Proposal - Bangkit 2023 Company-based Capstone

Company : Wowrack Indonesia

Problem Code : C-23 IT

Team Member :

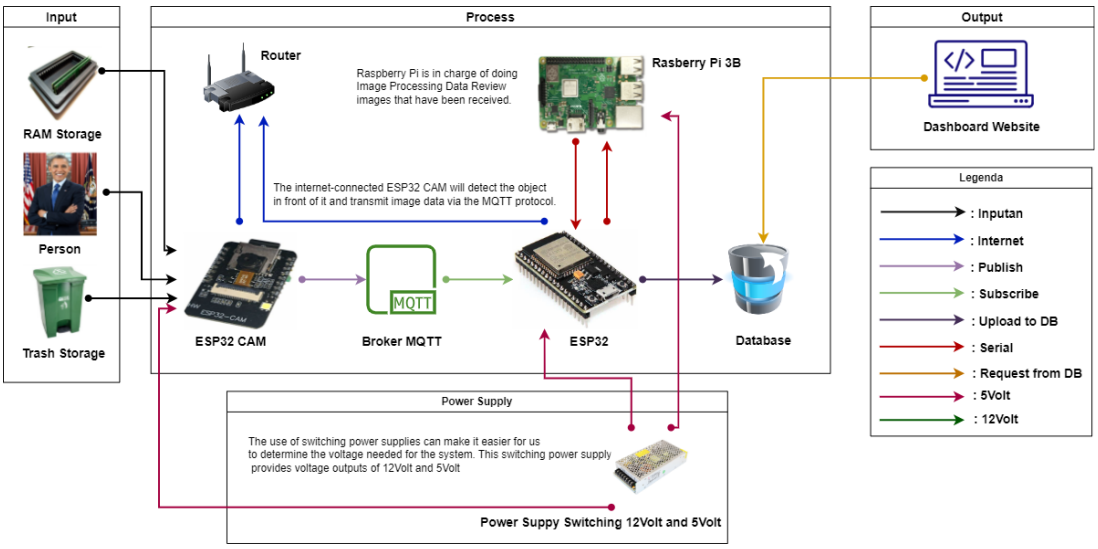
- **(ML) M261DKX4361- Arif Hidayat**
- **(ML) M151DSX2543 - Asnan Sabil**
- **(ML) M062DKX3801 - Iis Ismail**
- **(CC) C172DSX2880 - Luthfi Fadhlurrohman**

Why is this problem/project interesting for your team?

The use of cameras as replacements for sensors in several cases in the world of IOT automation is very popular and our team has found a solution to work on this project.

What's your group's initial idea to work on this project?

The idea that our team offers for making a capstone base company fully follows the instructions that were previously informed when the Wowrack company made a presentation, but there are several additional components and features that we offer for this capstone product. We use ESP32CAM, ESP32, Rasberry Pi 3B, 12 Volt and 5 Volt Switching Power Supply, MySQL database, MQTT Protocol, and output visualization through the website dashboard. so that the design we want to make easier to understand we attach a block diagram as follows:



Proposal - Bangkit 2023 Company-based Capstone

There are 3 main cases offered by the Wowrack company, namely detection of the amount of RAM in the RAM storage tray, detection of volume from the trash can, and facial recognition with the provision that facial data will be updated following changes that occur in the face of the person trying to be recognized. We integrate these three cases in one piece of hardware, but for algorithms and machine learning modeling, we will differentiate according to existing needs. The explanation of each process is attached in the block diagram that we have created as follows:

1. Block Input

The input needed for the capstone base company is in accordance with what has been explained by the Wowrack company, namely an image of the RAM storage tray, person, and trash can. These three inputs will be detected by ESP32CAM and then ESP32CAM will take one of the three input images according to the needs of the analysis.

2. Block Process

In this process block, we place several important components and interact with each other as follows: ESP32 CAM which is already connected to the internet network, and the MQTT protocol with the broker provided will detect the object in front of it and then take a picture of the object using the camera module, then ESP32 CAM serves as a publisher in the communication protocol using this MQTT. MQTT cannot send data in JPEG, PNG, or BMP format. Therefore, before sending the ESP32CAM, it must convert the captured image data into a byte array format. After that, the byte array format needs to be converted into JSON format according to the instructions. image data that is already in the form of JSON can be sent by ESP32CAM via the MQTT protocol to the Broker with the topic previously provided

Image data in JSON format that has been sent by ESP32CAM via the MQTT protocol to the available broker will be retrieved by ESP32 which is already connected to the internet network and the MQTT protocol with the same topic as ESP32CAM. This ESP32 serves as a subscriber in this data communication process, data that has been taken by ESP32 will be forwarded using serial communication to the Raspberry Pi 3B. JSON data that has been received by the Raspberry Pi 3B will be converted to an image data format so that the image data can be analyzed by the algorithms and modeling that our team has created before. After the image has been analyzed, the image data will be sent to the MySQL database using ESP32 which is connected to the internet network.

3. Blok Output Process

Image data that has been analyzed will be stored in the MySQL database with the attributes timestamp, image, and result. The attribute that will be created needs to be adjusted to the type of object to be detected, namely the amount of RAM in the RAM storage tray, the volume of garbage, and face detection. then the data will be displayed on the website dashboard that our team has previously created to make it easier to use and monitor objects.

Does your team have unique solutions to be proposed?

Our group has several Unix solutions and features to offer to address existing problems, including the following:

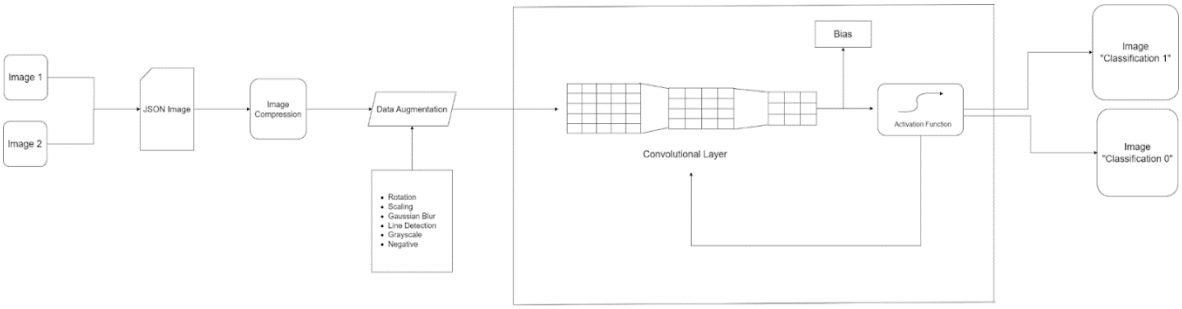
1. **Hardware**

   Seeing the company's need to solve existing problems, our group offers to use the IOT automation system attached as shown in the block diagram above, but if the company allows it to replace some of the hardware that has been a requirement before we want to replace the ESP32 CAM with a webcam and eliminate the MQTT protocol directly uses the Raspberry pi 3B hardware which is connected to a webcam camera and connected serially with ESP32 to connect to the internet network and database, in our opinion changing the existing composition of the IOT automation system hardware will make the system more efficient. We realize that the limited image quality that can be taken by the ESP32CAM can be annoying if the quality of light around the object is reduced and the use of MQTT for sending real-time video files is also limited and this can burden the performance of the ESP32CAM as a detector and publisher. Sending image data needs to be converted first to a byte array format and then converted again to the JSON type according to the provisions, this complicated communication network can be replaced by using a webcam camera which directly sends image or video data to raspberry pi then with the open cv library, image data can be directly analyzed and produce output as desired, but if the company does not allow it, we are ready and will comply with the existing regulations and work on this capstone wholeheartedly.

Proposal - Bangkit 2023 Company-based Capstone

2. Image Processing

In processing images that have been captured and transmitted by the system, we have an image processing plan as follows:



This process diagram is about processing and transforming images using augmentation techniques, then processing images through a Convolutional Neural Network (CNN) to perform binary image classification. The following is a description for each stage in the process diagram:

1. The process of reading the JSON data format: At this stage, it is necessary to make sure that the original image is in a byte array format and has been transmitted and converted to the JSON format so that it can be processed by the computer. This format allows us to read, manipulate, and change images using computer code.
2. Process JSON Converted Into Image: After the image is converted into JSON format, the next step is to return the image to its original format. This can be done using program code that extracts the required information from the JSON file and converts it back into a processable image format.
3. Image Augmentation: This stage involves changing several features in the image such as rotation, magnification, Gaussian blur, negative, grayscale, and others. Image augmentation can increase the amount of training data, as well as help improves the ability of the CNN model to recognize patterns and features in images.
4. CNN Process: At this stage, the modified and enhanced images will be processed through a Convolutional Neural Network (CNN) which has many layers, such as the convolutional layer, pooling layer, and normalization layer. CNN will study the patterns and features in the image to

produce the desired final result. CNN uses the TensorFlow library to do the processing from start to finish

5. Activation Function: The Activation function is used on each neuron in the CNN to add non-linearity to the model. This allows CNN to study more complex patterns and features in images.
6. Bias: Bias is used as part of each neuron in CNN. This helps improve CNN's ability to better learn patterns and features in images.
7. Binary Image Classification: The final stage in this process diagram is descriptive image classification. After the image is processed through CNN, the model will provide results that explain what is contained in the image. These results can be used for various purposes, such as recognizing objects in images or classifying images into certain categories.

Libraries used for image processing include the following:

1. TensorFlow

   Tensorflow is a deep learning framework that is often used to implement Convolutional Neural Networks (CNN). At the CNN Process stage in the process diagram above, TensorFlow can be used to easily and efficiently build and train CNN models.

   TensorFlow provides many useful features in building and training CNN models, such as optimizers, loss functions, and activation functions which can be selected as needed. In addition, TensorFlow also provides many built-in CNN layers that can be used to speed up model building. At the CNN Process stage, TensorFlow can be used to build the CNN model architecture easily and quickly using the TensorFlow Keras API. After the model is built, TensorFlow can be used to train the model using the backpropagation algorithm by optimizing a predetermined loss function.

2. OpenCV

   OpenCV (Open Source Computer Vision) is a library that is often used in image processing. At the image data capture stage, OpenCV can be used to read images from various image file formats, such as .jpg, .png, .bmp, and others.

3. PIL

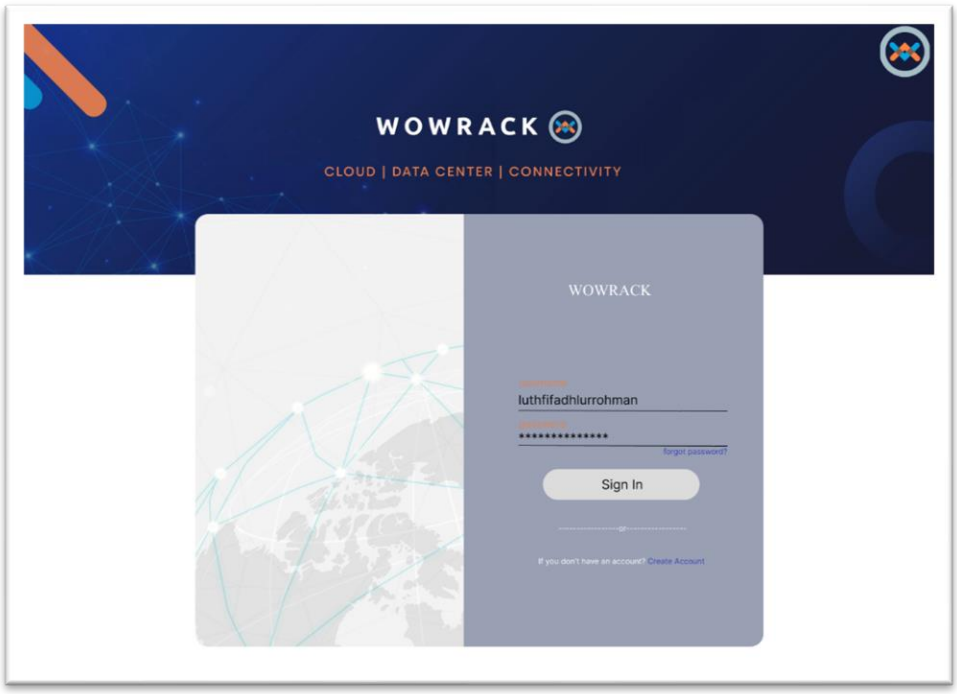
PIL (Python Imaging Library) is a Python library used to process and manipulate images. This library provides various functions for opening, saving, and manipulating images in various formats, such as JPEG, PNG, GIF, BMP, and many more.

PIL allows users to perform various actions on images, such as resizing, cropping, changing color, adjusting brightness, converting formats, and so on. In addition, PIL can also be used to create images from scratch.

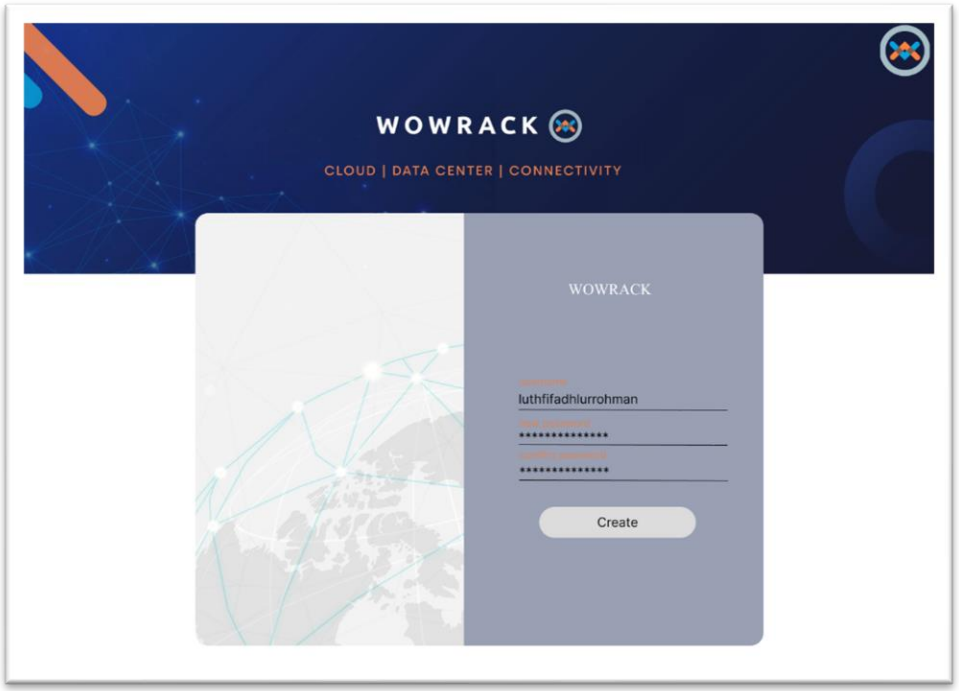
3. Dashboard

This dashboard is made on a website basis which will be intended to be able to visualize IoT results. IoT results will be stored in a database. This dashboard will later retrieve data from the database and IoT results will be displayed on the Website-based Dashboard. This website will later have features in the form of account registration, login, and history of the visualizations made. So that with this feature, data generated from IoT can be grouped according to needs. The advantage of our idea is that to display the results of the analysis, it will adjust to the modeling that was previously set according to the needs. The following is an example of the dashboard display that we will create later for this capstone:

Proposal - Bangkit 2023 Company-based Capstone



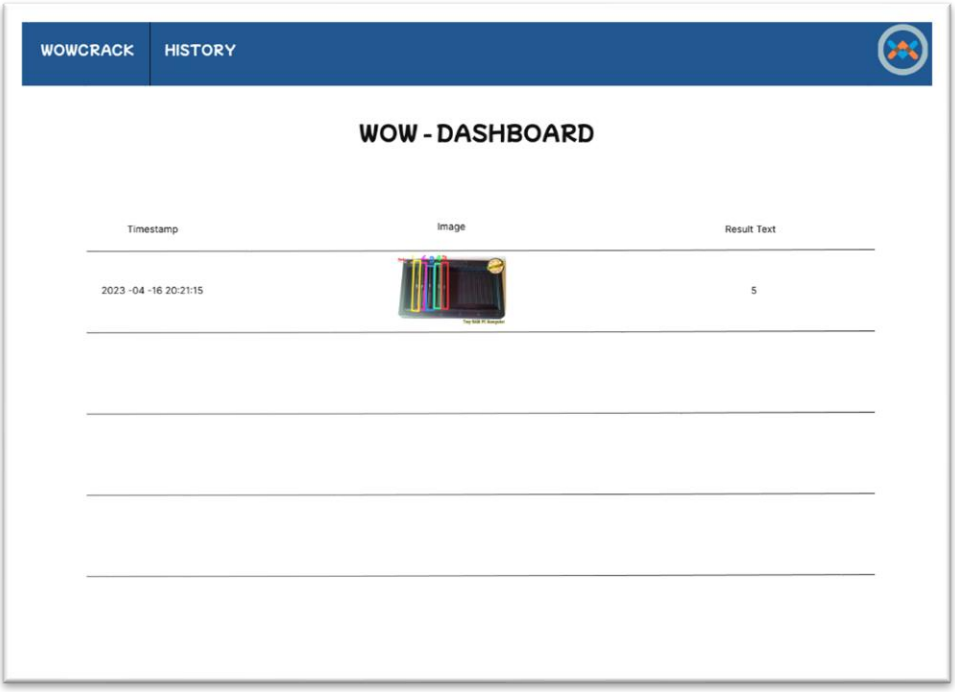
Picture 1 Screen Login



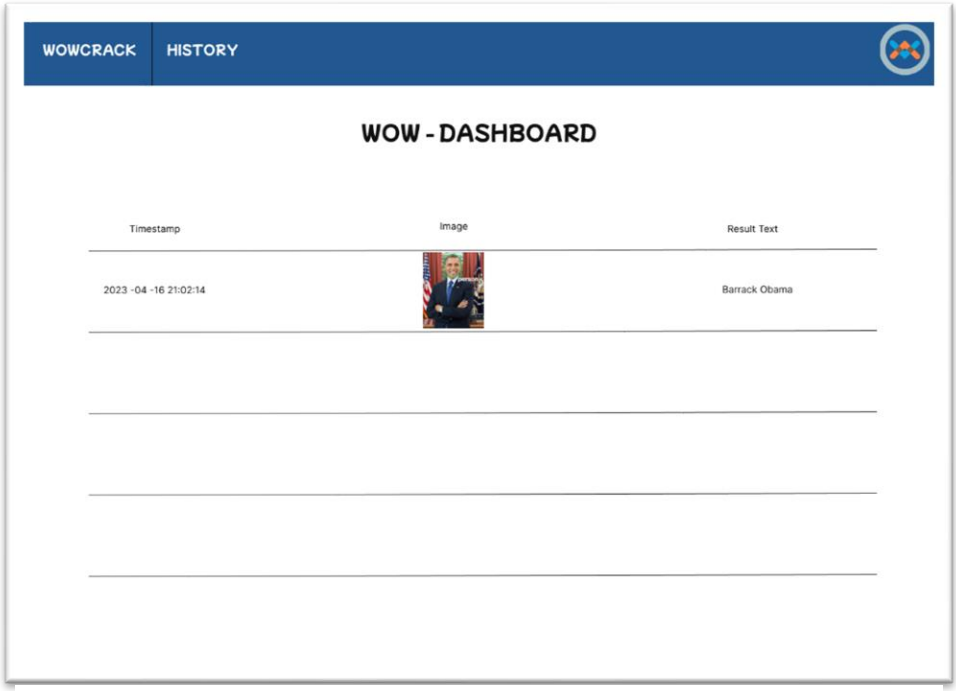
Picture 2 Screen Create Account

Proposal - Bangkit 2023 Company-based Capstone

Note: The dashboard display will adjust to the case provided by the company, for example as follows

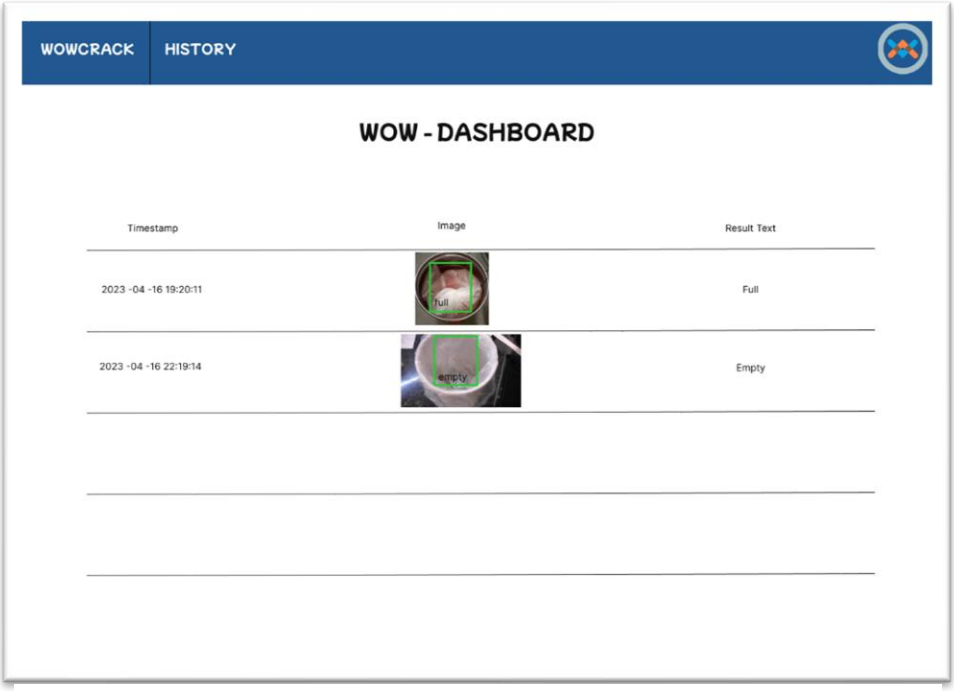


Picture 3 Dashboard for Detection RAM Tray



Picture 4 Dashboard for Detection Person

Proposal - Bangkit 2023 Company-based Capstone



Picture 5 Dashboard for Dectection volume Trash

The dashboard display will adjust to the case provided by the company, for example as follows

Based on your knowledge, what tools/IDE/Library will your team use to solve the problem?

1. ESP32 CAM
    - Hardware EP32CAM
    - Arduino IDE
    - Library MQTT
2. ESP32
    - Arduino IDE
    - WiFi Library
    - Library SoftwareSerial
3. MQTT
    - IP Adress
    - Broker
    - Topic Name

- Publisher
- Subscriber

4. Proses Analisis Image Processing on Raspberry Pi 3B
    - Pyserial
    - Raspberry Pi or Personal Laptop for System Subscriber
    - Visual Studio Code
    - Library OpenCV
    - Library TensorFlow
    - Library PIL
    - Library Matplotlib
    - Library Numpy
5. Storage
    - PHPMyAdmin - MySQL
6. Visualisasi Data atau Dashboard Website
    - Dashboard Hiperaktif
    - Bootstrap 5
    - ReactJS
    - HTML
    - Laravel
    - Figma
    - CSS
    - API

Based on your knowledge and explorations, what will your team need support for?

1. Mentors
2. Hardware availability whenever possible
3. Flexible work time
4. Dataset object from company
5. Hosting with PHPMyAdmin

Proposal - Bangkit 2023 Company-based Capstone

Any other notes/remarks we should consider on your team's application.

Our group will create an IOT automation system that will fully assist and solve existing problems in the Wowrack company more efficiently by offering data visualization features through a dashboard that our group has designed to make it easier for users.