

WEBSITE AND DATABASE IMPLEMENTATION FOR VEHICLE VIOLATION DATA CHECK BASED ON GOOGLE FIREBASE

Judha Ananda Pratama¹, Ridha Muldina Negara², Nyoman Bogi Aditya Karna³

^{1,2,3} Telecommunication Engineering , School of Electrical Engineering , Telkom University

¹yudhaananda@student.telkomuniversity.ac.id, ²ridhanegara@telkomuniveristy.co.id,

³aditya@telkomuniversity.ac.id

Abstract

Transportation is one of the most important aspect in human daily activity that used almost every day. In Indonesia, the growth of vehicles numbers each year has always increased in line with population growth increase as well. As the growth of vehicles number making space on the street diminish, many vehicle users commit violation and seize pedestrian rights. In order to overcome these violations, a tool is needed to discipline vehicle users to comply with regulation. Amans is a solution for the problem, it is a mini Internet of Things (IoT) device that record vehicle plates every time it violates the regulations. This paper demonstrates the data processing obtained from Amans device which is the vehicle plate number, this data stored in Firebase Realtime Database. Furthermore, the database is accessible for the vehicle user as they are able to access it through a website. The aim of this paper is to demonstrates the creating of fully functional system that able to store data to the database in less than two seconds.

Keywords: Vehicle, Database, Website, Firebase, Python

1. Introduction

Indonesia is one of few countries with the highest population in the world with population of around 267 million. From such a large population, in 2017 it is estimated that the number of vehicles in Indonesia is 138 million based on data from Badan Pusat Statistik Indonesia[1]. Around 80% of the vehicles is motorcycle, it is in line with police report every year which says that motorcycle is the vehicle with the highest accident rate. In the future, it is expected that the number of vehicles will keep increasing as population keep increasing uncontrollably. Infrastructure and traffic regulations are two important things in transportation. The growth in the number of vehicles that are increasingly booming has diminished space on the streets that other street users, such as pedestrians, should be able to use.

Reduced space on the streets encourages vehicle users to commit violations such as crossing the boundary line at traffic lights and using sidewalks as road for vehicles. That kind of violations must be dealt with according to regulations by responsible party, namely the traffic police. In daily life, the police were overwhelmed to take action for violation as there are many vehicle users commit violation. This condition also caused vehicle user to commit violations more frequently. Amans is the solution for the police to take action against violators of the law without the need to deal directly with the violators and it directly facilitate the work of traffic police.

Amans is a mini IoT device equipped with a camera that serves to detect vehicle plates. If the device detects any violation, it will automatically record the vehicle plate number then send it to the database. Violation can be recognized if vehicle users passes the parameters specified in Amans device configuration. As the vehicle plate number sent to the database, it will be matched with data in the database inputted by the authorities. In case, the vehicle plate number matches the data, database will automatically add the number of violations of that vehicle users. When vehicle users have committed three violations, they will receive a notification and warning email. This Final Project is expected to develop the database so that it will work as it should be.

2. Basic Concept

2.1 Database

Database is an organized collection of data, generally stored and accessed electronically from a computer system. It is essentially a computerized record-keeping system [2]. In order to gain access to the data inside a database, it needs Database Management System (DBMS) consisting of an integrated set of computer software that allows users to interact with one or more databases and provides access to all of the data contained in the database. The database itself can be considered as a kind of electronic filing cabinet, it is a repository for a collection of computerized data files [3]. Some examples of database software are Oracle, Filemaker Pro, Microsoft Access, Microsoft SQL Server SAP, and MySQL. Database software provides an interface for the users and the database. The interactions facilitated by DBMS include data definition and update, retrieval for reports or queries, and administration of data security and recovery [4].

In this paper, database that will be used is Realtime Database from Google Firebase. Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011 which then acquired by Google in 2014 [5]. Firebase service that is used in this paper is Realtime Database which is a service that provides application developers an Application Programming Interface (API) that allows application data to be synchronized across clients and stored in Firebase's cloud.

2.2 Python

Python is an interpreted and object oriented high-level programming language with automatic management of memory and dynamic semantics. Guido van Rossum created Python back in early 1990s at Stichting Mathematisch Centrum (CWI) as a successor of ABC programming language [6]. Python is a computer programming language that widely used worldwide not only for natural sciences but also for machine learning and data mining. Python is open source, free, and also flexible as it can run on various platform such as Windows, MAC OS and Linux/Unix.

2.3 Website

Website is a group of World Wide Web (WWW) pages usually containing hyper-link to each other and made available online in order to be accessed by many users. Website can have many functions such as personal website, corporate website, government website, etc. All publicly accessible websites collectively constitute the WWW, whereas private websites, such as a company's website, are typically part of an intranet [7]. Web Pages, which are the building blocks of websites, are document composed in plain text interspersed with formatting instruction of Hypertext Markup Language (HTML) or Cascading Style Sheets (CSS). Website can be accessed using software called web browser such as Mozilla Firefox, Safari, Chrome, and Opera by typing the web address into the browser. Website is hosted in special computers that are constantly connected to the internet known as web server where user send requests to access the website.

2.4 Hypertext Markup Language

Hypertext Markup Language (HTML) is a standard markup language mostly used for documents that displayed in web pages. HTML first version was written by Tim Berners-Lee, a contractor in CERN in 1990s and has been developed since then with its latest version is HTML5 [8]. HTML documents are able to be presented in Graphical User Interface (GUI), text-to-speech devices, text-only systems, and so forth. The word Hypertext refers to how HTML documents in web pages are linked together while the words Markup Language refer to how HTML is used to mark-up a text document with tags that will be translated by web browser to interpret the structure of the web pages display and compose materials into audible and visual web pages [9].

HTML was designed to define a documents structure such as heading, subheading, paragraphs, and so forth in order to facilitate documents exchange between researchers. HTML can be assisted by other technologies such as CSS and scripting language such as JavaScript in order to cover the drawbacks of HTML. In this paper, HTML will be used to create web pages for registration menu, login menu, and violation data display page. Registration menu will be an online form that used to handle registration process.

3. Work System

Objectives of this project are to manage the output data from Raspberry Pi and create the database to store those data. The database will be updated and synchronized every time there is new data from Amans device. In this thesis, most of the explanations will focus on processing output data from Raspberry Pi that will be sent to Firebase Realtime Database. As shown in Fig. 3.1, output data from Raspberry Pi will be sent to the database through internet. Users are able to check their own violation data through the website that can be accessed through the internet. The parts such as Tools Placement, Object Detection, and Object Recognition that is inside the box with dotted line are the focus of writer partner's thesis whereas the focus of this thesis is everything outside the dotted line box.

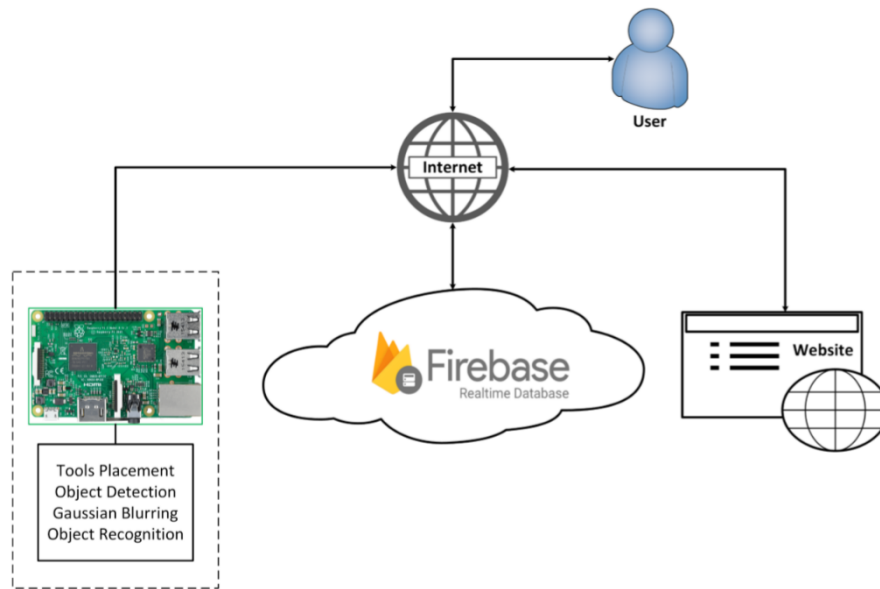


Figure 3.1 System Block Diagram.

In general, there are two parts that are going to be planned for this thesis, design of the database and the website. In the database design, writer will determine the Create, Read, Update, Delete (CRUD) rules to be applied in the database. In order to access the database, a web browser will be needed to open the interface of Firebase Realtime Database. In the website design, writer will write a HTML codes for Registration form, Login menu, and web page to display violation data. In order to integrate the database with the website, all of the codes are written in Python and HTML using Notepad++ which then will be compiled through Command Prompt from Windows. HTML codes are written for web pages only as it will be inserted into Python program.

3.1 Database System

Database system work flow starts when a plate number data in the form of text or string sent to Firebase Realtime Database, where the vehicle identification data were already inputted by the authorized person. The database has also been installed with some basic rules to regulate the limit to Create, Read, Update, and Delete. As shown in Fig. 3.2 left picture, database will check whether the plate number match the existed vehicle identification in database or not. If the plate number match with the data in database, the violation amount will be added by one whereas if the plate number does not match with the data in database, it will be treated as a new data where the violation amount declared as one. After that, database will update and synchronized the data and will send specific data to the website as user requested.

3.2 Website System

Website system work flow starts as vehicle user request for the website through their web browser, then web hosting service will send the web pages as requested. When the user open the website, the first thing they see will be the homepage which is a welcome page with three button that represent menus provided. The three menus are, admin login, user login, and registration menu. Admin login is the feature for authorized person to manually input vehicle identification data to the database. User login is the feature for vehicle user to check their own violation data while registration menu is the menu to register vehicle user which the username and password will be used to log in.

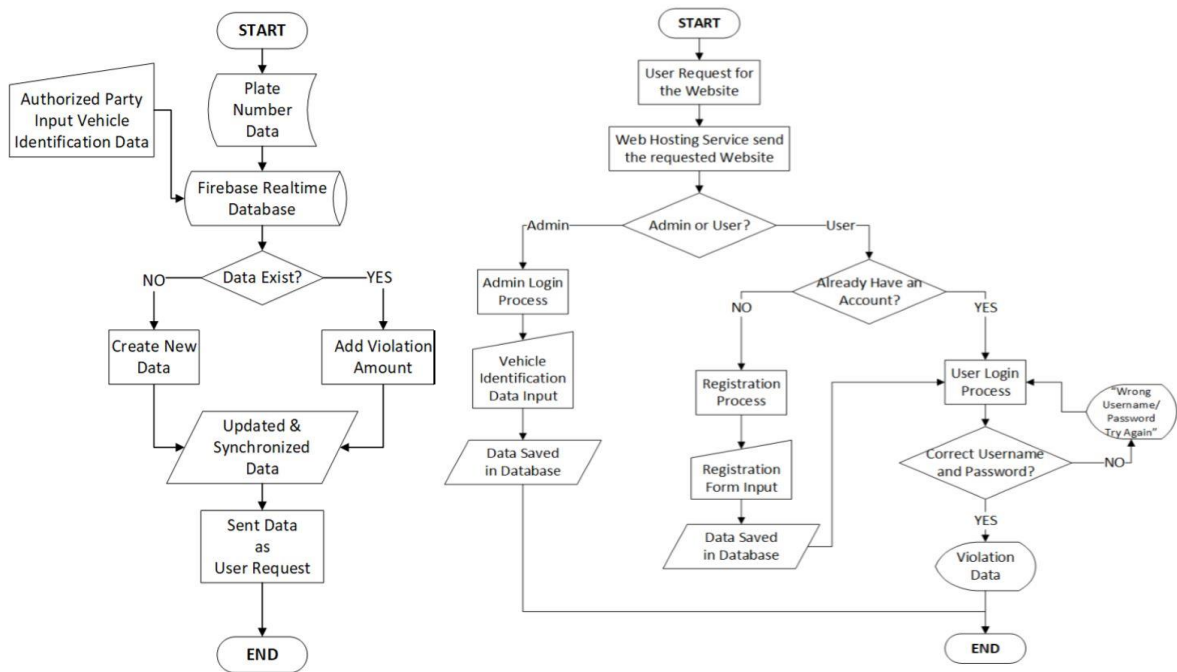


Figure 3.2 Database Work System (Left) and Website Work System (Right).

User login menu intended to be login portal for registered vehicle user to check their own violation data whereas for unregistered user, they have to register themselves through registration menu where they will obtain username and password to log in. In registration menu, users have to fill out an online form which will then be stored in Firebase Realtime Database. After finishing registration process, user can log in through user login menu with username and password obtained. As seen in Fig. 3.3, user has to fill in correct username and password in order for violation data to be displayed. In case the user using the wrong username and password, the screen will display “Wrong Username/Password, Try Again”. If the users fill in the correct username and password, the screen will display their violation data.

Process of website interface design and making process can be elaborated as follows:

- 1) First step to be done in website interface making process is to install all the software needed such as Notepad++ which is a free source code editor and Python version 3.8.1 with all the libraries needed to finish this thesis such as Flask and Firebase libraries.
- 2) Second step of this phase is to design an interface of each web page which are Homepage, Login page, Registration page, and Admin Page. These pages are written in HTML that will be executed using Python.
- 3) Third step is to connect registration page with Firebase Realtime Database. Registration page is an online form that will be filled in by unregistered user to get their username and password. In this step, method used is Firebase.Post and Firebase.Get. Post method is a method to write data to the database and Get method is a method to read data from the database.
- 4) Fourth step of this phase is to connect login page with Firebase Realtime Database. Login page is a login form that needs user’s username and password in order to show their violation data. In this step, method used is Firebase.Get as the data from the database need to be read and displayed on the website.
- 5) Fifth step of this phase is to connect admin page with Firebase Realtime Database. Admin page is a registration form for vehicle identification data that used Firebase.Post method to write the registration data to the database.
- 6) Last step of this phase is to check the functionality of each features on the website such as the function to read and write data to the database. This functionality check has a purpose to make sure the website interface is working as expected.

3.3 System Testing

After the implementation process has been carried out, the next phase is to do a series of tests on the system that has been built. These tests are intended to check the readiness of the system and by doing these tests, the writer is expected to find out which parts that still need to be developed and which ones that have not work as it should. Some of the tests that has been carried out are functionality test and time measurement test that will be explained in the subsections below.

3.3.1 Functionality Test

Functionality Test is a test where every feature on the website will be checked whether the features are already fulfil the parameters specified or not. In Table. 3.1, can be seen the expected results of each features along with the test result and assessment of the success of the feature.

Table 3.1 Functionality Test Parameters.

Features	Expected Result	Result After Test	Success (Yes/No)
Homepage	All the button are click-able and will direct users and admin to the relevant web page.	All the button are able to be clicked and automatically direct users and admin to the relevant web page.	Yes
Login Page	<ul style="list-style-type: none"> - Log In, Back to Homepage, and Sign Up button will direct users and admin to the relevant web page. - Log in activity with admin username and password will direct admin to Vehicle ID Input Page. - Log in activity with registered username and password will direct users to User Login Success Page. 	<ul style="list-style-type: none"> - All the button on the Log In page automatically direct users and admin to the relevant web page. - Logging in using admin username and password will be directed to Vehicle ID Input Page. - Logging in using registered username and password will be directed to User Login Success Page. 	Yes
User Login Success Page	Display vehicle data of user such as username, password, vehicle number, color, owner name, owner contact, violation time(s), and account register time.	Displayed vehicle data of user such as username, password, vehicle number, color, owner name, owner contact, violation time(s), and account register time	Yes
Vehicle ID Input Page	<ul style="list-style-type: none"> - Display a vehicle registration form. - Register button will direct admin to Vehicle ID Input Success Page. 	<ul style="list-style-type: none"> - Vehicle Registration Form displayed. - Register button automatically direct admin to Vehicle ID Input Success Page. 	Yes
Vehicle ID Input Success Page	Display the recent Vehicle ID input for confirmation.	Recent Vehicle ID input displayed.	Yes
Registration Page	<ul style="list-style-type: none"> - Display User Registration form. - Register button will direct users to Registration Success Page. 	<ul style="list-style-type: none"> - User Registration Form displayed. - Register button automatically direct users to Registration Success Page. 	Yes
Registration Success Page	<ul style="list-style-type: none"> - Display the recent user registration for confirmation. - Back to Homepage and Log In button will direct to the relevant web page. 	<ul style="list-style-type: none"> - Recent user registration displayed. - All button on the page automatically direct to the relevant web page. 	Yes
Vehicle Plate Data Delivery from Raspberry Pi	Vehicle Plate Number data sent to the database.	Vehicle Plate Number data succesfully sent to the database.	Yes
Notification email when the user has violated the regulations 3 times	Notification email sent to the user who violates 3 times	Notification email sent to the user who violates 3 times	Yes

3.3.2 Time Measurement Test

Time Measurement Test is a test to measure the time taken by the system to store data to the database. This test is intended to determine the average time to send data to the database. This test is carried out twice namely to measure the time needed to send user registration data to the database and also measure the time needed to send vehicle identification data. In this measurement, there is the possibility of human error due to measurements using stopwatch.

A. User Registration Data

As seen in Table. 3.2, there are ten trials carried out in order to measure the average time taken to store the registration data inputted by the user through the website to the Google Firebase Realtime Database. From ten trials carried out, all of the data sent was successfully stored in the database with average time of 1,541s.

Table 3.2 Vehicle Registration Time Measurement.

Experiment	Time Taken (s)	Success
1	1,54	Yes
2	1,51	Yes
3	1,53	Yes
4	1,48	Yes
5	1,55	Yes
6	1,50	Yes
7	1,53	Yes
8	1,58	Yes
9	1,65	Yes
10	1,54	Yes
Average Time		1,541

B. Vehicle Identification Data

In order to measure the average time taken to store the vehicle identification data inputted by the admin through the website to the Google Firebase Realtime Database, writer carried out ten trials as seen in Table. 4.3. All the data sent in the trials are successfully sent and stored in the database with average time of 1,965s.

Table 3.3 Vehicle ID Input Time Measurement.

Experiment	Time Taken (s)	Success
1	2,06	Yes
2	2,03	Yes
3	1,73	Yes
4	2,15	Yes
5	2,42	Yes
6	1,60	Yes
7	1,71	Yes
8	1,72	Yes
9	2,15	Yes
10	2,08	Yes
Average Time		1,965

4. Conclusion & Suggestion

4.1 Conclusion

From the system design, experimentation, testing, up to system implementation process, it can be concluded that:

- 1) All of the features on the website have worked properly after the functionality test as all the features have fulfilled the test parameters specified. Those features also already well connected to the database.
- 2) In the database, the data is separated in groups in order to ease the data entry to the database. On the website, all of the data will be retrieved for display on the web page.
- 3) The average time taken to store the user registration data to the database is 1,541s while the average time taken to store the vehicle identification data is 1,965s. The average time to store any data to database is 1,753s.
- 4) In order to create a reliable data collection, vehicle plate number text from Raspberry Pi has to match with the data in database otherwise there will be an error in data placement.
- 5) As known Firebase Realtime Database is a NoSQL-based database that store data as JSON objects, it is compatible for IoT system where the data is not relational. In this thesis, all of the data is relational with each other that makes the use of Firebase considered not suitable by the writer for this kind of system.

4.2 Suggestion

In order to improve the system, suggestions for further development of the system are as follows:

- 1) Experiments using other databases specially SQL-based database can be carried out to find which database is more suitable for the system.
- 2) A software is needed in order to measure the time taken to store the data to the database to decrease the human error possibility.

References:

- [1] "Data Jumlah Kendaraan Bermotor Menurut Jenis, 1949 - 2017." Badan Pusat Statistik, 2018, www.bps.go.id/linkTableDinamis/view/id/1133.
- [2] "Handbook of Research on Digital Content Management and Development in Modern Libraries." Handbook of Research on Digital Content Management and Development in Modern Libraries, by S. Thanuskodi, Information Science Reference, 2020. [3] Hanugra Aulia Sidharta, "Introduction to Open CV" BINUS Malang University. Indonesia, pp. 1, October 2017.
- [3] "Data and Databases." Information Systems for Business and Beyond, by David T. Bourgeois, Saylor Foundation, 2014.
- [4] Vasquez, Mike, et al. The Official (ISC)2 Guide to the CISSP CBK Reference. asdaWiley, 2019.
- [5] Husni, Nando Septian, and Muhamad Alam Syah. Membangun Ojek Online sadsMenggunakan Firebase. UDACODING, 2019.
- [6] "History and License of Python." History and License - Python 3.8.2 Documentation, Python Software Foundation, docs.python.org/3.8/license.html.
- [7] Sheikh, Tariq Hussain, and Naresh Kumar. "Internet and Web Technology". Booksclinic Publishing, 2020, Google Books.
- [8] Jurkowski, Odin L. Technology and the School Library: a Comprehensive Guide for Media Specialists and Other Educators. Rowman & Littlefield, 2017, Google Books.
- [9] Robbins, Jennifer Niederst. Learning Web Design a Beginner's Guide to HTML, asds CSS, JavaScript, and Web Graphics. O'Reilly, 2018.