# CHAPTER 4
# ANALYSIS AND DESIGN

## 4.1. System Design

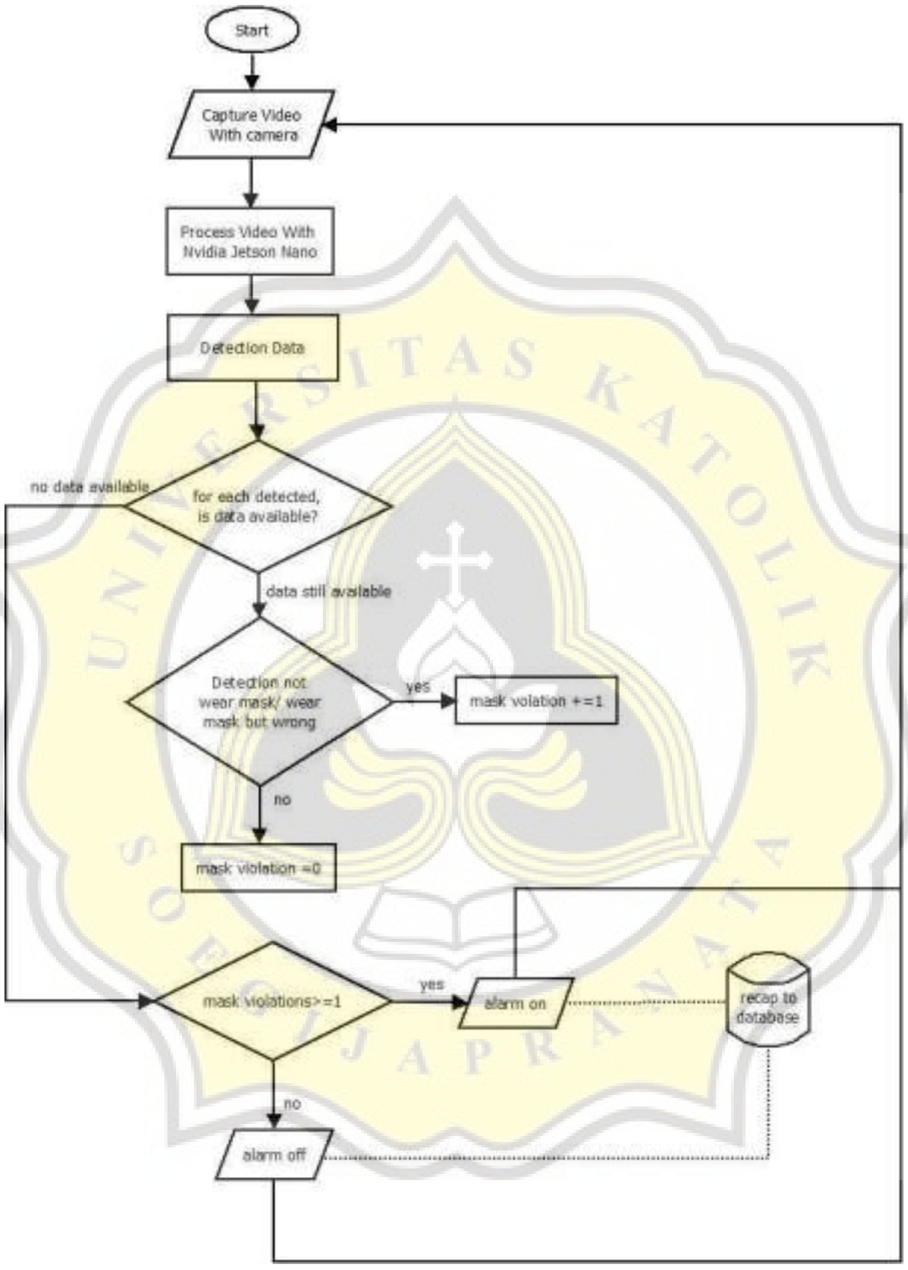


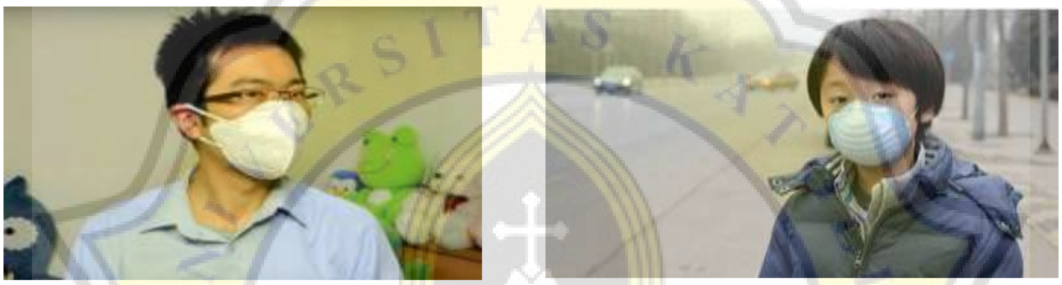
**Figure 4.1** Shows System Design

Above shows the flow chart of the system design showed on Figure 4.1 .When the system starts, the video will be captured by the camera. After that, the image will be processed with Nvidia Jetson-Nano. After detecting, initially, the data is still = 0, but after the system captures the image of the mask violation, whether it's not using a mask or using a mask but wrong, the data will increase +=1. If the data is not equal to 0, then the alarm will fire and enter the data in the system database.

## 4.2. Dataset

The dataset obtained from the Kaggle web was used as the system model training data. This dataset is divided into three classes, namely using a mask properly , not using a mask & using a mask but wrong . Below is example of data using mask properly showed on Figure 4.2(a) and 4.2(b), data of not using a mask showed on Figure 4.2(c) and 4.2(d), data of using mask but wrong showed on Figure 4.2(e) and 4.2(f).

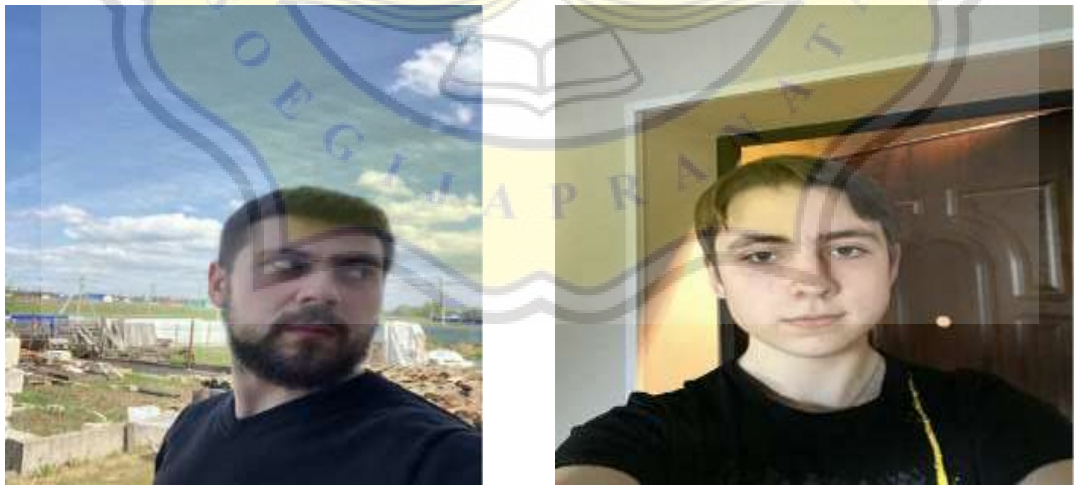
- Using Masks Properly:



(a) (b)

**Figure 4.2**(a) Example of Using Mask Properly 1 (b) and Using Mask Properly 2

- Not Using a Mask



(c) (d)

**Figure 4.2**(c) Example of  Not Using a Mask 1 (d) and Using Mask Properly 2

- using a mask but wrong

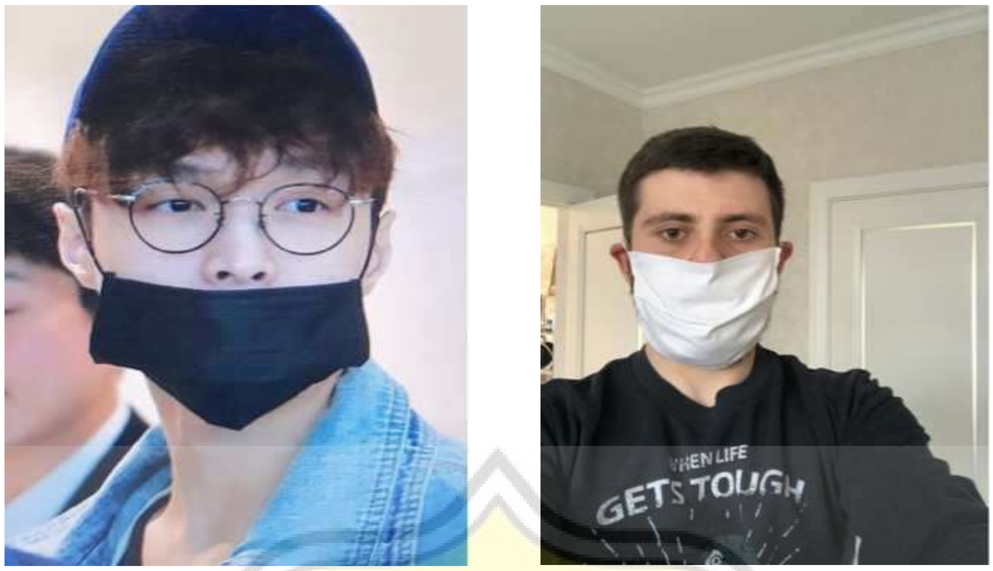

(e) (f)

**Figure 4.2**(e) Example of Using Mask but Wrong 1 and (f) Using Mask but Wrong 2

## 4.3. Pre-Processing Data

Before the image will be processed by the system, the image will be labeled. The first thing to do is convert the xml data to csv. Next, the label map file recap the classes using generate_labelmap.py and the output is label_map.pbtxt. And the last processed to tf record file using generate_tfrecord.py. The tf record is a step for converting data to a binary file so that the stored data is not large which affects the training step. Below is example of Image labeling showed in Figure 4.3.

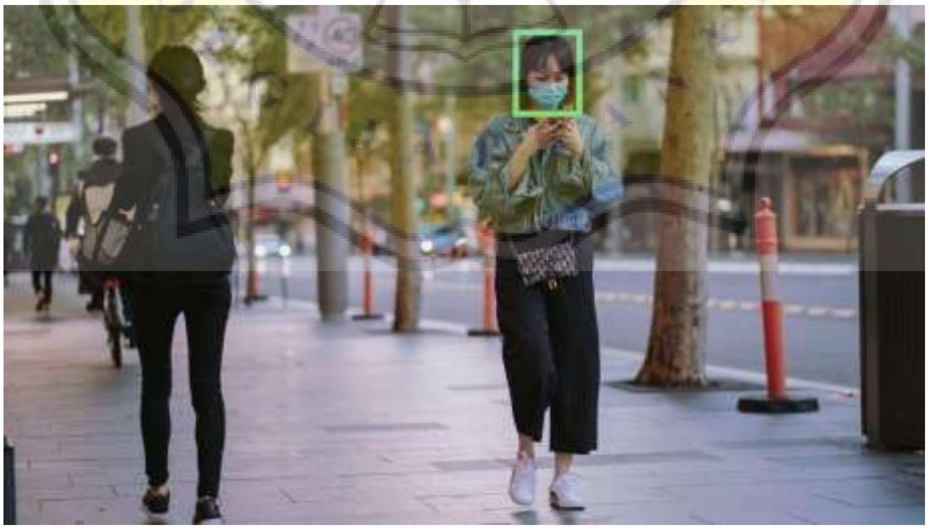


**Figure 4.3** Shows Image Labeling

## 4.4. Software

The software used in the system is Single Shot Multibox Detector SSD algorithm. SSD is a single deep neural network algorithm that applies the bounding boxes feature to estimate the location of the detected object. The SSD model used in this experiment is MobileNet V2 320x320 SSD. Below is SSD Architecture show in Figure 4.4:

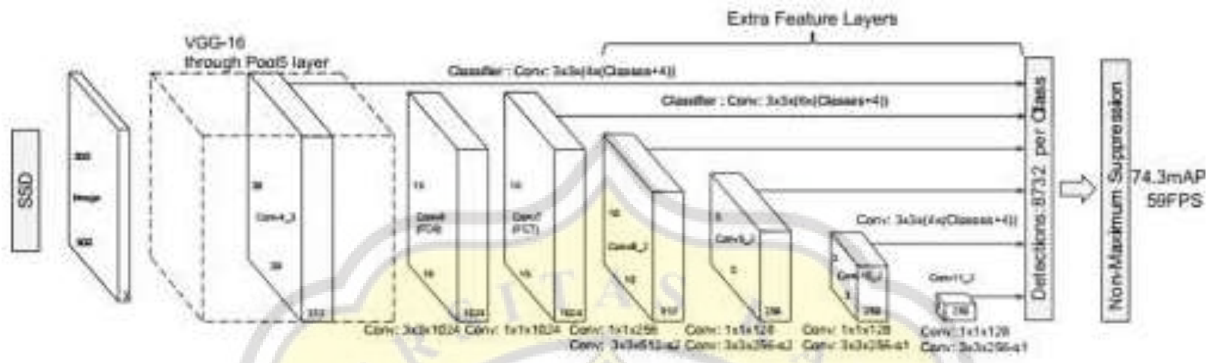


**Figure 4.4** Shows SSD Architecture [11]

In the picture can be seen that the SSD has 2 network architectures, namely the VGG-16 network and the network at the extra feature layer. The VGG-16 network is used as the base network because VGG-16 has a strong performance for high-quality images. Additionally, SSD adds a convolutional feature to the end of the base network that predicts different aspect ratios. SSD is used as an object detection process and then transfer learning is carried out. Transfer learning is a method that results in previously desired knowledge that will be needed again to get the desired results.

The deep learning model used in this research is Tensorflow. Tensorflow is a library capable of performing numerical computations and large-scale machine learning projects. The model used in TensorFlow in this study is the garden model. Tensorflow model garden is a repository with several different state-of-the-art (SOTA) model implementations and modeling solutions for TensorFlow users.

## 4.5. Hardware

The hardware used in this system is the Nvidia Jetson Nano. The Nvidia Jetson Nano is an Artificial Intelligence (AI) development kit that can be used to run a variety of modern AI loads with amazing performance. With the Nvidia Jetson Nano, systems can run AI frameworks and models for image recognition, object detection, segmentation, speech recognition, and more. In addition to the Nvidia Jetson Nano, it also uses a camera module. This camera functions to capture images that will be input to the Nvidia Jetson Nano. Micro SD was added so that the captured images have a high resolution. Adding a cooling fan to flow air into the engine through the grille on the radiator. Add hdmi video capture to record video and audio by connecting an HDMI cable from a video device to a computer to preview or save.  It also adds a wifi adapter to capture wifi signals. These tools are put together in such a way as to form a unified whole. Below is Design of Hardware showed on Figure 4.5:

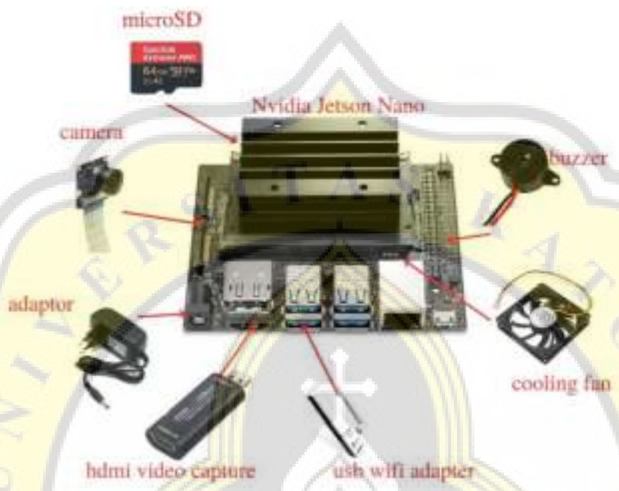


**Figure 4.5** Shows Design of Hardware

## 4.6. Testing

Use pictures in the dataset and real conditions. The results will be calculated for accuracy, precision, and recall. Accuracy is the ratio of correct predictions (positive and negative) to the entire data. Accuracy will be calculated by the formula

$$\frac{TP + TN}{TP + TN + FP + FN} \quad (1).$$

. Precision Is the ratio of positive correct predictions compared to the overall positive predicted outcome. Precision will be calculated by the formula

$$\frac{TP}{TP + FP} \quad (2).$$

Recall Is the ratio of predictions that are positive correctly compared to all data that is positive correctly. The recall will be calculated by the formula

$$\frac{TP}{TP + FN} \quad (3).$$

TP is positive data that is detected correctly. TN is the amount of negative data that is detected correctly. FP is negative data but detected as positive data. FN is positive data but detected as negative data. Below is an explanation table based on mask violations showed on Table 4.1. :

**Table 4.1.** Description of Values

| Values | Description |
|---|---|
| TP | when a mask violation is detected as a mask violation |
| FP | when a mask violation is detected as a non-mask violation |
| TN | when a non-mask violation is detected as a non-mask violation |
| FN | when a mask violation is detected as a non-mask violation. |