

CHAPTER IV

ANALYSIS AND DESIGN

4.1. Analysis

Problems to be faced in this program is to insert a message / information to the edge pixel of images. But before inserting the message into the images, must go through a series of processes. First, the input image should be converted first into grayscale image, then smoothed to eliminate noise. Noise is smooth lines contained in the image which can be detected as the edges when it is not. Therefore, to produce perfect images edges must be through images smoothing method first. After that image is converted into edge images. Then begin to insert a message into the edges pixel in random start. The number of messages that can be inserted into images is also limited by the amount of edges pixel.

The second problem is when they want to take the message in the image. Are messages decrypted accordance with the original message or not.

4.2. Design

4.2.1. Use Case Diagram

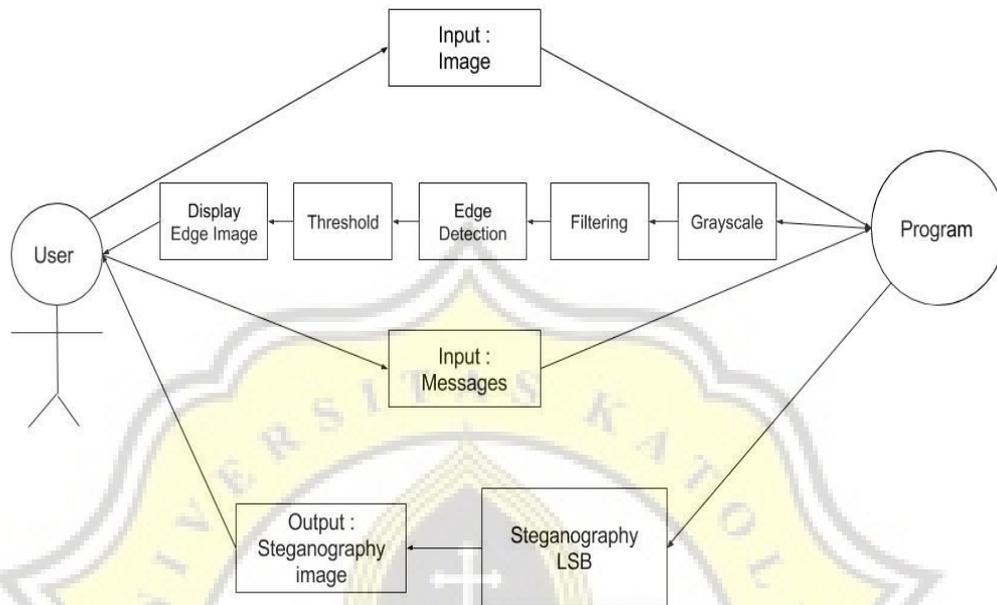


Figure 1: Case Diagram Encode

The first thing to do by the user is entering the input images. After that the program will process the input to be converted into form edges images, then the user can enter the message and the program will proceed to hide the message in the edges pixels of images. After that the user will get the output an image that has been inserted message.

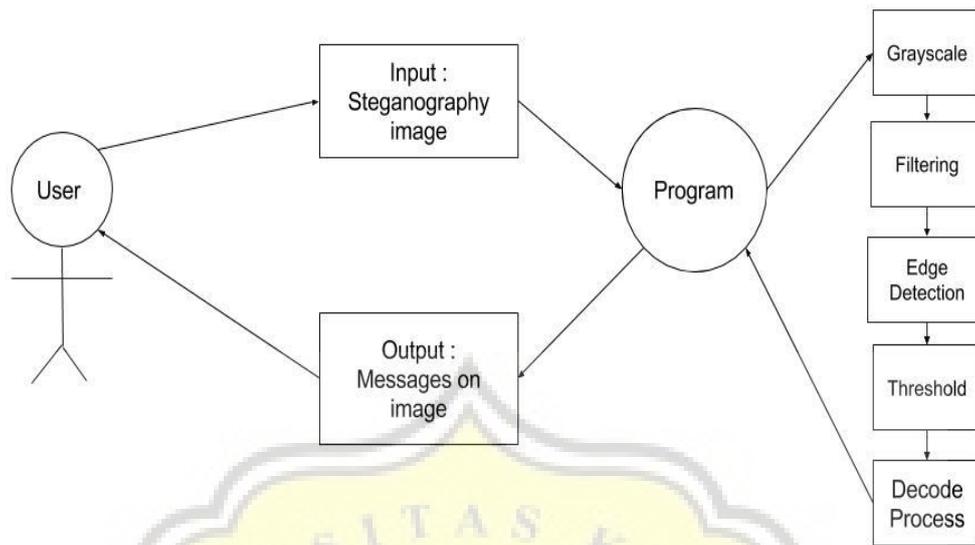


Figure 2 : Case Diagram Decode

Users can insert images that has been inserted message. Image will go through several process until threshold and program will do decode process. The program will display the output messages from the images.

4.2.2. Flow Chart

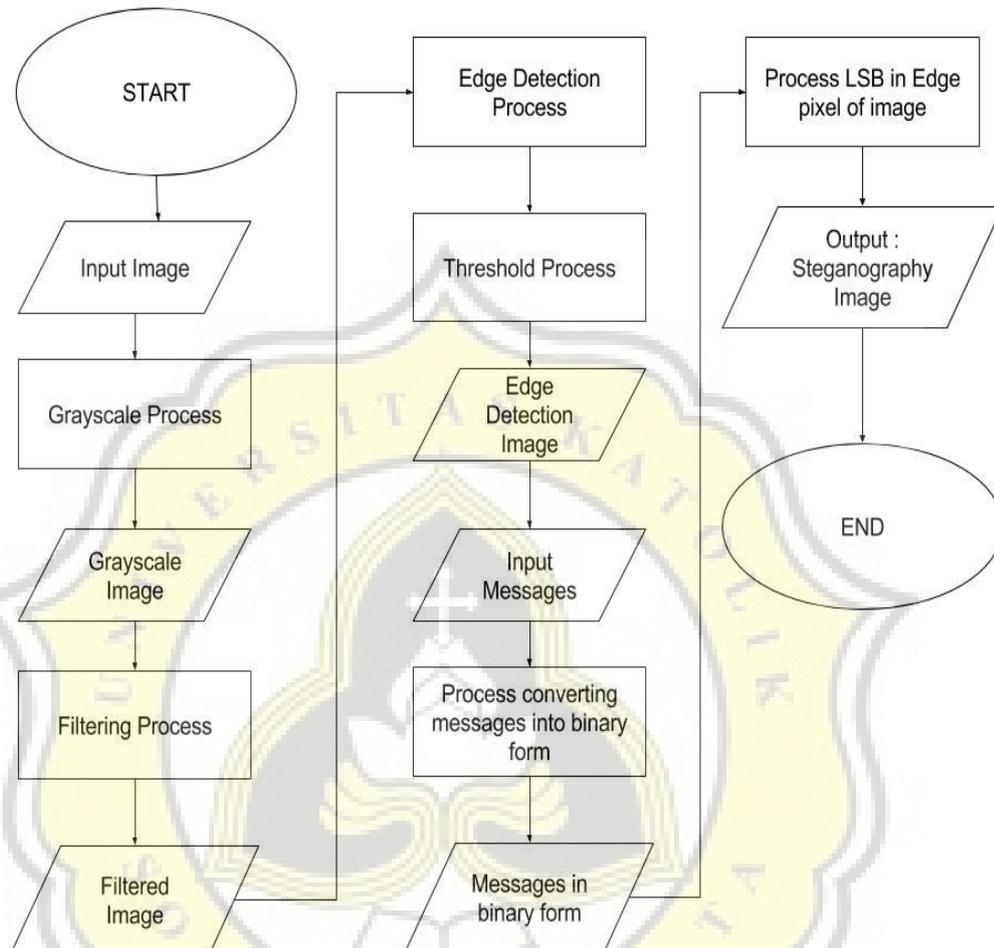


Figure 3: Flow Chart Encode Process

The program will display the user interface that asks the user to choose a menu that Encode and Decode. In the Encode menu, the user is prompted to enter an images, after that is converted to grayscale and then smoothed images. After that change the grayscale image that has been smoothed into the edge images using Sobel method. After images has been formed into the edge images the user will be prompted to enter the message that want to hide. Then the program will change each letter of the message into an ascii form, then into binary. Program will random the start of edge pixel to hide the message. After that insert every bit of message in the every last bit of edges pixel image.

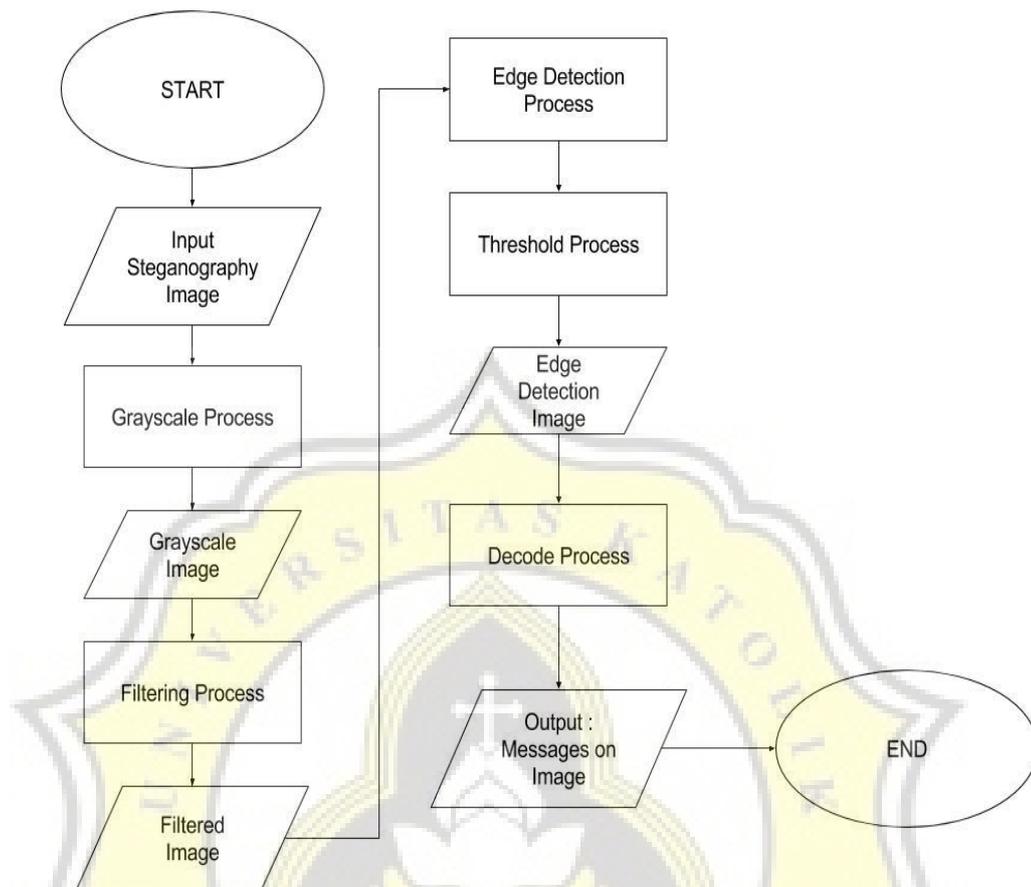


Figure 4: Flow Chart Decode Process

Decode, the user is prompted to insert an image that previously has entered the message. Then the image will go through process grayscale, filtering, edge detection, threshold. When the user clicks the button decodes, the program will search for the hidden messages on the edge pixel image. After that the messages will show in display interface.

4.2.3. Class Diagram

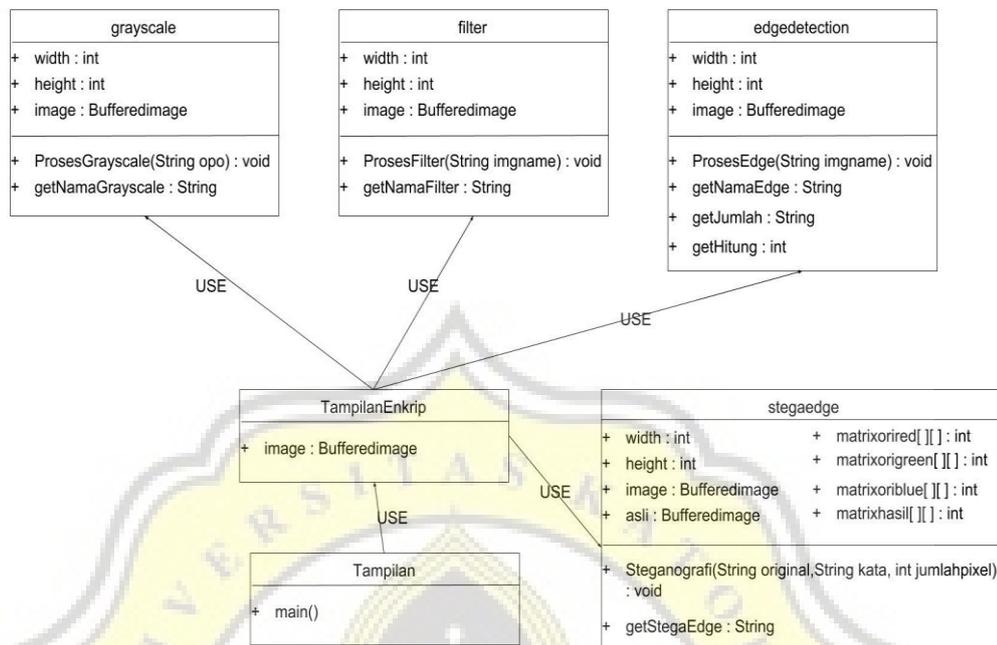


Figure 5: Class Diagram Encryption

1. Tampilan

The main class of program, will run in the first time.

2. TampilanEnkrip

Class that allows user to input a message to be hidden in the image.

3. grayscale

Class that process the image into grayscale image.

4. filter

Class that process the output of grayscale image to filtered image.

5. edgedetection

Class that process the output of filtered image to edge detection and threshold image.

6. stegaedge

Class that process the input message to be hidden in the input image.

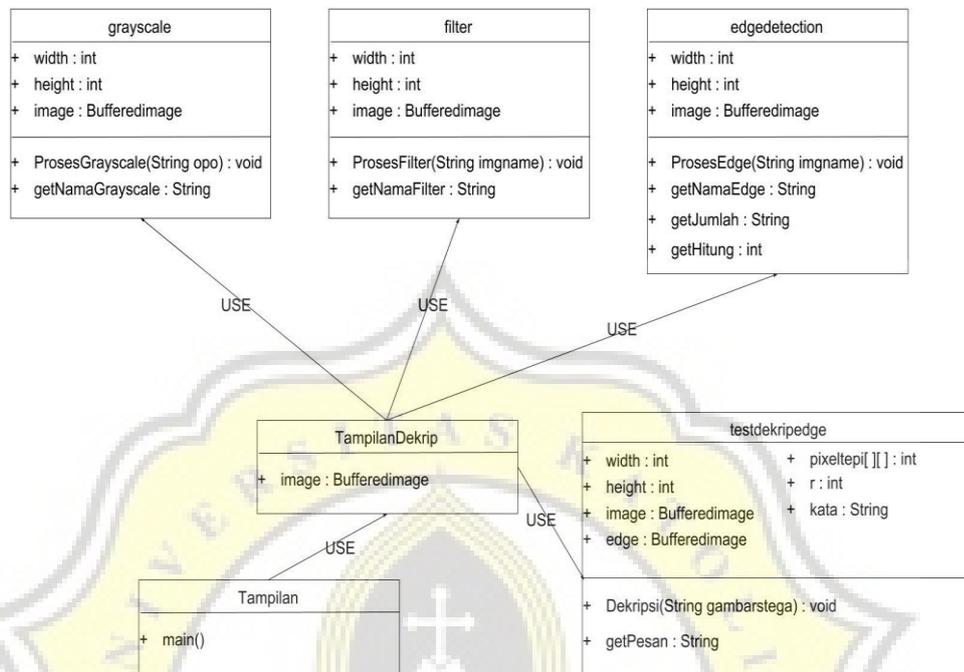


Figure 6: Class Diagram Decryption

1. Tampilan

The main class of program, will run in the first time.

2. TampilanDekrip

Class that allows user to input an image that contain a message to decrypt.

3. grayscale

Class that process the image into grayscale image.

4. filter

Class that process the output of grayscale image to filtered image.

5. edgedetection

Class that process the output of filtered image to edge detection and threshold image.

6. testdekripedge

Class that processes the input image to see messages that contained on the image.

