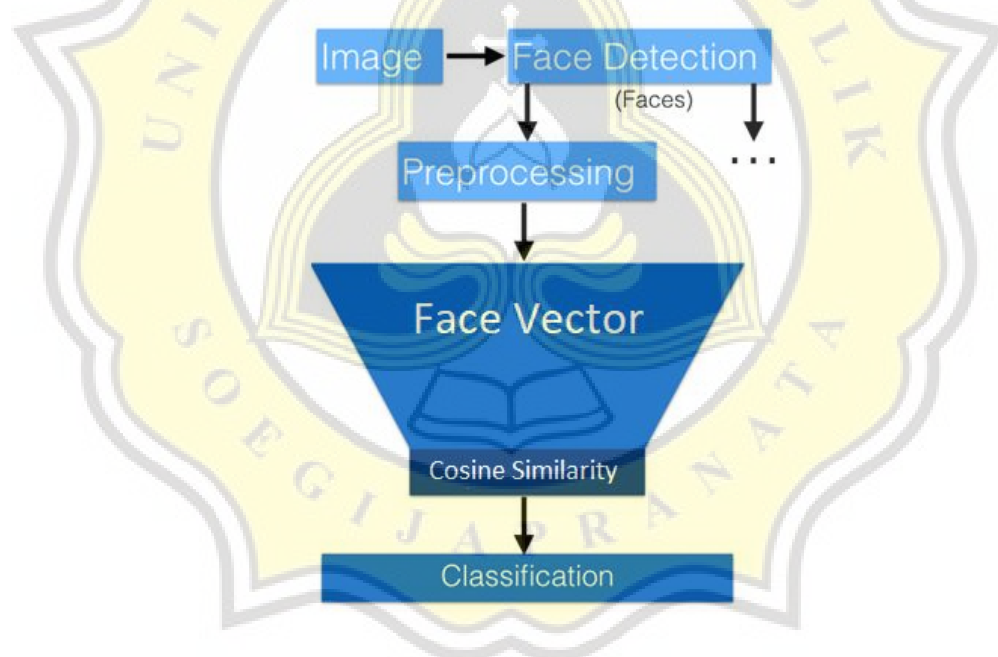


## CHAPTER 3

### RESEARCH METHODOLOGY

The steps that will be taken to complete this project are described in this section. Face recognition is a method that combines face detection, preprocessing, and classification systems. The face that will be used as training will be extracted from an image using Openface. This method is called the preprocessing process, the purpose of this process is that the face can be represented as a vector so it can be read by a computer system. After that, the vector formed from the training data will be measured for its similarity to the testing data using cosine similarity. The smaller the result, the more similar the data.



*Figure 3. 1 How Face Recognition works*

In working on this project, there are many things that need to be prepared. This is the step that prepared to doing this project :

### 3.1 Collecting Dataset

In this research uses “Celebrity Face Image Dataset” which is contains 17 person with many different poses each person for testing the Accuracy of OpenFace. This dataset downloaded from Kaggle.

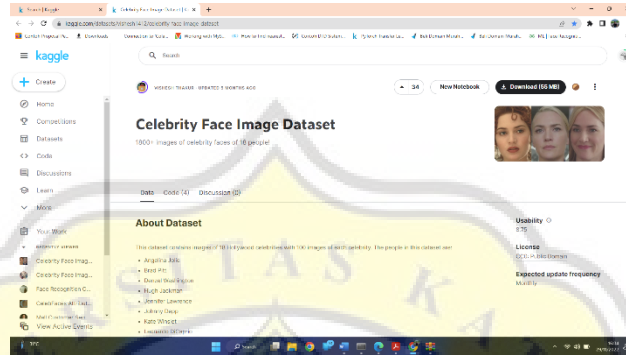


Figure 3. 2 Dataset from Kaggle

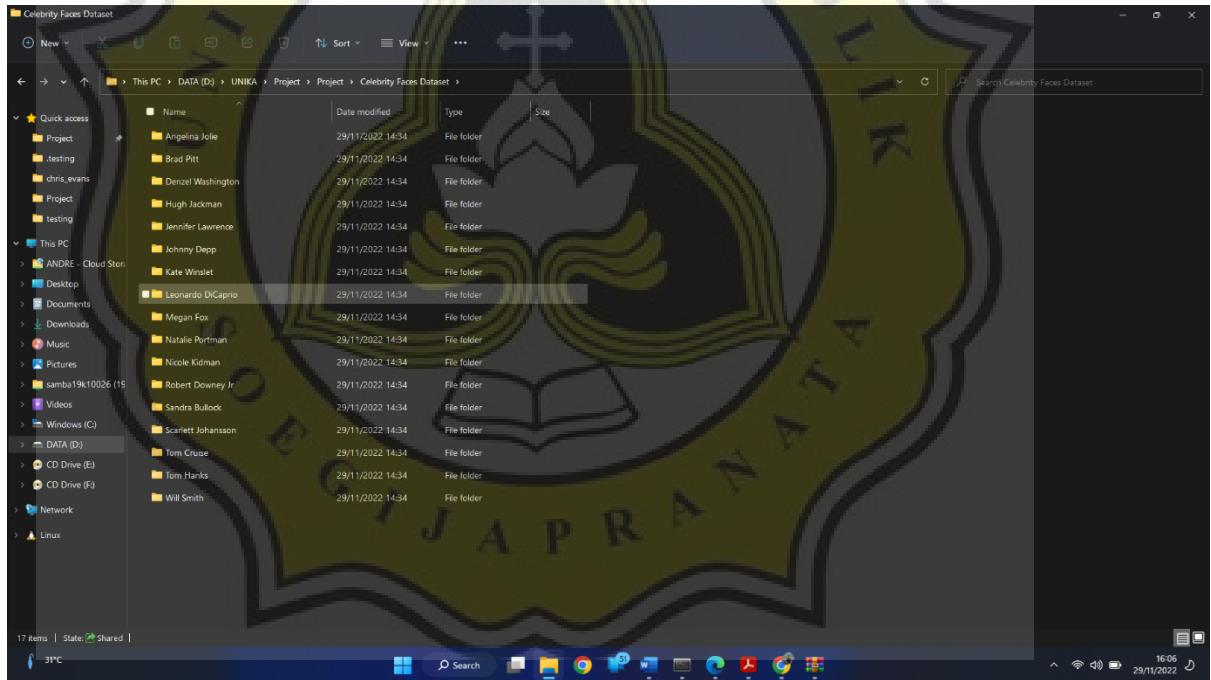
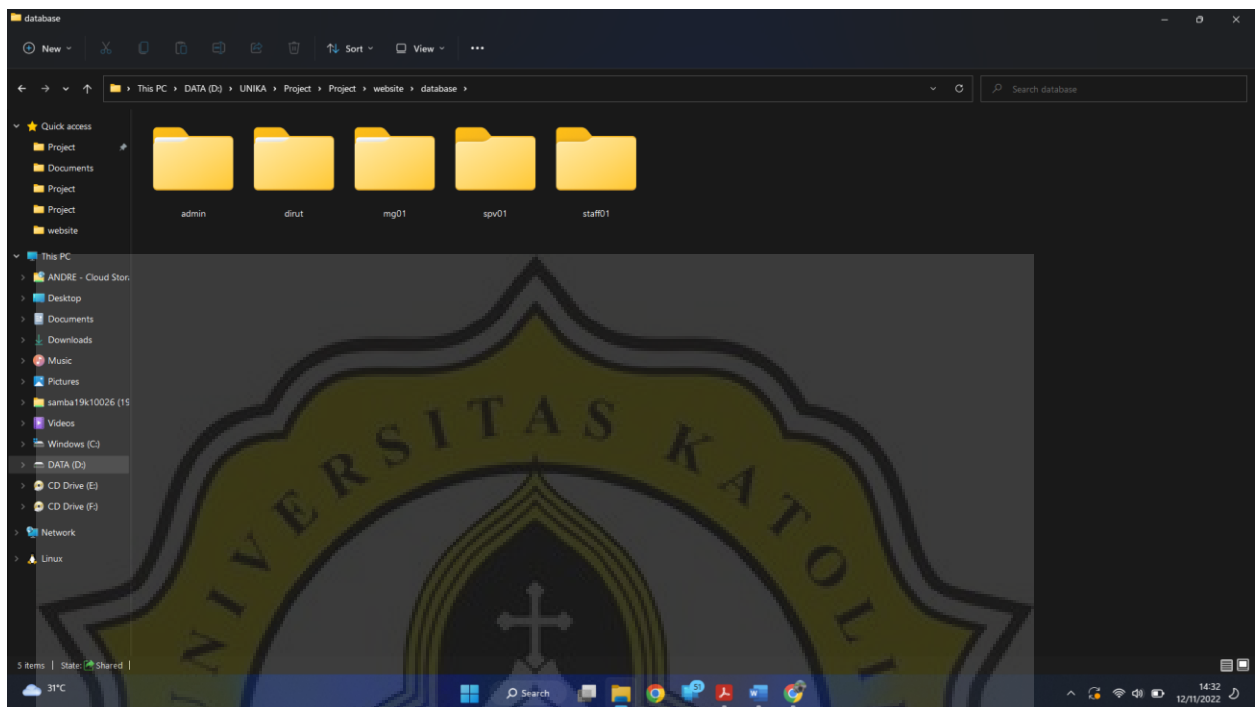


Figure 3. 3 Downloaded “Avengers Dataset”

While for testing on the attendance website, I collected facial training face from my friends and family which is each folder / username contains 5 images face with different poses.



*Figure 3. 4 Dataset for Attendance Website*

### **3.2 Extracting Face from an Image**

In an image, there are various kinds of objects, such as faces, backgrounds, etc. This process jobs are gets rid of unnecessary objects and then represents them as a vector. In this research, I used Haarcascade to do face detection job and OpenFace algorithm to do feature extraction process. After the face extraction process , 128D vector of face embedding are produced. The 128D face embedding vector that used for training dataset are save into .npy file.

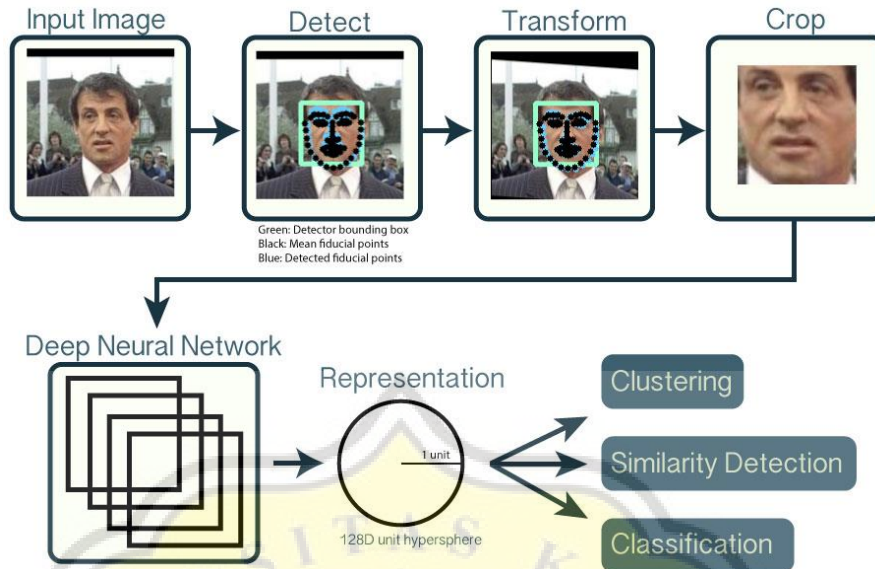


Figure 3. 5 Openface Algorithm [11]

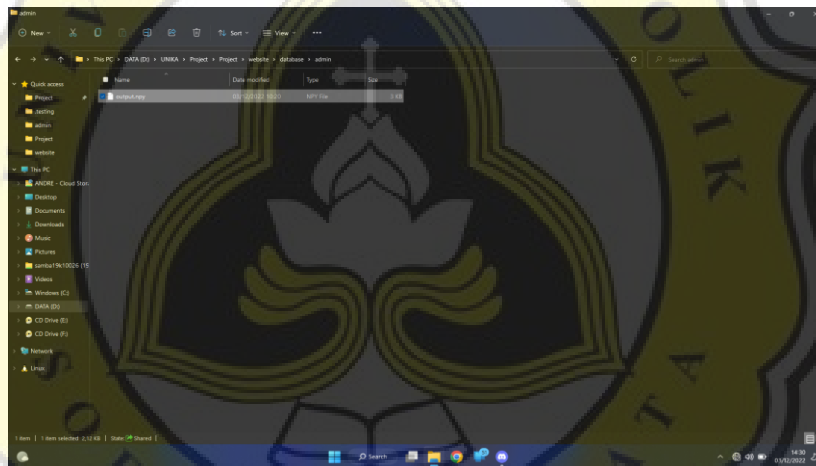


Figure 3. 6 Numpy Save File

The 128D face embedding vector that used for training dataset are save into .npy file. This .npy file will be used for the face matching process.

### 3.3 Euclidean Distance

The Euclidean distance is a term used in mathematics to define the length of a line segment that connects two points in vectors. In either a two-dimensional or three-dimensional space, the distance of a segment connecting the two points is measured. It is generally used to find similarities between two objects.

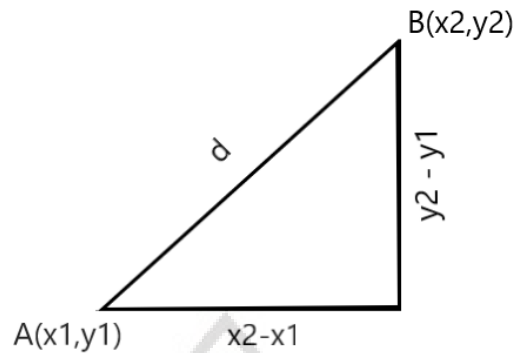


Figure 3. 7 Euclidean Distance [12]

$$D = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2 + \dots ]}.$$

Like Pythagoras, Euclidean distance measures the distance between two point using this formulas. The x and y coordinates of point A are X1 and Y1, and the x and y coordinates of point B are X2 and Y2. This formula will produce the distance from point A to point B (d). The lower the distance, the more similar the object is.

This study uses Euclidean distance as an algorithm to compare two vectors that represented 128 dimensional vectors which is obtained from the Openface feature extraction

### 3.4 Cosine Similarity

Different from Euclidean distance, cosine similarity calculates the angle of two objects to calculate the distance. This study also uses cosine similarity as an algorithm to compare two 128 dimensional vectors which is obtained from Openface feature extraction

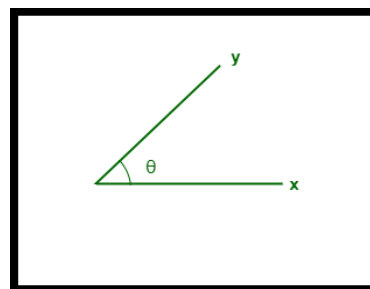


Figure 3. 8 Cosine Similarity [14]

---

$$\text{Cos}(x, y) = x \cdot y / \|x\| * \|y\|$$

$$\theta(x, y) = 1 - \text{Cos}(x, y)$$

---

$$\|x\| = \sqrt{x_1^2 + x_2^2 + x_3^2 + \dots}$$

$$\|y\| = \sqrt{y_1^2 + y_2^2 + y_3^2 + \dots}$$

$$x \cdot y = (x_1 * y_1) + (x_2 * y_2) + (x_3 * y_3) + \dots$$

### 3.5 Deploy Website using Streamlit Library

This web-apps using a streamlit library and connected with Database MYSQL that save face and login credentials each employee. If the face cosine similarity score less than 0.014 or the face euclidean distance score less than 0.17 then the attendance are success.

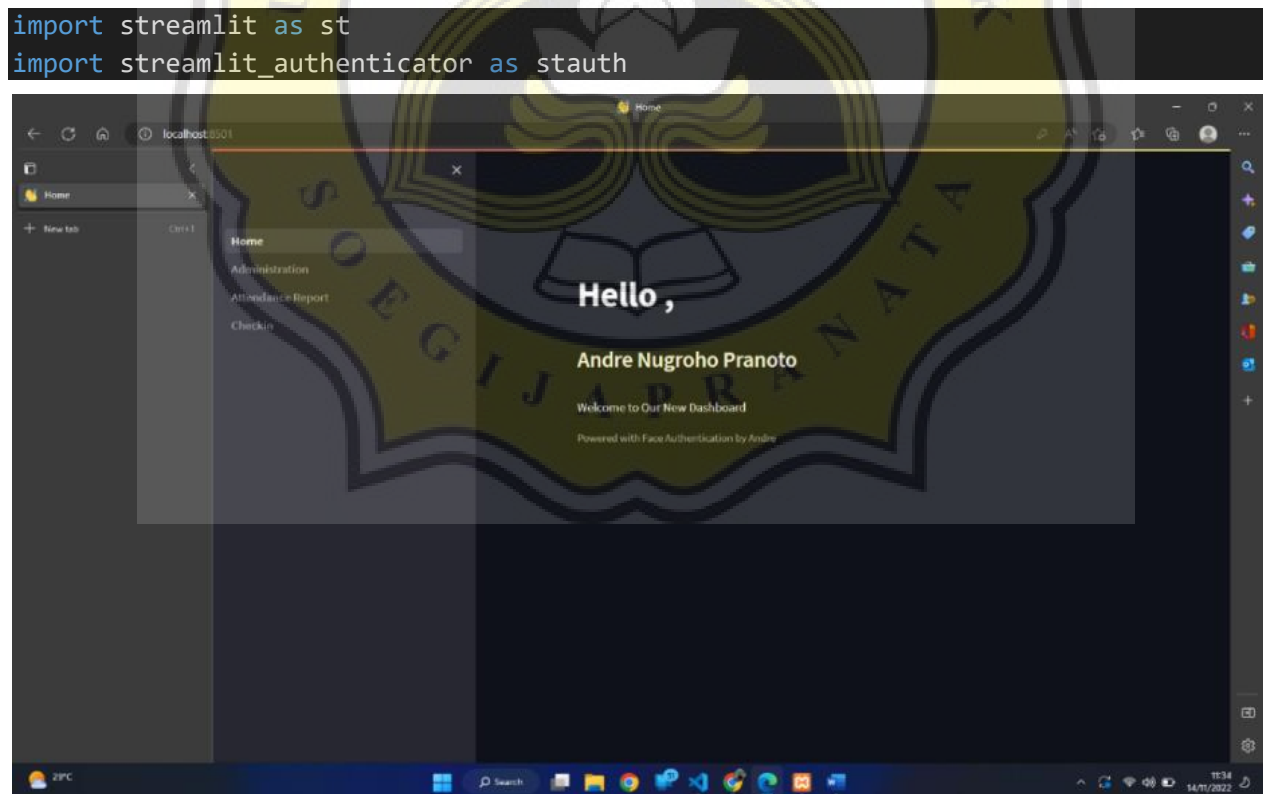


Figure 3. 9 Attendance Web-Apps