



# Real-Time Web-based Dashboard using Firebase for Automated Object Detection Applied on Conveyor

Fadhillah Afira, Joni Welman Simatupang\*

Electrical Engineering Study Program, President University, Cikarang, West Java, 17530, Indonesia

\*Correspondence: [joniwsmt@president.ac.id](mailto:joniwsmt@president.ac.id)

SUBMITTED: 26 April 2023; REVISED: 5 June 2023; ACCEPTED: 7 June 2023

**ABSTRACT:** Conveyors are used by many factories in the industrial sector as tools to move some materials through various processes. Currently, it is necessary to have a device which is connected to a conveyor using a digital system. In this study, a conveyor is designed to use a webcam with a deep learning image classification system, Firebase real-time database, and a web-based dashboard. The webcam is used to capture and classify objects based on shape, color, and status, as well as counting objects that run on the conveyor. Firebase real-time database will receive and store data from the webcam system in real-time so that the data can be displayed on the dashboard. The dashboard used is a website-based design using two web development systems: front-end and back-end. Data displayed on the dashboard uses a real-time data table which is capable of displaying real-time data. Testing is conducted to analyze the performance of the full prototype. Testing methods used are One-by-one Object Test and Sequential Object Test, with total of 20 tests. One-by-one Object test is conducted five times, with a total of 168 data and a total time of 12 minutes and 15 seconds. Meanwhile, Sequential Object test is conducted 15 times, with a total of 546 data and a total time of 7 minutes and 19 seconds. Based on the observations of functional dashboard test, in fact all features and buttons on the dashboard are functioned well.

**KEYWORDS:** Conveyor; webcam; firebase; database; image classification

---

## 1. Introduction

Industry is one of the fields where raw materials are processed into finished materials by factories. To carry out these processes efficiently in terms of time, labor, and operational costs, factories require manpower, machines, and production lines. One commonly used production line in various types of factories is the conveyor [1]. Conveyors work by moving objects of certain weights from one end to the other, making it easier for workers to transport materials between different processes.

However, the use of conveyors in the industry is still relatively simple, primarily serving as a tool to move objects in large quantities with the same or varying loads through different processes [2, 3]. In this digital era, it is crucial for industry workers to adapt and incorporate internet-based applications into conveyors [4, 5].

Sensors are commonly applied to conveyors to detect and determine the suitability of objects. While using multiple sensors can be time-consuming, it is an effective method for data

collection. Alternatively, cameras can be used to replace sensors, as they can capture data from a wider range of objects [6]. Data retrieval from cameras requires additional programs, such as deep learning image classification methods. Deep learning is an algorithmic approach used for object classification, information processing, speech recognition, and object perception [7]. Furthermore, the obtained data needs to be processed, stored, and presented in an attractive manner using databases and dashboards. The database on the conveyor stores the data from the camera, which is usually short-lived until the next data comes in [8]. To visualize the data clearly, a dashboard is used [9].

The objective of this research is to facilitate image classification, data processing, and information display on conveyors. Three components are utilized: a webcam equipped with a deep learning image classification system, a real-time firebase database, and a web-based dashboard that displays data in real-time. The goal is to develop a web-based application that provides easy access to real-time data information.

## 2. Materials and Methods

This research work follows the classic life cycle method, consisting of five stages: planning stage, analysis stage, design stage, implementation stage, and test stage [8]. The planning and analysis stages were completed after conducting extensive literature studies. Subsequently, this research work proceeded with the design, implementation, and testing of the prototype, with the aim of developing an innovative product to achieve its ultimate goal [11]. In this section, the focus will be on the design stage and implementation stage. Figure 1 presents the block diagram of the entire system, starting with the setup of the webcam as the input. The webcam will automatically detect objects, recognizing four values: object data based on color, shape, object count, and a TRUE/FALSE value. The data will be processed, stored, and managed directly by the Firebase real-time database, which serves as the database system. The dashboard will display the data in a table format with six rows, and the number of columns will depend on the amount of data entered into the dashboard.



**Figure 1.** System block diagram.

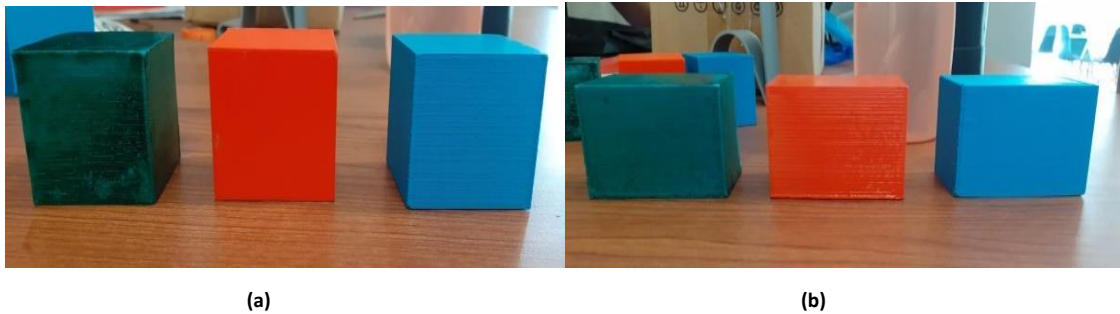
### 2.1. Conveyor setup

The design of the 3D conveyor body is created using wood, while the conveyor roller is 3D printed using PLA filament material with a 3D printer. A polyurethane conveyor belt is then installed based on the size of the conveyor and the motion of the rollers. Subsequently, the entire conveyor is assembled into a single unit and positioned on a legged table, as illustrated in Figure 2. Camera is positioned next to the conveyor at a distance of approximately 15 cm.



**Figure 2.** Conveyor and camera setup side view.

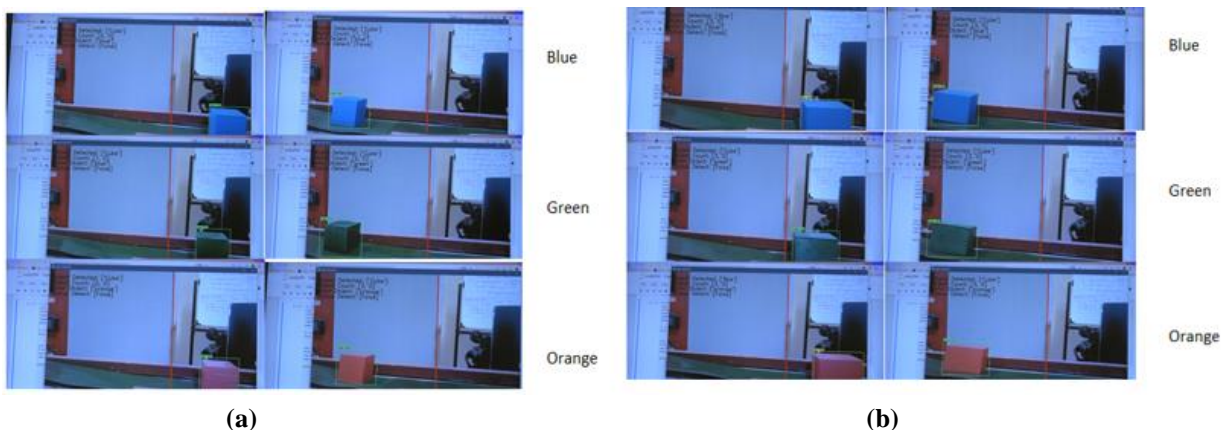
Objects were placed in order according to size and color. The objects used were dummy objects that had been printed with PLA filament material using a 3D Printer machine, as shown in Figure 3a and Figure 3b. There were a total of six dummy objects available in two shapes, Cube and Box, and three colors, Blue, Green, and Orange. The dummy objects were lightweight, so the belt did not need to be too tight for the roller conveyor.



**Figure 3.** Cube dummy objects front view (a) and box dummy objects front view (b).

## 2.2. Webcam system setup.

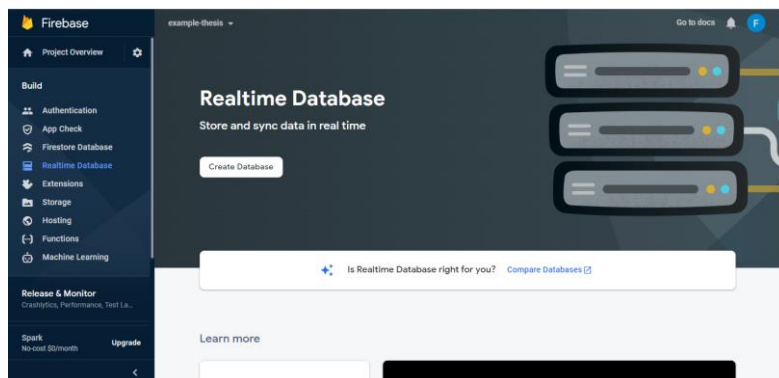
The next step was to set up the webcam and the detection system, which had been programmed using the deep learning image classification method. Once the system was ready to run, the program was tested by turning on the conveyor and placing objects on it. The objects moved slowly at a steady speed towards the camera. The camera began capturing the objects and sending data to the system, which then displayed the data in order. Figure 4a and Figure 4b illustrate the process of the camera capturing all six objects and the system displaying the data.



**Figure 4.** Camera test on cube objects (a) and box objects.

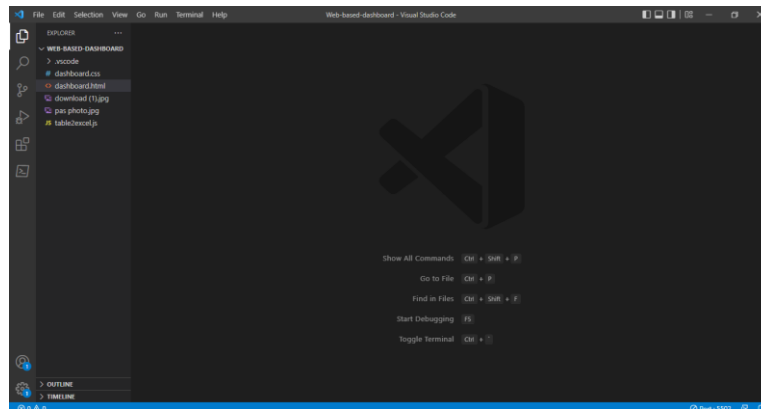
### 2.3. Firebase setup.

Data can be defined as the value of an attribute of an entity, and a Database can be defined as a collection of data entities. A database serves as an information system for storing, processing, and editing data for various purposes [12]. After setting up the conveyor and camera, the next step was to set up Firebase, a web-based application provided by Google. Every Google account had access to all the features of Firebase for free. Before accessing Firebase, the browser application was linked to a Google/Gmail account, automatically creating a Firebase account. Users were able to create a database by using the Create Database button. We utilized the Realtime Database feature, as shown in Figure 5.



**Figure 5.** Firebase realtime database main page.

### 2.4. Web-based dashboard setup.

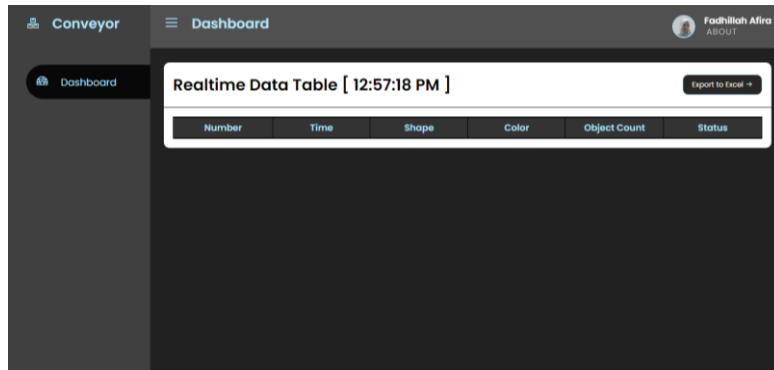


**Figure 6.** Main page on VS code.

In this project, a dashboard was built and designed using the HTML, CSS, and JavaScript programming languages. Hypertext Markup Language (HTML) was a text document used to display the entire website content to the browser [13]. CSS served as the foundation of the website, handling the visual appearance of web pages to make them attractive. CSS was easy to use and provided powerful control over the design of an HTML document [14]. JavaScript played a role in enabling client-side interactions with the user through the website [15]. The dashboard served as a form of data visualization, capable of displaying a large amount of information on one or more pages on a single screen [16]. The dashboard used was website-based, although it remained a private-user dashboard inaccessible freely on the internet. The website contained documents written in HTML format and accessed through the HTTP website server protocol. The pages of the website were accessed through a URL known as the

homepage [17]. The software used to build this dashboard was Visual Studio Code. In this project, a total of 5 files were utilized, including 1 .html file, 1 .css file, 1 .js file, and 2 .jpg photo files. Each file was linked to the others. In dashboard.html, all files were linked together to be displayed on one website page, as shown in Figure 6.

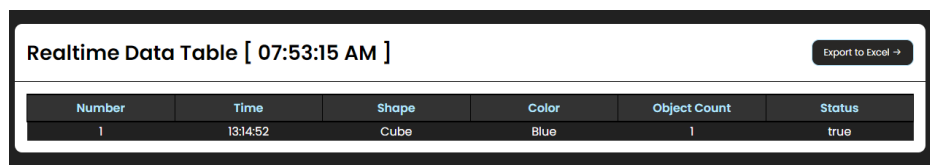
The entire dashboard design was successfully programmed, and the next step was to display the final result. In VS Code, HTML could be displayed using the Open with Live Server feature, which opened the HTML as a complete website in the computer's default browser application. The final dashboard view is shown in Figure 7.



**Figure 7.** Overall dashboard view.

### 2.5. Additional feature setup.

When data was entered and displayed on the dashboard, there was no guarantee that the data would be saved automatically when new data arrived. Therefore, it was necessary to have a feature where the completed data was displayed on the dashboard and then stored neatly in accordance with the table format on the dashboard. As a result, an additional feature was built using JavaScript to be added later to the main page of the dashboard. This feature was used to download the contents of the Realtime Data Table in .xlsx or Excel format. The first step involved creating an additional button on the table for downloading files. After the entire code was built, the final design of the button on the main dashboard page was shown in Figure 8.



**Figure 8.** Overall view of export to excel button.

## 3. Results and Discussion

A test was conducted on the performance of the camera system, Firebase Realtime, and Web-based Dashboard to determine if the system was functioning properly and identify areas for development in future projects. The test involved 6 objects with 2 shapes, Cube and Box, and 3 colors, Blue, Green, and Orange. The test was divided into 2 parts, totaling 20 tests. In the first test, the objects were placed one by one on the conveyor. Each subsequent object was placed after the previous object had moved down from the conveyor. There were a total of five system trials. In the second test, the objects were placed on the conveyor sequentially with a time gap between each placement. Each subsequent object was placed some time after the

previous object had been placed. This test was conducted five times, with three tests for each time. The final test evaluated the entire system and the features on the dashboard. This test, known as the Functional test, aimed to assess the performance of all features and determine if any further development was required. All features were evaluated based on their performance. The features included in the test were the Dashboard Homepage, Sidebar Menu, Header, and Realtime Data Table. The Dashboard Homepage was tested for overall performance, including the functionality of available buttons.

### 3.1. One-by-one object test.

As previously explained, this chapter will provide an overview of the data results obtained from the first test conducted using the one-by-one method. In this test, each incoming object moved until the previous object had moved down from the conveyor, and then new objects entered alternately. The order of objects that entered first was Cube Blue, followed by Cube Green, Cube Orange, Box Blue, Box Green, and Box Orange. The test was conducted 5 times, with a time gap of 5 seconds between each object's departure from the conveyor.

**Table 1.** One-by-one Object Test Results.

Test	Webcam	Total Data		Total Time
		Database	Dashboard	
<b>One-by-one Object Test</b>				
1	31	31	31	00.01.54
2	48	48	48	00.02.30
3	30	30	30	00.02.35
4	30	30	30	00.02.48
5	29	29	29	00.02.28
<b>Total Data</b>	168	168	168	0.12.15

In the first test, a total of 31 data were obtained by the webcam system for 6 objects, with detection time ranging from 13:14:48 to 13:16:42, as shown in Table 1. This test took a total time of 1 minute and 54 seconds. The average time to retrieve 1 data from 1 object was 4 seconds. After evaluating the webcam data, the test was then conducted on the 2 main systems in this project.

The second test yielded a total of 48 data from 6 objects. The amount of data in the second test was higher compared to the first test. This could be attributed to the effects of lighting, object motion, and the positioning of objects around the camera and conveyor. The webcam detected objects from 13:17:52 to 13:20:22. The total duration for object detection by the webcam in this test was 2 minutes and 30 seconds. The average time to produce 1 data per object in this second test was 3 seconds. The variations in total and average durations between each test were influenced by factors such as data quantity, conveyor speed, and the time spent by the webcam in object detection.

The third test took place from 13:21:23 to 13:23:28. Based on the duration, it can be calculated that the camera system took 3 minutes and 35 seconds to retrieve data from 6 objects. This duration was 5 seconds longer than the second test. With a total of 30 data obtained, the webcam required at least 5 seconds to retrieve 1 data from 1 object. This duration was longer compared to the first and second tests, despite obtaining fewer data. It can be concluded that the webcam sometimes takes a longer time to recognize objects. Additionally, the webcam system only counted 3 incoming objects, while all 6 objects were used.

In the fourth test, a total of 30 data were obtained, the same as the third test. The test started at 13:25:08 and ended at 13:27:56, with a total duration of 2 minutes and 48 seconds. The time to retrieve 1 data from 1 object in this test was 6 seconds. The total duration of this test was longer than the third test, despite having the same total data. Factors such as the webcam system requiring more time to collect data, the conditions around the camera, or the objects remaining on the conveyor for a longer period could have influenced the results.

The fifth test was the final test in the One-by-one Object Test series. It resulted in a total of 29 data, which was 1 data less than the fourth test. The test was conducted from 13:28:31 to 13:30:59, with a total duration of 2 minutes and 28 seconds. The time to retrieve 1 data from 1 object was 5 seconds, the same as in the third test but with 1 more data. Comparing the last test to the second and third tests, it had a faster total time despite having 1 data less. Thus, this last test was approximately 10-20 seconds faster with 1 less data.

Based on Table 1, it can be concluded that Firebase, as a database, was able to accommodate all data accurately without missing any. The performance of Firebase in receiving and storing data was reliable. Reviewing the entire table above, there were no differences observed between Webcam data, Firebase data, and Dashboard data. The dashboard performed well in this test, as depicted in Table 1, with no discrepancies found among the Webcam, Firebase, and Dashboard data.

**Table 2.** One-by-one object test specific results.

Test	One-by-one Object Test				
	1	2	3	4	5
	<b>Shape</b>				
Cube	28	44	27	26	25
Box	3	4	3	4	4
	<b>Color</b>				
Blue	12	20	11	7	9
Green	10	15	11	11	8
Orange	6	5	4	5	8
Other	3	8	4	7	4
	<b>Status</b>				
TRUE	3	5	2	2	3
FALSE	28	43	28	28	26
	<b>Object Count</b>				
Object Count	4	6	3	3	5

As it is shown in Table 2, in the Shape section of the first test, 10% of the data is a Box-shaped object. While 90% of the data is the Cube Object with a total of 28 data. This means that the webcam reads that the object has more Cubes than Boxes. Then, in the Color data section, the webcam has the ability to read more colors. 39% data of 31 are Blue Objects. Then, the Green object with total of 10 data and the orange object has a total of 6 data. In the color section, there are a total of 3 data obtained with details that the Webcam classifies objects as White objects. In the Status section, the highest percentage is the FALSE status with total is 28 data. While the TRUE status only gets 3 data. However, there are many objects that meet the 4 criteria, but the system reads that they did not meet the criteria. The last data is Object Count. The system is expected to read the entire object with a total of 6 objects. However, the system counts there are only 3 incoming objects.

The highest percentage in the Shape section of the second test is obtained by Cube with 92%. Data with the label Box only gets 4 data. Then in Color section, the highest percentage is Blue with 42% of the 48 total data. Green has 15 data and Orange has 5 data. The remaining 8 total data are from Other. In the Status section, FALSE got the highest percentage, 90%. The



last part is the Object Count with total count is 6 objects. It can be concluded that the level of data accuracy on objects is also influenced by the webcam system in classifying data.

For the third test, the total data of Cube objects are 27 data. While the Box object only has 3 data. If the data obtained is less than 3, it can be assumed that the system is difficult to read Box objects. In the Color section, the system reads objects with 4 other color data, which are 2 White and 2 Red. Then, in the Status section, the FALSE reached the highest percentage 93%. However, there are a lot of data that meets the four criteria.

In the fourth test, the highest percentage of the shape was regained by Cube objects with 87%. While the Box has only 4 data. Then, in the color section, the highest percentage was achieved by Green, which was 37%. However, it should be noted that the number of Other data is higher than Orange. The camera system reads 7 data on Other color, which are 5 White, 1 Black, and 1 Yellow. In the Status section, the FALSE value still achieved the highest percentage, which was 93% or 28 data. Meanwhile, the remaining 2 data are TRUE values. In the Object Count section, the camera system only counts a total of 3 objects from 6 objects that walk on the conveyor.

For the last test, the highest percentage in the Shape section was still achieved by Cube with 86%, lower than the previous test. However, there is an increase in the percentage and number in Box with 14% of 29 data, 4 data. It means there are 4 data that have been successfully read by the camera system as Box objects. However, this data is still relatively low considering the total existing data and the objects used. Blue, Green, and Orange each have a contiguous amount of data. Meanwhile, the Other color has only 4 data.

In the Status section, FALSE still excels as an object detection state, with 90%. While the value of TRUE only gets 3 data. Then, in the Object Count section, the camera system counts the number of objects that enter as many as 5 objects. There is a fairly good increase in results, considering that in previous tests, the camera system can only count up to 3 objects. In this last test, the camera system performed well in counting incoming objects.

Through the five tests above, it can be concluded that each test is conducted with the same sequence of objects and test methods can produce different data and amounts. Each amount of data in the five tests above is not much difference in value, except for the second test. A significant increase in the amount of data occurred until the data reached 48 total data. This can be affected by the performance of the camera system in recognizing and classifying objects, and the conditions around the conveyor and camera.

### 3.2. *Sequential object test.*

As previously explained, this chapter will explain the overall data results from the first test conducted using the Sequential Object Test. The first object will be placed on a conveyor and then followed by the next object at a different time for each test. So, the camera will capture 1-2 more objects in one camera capture.

There is a possibility that the data read more than the previous test because the objects run at almost close distances. This test will be conducted 5 times with the following time differences. The order of objects in this test is the same as in the previous test, starting with Cube Blue-Cube Green-Cube Orange, Box Blue, Box Green, and Box Orange.

In this Sequential Object Test, there is an interesting phenomenon to be discussed. In the last two tests of the 25-seconds Sequential Object Test, there was a very significant increase in the amount of data. As shown in Table 3, the amount of data of the last test with the



difference in the amount of data that is quite far compared to other test. These are also 3 tests with the highest total data results, amounting to 171 data.

**Table 3.** Sequential object test results.

Test	Total Data			Total Time
	Webcam	Database	Dashboard	
Sequential Object Test				
1st Test (Duration : 5 s)				
1	42	42	42	00.00.25
2	30	30	30	00.00.19
3	38	38	38	00.00.19
Total Data	110	110	110	0.01.03
2nd Test (Duration : 10 s)				
1	45	45	45	00.00.24
2	40	40	40	00.00.18
3	43	43	43	00.00.22
Total Data	128	128	128	0.01.04
3rd Test (Duration : 15 s)				
1	43	43	43	00.00.33
2	21	21	21	00.00.21
3	19	19	19	00.00.26
Total Data	83	83	83	0.01.20
4th Test (Duration : 20 s)				
1	22	22	22	00.00.29
2	16	16	16	00.00.30
3	16	16	16	00.00.31
Total Data	54	54	54	0.01.30
5th Test (Duration : 25 s)				
1	24	24	24	00.00.39
2	68	68	68	00.01.02
3	79	79	79	00.00.41
Total Data	171	171	171	0.02.22

In the first test with a total of 3 tests, the total number of data obtained was 110. The second test was 128. The third test was 83 and the fourth test was 54. From the first to the fourth test there was a decrease in the amount of data, but in 3 tests The last time there was a very high increase. In the fifth test of sequential object test, another object is detected by the camera. As long as the system is running to detect 6 objects, the camera also detects the others objects around the conveyor. This causes the data on the camera is working to produce the data many times more than the previous test.

The test cannot be stopped at that time because the system has logged into the database and has calculated all objects. If the test is stopped, the object will immediately jump to the next count. This is not in accordance with the purpose of making the system, which is to provide real-time and accurate data. However, it also proves that the camera system can detect more objects and capture more data in less time. The system is also very good when used in scanning systems that require fast time and accurate data.

The highest level of data accuracy is in the third test of the Sequential Object Test, which is 15 seconds time gap. Although there was a significant decrease in the amount of data in the second and third tests, the level of data accuracy in each object was of high value. It can be seen from the amount of incoming data which almost has the same amount for each object. While the data results for the Shape and Color categories did not show a significant difference in number, the amount of data between one and another is almost the same. This is caused by the time gap in this test being 15 seconds, so the distance between objects is ideal and the camera is able to detect all objects accurately compared to other tests.

There are a lot of data entered at the same time in the Sequential Object tests. In 1 second, the camera system is able to take 3 data at once. It can be assumed that the short

duration of data collection is due to the large number of data taken at the same time. This is influenced by the location of the object on the conveyor. Because of its sequential object order, the camera can capture 2-3 objects at once in 1 capture.

Overall data results from 3 systems, which are webcam, database, and dashboard can be seen in Table 3 to compare and match each data. The firebase realtime database is still running well even though there are 2-3 data entered at the same time. Even when 2 to 3 data come together, all the data is entered with a different ID number.

One of the drawbacks of firebase is that it is not able to receive dozens of data at the same time. Also, firebase realtime database is not yet capable of working with larger or larger data sets. Thus, this database is more suitable for use in projects such as this project. In the dashboard, there were no errors or data loss during the test. Dashboard are capable of receiving and displaying data to the main page in real-time. Even the table on the dashboard is able to display data from the database neatly and according to the type of data.

Besides, the hosting system cannot be used in this research work due to several errors while installing the hosting system from firebase using the command prompt. It is hoped that the hosting system on the dashboard will be developed, so that users outside this network can access the dashboard online.

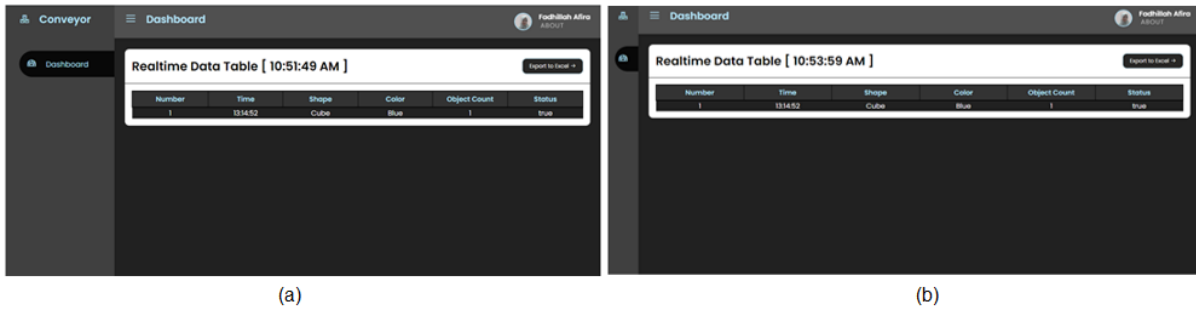
### 3.3 Functional dashboard test.

In Table 4, it can be explained that all the features on the dashboard work according to the command. Overall buttons and contents on the dashboard homepage are built according to the required features. The buttons on the homepage are clickable.

**Table 4.** Dashboard functional test results.

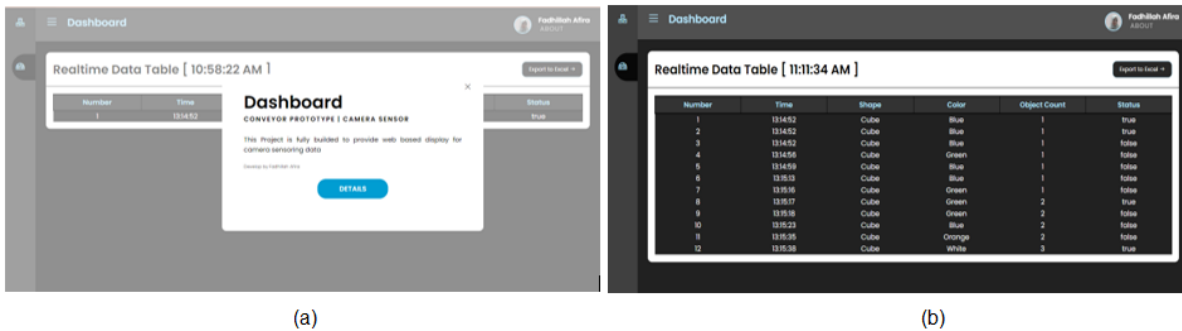
Name of the Feature	Description	Success (Yes/No)
<b>Dashboard Homepage</b>	All of the buttons are clickable and have each command	Yes
<b>Sidebar Menu</b>	To show menus available on the dashboard, such as Dashboard and Access Menu using the sidebar system	Yes
	The 'three-line' and 'About' buttons are clickable and display a sidebar menu	Yes
<b>Header</b>	The pop-up card has the title of the project, brief explanation, name of the project developer, and the 'Details' button	Yes
	Details' button connects to the Youtube link	Yes
	X' button to bring the page back to home	Yes
<b>Real-time Data Table</b>	Display overall data from Firebase in the table	Yes
<b>Additional Feature</b>	Export to Excel button is clickable to download the table into .xlsx format	Yes

In the Sidebar Menu, all buttons and content are in accordance with those in the previous program. There is only 1 available menu, so to move to another menu, it require additional menu as the program on the first menu. Sidebar Menu is flexible so it can be opened and closed. This feature also successfully opens and closes the menu so that when the menu is open, all the text and icons from the menu are visible and when closed, only the icons on the menu are visible. The remaining pages which are Header and Realtime Data Table are also flexible since it follows the movement of the Sidebar Menu as it is shown in Figure 9a and 9b.



**Figure 9.** Sidebar menu view when it is closed (a) and when it is open (b).

In the Header section, there is a page title and profile. In the title of the page, the text is large and in blue color, the text looks lit and matches the background color. The page title is not clickable, but the 3 lines next to it are clickable. The line is connected to the motion of the Sidebar Menu. In the profile section, there are names, photos, and About. About section is a clickable link that links to a pop-up menu containing a brief description of the project as shown in Figure 10a.



In the Realtime Data Table section, this table is built to display all data from the database. As shown in Figure 10b, the table successfully displays detailed data according to the rows and columns that have been programmed in the order of Number, Time, Shape, Color, Object Count, and Status. The last features is the additional features. In this feature, the Export to Button is used to download data on the Realtime Data Table in .xlsx format. This feature is successful in automatically downloading all the data entered on the dashboard. All data entered on the dashboard was successfully copied to the excel workbook as it is shown in Table 10b.

#### 4. Conclusions

After testing the database and dashboard, the entire project has been successfully conducted. Thorough One-by-one Object Test with a total of 5 tests and a time difference of 5 seconds for each data, the total number of data obtained from 5 tests is 168 data with a total duration of 12 minutes 15 seconds. This research work also conducted 15 Sequential Object Tests using 6 objects with each of the 5 tests performed 3 times. Overall tests are conducted with a length of time of 7 minutes 19 seconds. The resulting data amounted to 546 data. High data total is caused by the last 6 tests of the 20 second and 25 second test experiencing a very high increase in the number of data. All features and buttons on the dashboard run and work according to their functions. The total available features are 5 features and additional features with the shape of a button named “Export to Excel”.

## Acknowledgments

The authors deeply acknowledge The Research Institute and Community Service (RICS) of President University for supporting this work through the incentive scheme of publication.

## Competing Interest

The authors declare that there is no any financial or personal conflict of interest with others.

## References

- [1] Erinofiardi (2012). Analisa Kerja Belt Conveyor 5857-V Kapasitas 600 Ton/Jam. *Jurnal Rekayasa Mesin*, 3, 450-458.
- [2] Ferdoko, G. (2019). Implementation of Industry 4.0 in the belt conveyor transport. *MATEC Web of Conferences*, 263, 3-4. <https://doi.org/10.1051/mateconf/201926301001>.
- [3] Rospawan, A.; Simatupang, J.W.; Purnama, I. (2022). Development of Hot Air Dryer Conveyor for Automotive Tampo Printing Parts. *Green Intelligent Systems and Applications*, 2, 34-41. <https://doi.org/10.53623/gisa.v2i1.69>.
- [4] Ali, Z.H.; Ali, H.A.; Badawy, M.M. (2015). Internet of Things (IoT): Definitions, Challenges and. *International Journal of Computer Applications*, 128, 37-46. <https://doi.org/10.5120/ijca2015906430>.
- [5] Rospawan, A.; Simatupang, J.W.; Purnama, I. (2022). IOT Application for Conveyor Motor Load Current and Temperature Monitoring Device for Factory Acceptance Test in Industrial Application. *Jurnal ELTIKOM: Jurnal Teknik Elektro, Teknologi Informasi dan Komputer*, 6, 152-162. <http://doi.org/10.31961/eltikom.v6i2.553>.
- [6] Fadhil, A.T.; Abbar, K.A.; Qusay, A.M. (2020). Computer Vision-Based System for Classification and Sorting Color Objects. *The Fourth Postgraduate Engineering Conference*, 745, 012030. <http://doi.org/10.1088/1757-899X/745/1/012030>.
- [7] Aggarwal, V.; Gagandeep. (2018). A review:deep learning technique for image classification. *ACCENTS Transactions on Image Processing and Computer Vision*, 21-24. <http://doi.org/10.19101/TIPCV.2018.411003>.
- [8] Kraveva, R.; Kravev, V.; Sinyagina, N.; Koprinkova-Hristova, P.; Bocheva, N. (2018). Design and Analysis of a Relational Database for Behavioral Experiments Data Processing. *International Journal of Online Engineering*, 14, 117-129. <https://doi.org/10.3991/ijoe.v14i02.7988>.
- [9] Kumar, S.; Tiwar, P.; Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: a review. *Journal of Big Data*, 6, 111. <https://doi.org/10.1186/s40537-019-0268-2>.
- [10] 5 Metode Penelitian Rekayasa Perangkat Lunak. (accessed on 8 February 2022) Available online: <https://sites.google.com/a/student.unsika.ac.id/alex-mulyana-11070/assignments/5metodepenelitian/5-metode-penelitian-rekayasa-perangkat-lunak>.
- [11] Ramdhani, A.; Ramdhani, M.A.; Amin, A.S. (2014). Writing a Literature Review Research Paper: A step-by-step approach. *International Journal of Basics and Applied Sciences*, 47-55.
- [12] Berg, K.L.; Seymour, T.; Goel, R. (2013). History Of Databases. *International Journal of Management & Information Systems*, 17, 29-34. <https://doi.org/10.19030/ijmis.v17i1.7587>.
- [13] What's the Difference Between the Front-end and Backend. (accessed on 8 February 2022) Available online: [https://www.uobabylon.edu.iq/eprints/publication\\_4\\_27425\\_1402.pdf](https://www.uobabylon.edu.iq/eprints/publication_4_27425_1402.pdf)
- [14] CSS Cascading Style Sheets. (accessed on 21 January 2022) Available online: [https://www.tutorialspoint.com/css/css\\_tutorial.pdf](https://www.tutorialspoint.com/css/css_tutorial.pdf).
- [15] JavaScript javascript language. (accessed on 21 January 2022) Available online: [https://www.tutorialspoint.com/javascript/javascript\\_tutorial.pdf](https://www.tutorialspoint.com/javascript/javascript_tutorial.pdf).

- [16] Sadiku, M.N.; Shadare, A.E.; Musa, S.M.; Akujuobi, C.M. (2016). Data Visualization. *International Journal of Engineering Research and Advanced Technology*, 2, 11-14.
- [17] Sihombing, W.W.; Aryadita, H.; Rusdianto, D.S. (2018). Perancangan Dashboard untuk Monitoring dan Evaluasi. *Journal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3, 435-436.



© 2023 by the authors. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).