CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation

The chapter five describes implementation and testing where the implementation would discover about the code and the explanation and the testing would discover about trial results of algorithms and data structures.

```
1. for scale in np.linspace(0.2, 1.0, 20)[::-1]:
2. resized = imutils.resize(gray, width = int(gray.shape[1] *
   scale))
3. r = gray.shape[1] / float(resized.shape[1])
4. if resized.shape[0] < tH or resized.shape[1] < tW:
5. break
6. edged = cv2.Canny(resized, 50, 200)
7. result = cv2.matchTemplate(edged, template, cv2.TM_CCOEFF)
```

As seen on the list of source code above, the explanation is as follows. In the first line, the code means to looping for 20 times periodically in range 0.2 to 1.0. The image also resized along with the 20 looping times. In the line 6, the image modified into canny edge detection form. In the last line, after modifying the image, it processed into match template method according to Cross Coefficient template matching method by using openCV.

```
1. (_, maxVal, _, maxLoc) = cv2.minMaxLoc(result)
2. if found is None or maxVal > found[0]:
3. found = (maxVal, maxLoc, r)
4. (maxVal, maxLoc, r) = found
5. (startA[i], startC[i]) = (int(maxLoc[0] * r), int(maxLoc[1] * r))
6. (startD[i], StartB[i]) = (int((maxLoc[0] + tW) * r),
   int((maxLoc[1] + tH) * r))
```

By using openCV privilege, the program utilized minMaxLov provided by openCV to search min max detected pixel as seen on line 1. In the 2-3 lines, if part of the image detected and having the same form with the template, the code would generate the maxVal because in Cross Coefficient template matching method we only using maxVal as distinguishing value. In the following line, the maxVal and the maxLoc that have found generate into edge edges output detected.

5.2 Testing

After having such analysis and implementation, some testing according to having some another point of view from the project have done. There are 4 types of testing, the first and the second testing worked with circular form of template with different form of dataset. The first testing having circular form of image input and the second testing having square form of image input. The third and the fourth testing worked with square form of template with different form of dataset. The third testing having square form of image input and the fourth testing having circular form of image input.

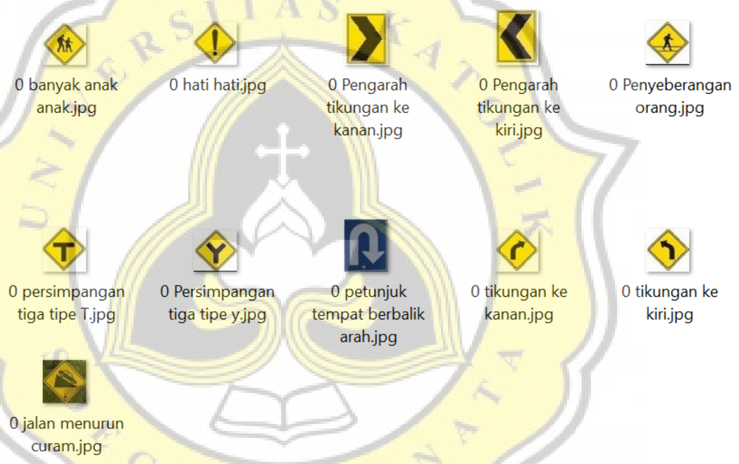
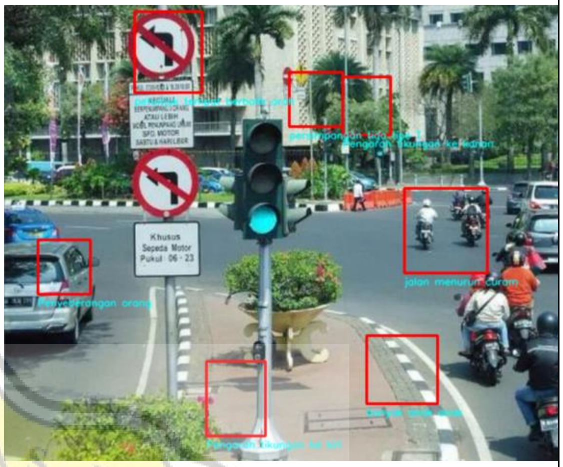
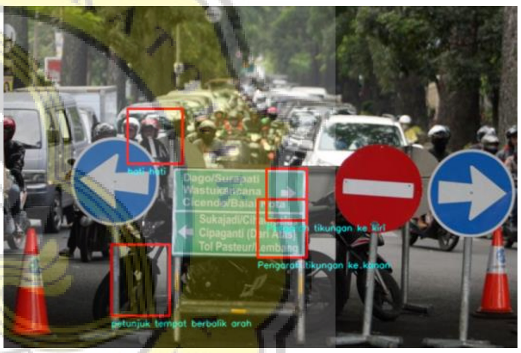
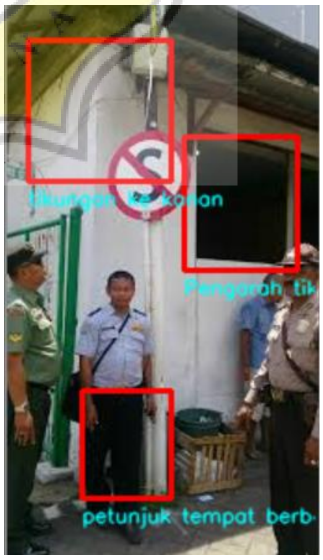

0 banyak anak anak.jpg
0 hati hati.jpg
0 Pengarah tikungan ke kanan.jpg
0 Pengarah tikungan ke kiri.jpg
0 Penyeberangan orang.jpg
0 persimpangan tiga tipe T.jpg
0 Persimpangan tiga tipe y.jpg
0 petunjuk tempat berbalik arah.jpg
0 tikungan ke kanan.jpg
0 tikungan ke kiri.jpg
0 jalan menurun curam.jpg

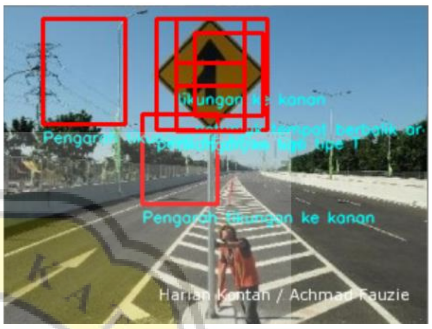
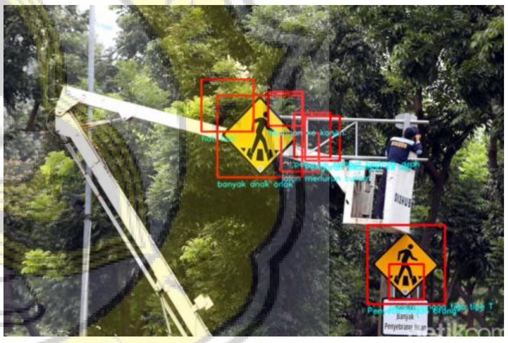
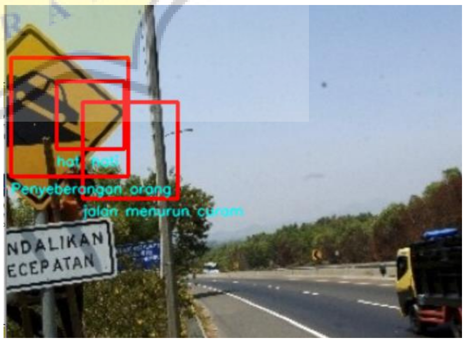
Illustration 5.1: Square form of template

Table 5.1: Testing Table of Some Data

Real Image	Output Image
	 petunjuk tempat berbalik arah Pengarah tikungan ke kanan penyeberangan orang jalan menurun curam Pengarah tikungan ke kiri Khusus Sepeda Motor Pukul 06 - 23
Dago/Surapati Wastukancana Cicendo/Balai Kota Sukajadi/Cihampelas Cipaganti (Dari Atas) Tol Pasteur/Lembang	 hati-hati tikungan ke kiri Pengarah tikungan ke kanan petunjuk tempat berbalik arah
	

The first testing worked with 11 image template with 10 image input. The output came up with the chaos of randomly detected match output. But, the program can't find the same form or shape between the undetected part with the templates that processed in.

Table 5.2: Testing Table of Some Data

Real Image	Output Image
	 tikungan ke kanan Pengarah tikungan ke kanan Harian Kontan / Achmad Fauzie
	
	 Penyeberangan orang jalan menurun curam NDALIKAN ECEPATAN

The second testing processed with 11 image template and 10 image input. After had some chaos with the circular form of dataset. Later on the second testing, tested with the square form of dataset along with the square form of template. But it did not come as well as hoped.

The engine just could not differentiate between each square form of template and the output finally piled up with one and another.

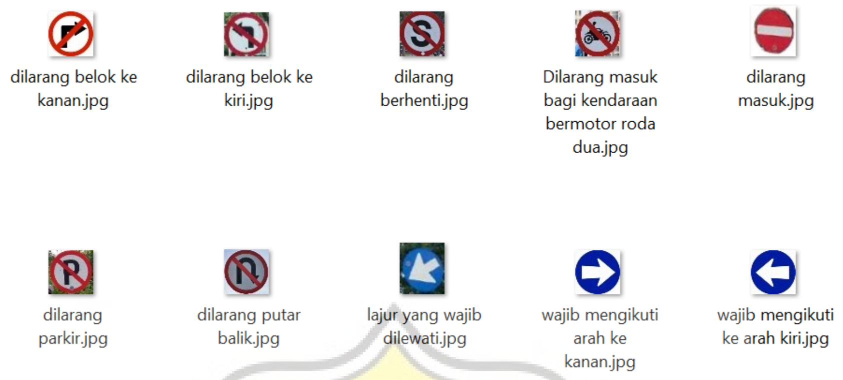
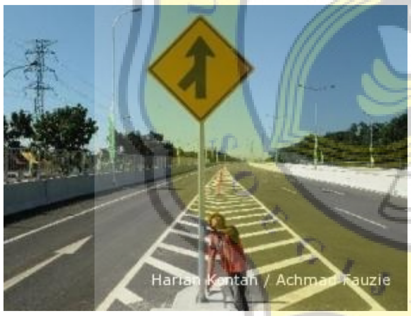
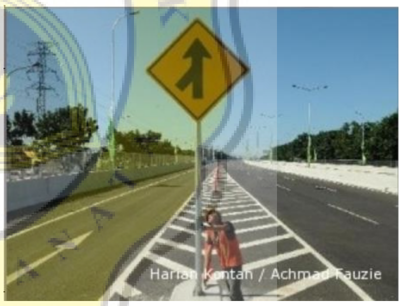
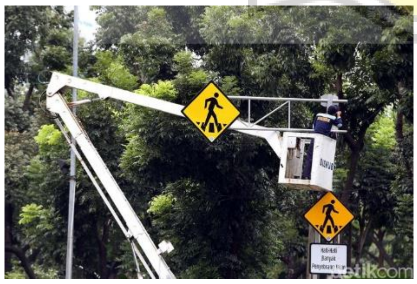
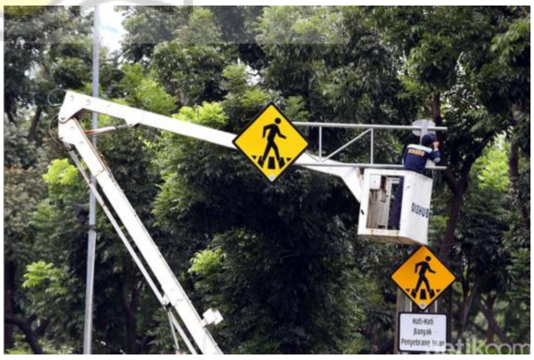
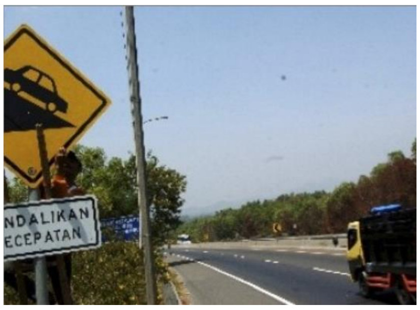

dilarang belok ke kanan.jpg
dilarang belok ke kiri.jpg
dilarang berhenti.jpg
Dilarang masuk bagi kendaraan bermotor roda dua.jpg
dilarang masuk.jpg
dilarang parkir.jpg
dilarang putar balik.jpg
lajur yang wajib dilewati.jpg
wajib mengikuti arah ke kanan.jpg
wajib mengikuti ke arah kiri.jpg

Illustration 5.2: Circular form of template

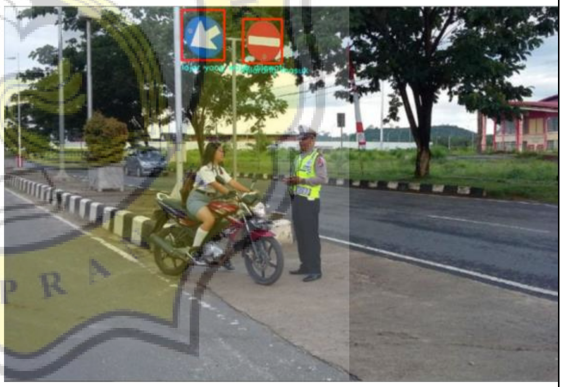
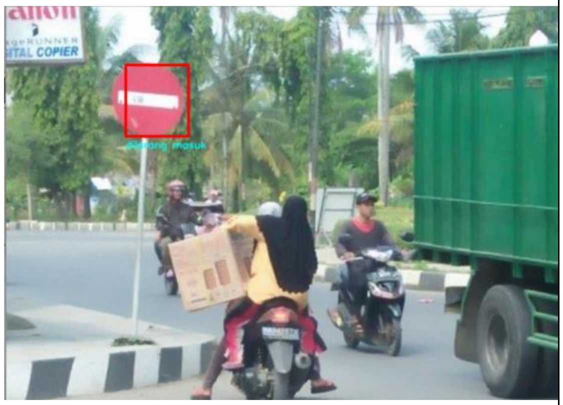
Table 5.3: Testing Table of Some Data

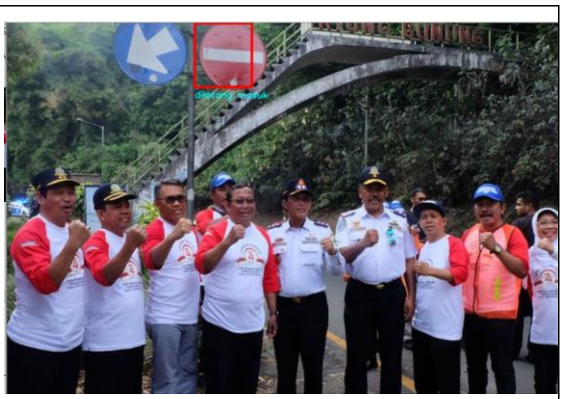
Real Image	Output Image
	
	

The third testing worked with 10 image of template and 10 image input. This testing just come up with as well as hope result. The circular form of template where did not match with any part of image that processed. It is making higher reason why this project ended up using circular form of template as reference.

Table 5.4: Testing Table of Some Data

Real Image	Output Image
	
	

After having some varied point of testing, this forth testing worked with 10 image of template and 10 image input. The success percentage scored around 70% of success. However, after having a lot of angle in datasets as said on previous chapter, the engine having flurry detected output which make sense with the testing on this chapter that have done.

The cross coefficient template matching method having such tricky detect process. Each template had to have really different form of shape. The sign color did not very influential in the process, because the template and the image would turn into grayscale. After done some searching and analysis on the internet, many people done template matching method with a really look alike template or they just screenshot part of image dataset and used it as template which did not work on this project. The Image Denoising is already implemented in input image, but it did not give the result as expected.

The problem of this research is how to detect Indonesia's traffic signs properly, describe what the sign it is. The analysis describe about what was the reason that making failure in the detection process, create several case comparisons such as different form of traffic sign, various kind of template, combine a few datasets with some different noises.

5.2.1 Templates of Dataset

After doing some research and various kind of testing, this study came up with two kind of different template, circular and square template. The templates must be having under 50x50 pixel, because if the templates having higher pixels that it must be, it may affect in detection progress. The error that produced by executing with templates that having pixels higher than 50x50 pixels reached up to 70%. In conclusion, the templates must be having pixels in range between 47 and 49.

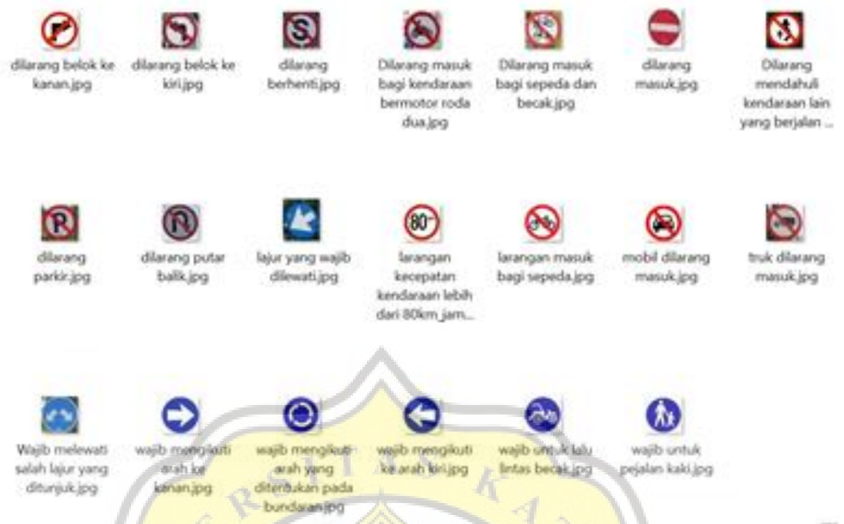

dilarang belok ke kanan.jpg
dilarang belok ke kiri.jpg
dilarang berhenti.jpg
Dilarang masuk bagi kendaraan bermotor roda dua.jpg
Dilarang masuk bagi sepeda dan becak.jpg
dilarang masuk.jpg
Dilarang mendahului kendaraan lain yang berjalan ...
dilarang parkir.jpg
dilarang putar balik.jpg
lajur yang wajib dilewati.jpg
larangan kecepatan kendaraan lebih dari 80km_jam...
larangan masuk bagi sepeda.jpg
mobil dilarang masuk.jpg
truk dilarang masuk.jpg
Wajib melewati salah lajur yang ditunjuk.jpg
wajib mengikuti arah ke kanan.jpg
wajib mengikuti arah yang ditentukan pada bundaran.jpg
wajib mengikuti ke arah kiri.jpg
wajib untuk lalu lintas becak.jpg
wajib untuk pejalan kaki.jpg

Illustration 5.3: List of template picture

The picture above is the list of template in circular form of traffic sign in Indonesia that used in this research.

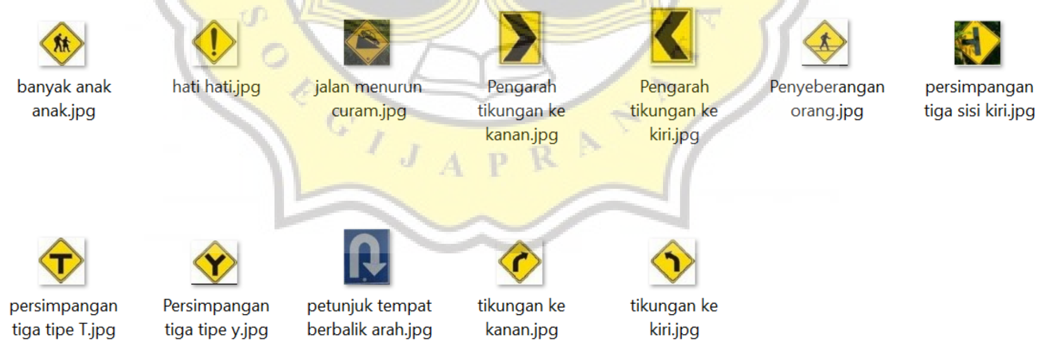

banyak anak anak.jpg
hati hati.jpg
jalan menurun curam.jpg
Pengarah tikungan ke kanan.jpg
Pengarah tikungan ke kiri.jpg
Penyeberangan orang.jpg
persimpangan tiga sisi kiri.jpg
persimpangan tiga tipe T.jpg
Persimpangan tiga tipe y.jpg
petunjuk tempat berbalik arah.jpg
tikungan ke kanan.jpg
tikungan ke kiri.jpg

Illustration 5.4: List of template picture

The picture above is the list of template in square form of traffic sign in Indonesia that used in this research.

5.2.2 Datasets

The total of image that have used as datasets is 500 image. As said on the previous chapter, this study not only using the pure image as datasets, but adding some noises in a few image as datasets for comparison, some variation of datasets and to analyze does noises matter in the detection progress. To sum up, this project using 200 pure images, 100 images with Gaussian noise, 100 images with Salt and Pepper noise and 100 images with Speckle noise. In the course of searching datasets, researcher try as much as possible to find the datasets that having various kind of location, distance, tilt, exposure and angle of the image.

Illustration 5.5: Example of some dataset that contain pure images

Illustration 5.6: Example of some dataset with Gaussian blur

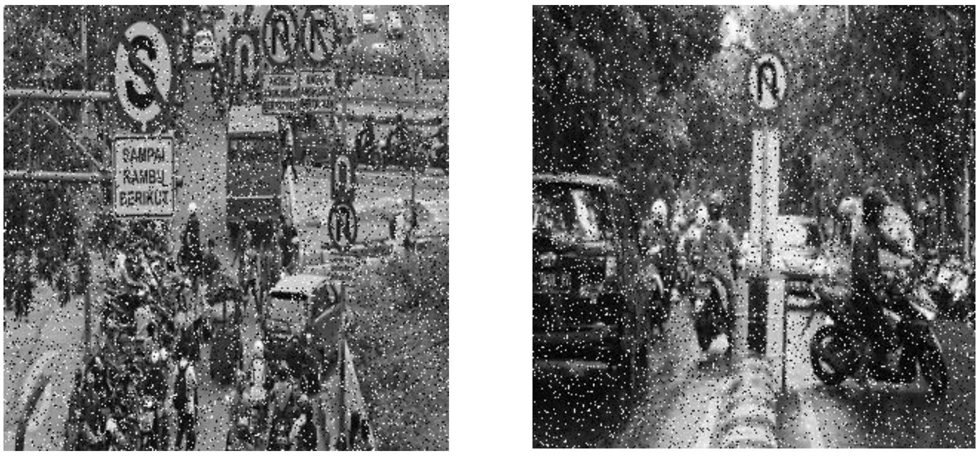


Illustration 5.7: Example of some dataset with Salt and Pepper noise

Illustration 5.8: Example of some dataset with Speckle noise

5.2.3 Diagram of Analysis

For the first diagram of analysis, analyzed with 200 pure images of dataset with 20 circular form of template. Dataset picked with various kind of angle, like what it said in previous sub chapter.

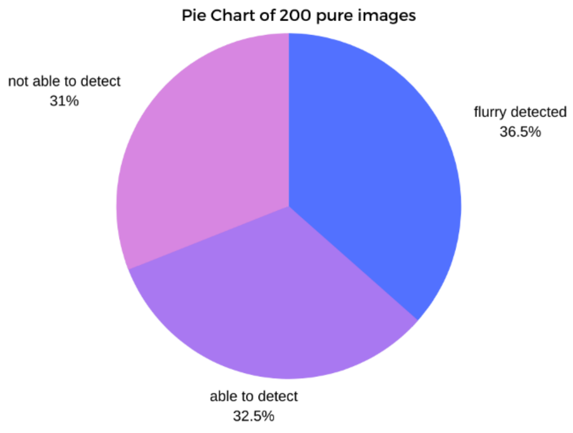


Illustration 5.9: analysis diagram with 200 pure images

The illustration above turned out with 32,5% able to detected, 31% not able to detect and 36,5% flurry detected. Flurry detected means, the engine randomly assigned the picture with randomly name of template along with the detected square. For example, the engine assigned the trees or the tire of cars or the pole posts as detected sign with randomly name of template which does not look the same at all or not what they should be.

There is some list of problem that has found which affect the way the engine processed in detection system. First, the clarity of image. The clarity of image, affecting almost 60% of the detection process. The image that having lack of clarity, has more than 80% would step into not able to detect group. However, there is small chance that the image would detect perfectly with some lack of clarity, such as the color of sign worn out, the image just blurry and the sign is ruined, but only around 3% of chance. Secondly, image size. The engine just could not stand with the smaller or bigger image, the smaller the image, the template could not be matched, the bigger the image, would having some confusion with the detection. It turns out, that the output is enter not able to detect groups. Thirdly, similarity between templates. As seen on the previous sub chapter that having the list of circular template, the detection process having confusion between prohibited to turn left and prohibited to U-turn. Lastly, tilt and distance of the signs on the image. The distance between the sign and where the picture was taken also affects the detection process.

The second diagram of analysis, analyzed with 100 image for each noises with 20 circular form of template.

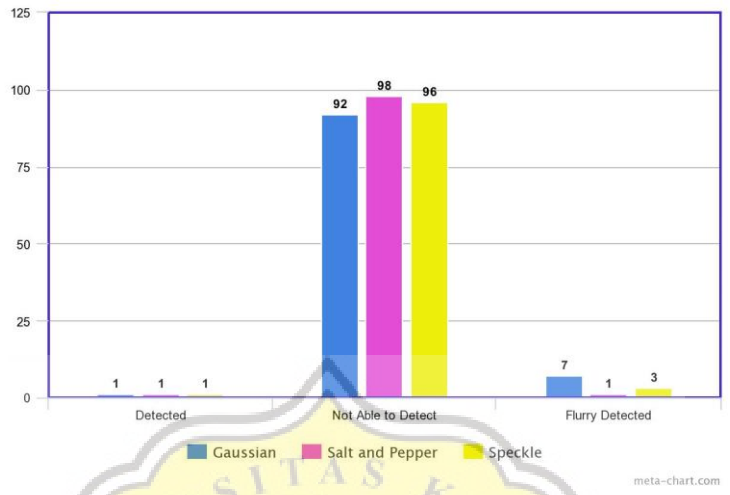


Illustration 5.10: Analysis Diagram with 100 image with each noises

In conclusion, the researcher came up with some reason why the engine could not detect the dataset and generate such a high amount of not able to detect in column chart. Noises do really affect the detection process, because the image that processed with noise either Gaussian or salt and pepper or speckle, would came up with the lack of clarity image and ruined out the form of sign in the image of dataset.

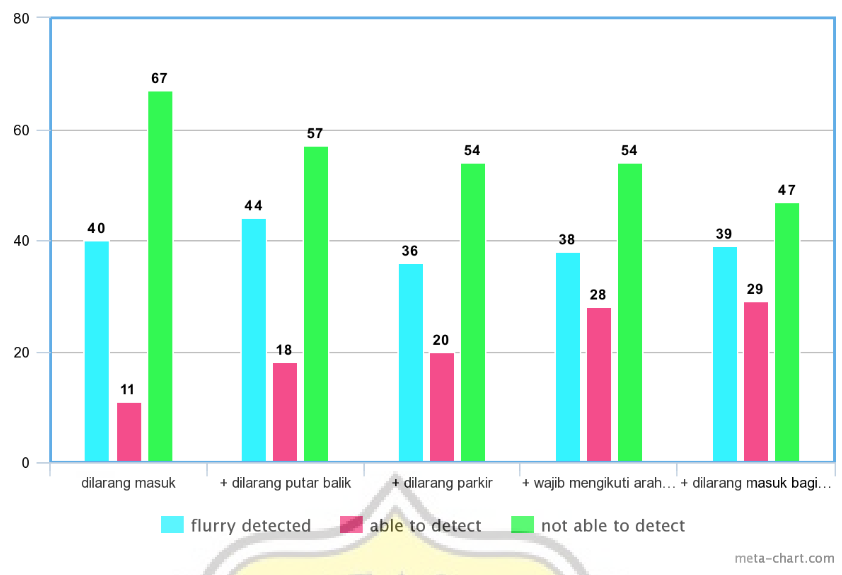


Illustration 5.11: Analysis Diagram with 100 image where The Template Added One by One

As shown by diagram above, for each multi bar chart, the template added one by one. For example, on the first multi bar chart shown that the template that used was “dilarang masuk” only and for the second bar, the template that used still “dilarang masuk” but added with “dilarang putar balik” template and so on. For the fourth multi bar chart, the template that added were “wajib mengikuti arah ke kanan” and “wajib mengikuti arah ke kiri”. For the last multi bar chart, the template that added were “dilarang masuk untuk mobil” and “dilarang masuk bagi kendaraan roda dua”. As the template added, the engine started to search the possibility of similarity for each template from the input image. That’s how the percentage of flurry detected have been so high. The more templates that are processed, the more automatically the engine would scan the input images to find the similarity between them.