web or the web alone is a system that is interrelated in a hypertext format document that contains a variety of information, including text, images, sound, video, and other multimedia information and can be accessed through a device called a web browser. To translate documents in hypertext form into understandable documents, the web browser through the web client will read the web pages stored on a web server through a protocol known as HTTP or Hypertext Transfer Protocol (Morita et al., 1974).

C. *PHP (Hypertext Preprocessor)*

PHP is a programming language that allows web developers to create dynamic web applications quickly and easily. PHP was pioneered and first introduced around 1994 by Rasmus Lerdorf through his website to find out who had accessed his online summary. PHP is a server-side scripting language that integrates HTML tags. Server-side scripting is the syntax and commands that are executed on the server and included in HTML documents. On a dynamic web page, PHP functions as a programming language that carries out certain commands, while HTML functions as the structure of the web page design. When a dynamic web page is opened on a web browser page, the first time that happens is the server processes all existing PHP commands then displays the results in HTML format to the web browser, so that what is displayed to the web browser is only the design view of HTML, while PHP scripts working behind the scenes.

D. *Laravel*

Laravel is a PHP framework with open code (open source) with an MVC (Model-View-Controller) design which is used to build website applications. This framework was first built by Taylor Otwell on February 22, 2012 (Junirianto, 2018). Laravel provides an updated tool for interacting with databases called migrations. With migration, developers can easily modify a database on a platform independently because the database schema implementation is represented in a class. Migration can run on several databases that have been supported by Laravel (MySQL, PostgreSQL, MSSQL, and SQLITE) and for implementation of Active Record on Laravel called Eloquent which uses the modern OOP standard.

E. *API (Application Program Interface)*

API stands for Application Programming Interface, or often translated into an application programming interface. There are many definitions put forward about API. One of the briefs was conveyed by the Konkani NLP Team, from Goa University, in a presentation entitled API or Application Programming Interface.

According to them, "API is a set of commands, functions, and protocols that can be used by programmers when building software. The API allows programmers to use standard functions to interact with other operating systems. Because of its function as a Lego translator, bridge, and game, this technology called API makes complex problems simpler and easier. API consists of various elements such as functions, protocols, and other tools that allow developers to create applications. The purpose of using the API is to speed up the development process by providing separate functions so that developers don't have to build similar features. You can feel the application of the API if the desired feature is very complex, of course it takes time to build something similar to it. For example: integration with a payment gateway. There are various types of system APIs that can be used, including operating systems, libraries, and the web. API can be accessed via HTTP request protocol, this is a concept not a technology. We can create a Web API using different technologies such as PHP, Java, .NET, etc. (Sandi, 2017). In accessing the API, there are several HTTP request methods that have specific functions. The following types of HTTP request methods for accessing the API are presented in Table 1.

Table 1. Types of HTTP Request Methods

No	Method	Information
1.	<i>HTTP GET</i>	This method is used to take representational information only and not change it.
2.	<i>HTTP POST</i>	This method is used to send the entity to the specified resource, and make additions to a resource.
3.	<i>HTTP PUT</i>	This method is used to replace all current representations of the target source with the request payload.
4.	<i>HTTP DELETE</i>	This method is used to delete the specified resource.
5.	<i>HTTP PATCH</i>	This method is used to apply partial modifications to the resource.

After sending a data or request via HTTP Request, then the server responds via HTTP Response where the response is in the form of Json (Javascript Object Notation) which contains data or information and has a status code that describes the server status of the requested resource. The HTTP response status code indicates whether the specified HTTP request has completed successfully. Responses are classified into five classes consisting of informational responses, successful responses, redirects, client errors, and server errors (Admin, 2018). The following are several types of status codes in HTTP responses which are presented in Table 2

Table 2. Various kinds of status code in HTTP Response

No	Status Code	Information
1.	<i>100</i>	Continue, the temporary response so far is OK, and the client should either continue with the request or ignore it when it's done.
2.	<i>200</i>	OK, request success.
3.	<i>201</i>	Created, The request was successful and a new resource was created as a result.
4.	<i>202</i>	Accepted, the request has been received but has not been followed up.
5.	<i>204</i>	No Content, There is no content to post for this request.
6.	<i>300</i>	Multiple Choices, Requests have more than one possible response. The user agent or user must choose one of them.
7.	<i>301</i>	Moved Permanently, This response code means that the URI of the requested resource has been changed.
8.	<i>304</i>	Not Modified, This is used for caching purposes. Notifies the client that the response has not been modified.
9.	<i>400</i>	Bad Request, This response means the server cannot understand the request due to invalid syntax.
10.	<i>401</i>	Unauthorized, Authentication required getting the requested response.
11.	<i>403</i>	Forbidden, the client does not have access rights to the content so the server refuses to provide an appropriate response.
12.	<i>404</i>	Not Found, the server cannot find the requested resource.
13.	<i>500</i>	Internal Server Error, Server has encountered a situation that cannot be handled.
14.	<i>501</i>	Not Implemented, The request method is not supported by the server and cannot be handled.
15.	<i>502</i>	Bad Gateway, This error response means that the server, while working as a gateway to get the response needed to handle the request, got an invalid response.
16.	<i>503</i>	Service Unavailable, the server is not ready to handle the request.

F. System Usability Scale (SUS)

System usability scale is scale that used to assessment of usability system. This method has been used by so many people to testing software, because it is easy to use and the result could be consideration. Another reason using this method is easy to understanding and easy to use, so that this method is popular to usability testing (Santoso, 2016)

This method has 10 questions and 5 answer options. The maximal score is 100 and minimal is 0. The 10 questions can be seen in table 3 and the 5 answer questions can be seen in table 4.

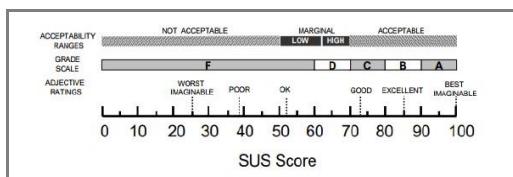
Table 3. List Question of SUS

No.	Question
1	I think that I would like to use this system frequently.
2	I found the system unnecessarily complex.
3	I thought the system was easy to use.
4	I think that I would need the support of a technical person to be able to use this system.
5	I found the various functions in this system were well integrated.
6	I thought there was too much inconsistency in this system.
7	I would imagine that most people would learn to use this system very quickly.
8	I found the system very cumbersome to use.
9	I felt very confident using the system.
10	I needed to learn a lot of things before I could get going with this system.

Table 4. List Answer option of SUS

No.	Option	Score
1	Strongly Disagree	1
2	Disagree	2
3	Half and half	3
4	Agree	4
5	Strongly Agree	5

After collecting data from respondents, data will be measured to find out the average score from all respondents. The average score will be concluding by SUS score to find out in which grade. SUS score can be seen in picture 1.



Picture 1. SUS Score

III. RESEARCH METHODS

A. Research Time and Place

This research conducted at Samarinda City especially in parking are of Agriculture Polytechnic of Samarinda. It takes duration from October 2019 – Maret 2020. Includes making proposal, collecting data, application development, and reporting.

B. Tools and Materials

Hardware that used in development is as follows:

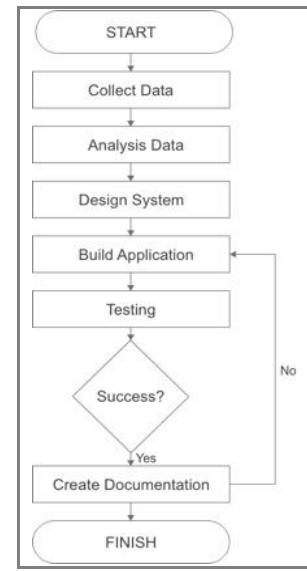
1. HP Notebook Laptop 14-an017AI (AMD A6, CPU 2.0 GHz, RAM 10GB DDR3L, 500GB HDD, 256GB SSD)
2. Printer for printing some data paper

Software that used in development is as follows:

1. PhpStorm 2017
2. Xampp
3. Postman
4. Browser

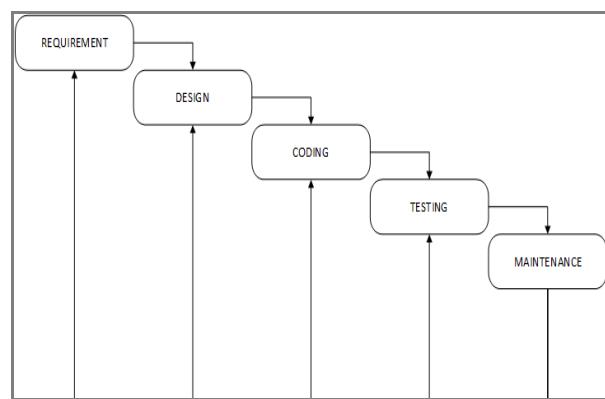
C. Research Procedure

Research procedure can be seen in picture 2. This research procedure is shown as flow chart diagram.



Picture 2. Research Procedure

In this research, the system making technique uses the waterfall method. The waterfall method is a method of developing information systems that is systematic and sequential, meaning that each stage in this method is carried out sequentially and continuously (Rusmawan, 2019). Method that used in developing the system is Waterfall. This method can be seen in picture 3.



Picture 3. Waterfall Method

D. System Design

Designing User Interface (UI) is a process to create effective communication media between human and computer or mobile application. UI is connector between user and system (Susilo et al, 2018). UI design of this system can be seen in picture 4, 5, 6, 7.



Picture 4. User Interface Design Landing Page



Picture 5. User Interface Design Login Page



Picture 6. User Interface Design Register Page



Picture 7. User Interface Design Dashboard Page

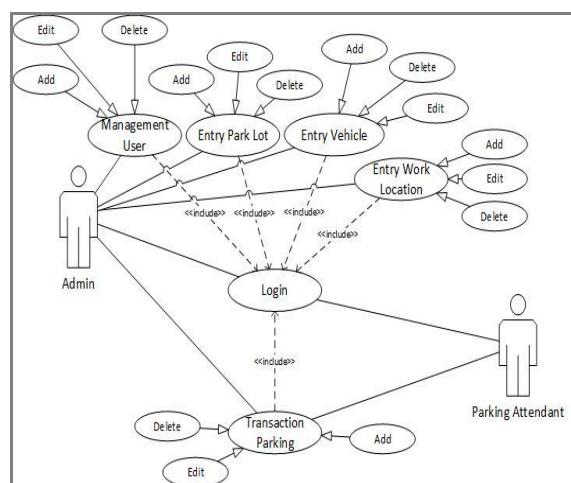
are the focus of developing the android system of this research can be seen in Table 5.

Table 5. API & Entity

#	Process	Candidate API		Entity
		Method	URL	
1	Create and display logins	POST	api/login	User
2	Obtain and display transaction data per parking attendant day	GET	api/transaction_day	Transaction
3	Obtain and display parking location data according to parking attendant users	GET	api/park_location	parking area
4	Obtain and display parking attendant user transaction data	GET	api/transaction	Transaction
5	Obtain and display parking attendant user transaction data	POST	api/transaction /store	Transaction

F. Use Case Diagram

The UML designs are made in the form of use case diagrams, activity diagrams, sequence diagrams, and class diagrams. The following is the use case design on the system which can be seen in Picture 8.



Picture 8. Use Case Diagram

E. Candidate API

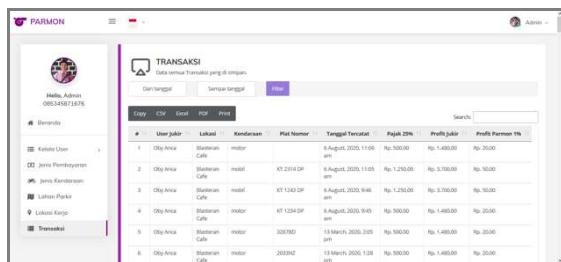
Based on use case analysis, ERD, and database design, it can be concluded that all API candidates who

IV. RESULT AND DISCUSSION

A. Implementation Interface

1. Landing Page

Picture 9 shows the landing page of the "Parmon" web. On this page there are 4 sections, namely MainHeader, where there is a brief explanation of "Parmon" and there is an about and contact menu on the top right and an admin login link on the top right of MainHeader that says "Parmon". The second part is about "Parmon", which contains persuasive content to invite parking managers to cooperate with "Parmon". The third part is content that contains the terms and conditions for registering as a "Parmon" partner or parking attendant. And the fourth part is content that contains contacts in the form of email and phone numbers to contact and request more information regarding the "Parmon" application.



Picture 9. Landing Page Web

2. Login Page

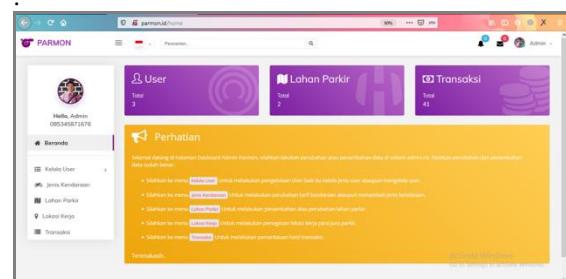
Display of the "Parmon" web admin login page where the admin in charge and has been registered, logs in by entering the email and password before entering the "Parmon" admin dashboard page just.



Picture 10. Login Page

3. Dashboard Page

"Parmon" web dashboard display where the admin can manage parking attendant users, provide prices for each vehicle in the vehicle type menu, manage parking lots, provide parking attendant users' work locations according to the parking area and view transactions made by parking attendants. In addition, there are statistical data regarding the number of registered parking attendant users, active parking lots and the number of transactions that have been successfully recorded



Picture 11. Dashboard Page

4. Transaction Page

On the transaction page, the results of recording transactions that have been successfully carried out by the parking attendants where the admin can see the vehicle plate number, the type of vehicle and the recorded date of the transaction made by the parking attendants.

Picture 11. Transaction Page

B. API Application "Parmon"

1. API Login

The "Parmon" login API is a parking attendant user authentication API with JWT security. To request the API login using Postman with the POST method and the URL <http://Parmon.id/api/login> and fill in the body form-urlencoded in the email key and password. The following is the input process in.

1. POST /api/login HTTP/1.1
2. Host: "Parmon".id
3. Content-Type: application/x-www-form-urlencoded
4. Accept: application/json
5. email=thered@gmail.com&password=1234567890

2. API Parking attendant transaction data

The transaction data API with the GET method is an API that displays transaction information that has been successfully recorded by the parking attendant. To request a transaction data API, use Postman with the GET method and URL <http://Parmon.id/api/transaksis> and fill in the Header in the Authorization, Accept and Content-Type keys, where the value of the Authorization key is in the form of a token bearer that we get after logging in . The following is the input process in Postman

1. GET /api/transaksis HTTP/1.1
2. Host: "Parmon".id
3. Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwO1wvXC9wYXJtb24uaWRL2FwaWvbG9naW4iLCJpYXQiOjE1NzK3OTUxNzEsImV4cCI6MTU30Tc50Dc3MSwibmJmIjoxNTc5Nzk1MTcxLCJqdGkiOiJVNEJKYmRGRXBjQUMxN1lVIiwi c3ViIjo0LCJwcnYi0iI4N2UwYWYxZWY5ZmQxNTgxMmZkZW5NzE1M2ExNGUwYjA0NzU0NmFhIn0.nmHTx2rBla mFGv6tEL_rpDxue604ml0pILMVW_wp70
4. Accept: application/json
5. Content-Type: application/json

3. API Parking attendant transaction data/day

The parking attendant transaction data API per day with the GET method is an API that displays transaction information that has been successfully recorded by the parking attendant per day. For daily transaction data API requests, use Postman with the GET method and URL http://Parmon.id/api/today_transaksi and fill in the Header in the Authorization, Accept and Content-Type key sections. The following is the input process in Postman.

1. GET /api/today_transaksi HTTP/1.1
2. Host: "Parmon".id
3. Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9wYXJtb24uaWRcL2FwaWvbG9naW4iLCJpYXQiOjE1NzU5MDY40TcsImV4cCI6MTU3NTkxMDQ5NywibmJmIjoxNTc10TA20DK3LCJqdGkiOiJncFRFZENCZ3Y30DVPeE9FIiwi c3ViIj00LCJwcnYi0iI4N2UwYWyxZNY5ZmQxNTgxMmzkZWM5NzE1M2ExNGUwYjA0NzU0NmFhIn0.R8Nry-AYz2LkmsI6T6mHiy6hbpQcNHVYiF-a84gd_ow
4. Accept: application/json
5. Content-Type: application/json

4. API Post Data transaction

API post transaction data with POST method is an API that stores transaction data to the database server by parking attendants. To request the API post transaction data, use postman with the POST method and URL <http://Parmon.id/api/transaksis> and fill in the Header in the Authorization, Accept and Content-Type keys. Then fill in the Body in the key plate number and vehicle id_jenis_filled according to the vehicle parked at the location. The input process in Postman is as follows

1. POST /api/transaksi HTTP/1.1
2. Host: "Parmon".id
3. Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9wYXJtb24uaWRcL2FwaWvbG9naW4iLCJpYXQiOjE1NzK30Tg40DcsImV4cCI6MTU3NTgMjQ4NywibmJmIjoxNTc5Nzk40Dg3LCJqdGkiOiIy0VpRUEhobk9YUfVvjd1Iiwi c3ViIj00LCJwcnYi0iI4N2UwYWyxZNY5ZmQxNTgxMmzkZWM5NzE1M2ExNGUwYjA0NzU0NmFhIn0.rREPWPQ-SioJDacn-nkZMqB15PUPUe8dKe8GAqop_Ds
4. Content-Type: application/x-www-form-urlencoded
5. Accept: application/json
6. plat_nomor=KT 000 AA&id_jenis_kendaraan=2

C. "Parmon" Web UI / UX Testing

UI / UX testing for the "Parmon" web was carried out using the SUS (System Usability Scale) questionnaire. The following data on the results of the SUS calculation are shown in Table 6.

Table 6. Data the calculation of the "Parmon" web SUS

Respondents	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10	Total	Score SUS
R1	3	3	3	3	4	4	3	3	3	2	31	77,5
R2	3	3	3	4	3	4	3	3	3	3	32	80
R3	3	4	3	3	4	3	2	3	3	3	31	77,5
R4	3	4	3	3	3	4	2	3	4	3	32	80
R5	4	3	4	3	3	4	3	3	3	3	33	82,5

Respondents	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10	Total	Score SUS
R6	3	4	3	4	4	3	3	3	3	3	33	82,5
R7	3	3	3	2	4	4	3	3	3	3	31	77,5
R8	4	3	4	3	3	4	3	4	4	3	35	87,5
R9	4	3	3	3	4	3	3	3	3	3	32	80
R10	3	4	3	3	3	4	3	3	3	3	32	80
R11	3	3	3	3	4	4	3	3	3	3	32	80
R12	3	3	3	4	3	4	3	3	3	3	32	80
R13	3	3	3	3	3	3	3	3	3	3	30	75
R14	3	3	3	3	3	3	3	3	3	3	30	75
R15	3	3	3	3	3	3	3	4	3	3	31	77,5
R16	3	3	3	3	3	4	3	3	3	3	31	77,5
R17	4	3	3	3	3	4	2	3	3	3	31	77,5
R18	3	3	3	3	3	3	3	3	3	3	30	75
R19	3	4	3	3	3	4	3	3	3	3	32	80
R20	3	3	3	3	3	4	2	3	3	3	30	75
R21	3	3	4	3	3	3	3	4	3	3	32	80
R22	3	3	3	3	3	3	2	3	3	3	29	72,5
R23	3	3	3	3	3	4	2	3	3	4	31	77,5
R24	3	3	3	3	3	3	2	3	3	3	29	72,5
R25	3	3	3	3	3	2	4	3	3	3	30	75
Average Score												80,36

After calculating the data from the respondent's results, in order to calculate the SUS score, the calculation for each question value is calculated in 2.5 times and then the average value of the SUS score is calculated, the value is 80.36. This proves that the test from the "Parmon" web is acceptable because the value obtained is above average and gets a grade scale B value as shown in Figure 4 and gets an excellent rating.

V. CONCLUSION

From the results of testing and analysis of the "Parmon" Web and API, it can be concluded that the "Parmon" web was successfully created with the Laravel framework where the parking attendant account registration system is registered by the admin on the user management web page and has a price setting for each vehicle that can be done on the vehicle type web page. Each transaction is successfully recorded and stored in the database and can be viewed on the transaction web page. On the transaction webpage, it also manages to display the results of sharing for levies, parking attendants, and the "Parmon" application manager. For the API login, "Parmon" was successfully created with JWT security where the authentication process for the API login uses email and password. The transaction data API succeeds in displaying data based on the parking attendant user along with the details of each recorded transaction and the transaction data API successfully displays data per day based on the parking attendant user and the API for posting transaction data is successful to save each transaction according to the parking attendant user.

REFERENCES

Admin. (2018). *Daftar Http Status Code*. PT Proweb Indonesia.

Budiarto, S. (2016). Studi Tentang Pemungutan Retribusi Parkir. *Studi Tentang Pemungutan Retribusi Parkir Dan Kontribusinya Terhadap Pendapatan Asli*

Derah Di Kota Samarinda, 4(3), 1–7.

Hanaf, F. (2015). Rancang Bangun Sistem Informasi Bengkel Mobil Berbasis Web (Studi Kasus Pt . Surya Kencana). *Creative Information Technology Journal*, 3(1), 1–13.

Junirianto, E. (2018). *Pemrograman Web Dengan Framework Laravel* (T. W. Publish (ed.); 1st ed.). Wade Group.

Kurniawan, F., & Puspita, E. (2010). Sistem Informasi Pelayanan Parkir Yang di Lengkapi Dengan Kamera. *EEPIS Final Project*. <http://repo.eepis-its.edu/405/>

Morita, N., Shimizu, M., Arisawa, M., & Koshi, M. (1974). Studies on the Medicinal Resources. XXXVI. The Constituents of the Leaves of *Saxifraga stolonifera* Meerburg (Saxifragaceae). *Chemical and Pharmaceutical Bulletin*, 22(7), 1487–1489. <https://doi.org/10.1248/cpb.22.1487>

Perkasa, M. I., & Setiawan, E. B. (2018). Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token. *Jurnal ULTIMA Computing*, 10(1), 19–26. <https://doi.org/10.31937/sk.v10i1.838>

Ramadiani, Widagdo, P. P., & Junirianto, E. (2016). Tracer Study Menggunakan Framework Bootstrap. *Prosiding Seminar Nasional Ilmu Komputer Dan Teknologi Informasi*, 1(1), 1–7.

Rusmawan, U. (2019). *Teknik Penulisan Tugas Akhir dan Skripsi Pemograman*. Elex Media Computindo.

Sandi, A. (2017). *Mengenal Apa itu Web API*. Codepolitan.

Santoso, Z. S. and H. B. (2016). An Indonesian adaptation of the System Usability Scale (SUS). *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 145–148. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7872776&isnumber=7872705>

Solihin, M., & Kurniadi, D. (2017). Rancang Bangun Sistem Informasi Parkir Sekolah Tinggi Teknologi Garut. *Jurnal Algoritma*, 14(2), 9–16.

Sundari, J. (2016). Sistem Informasi Pelayanan Puskesmas Berbasis Web. *IJSE – Indonesian Journal on Software Engineering*, 2(1), 44–49. <http://ejournal.bsi.ac.id/jurnal/index.php/ijse/article/view/665/556>

Susilo, E., Wijaya, F. D., & Hartanto, R. (2018). Perancangan dan Evaluasi User Interface Aplikasi Smart Grid Berbasis Mobile Application. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 7(2), 150–157. <https://doi.org/10.22146/jnteti.v7i2.416>

Syarwani, A. (2019). *Perancangan Api Point Of Sale Dengan Keamanan Json Web Token (JWT)*.