message into thinking that the image sent is just a plain image that does not contain anything suspicious. However, the traditional image steganography method generally results in some image distortion that can cause the presence of hidden information to be successfully detected by a Steganalyzer [2].

Steganalysis is a technique that aims to detect the presence of hidden information in the stego-image. Various methods have been proposed for steganalysis detection which combine handcrafted feature extraction with machine learning algorithm [2]. Although some conventional steganographic approaches can deceive one or several steganalyzers, it is very di cult to resist the detection of a well-designed one [3,4]. The main reason is that the conventional steganographic operation, which mainly embeds the secret message by slightly modifying some insensitive image features, will inevitably cause some distortion in the stego-image, especially under high message payload. Such distortion will cause the presence of hidden information to be detected.

Meanwhile, recent developments in the eld of image generation have prompted the development of another paradigm called Generative Image Steganography [1,5]. Compared to most traditional image steganography which embed the secret message into a cover image, this technique transforms the message directly into a generated image. This makes the technique more resistant to steganalysis detection because the resulting image is generated from scratch, so there is no trace of image manipulation as occurred in traditional image steganography [6{9].

In this presentation, we further explored the Generative Image Steganography based on sequence processing and Generative Adversarial Network (GAN) [10]. We are pursuing the intention to improve the stegosampling algorithm to produce a simple but e ective solution for Generative Image Steganography. For this, we propose a method combining Seq2seq processing with GAN to convert input text into generated images to hide the message contents. We will also introduce our challenge to implement the concept of private key into the proposed method.

2. Related Work

2.1 Image Steganography

To date, many conventional image steganography approaches have been proposed. They generally use an existing

[1] Nagoya University
[2] Telkom University
[3] RIKEN
[a] anditya@telkomuniversity.ac.id
[b] taka-coma@acm.org
[c] yasutomo.kawanishi@riken.jp
[d] ide@i.nagoya-u.ac.jp

Fig. 5 Generator Decoder Architecture

model is trained, the sender can use the Encoder Generator to generate an image for a speci c recipient according to the receiver's secret key. The image then can be sent and can only be read back by the correct Generator Decoder owned by the recipient. The proposed framework is shown in Fig. 2.

3.2 Model Architecture

For the LSTM Encoder, we use an LSTM layer with embedding layer. The cell state and the hidden state of the LSTM is continued to a dense layer for linear transformation into a 256-dimensional vector each as depicted in Fig. 3a.

The Generator Encoder receives both states and reshapes it into a $(8; 8; 8)$ tensor, before it is fed into a Convolution block. The output then proceeds to two Residual blocks and a nal Transpose Convolution block. The complete architecture can be seen in Fig. 4.

The Discriminator model consists of four layers of 4 4 stride 2 Convolution, and a Dense layer at the end. The architecture can be seen in Fig. 5a. Similarly, the Generator Decoder also consists of four layers of Convolution block. The di erence with the Discriminator is the addition of Batch Normalization layer into each block. The output is then continued to a nal convolution block with hyperbolic tangent (tanh) activation and then reshaped to a size $(2; 256)$ array as shown in Fig. 5b.

The LSTM Decoder receives the output from the Generator Decoder as two vector states and feed them into an LSTM layer with a dense layer for word prediction as depicted in Fig. 3b.

4. Evaluation

4.1 Generative Steganography Experiments

We conduct our experiments using the CelebA dataset [17] to produce \realistic-looking" images. For the sake of training e ciency, we rescale the sizes of images to 64 64 pixels and train the model on the rescaled images. For the payload, we use English texts with a length of up to 30 words per sentence taken from the Tatoeba dataset [18].

We trained our model end-to-end with 100,000 sentences and 162,770 images for training and 20,000 sentences for testing. For the evaluation criteria, the BLEU metric [19] and the Word Error Rate are used to measure the retrieval

Fig. 6 Example of the generative steganography result.

Fig. 7 Comparison of (a) the generated images and (b) CelebA dataset

performance, and the Fréchet Inception Distance (FID) to measure the image quality.

4.2 Steganography Results

Our initial observations yielded moderate but encouraging results. The proposed model achieve 0.59 in BLEU metric and 0.26 Word Error Rate when retrieving text messages. Examples of the input message, the resulting stego-image, and the retrieved message can be seen in Fig. 6.

We can see that the proposed method is able to produce a stego-image with a fairly good performance, but there are still some word prediction errors that must be corrected.