TESIS - IF 185401

FACIAL INPAINTING MENGGUNAKAN *GENERATIVE ADVERSARIAL NETWORK* DENGAN MEMPERTAHANKAN KETERKAITAN SPASIAL

AVIN MAULANA
05111850010029

Dosen Pembimbing
Dr. Eng. Chastine Fatichah, S.Kom, M.Kom
Dr. Eng. Nanik Suciati, S.Kom, M.Kom

Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
2020

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M. Kom)
di
Institut Teknologi Sepuluh Nopember
Oleh:

AVIN MAULANA
NRP: 05111850010029

Tanggal Ujian: 17 Juli 2020
Periode Wisuda: September 2020

Disetujui oleh:
Pembimbing:

1. Dr. Eng. Chastine Fatichah, S.Kom., M.Kom. .....
NIP: 19751220 200112 2 002

2. Dr. Eng. Nanik Suciati, S.Kom., M.Kom. .....
NIP: 19710428 199412 2 001

Penguji:

1. Dr. Ahmad Saikhu, S.Si., M.T. .....
NIP: 19710718 200604 1 001

2. Shintami Chusnul Hidayati, S.Kom, M. Sc., Ph. D .....
NPP: 1987202012004

3. Dr. Radityo Anggoro, S.Kom, M.Sc. .....
NIP: 19841016 200812 1 002

Kepala Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN INSTITUT TEKNOLOGI SEPULUH NOPEMBER FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS DEPARTEMEN TEKNIK INFORMATIKA

Dr. Eng. Chastine Fatichah, S.Kom., M.Kom
NIP: 19751220 200112 2 002

[*Halaman ini sengaja dikosongkan*]

FACIAL INPAINTING MENGGUNAKAN *GENERATIVE ADVERSARIAL NETWORK* DENGAN MEMPERTAHANKAN KETERKAITAN SPASIAL

Nama Mahasiswa : Avin Maulana
NRP : 05111850010029
Pembimbing I : Dr. Eng. Chastine Fatichah, S.Kom, M.Kom
Pembimbing II : Dr. Eng. Nanik Suciati, S.Kom, M.Kom

ABSTRAK

Facial inpainting merupakan proeses merekonstruksi kembali bagian yang hilang pada citra wajah sedemikian sehingga citra hasil rekonstruksi dapat tetap terlihat realistis, serta pihak pengamat tidak dapat mengenali bagian yang merupakan hasil rekontsruksi. *Facial inpainting* dapat menjadi masalah yang menantang, karena untuk melakukan proses rekonstruksi diperlukan pengetahuan perseptual dari wajah, tidak cukup jika hanya dengan melihat kemiripan dengan bagian sekitar dari bagian yang hilang, seperti algoritma *inpainting* konvensional. Seiring dengan perkembangan teknologi dan ketersediaan data, *facial inpainting* dapat dilakukan dengan menggunakan konsep *deep learning*.

Penelitian sebelumnya melakukan *inpainting Generative Adversarial Network* (GAN). Namun terdapat masalah yang timbul pada saat *inpainting* dilakukan. Masalah pertama adalah piksel yang tidak konsisten antara bagian hasil *inpainting* dengan bagian sekitarnya ketika proses *inpainting* dilakukan pada citra wajah *unaligned*. Hasil *inpainting* terlihat tidak realistis karena perbedaan yang dapat terlihat antara bagian hasil rekonstruksi dengan bagian asli. Masalah ini dapat dilihat sebagai masalah keterkaitan spasial. Piksel hasil *inpainting* juga mungkin menunjukkan perbedaan warna dengan bagian piksel sekitarnya, seperti ketika bagian citra yang dilakukan rekonstruksi terletak pada setengah bibir.

Penelitian ini bertujuan mengembangkan metode *inpainting* berbasis GAN dengan tambahan *loss* berupa *feature reconstruction loss* dan *face landmark loss* untuk mengatasi hasil inpainting yang terlihat tidak realistis pada citra wajah *unaligned*. *Feature reconstruction loss* adalah *loss* yang diperoleh dari *pre-trained network* VGG-Net. *Loss* ini dapat digunakan untuk membantu mempertahankan keterkaitan spasial pada citra, terlebih ketika citra yang digunakan merupakan citra wajah *unaligned*. *Face landmark loss* juga merupakan *loss* dari *pre-trained network*, dan dapat digunakan untuk membantu meningkatkan kualitas perseptual dari citra hasil *inpainting*.

Proses *training* dilakukan dengan skenario *curriculum learning*. Secara kualitatif hasil yang diperleh menunjukkan bahwa metode *inpainting* yang diajukan tetap dapat dilakukan pada citra wajah *unaligned* dengan tetap mempertahankan keterkaitan spasial. Secara kuantitatif, metode yang diajukan mampu memperoleh rata-rata PSNR dan SSIM 21.528 dan 0.665, sementara maksimum PSNR dan SSIM yang diperoleh 29.922 dan 0.908.

Kata kunci: *curriculum learning, facial inpainting, feature reconstruction loss, generative adversarial network,* keterkaitan spasial

[*Halaman ini sengaja dikosongkan*]

FACIAL INPAINTING USING GENERATIVE ADVERSARIAL NETWORK WITH PRESERVING SPATIAL CORRELATION

By : Avin Maulana
Student Identity Number : 05111850010029
Supervisor I : Dr. Eng. Chastine Fatichah, S.Kom, M.Kom
Supervisor II : Dr. Eng. Nanik Suciati, S.Kom, M.Kom

ABSTRACT

Facial inpainting is a process to reconstruct some missing or damaged pixels in the facial image, and the reconstructed pixels should still be realistic so the observer could not differentiate between the reconstructed pixels or the original one. Facial inpainting can be a challenging problem to solve in machine learning. The reconstruction process on the human face's image should consider perceptual knowledge of faces, besides its similarity with its neighbor region, unlike another conventional inpainting algorithm. Along with technology development and data availability, facial inpainting can be done using deep learning methods.

Some of the previous researches have done inpainting using generative network, such as Generative Adversarial Network. However, there are a few problems that may arise when the inpainting algorithm has been done. The first problem was an inconsistency between adjacent pixels when facial inpainting was done on unaligned face images. The inpainting result would be seen as unrealistic because of its difference between the reconstructed and original regions. It can be seen as a spatial correlation problem between adjacent pixels. Inpainting results may also show different colors between the generated region and its adjacent original regions, for instance, when the missing regions were half of the lips area. Therefore, an improvement method in facial inpainting based on deep-learning is proposed to reduce the effect of the stated problems before, using GAN with additional loss from feature reconstruction loss and face landmark loss. Feature reconstruction loss is a loss obtained by using pre-trained network VGG-Net. It can be used to help in preserving spatial consistency within images, especially when it comes to unaligned faces. Face landmark loss is also a loss from a pre-trained network, and able to help the perceptual quality of the inpainting result.

Training process have been done using curriculum learning scenario. Qualitative results show that our inpainting method can reconstruct missing area on unaligned face images, and still preserve consistent colours between its adjacent pixels. From the quantitative results, our proposed method can achieve average score 21.528 and 0.665 on PSNR and SSIM metrics, respectively. While the maximum score achieved for PSNR and SSIM are 29.922 and 0.908, respectively.

Keyword: curriculum learning, facial inpainting, feature reconstruction loss, generative adversarial network, spatial correlation

[*Halaman ini sengaja dikosongkan*]

KATA PENGANTAR

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

Bismillahirrohmanirrohim.

Alhamdulillaahirobbil'aalamiin. Puji syukur penulis panjatkan ke hadirat Allah SWT., *shalawat* serta salam semoga senantiasa tercurahkan kepada *Rasulullah* Muhammad SAW., karena atas berkah dan rohmah-nya penulis dapat menyelesaikan penelitian yang berjudul **"*Facial Inpainting* Menggunakan *Generative Adversarial Network* dengan Mempertahankan Keterkaitan Spasial"** dengan baik. Tanpa doa, dukungan, bimbingan dan bantuan, penulis tidak dapat menyelesaikan penelitian ini dengan baik. Sehingga pada kesempatan ini, dengan segala kerendahan hati penulis ingin menyampaikan rasa terima kasih dan penghargaan sebesar-besarnya, kepada:

1. Kedua orang tua: Pak Ahmadi dan Ibu Mulyani selalu memberi waktu, bantuan, dorongan, semangat, serta do'a kepada penulis dalam hal apapun,
2. Ketiga adik: Devi Aulia, Bintang Ramadani, dan Fatan Jadid Akbar yang memberi semangat dalam menempuh pendidikan,
3. Ibu Dr. Eng. Chastine Fatichah, S.Kom, M.Kom selaku Pembimbing sekaligus Dosen Wali yang telah memberikan banyak masukan dan dorongan selama proses penelitian ini berlangsung, serta meluangkan waktu dan kesempatan selama proses bimbingan dan proses perwalian selama penulis menempuh studi di Informatika iTS,
4. Ibu Dr. Eng. Nanik Suciati, S.Kom, M.Kom selaku Pembimbing yang telah meluangkan waktu dan kesempatan, memberi banyak masukan, dorongan, bahkan lagu selama proses penelitian berlangsung,
5. Bapak Dr. Dr. Ahmad Saikhu, S.Si., M.T., Bapak Dr. Radityo Anggoro, S.Kom, M.Sc., serta Ibu Shintami Chusnul Hidayati, S. Kom., M. Sc., Ph. D selaku Dosen Penguji yang telah memberi saran sehingga penelitian yang dilakukan bisa menjadi lebih baik lagi,

6. Dekan Fakultas Teknologi Elektro dan Informatika Cerdas, Ketua Departemen Teknik Informatika, Ketua dan Sekretaris Program Pascasarjana Informatika Institut Teknologi Sepuluh Nopember, yang telah memberikan izin menggunakan fasilitas kampus kepada penulis sehingga dapat menyelesaikan penidikan,
7. Seluruh Dosen, staf laboratorium, staf tata usaha, dan segenap karyawan Departemen Teknik Informatika iTS,
8. Keluarga W5 yang terdiri dari Awwib, Dino, Riko dan Mudi yang selalu saling menopang dan menghibur siang dan malam,
9. Adam Safri, S.Kom, M.Kom yang telah menjadi rekan dan memberi motivasi selama proses penelitian,
10. Teman seperjuangan mahasiswa Magister Teknik Informatika iTS tahun Angkatan 2018,
11. Seluruh pihak yang senantiasa memberi semangat, motivasi dan dukungan.

Semoga Allah SWT. senantiasa memberikan rahmat, barokah, kesehatan dan balasan pahala yang melimpah kepada semua pihak atas kebaikan-kebaikan yang diberikan kepada penulis dari sejak sebelum dan selama penelitian ini berlangsung. Akhirnya, semoga penelitian ini dapat memberikan kontribusi bagi perkembangan ilmu pengetahuan dan dapat memberikan manfaat bagi penelitian selanjutnya.

Kuta, 20 Juli 2020

Avin Maulana

DAFTAR ISI

LEMBAR PENGESAHAN TESIS iii
ABSTRAK v
ABSTRACT vii
KATA PENGANTAR ix
DAFTAR ISI xi
DAFTAR GAMBAR xiii
DAFTAR TABEL xvii
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Perumusan Masalah 6
1.3 Tujuan Penelitian 7
1.4 Manfaat Penelitian 7
1.5 Kontribusi Penelitian 7
1.6 Batasan Masalah 7
BAB 2 DASAR TEORI 9
2.1 *Image Inpainting* 9
2.2 *Convolutional Neural Network* (CNN) 12
2.3 *Auto-Encoder* (AE) 13
2.4 *Generative Adversarial Network* (GAN) 16
2.5 *Curriculum Learning* 19
2.6 Metode Evaluasi 20
BAB 3 METODOLOGI PENELITIAN 23
3.1 Studi Literatur 23
3.2 Deskripsi *Dataset* 24
3.3 Perancangan dan Implementasi Metode 25
3.3.1 Praproses 27
3.3.2 *Splitting* Data 28
3.3.3 Augmentasi Data 28
3.3.4 *Generator Network* 29
3.3.5 *Discriminator Network* 32
3.3.6 *Facial Landmark Network* 35
3.3.7 *Semantic Parsing Network* 36

3.3.8 Skenario Uji Coba 37
BAB 4 HASIL DAN PEMBAHASAN 39
4.1 Hasil Penelitian 39
4.1.1 Lingkungan Uji Coba 39
4.1.2 Skenario Uji Coba 39
4.1.3 Proses *Training Network* 41
4.1.4 Hasil *Inpainting Testing Network* Berdasarkan Kualitas Gambar 49
4.1.5 Hasil *Inpainting Testing Network* Secara Visual 51
4.1.6 Pembahasan Masalah Keterkaitan Spasial 53
4.1.7 Pembahasan Masalah Citra Wajah *Unaligned* 55
BAB 5 KESIMPULAN 57
5.1 Kesimpulan 57
5.2 Saran 57
DAFTAR PUSTAKA 59
LAMPIRAN 63
BIODATA PENULIS 65

DAFTAR GAMBAR

Gambar 1.1 Image *restoration*. (a) citra sebelum restorasi, (b) citra hasil *inpainting* (Qureshi, dkk., 2017) 1

Gambar 1.2 *Facial inpainting* pada citra wajah *unaligned* (a) citra asli, (b) citra dengan region yang hilang, (c) hasil *inpainting*. (Yijun, dkk., 2017).....5

Gambar 1.3 *Facial inpainting* pada citra yang memperhatikan keterkaitan dengan bagian sekitarnya (a) citra asli, (b) citra dengan bagian yang hilang, (c) hasil *inpainting*. (Yijun, dkk., 2017).....5

Gambar 2.1 *Image inpainting.* (a) citra asli, (b) citra yang rusak karena proses trasmisi, (c) citra hasil rekonstruksi dengan teknik *inpainting*. (Furht, 2008) 9

Gambar 2.2 Pembagian metode *inpainting* (Qureshi, dkk., 2017) 10

Gambar 2.3 Arsitektur *Network Generative Face Completion* (GFC) (Yijun, dkk., 2017) 11

Gambar 2.4 Contoh CNN dalam menyelesaikan masalah klasifikasi citra (Stoll, 2017) 13

Gambar 2.5 Model *network Auto Encoder* (AE) (T. Ma, dkk., 2016).....14

Gambar 2.6 Alur kerja AE pada domain citra (Chollet, 2016).....15

Gambar 2.7 Model *Variational Auto-Encoder* (VAE) (Frans, 2016).....16

Gambar 2.8 *Generative Adversarial Network* (GAN) 17

Gambar 2.9 Proses *training* GAN (I. J. Goodfellow, dkk., 2014).....18

Gambar 2.10 Hasil dari GAN dengan data *training* TFD. Kolom paling kanan menunjukkan hasil model G setelah melalui proses *training* (I. J. Goodfellow, dkk., 2014) 18

Gambar 3.1 Diagram alir penelitian.....23

Gambar 3.2 Contoh gambar pada CelebA 24

Gambar 3.3 Arsitektur *network* yang diajukan 25

Gambar 3.4 Citra *input*. (a) Citra asli CelebA, (b) setelah dilakukan proses *cropping* dan *resize,* (c) Citra I stelah diberi *masking.* 27

Gambar 3.5 Susunan *layer network Generator* G, terdiri dari *encoder* dan *decoder* 29
Gambar 3.6 Susuan *layer* pada *network local* dan *global discriminator*.....34
Gambar 3.7 *Heat-map landmark* pada metode Liao (Liao dkk., 2018). (a) Citra input asli, (b) citra hasil *network landmark* dari (a), (c) citra *ground truth heat-map landmark* dari (a) 35
Gambar 3.8 Contoh s*emantic parsing* pada GFC. (a) Citra input asli, (b) citra hasil *network semantic parsing*. Nilai Ls merupakan $L2$ dari citra hasil *network semantic parsing* dengan *ground truth* (Yijun, dkk., 2017). 36
Gambar 4.1 *Landmark* wajah. Baris pertama merupakan citra masukan, baris kedua adalah hasil *landmark* menggunakan DLIB, baris ketiga merupakan hasil *network landmark* yang dibuat 41
Gambar 4.2 Hasil *semantic parsing* dengan BiSeNet. Data CelebA, dengan jumlah kelas 19 42
Gambar 4.3 Hasil *Network Generator* saat *training* dimulai.....43
Gambar 4.4 Hasil *Network Generator* menggunakan dua *loss*: *KL-Divergence* dan *Feature Reconstruction Loss* 44
Gambar 4.5 Hasil *Network Generator* tahap pertama pada metode GFC (Yijun dkk., 2017) 45
Gambar 4.6 Nilai loss dari *network generator* pada 15000 *step* awal, kiri menunjukkan *feature reconstruction loss,* kanan menunjukkan *KL-Divergence Loss*.....45
Gambar 4.7 Hasil *network generator* pada GAN standar ketika *discriminator* tidak seimbang, terlalu lemah atau terlalu kuat dari *generator* 47
Gambar 4.8 Hasil *network generator* step 25,000, tahap kedua dengan tambahan *loss adversarial* dari *local* dan *global discriminator*.....48
Gambar 4.9 Hasil *network generator* pada step 40,000, tahap ketiga dengan tambahan *loss landmark* dan *semantic parsing* 49
Gambar 4.10 Hasil *network generator* pada beberapa masukan data *test* 52
Gambar 4.11 Perbandingan hasil yang diperoleh dalam mempertahankan keterkaitan spasial.....53

Gambar 4.12 Hasil *network generator* pada kasus *inpainting* yang memperhatikan keterkaitan spasial.....54
Gambar 4.13 Perbandingan hasil yang diperoleh pada kasus citra wajah *unalgined* 55
Gambar 4.14 Hasil *network generator* pada *masukan* citra wajah *unaligned* 56

[*Halaman ini sengaja dikosongkan*]

DAFTAR TABEL

Tabel 3.1 Perbedaan fungsi *loss* GAN dan WGAN.....33
Tabel 4.1 Lingkungan Uji Coba.....39
Tabel 4.2 Konfigurasi parameter 40
Tabel 4.3 Hasil SSIM antara citra keluaran dengan masukan (SSIM semakin besar semakin baik).....50
Tabel 4.4 Hasil PSNR antara citra keluaran dengan masukan (PSNR semakin besar semakin baik).....50
Tabel 4.5 Perbandingan hasil yang diperoleh dengan metode terdahulu.....51
Tabel 4.6 Hasil penilaian responden 52

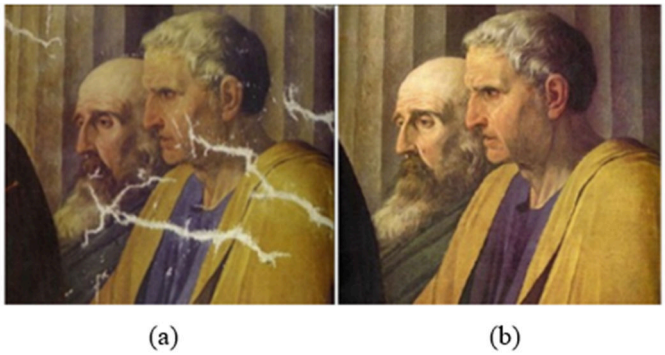
[*Halaman ini sengaja dikosongkan*]

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Image inpainting, merupakan salah satu masalah yang terdapat di domain citra. *Inpainting*, merupakan proses untuk merekonstruksi kembali region yang hilang pada suatu citra sedemikian sehingga region hasil rekonstruksi tetap konsisten secara visual dengan region selain region yang hilang, sehingga citra keseluruhan tetap terlihat realistis (Liao dkk., 2018). Image *inpainting* dilakukan ketika citra yang dimiliki terdapat kerusakan pada region tertentu. Penerapan *image inpainting* ini dapat dilakukan dengan tujuan untuk menghilangkan bagian objek yang tidak diinginkan, atau restorasi citra yang mengalami kerusakan (Qureshi, dkk., 2017). Gambar 1.1 menunjukkan contoh penerapan *inpainting* dalam restorasi citra.

Menurut Qureshi, dkk. (Qureshi, dkk., 2017), metode untuk melakukan *image inpainting* dapat dibagi menjadi 4 bagian, yaitu: 1) e*xemplar-based*; 2) *sparsity-based*; 3) *PDE-based;* 4) *Hybrid*. Kawai dan Yokoya (Kawai & Yokoya, 2012) mengajukan sebuah metode *exemplar-based* untuk menyelesaikan masalah *inpainting*. Pendekatan yang diajukan oleh Kawai, menggunakan koherensi spasial untuk menemukan bagian yang memiliki *pattern* yang mirip dengan region yang akan diperbaiki, kemudian menentukan kelayakan nilai piksel baru. Seiring dengan perkembangan teknologi dan ketersediaan data, *inpainting* dapat dilakukan dengan



Gambar 1.1 Image *restoration*. (a) citra sebelum restorasi, (b) citra hasil *inpainting* (Qureshi, dkk., 2017)

menggunakan metode yang berbasis konsep *learning*. Konsep *learning* berarti sistem akan mempelajari/mengekstrak pola dari data, kemudian menyelesaikan masalah yang melibatkan pengetahuan berdasarkan pengetahuan yang diekstrak (I. Goodfellow, dkk., 2016).

Algoritma *learning* yang dikenal saat ini adalah *Artificial Neural Networks* (ANN). ANN menirukan konsep *learning* yang terjadi pada otak makhluk hidup, dan menggunakan teori fundamental di bidang matematika seperti aljabar linier, teori peluang, serta optimasi numerik. Dalam beberapa tahun terakhir, ANN memiliki perkembangan pesat seiring dengan ketersediaan data, komputer yang semakin canggih, serta keberagaman teknik yang dapat digunakan untuk ANN (I. Goodfellow, dkk., 2016).

Deep Neural Network (DNN) adalah jenis ANN yang terdiri dari banyak *hidden layer* atau neuron yang saling terhubung. Pada domain citra, operator konvolusi dapat digunakan pada *layer network* untuk berbagai keperluan dalam mengekstrak informasi dari citra. Jenis DNN yang menggunakan operator konvolusi ini disebut dengan *Convolutional Neural Network* (CNN). Saat ini, CNN merupakan topik yang banyak diteliti oleh peneliti di bidang komputer visi. Krizhevsky, dkk. (Krizhevsky, dkk., 2012) menggunakan CNN untuk menyelesaikan masalah klasifikasi dengan *input* citra, menggunakan data ImageNet sebagai data untuk *training*. Arsitektur *network* yang digunakan menggunakan operator konvolusi. Jumlah data yang digunakan untuk *training* akan mempengaruhi performa dari *network,* karena proses *learning network* dilakukan menggunakan data *training* sebagai acuan.

Tanaka, dkk. (Tanaka, dkk., 2018) mengajukan metode *inpainting* dengan menggabungkan algoritma *inpainting patch-based* dengan CNN. Pada metode yang diajukan Tanaka, CNN digunakan untuk mendeteksi region yang dianggap gagal / rusak secara otomatis, kemudian proses *inpainting* dilakukan dengan metode *patch-based*. Algoritma ini memperoleh hasil yang bagus digunakan jika proses *inpainting* dilakukan untuk merekonstruksi region yang memiliki kesamaan dengan region sekitarnya, seperti *background* laut, atau langit. Masalah muncul ketika proses *inpainting* dilakukan pada region yang memiliki karakteristik yang spesifik, seperti mata, mulut, atau hidung. Region disebut memiliki karakteristik spesifik

karena memiliki karakteristik yang tidak terdapat di region lain. Sehingga, butuh diperlukan pendekatan lain untuk melakukan *inpainting*, agar persepsi dari citra yang direkonstruksi tetap konsisten dan tetap realistis. Masalah *inpainting* pada wajah ini merupakan jenis khusus dari *inpainting*, yaitu *facial inpainting* atau *face completion*.

Jenis *network* yang dapat digunakan pada *inpainting* seperti *Variational Auto-Encoder* (VAE) atau *Generative Adversarial Network* (GAN). GAN pertama kali diajukan oleh Goodfellow, dkk. pada tahun 2014 (I. J. Goodfellow, dkk., 2014). GAN menggunakan model *generative* (G) dan *discriminative* (D), dengan prinsip *two-player minimax game*. Model *generative* digunakan untuk menemukan distribusi dari data, dan model *discriminative* digunakan untuk menentukan probabilitas data yang dihasilkan merupakan data asli atau data sintesis. Prosedur *training* pada model G dilakukan dengan memaksimalkan kemungkinan model D melakukan kesalahan dalam klasifikasi. Pertama kali diajukan, GAN digunakan untuk menghasilkan citra sintesis dari input vektor yang berupa *random noise*. Dari hasil yang diperoleh, model *generative* yang dimiliki oleh GAN dapat mensintesis citra dengan baik sesuai karakteristik data *training* yang digunakan.

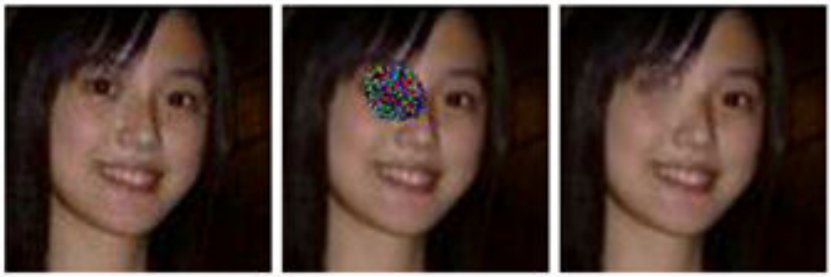
GAN juga digunakan untuk mengembangkan metode VAE dalam mensintesis citra. Metode ini diajukan oleh Hou, dkk. (Hou, dkk., 2019), VAE digunakan sebagai generator, kemudian model *discriminative* GAN digunakan agar data sintesis yang dihasilkan semakin menyerupai distribusi data *training*. Pada model yang diajukan oleh Hou, dkk., digunakan *pre-trained network Visual Geometry Group Network* (VGG-Net) (Simonyan & Zisserman, 2015) sebagai pembanding antara citra *input* dengan citra *output*. Hal ini dilakukan karena *hidden-representation* menyimpan fitur penting yang berhubungan dengan kualitas perseptual, seperti keterkaitan spasial pada citra. Namun, metode yang diajukan oleh Hou, dkk. tidak menjelaskan tentang kemampuan metode yang diajukan ketika digunakan untuk masalah *inpainting*.

Salah satu penelitian yang menggunakan konsep GAN pada masalah *facial inpainting* dilakukan oleh Liao, dkk. (Liao dkk., 2018), dengan konsep *multi-task learning*. Proses rekonstruksi citra bagian wajah dilakukan bersamaan sekaligus dengan klasifikasi diskriminator, serta ekstraksi *heat-map landmark* dan *semantic*

segmentation dari citra wajah. Ketiga hasil dari model digunakan langsung sebagai pembanding yang berikutnya akan digunakan untuk memperbarui bobot pada *network*. Haofu mengajukan skema *concentrated inpainting* untuk diskriminator, sehingga komputasi yang dilakukan oleh model terfokus pada rekonstruksi region yang rusak atau hilang saja, bukan keseluruhan dari citra. Pendekatan Haofu berhasil menghasilkan citra rekonstruksi dengan baik pada masalah *facial inpainting*, namun metode yang diajukan Haofu belum menggunakan VGG-Net. VGG-Net dapat dimanfaatkan untuk menentukan nilai *loss* dari *output* yang dihasilkan. Karena domain pada VGG tidak terbatas pada tiga *channel* seperti pada domain RGB, penggunaan VGG dapat meningkatkan kualitas perseptual citra yang direkonstruksi.

Yijun, dkk. (Yijun, dkk., 2017) memanfaatkan *semantic parsing* untuk meningkatkan kualitas hasil *facial inpainting*. Hasil segmentasi wajah dengan menggunakan *network* tambahan digunakan sebagai panduan untuk merekonstruksi region dengan karakteristik spesifik yang hilang, seperti mata, hidung, atau mulut. Pada penelitian Yijun, dkk., GAN digunakan untuk merekonstruksi bagian yang hilang. Berbeda dengan GAN yang diajukan oleh Ian Goodfellow (I. J. Goodfellow, dkk., 2014), Yijun menggunakan dua buah *discriminator*; *local discriminator* dan *global discriminator*. *Local discriminator* digunakan secara spesifik terbatas pada region yang hilang / rusak, sementara *global discriminator* digunakan untuk citra secara keseluruhan. Metode ini diberi nama *Generative Face Completion* (GFC). Dengan metode GFC, diharapkan hasil citra rekonstruksi yang detail, namun tetap realistis.

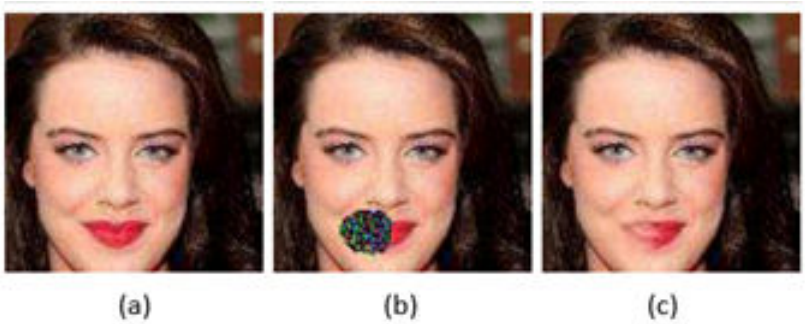
Namun, masalah timbul ketika metode GFC diterapkan pada citra *unaligned face*, yaitu citra wajah yang miring, atau orientasi wajah tidak tegak lurus terhadap sumbu horizontal. Contoh *inpainting* pada citra *unaligned face* disajikan pada Gambar 1.2. Terlihat bahwa ketika melakukan *inpainting* pada citra *unaligned face*, metode GFC gagal merekonstruksi kembali bagian citra wajah yang hilang. Masalah juga timbul ketika bagian yang hilang memiliki keterkaitan dengan bagian sekitarnya. Gambar 1.3 menunjukkan masalah yang timbul ketika bagian yang direkonstruksi memiliki keterkaitan warna dengan bagian sekitar supaya terlihat realistis. Terlihat pada citra hasil rekonstruksi (c), bagian yang direkonstruksi



Gambar 1.2 *Facial inpainting* pada citra wajah *unaligned* (a) citra asli, (b) citra dengan region yang hilang, (c) hasil *inpainting*. (Yijun, dkk., 2017)

memiliki warna yang tidak sesuai dengan bagian sekitarnya. Bagian yang hilang masih memiliki keterkaitan warna dengan bagian sekitar, dan metode GFC gagal mempertahankan keterkaitan warna yang ada agar kualitas perseptual tetap terjaga.

Berdasarkan penelitian yang telah dilakukan sebelumnya, terdapat dua masalah yang timbul ketika *facial inpainting* dilakukan menggunakan *deep learning*, khususnya *Generative Adversarial Network* (GAN) dengan dua *discriminator* yang diajukan Yijun pada metode GFC. Masalah pertama, hasil *inpainting* tidak realistis ketika citra wajah yang menjadi *masukan* merupakan citra wajah *unaligned.* Masalah kedua, bagian hasil *inpainting* menunjukkan bagian yang tidak konsisten dengan *warna* sekitarnya, masalah ini didefinisikan sebagai masalah keterkaitan spasial.



Gambar 1.3 *Facial inpainting* pada citra yang memperhatikan keterkaitan dengan bagian sekitarnya (a) citra asli, (b) citra dengan bagian yang hilang, (c) hasil *inpainting*. (Yijun, dkk., 2017)

Kedua masalah ini akan diatasi dengan tambahan *loss* dari beberapa kriteria, yaitu *feature reconstruction loss* berdasarkan *pre-trained network* VGG-Net (Simonyan & Zisserman, 2015), dan *landmark loss* berdasarkan *face landmark network.* VGG-Net akan digunakan pada penentuan *feature reconstruction loss,* karena VGG-Net dapat mempertahankan *hidden-feature* pada citra, terbukti pada penelitian yang telah dilakukan oleh Hou, dkk. (Hou, dkk., 2019). Sehingga, VGG-Net akan digunakan juga pada proses *inpainting* untuk mempertahankan *hidden-feature* yang berhubungan dengan kualitas perseptual, agar *facial inpainting* tetap dapat dilakukan pada citra wajah yang *unaligned* dan rekonstruksi bagian wajah yang hilang dapat dilakukan dengan memperhatikan keterkaitan spasial.

Penggunaan informasi *landmark* dari wajah dapat membantu meningkatkan kualitas perseptual citra wajah hasil rekonstruksi, seperti yang diajukan oleh Liao, dkk. (Liao dkk., 2018). Namun, pada penelitian ini akan digunakan *landmark* yang lebih umum, yaitu 68 titik *landmark* untuk memperoleh hasil yang lebih akurat, tidak terbatas 5 titik seperti pada metode yang diajukan Liao. *Face landmark network* yang akan digunakan mengadopsi *network* untuk *object contour detection* yang diajukan oleh Jimei, dkk. (Jimei dkk., 2016). Seperti pada metode GFC, *semantic segmentation network* juga akan digunakan untuk meningkatkan kualitas perseptual yang dihasilkan. Namun pada penelitian ini, *network* yang digunakan mengadopsi *network* yang diajukan oleh Yu, dkk. pada BiSeNet (Yu dkk., 2018).

Penelitian ini akan menggunakan GAN dengan lokal dan global *discriminator*, strategi *curriculum learning* (Bengio, dkk., 2009) secara bertahap dengan beberapa *loss* yang didefinisikan. *Curriculum learning* merupakan metode *learning* yang dilakukan secara bertahap dengan meningkatkan tingkat kesulitan dan ukuran *network* yang digunakan. Tingkat kesulitan yang dimaksud adalah definisi *loss* yang digunakan sebagai fungsi tujuan. Fungsi *loss* yang digunakan akan bertambah bersamaan dengan meningkatnya tingkat kesulitan pada tahap *learning* serta *layer* yang digunakan.

1.2 Perumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan sebelumnya, masalah yang dapat dirumuskan adalah:

1. Bagaimana melakukan *inpainting* pada citra wajah yang *unaligned* dengan menggunakan *GAN*?
2. Bagaimana mempertahankan masalah keterkaitan spasial pada proses *inpainting* citra wajah?
3. Bagaimana evaluasi metode yang diajukan untuk mengatasi masalah pada *inpainting* citra wajah?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengembangkan metode *inpainting* pada citra wajah yang *unaligned* yang memerhatikan keterkaitan spasial bagian yang direkontruksi dengan bagian sekitarnya.

1.4 Manfaat Penelitian

Manfaat dari hasil penelitian adalah membantu dalam perbaikan citra wajah yang mengalami kerusakan, serta dapat digunakan pada tahap *pre-processing* untuk penyelesaian tugas lain di bidang komputer visi, seperti *face recognition.*

1.5 Kontribusi Penelitian

Kontribusi pada penelitian ini adalah penggunaan *feature reconstruction loss* berdasarkan VGG-Net, dan *landmark loss* berdasarkan *landmark network* 68 titik pada proses *inpainting* citra wajah untuk mempertahankan keterkaitan spasial serta memungkinkan *inpainting* tetap dapat dilakukan pada citra wajah yang *unaligned*.

1.6 Batasan Masalah

Beberapa batasan yang terdapat pada penelitian ini yaitu:

1. Domain dari penelitian terbatas pada citra wajah manusia.
2. *Masking* yang digunakan berbentuk persegi.

[*Halaman ini sengaja dikosongkan*]

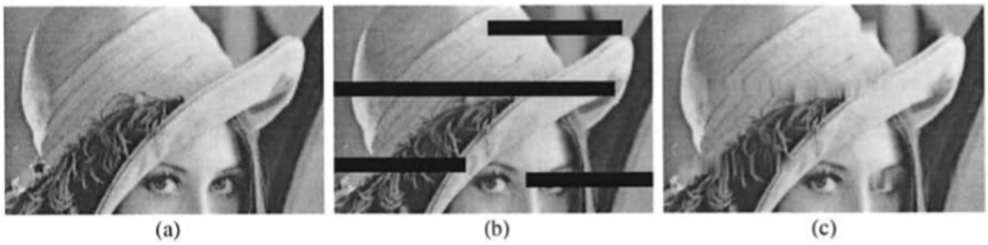
BAB 2
DASAR TEORI

Pada Bab ini akan dijelaskan dasar teori yang berkaitan dengan penelitian yang akan dilakukan.

2.1 *Image Inpainting*

Menurut Furht (Furht, 2008), *image inpainting* didefinisikan sebagai proses mengisi data yang hilang (*missing data*) pada region tertentu dari masukan citra. Tujuan dari proses ini adalah untuk merekonstruksi bagian yang rusak atau hilang dari suatu citra sedemikian sehingga bagian hasil *inpainting* (*inpainted region*) tidak dapat diketahui oleh pihak pengamat, bahwa telah dilakukan proses *inpainting*. *Image inpainting* mungkin dilakukan di berbagai domain, seperti citra pemandangan alam, atau objek makhluk hidup. *Facial inpainting*, merupakkan salah satu sub-domain dari *inpainting,* dimana domain masalah merupakan citra wajah. Artefak mungkin saja muncul setelah proses *inpainting* dilakukan, seperti region yang direkontruksi terlihat tidak realistis. Artefak yang umumnya paling sering ditemui seperti region yang buram (*blurring*), tekstur region tidak konsisten, atau *edge* yang terputus (*disconnected edges*) (Qureshi, dkk., 2017)

Region hilang yang akan dilakukan proses *inpainting* dapat disebabkan oleh beberapa penyebab, seperti *packet loss* saat proses transmisi nirkabel pengiriman citra, kerusakan yang tidak disengaja seperti retakan, goresan, kotoran atau kesalahan yang mungkin terjadi pada proses digitalisasi gambar. Proses *inpainting* juga dapat dilakukan untuk penghilangan suatu objek pada citra seperti logo, cap tanggal yang mungkin dibubuhkan pada citra, tulisan, ataupun

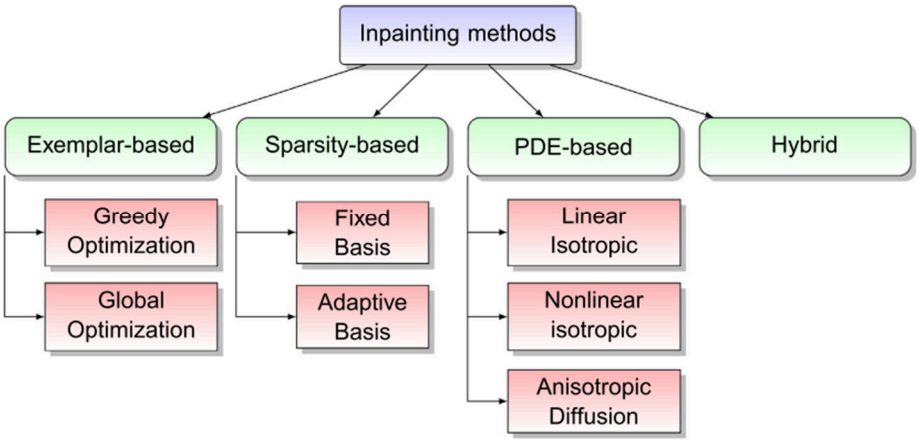


Gambar 2.1 *Image inpainting.* (a) citra asli, (b) citra yang rusak karena proses trasmisi, (c) citra hasil rekonstruksi dengan teknik *inpainting*. (Furht, 2008)

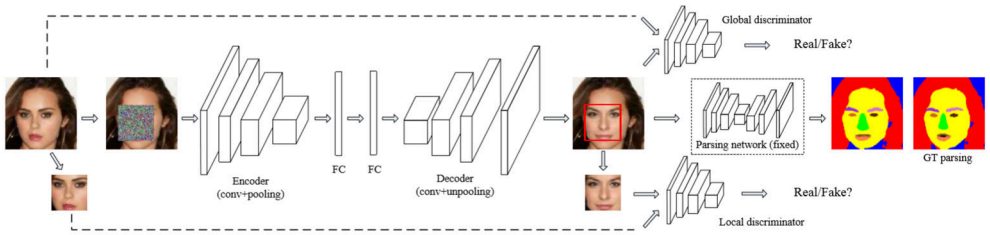
menghilangkan objek orang pada citra. Umumnya, setelah pengguna memilih region mana yang akan dilakukan proses *inpainting*, algoritma *inpainting* akan secara otomatis memperbaiki region yang dipilih dengan menggunakan metode yang berkaitan. Qureshi, dkk. menyatakan bahwa region yang direkonstruksi akan diestimasi menggunakan informasi berdasarkan piksel dari region yang diketahui dengan asumsi piksel yang diketahui memiliki struktur geometris dan sifat statistik yang sama (Qureshi, dkk., 2017). Contoh *inpainting* pada citra disajikan pada Gambar 2.1.

Salah satu faktor penentu kualitas perseptual dari citra adalah korelasi atau keterkaitan spasial. Keterkaitan spasial didefinisikan sebagai keterkaitan antara satu piksel dengan piksel sekitarnya yang mempengaruhi kualitas perseptual. Ketika proses *inpainting* dilakukan dengan mengabaikan keterkaitan spasial, citra yang dihasilkan mungkin saja memiliki kualitas perseptual yang tidak maksimal. Hasil *inpainting* masih menunjukkan artefak, sehingga bagian yang direkonstruksi masih bisa dibedakan dengan bagian lainnya. Gambar 1.3 menunjukkan proses *inpainting* pada bagian dengan keterkaitan spasial. Terlihat pada Gambar 1.3 (b), bagian yang hilang adalah bagian dari bibir berwarna merah. Bagian yang hilang ini dikatakan memiliki keterkaitan spasial dengan piksel sekitarnya karena penentuan hasil *inpainting* harus memperhatikan bagian bibir sekitarnya.

Terdapat beberapa teknik atau metode untuk melakukan *inpainting*. Menurut Qureshi, dkk., metode untuk melakukan *image inpainting* dapat dibagi



Gambar 2.2 Pembagian metode *inpainting* (Qureshi, dkk., 2017)



Gambar 2.3 Arsitektur *Network Generative Face Completion* (GFC) (Yijun, dkk., 2017)

menjadi 4 bagian, yaitu: 1) e*xemplar-based*; 2) *sparsity-based*; 3) *PDE-based;* 4) *Hybrid* (Qureshi, dkk., 2017). Pembagian metode ini disajikan dalam bentuk diagram pada Gambar 2.2. Pada metode *exemplar-based*, proses *inpainting* dilakukan dengan mensintesis tekstur *missing region* menggunakan informasi dari region yang diketahui dengan ketentuan memiliki struktur yang serupa. Metode ini memanfaatkan konsep optimasi, disebut *greedy optimization* jika proses *inpainting* dilakukan secara *greedy*, disebut *global optimization* jika menggunakan pendefinisian suatu fungsi tujuan yang akan diminimasi secara iteratif. *Exemplar-based* memiliki keunggulan jika *inpainting* dilakukan pada region yang besar, namun dengan *computational cost* yang tinggi. Artefak yang sering muncul pada metode *exemplar-based* adalah *edges* yang terputus (*disconnected edges*) serta region yang dihasilkan tidak konsisten dengan region sekitar dan terlihat. Penggunaan NN, dengan suatu fungsi objektif yang akan diminimasi, merupakan salah satu contoh metode *inpainting exemplar-based*.

Contoh arsitektur NN yang dibuat untuk menyelesaikan *facial inpainting* adalah *Generative Face Completion* (GFC), diajukan oleh Yijun dkk (Yijun, dkk., 2017). Arsitektur ini menggunakan model GAN, dengan *generator* merupakan *Variational Auto Encoder*, dua buah *discriminator,* serta *network semantic parsing* yang digunakan untuk meningkatkan kualitas perseptual dari hasil *inpainting*. Bentuk aristektur *network* GFC disajikan pada Gambar 2.3. Pada penelitian ini, penggunaan *network semantic parsing* terbukti dapat meningkatkan kualitas perseptual dari citra hasil *inpainting*.

Metode *sparsity-based*, memanfaatkan representasi *sparse* dari citra pada suatu basis tertentu, seperti *Discrete Cosine Transform* (DCT), *Discrete Wavelete*

Transform (DWT). Ide dasar dari *sparsity-based* adalah asumsi bahwa bagian *missing region* dan komplemennya pada citra memiliki representasi *sparse* yang sama, sehingga dapat dilakukan estimasi untuk mengisi *missing region* (Qureshi, dkk., 2017). Proses *inpainting* dilakukan dengan melakukan *transformasi* ke domain representasi yang lain.

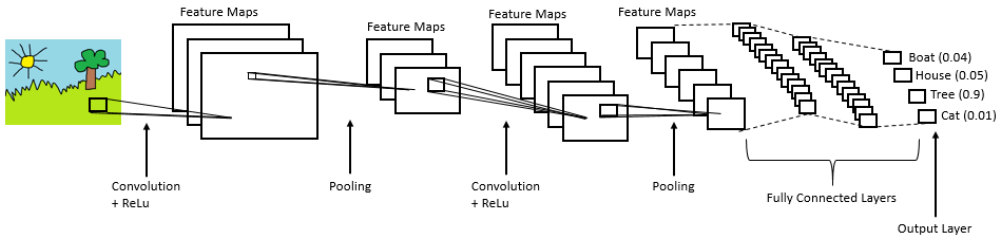
Partial Differential Equation (PDE)-based, memanfaatkan konsep persamaan diferensial parsial untuk melakukan estimasi nilai pada *missing region*. *PDE-based* ini dibagi menurut *flow* difusi yang digunakan, yaitu *linear, nonlinear, isotropic* dan *anisotropic*. Keunggulan dari metode *PDE-based* ini terletak pada waktu komputasi yang lebih rendah dibanding metode lainnya, serta kemampuannya yang bagus untuk memperbaiki kerusakan atau *missing region* dengan bentuk garis atau lurus panjang. *PDE-based* tidak bisa mengatasi *missing region* dengan ukuran besar, hasil rekonstruksi cenderung buram. Sementara Sementara metode *hybrid-based* menggabungkan antara beberapa metode yang telah didefinisikan sebelumnya.

Kualitas dari hasil rekonstruksi akan bergantung dari metode yang digunakan, dan hasil rekonstruksi dengan metode yang berbeda mungkin menghasilkan citra rekonstruksi yang berbeda, secara signifikan. Hal ini disebabkan karena teknik *inpainting* mungkin memfokuskan restorasi pada tekstur saja, atau terhadap struktur saja.

2.2 *Convolutional Neural Network* (CNN)

CNN adalah salah satu jenis khusus dari NN, untuk memproses data yang memiliki susunan topologi seperti *grid* (I. Goodfellow, dkk., 2016). Dimisalkan sebuah data *time-series*, data *time-series* merupakan data yang berbentuk *grid* 1D, yang menyatakan sampel data terhadap satuan waktu. Sementara, data gambar atau citra, dapat dilihat sebagai data *grid* 2D yang terdiri dari nilai piksel. Penamaan *"convolutional"* berasal dari penggunaan operator konvolusi pada salah satu *layer* NN yang menggantikan operasi aljabar matriks biasa. Tahapan pada *layer* CNN umumnya terbagi menjadi 3 tahap, yaitu: 1) *convolution stage*, 2) *detector stage,* 3) *pooling stage* (I. Goodfellow, dkk., 2016).

Tahap pertama merupakan tahap *convolution stage*. Pada tahap ini terdiri dari operasi konvolusi dengan kernel. Kernel pada CNN merupakan *array*



Gambar 2.4 Contoh CNN dalam menyelesaikan masalah klasifikasi citra (Stoll, 2017)

multidimensional yang menjadi operator untuk melakukan operasi tertentu dengan *array input*. Nilai yang terdapat pada kernel merupakan nilai yang diperoleh dari hasil *learning.* Hasil operasi aljabar antara *input* dengan kernel disebut dengan *feature map*. Proses operasi antara *input* dengan kernel ini disebut dengan proses ekstraksi fitur ke dalam *feature map*.

Tahap kedua merupakan tahap *detector stage*. Hasil dari tahap sebelumnya yang merupakan operasi linier, dijadikan sebagai *input* ke dalam suatu fungsi aktivasi nonlinier, seperti *rectified linear activation*. Tahap berikutnya merupakan tahap *pooling*. Pada tahap ini, dilakukan reduksi dimensi dari *input array* yang masuk dengan memanfaatkan ringkasan statistik, seperti *max-pooling, sum-pooling,* atau *average-pooling*, dengan harapan dimensi data menjadi lebih kecil (reduksi) namun tanpa menghilangkan informasi penting dari data yang diterima.

Setelah menggunakan beberapa *convolutional layer*, umumnya untuk kasus klasifikasi digunakan *layer* tambahan berupa *fully-connected layer* yang melakukan *flattening* dari matriks menjadi sebuah vektor, dan selanjutnya digunakan untuk melakukan tugas klasifikasi dengan fungsi aktivasi tertentu. Proses klasifikasi dengan CNN ini diilustrasikan pada Gambar 2.4.

2.3 *Auto-Encoder* (AE)

Auto-Encoder (AE) merupakan salah satu algoritma *self-supervised*, yaitu algoritma dengan target *output* yang dihasilkan adalah *input* itu sendiri (Chollet, 2016). Ng, dkk. (Ng, dkk., n.d.) mendefinisikan AE sebagai sebagai salah satu jenis *neural network*. Diberikan sebuah himpunan data training $\{x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(n)}\}$ dengan $x^{(i)} \in \Re^n$, AE adalah algoritma *learning* dengan *backpropagation*, dengan tujuan nilai dari *ouput* akan sama dengan *input*. Gambar 2.5 merupakan ilustrasi

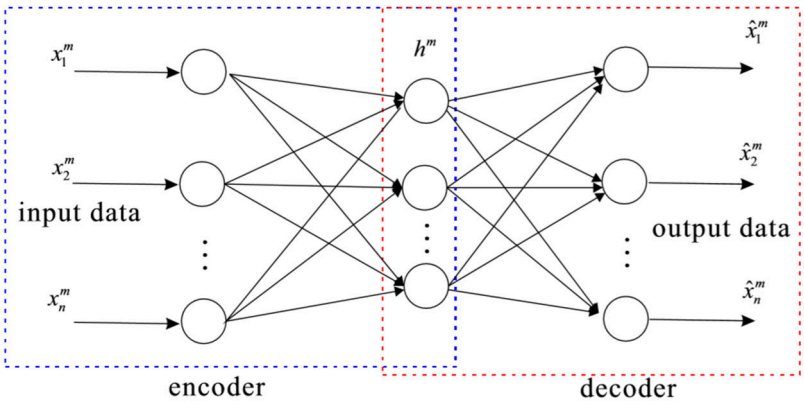
sederhana dari AE. Berdasarkan Gambar 2.5, tujuan dari AE adalah sebuah pemetaan sedemikian sehingga $x_i = \hat{x}_i$.

Network AE terdiri dari dua bagian *network*, yaitu *encoder* dan *decoder*. Misal diberikan *input* berupa citra berukuran 10×10, maka terdapat 100 unit *input* pada L_1, dengan 50 *hidden units* pada sebuah *hidden layer* L_2. Dalam kasus ini, AE melakukan proses pembelajaran untuk pemadatan (*compress*) terhadap *input* yang diberikan. Proses ini disebut dengan *encoding*, melakukan perubahan dari *input* input ke dalam representasi yang lain. *Network encoder* adalah *network* yang melakukan pemetaan dari *input* x menjadi vektor h^m, didefinisikan pada persamaan (2.1). Vektor $z = h^m$ hasil transformasi disebut dengan variabel *latent.*

$$h^m = e(x) \tag{2.1}$$

$$\hat{x} \quad = d(h^m) \tag{2.2}$$

Kebalikan dari *encoder, network decoder* adalah *network* yang bertugas melakukan pemetaan dari variabel *latent* z ke *output* $\hat{x}$ dengan $\hat{x} = x$, dinyatakan dalam persamaan (2.2). Sehingga, secara keseluruhan *network* AE adalah network yang mencari fungsi $e(x)$ dan $d(h^m)$ sedemikian sehingga diperoleh fungsi identitas $(d \circ e)(x) \approx x$, dengan vektor antara h^m. Goodfellow, dkk. (I. Goodfellow, dkk., 2016) menyebut AE ini sebagai algoritma *representation learning* karena proses *learning* dilakukan untuk menangkap suatu representasi dari *input.* Jenis AE disebut *undercomplete* AE jika terdapat proses reduksi dimensi dari dimensi input menjadi suatu representasi dengan dimensi yang lebih kecil, Proses



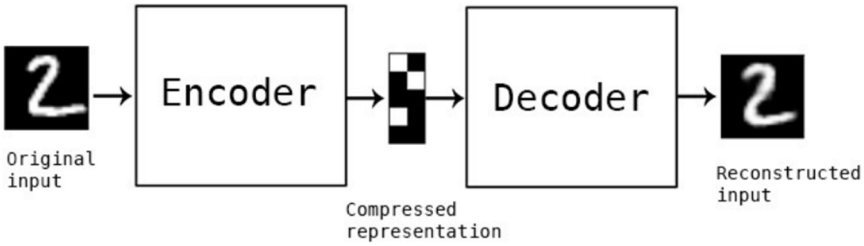
Gambar 2.5 Model *network Auto Encoder* (AE) (T. Ma, dkk., 2016)

learning dilakukan dengan melakukan minimasi nilai *loss function* L_R, pada persamaan (2.3). L_R memberi *penalty* jika nilai *output* berbeda dari *input* x, seperti *mean-squared error* (MSE). Jenis *loss* ini disebut dengan *reconstruction loss*.

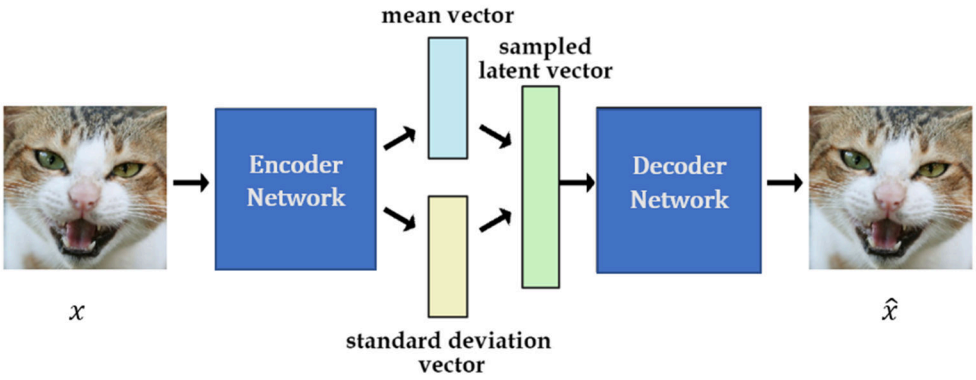
$$L_R = \Big(x, \big(d\big(e(x)\big)\big)\Big) \tag{2.3}$$

AE dapat diterapkan pada *input* berupa citra, menggunakan operator konvolusi. Sehinnga, AE dengan operator konvolusi ini dapat disebut dengan *convolutional auto-encoder*. Ilustrasi alur kerja dari AE pada citra ditampilkan pada Gambar 2.6. Penerapan dari AE saat ini adalah pada masalah penghilangan *noise* dari data, serta reduksi dimensi data. AE mampu mengenali proyeksi data lebih baik dari *Principial Component Analysis* (PCA) atau teknik dasar yang lain (Chollet, 2016).

Salah satu pengembangan dari AE, adalah *Variational Auto-Encoder* (VAE). Perbedaan mendasar dari AE dengan VAE terletak pada adanya parameterisasi sebelum transformasi menjadi vektor *latent* z, serta batasan (*constraint*) atau *loss* yang digunakan. Sehingga, berbeda dari AE yang hanya menentukan fungsi $x \mapsto e(x) = z$, VAE melakukan pemetaan dari x menjadi dua buah variabel, yaitu variabel z_μ dan z_σ, disebut *mean vector* dan *standard deviation vector*. Kemudian, ditentukan sebuah fungsi yang memetakan (z_μ, z_{sigma}) ke sebuah nilai di variabel *latent* z. Dengan kata lain, pada VAE terdapat proses *learning* sebuah model z, yang merupakan penentu probabilitas distribusi dari data *training*, pada persamaan (2.4). Variabel $\epsilon \sim \mathcal{N}(0,1)$ merupakan *auxiliary noise* (Kingma & Welling, 2014). Sehingga, dengan adanya model probabilitas distribusi ini, VAE disebut dengan *generative model*, karena dapat digunakan untuk



Gambar 2.6 Alur kerja AE pada domain citra (Chollet, 2016)



Gambar 2.7 Model *Variational Auto-Encoder* (VAE) (Frans, 2016)

melakukan sintesis data berdasarkan suatu inputan acak dengan distribusi normal. Model VAE disajikan pada Gambar 2.7.

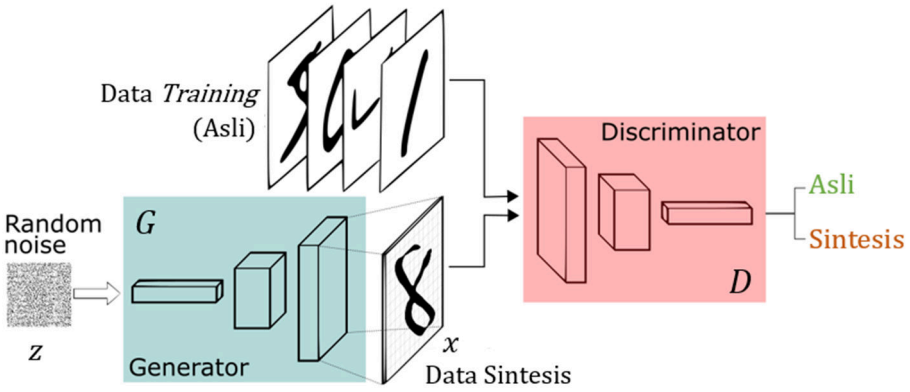
$$z = z_\mu + z_\sigma \epsilon \tag{2.4}$$

Selain adanya paremeterisasi, faktor pembeda VAE dengan AE sederhana adalah pada kriteria *loss* yang digunakan. Selain *reconstruction loss* seperti pada persamaan (2.3), VAE memiliki *loss* lain yaitu *Kullback–Leibler divergence* (*KL-divergence*) *loss*. *KL-divergence* menyatakan perbedaan dari dua buah distribusi, dinyatakan dalam persamaan (2.5). Sehingga, total *loss* yang menjadi fungsi objektif VAE menjadi $L = L_R + \mathfrak{D}_{KL}$.

$$\mathfrak{D}_{KL} = \frac{1}{2}\left[\sum_{i=1}^{n} z_{\mu,i}^2 + \sum_{i=1}^{n} z_{\sigma,i}^2 - \sum_{i=1}^{n} \log\left(z_{\sigma,i}^2 + 1\right)\right] \tag{2.5}$$

2.4 *Generative Adversarial Network* (GAN)

Network GAN pertama kali diajukan oleh Ian Goodfellow, dkk. (I. J. Goodfellow, dkk., 2014). Arsitektur dari GAN terdiri dari dua model, yaitu model *generative* (G) dan *discriminative* (D). Kedua model ini akan saling melakukan proses *training* dengan konsep permusuhan (*adversarial*). Model *generative, G,* bertugas untuk melakukan sintesis data x dari *input random* z. Kemudian, model *discriminator*, D, menjadi *adversarial* dari G, bertugas untuk menentukan data yang dihasilkan oleh model G merupakan data asli atau data sintesis. Umumnya, model D yang digunakan pada GAN adalah *network* untuk melakukan proses klasifikasi, sehingga hasil dari $D(x)$ berupa peluang keanggotaan. Bentuk arsitektur GAN



Gambar 2.8 *Generative Adversarial Network* (GAN)

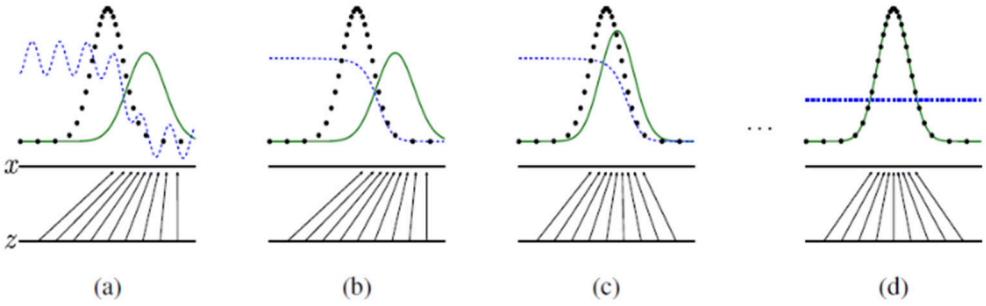
secara sederhana diilustrasikan pada Gambar 2.8. Nilai *loss* yang dihasilkan nantinya dipakai untuk pembaruan bobot dari masing-masing *network*.

Prinsip *adversarial* ini menyerupai konsep *mini-max game* dua pemain. Proses *training* pada D dilakukan dengan tujuan model D dapat mengklasifikasikan data asli atau data sintesis, yang berarti tingkat keberhasilan model D tinggi. Sementara model G bertujuan untuk menghasilkan data sintesis $x = G(z)$ sedemikian sehingga model D memiliki tingkat keberhasilan seminimum mungkin. Secara implisit, G akan mendefinisikan sebuah distribusi probabilitas p_g, karena distribusi data sintesis $G(z)$ diperoleh dengan $z \sim p(z)$.

Sehingga, fungsi objektif dari model GAN ini dapat dinyatakan dalam persamaan (2.6).

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \qquad (2.6)$$

Dimana p_{data} menyatakan distribusi data asli, $x \sim p_{data}(x)$ menyatakan sampel x mengikuti aturan distribusi $p_{data}(x)$. Tujuan dari proses *training* GAN adalah agar distribusi p_g semakin serupa dengan p_{data} sehingga D kesulitan dalam membedakan sebuah sampel dmengikuti distribusi p_{data} atau p_g.



Gambar 2.9 Proses *training* GAN (I. J. Goodfellow, dkk., 2014)

Secara konseptual, proses *training* yang terjadi pada GAN diilustrasikan pada Gambar 2.9. Garis biru, hitam putus-putus, dan hijau menyatakan distribusi dari model *discriminative* D, data asli p_{data}, dan model generative G, yaitu p_g. Garis atas, z, menunjukkan domain data *random input* dihasilkan, sementara garis bawah, x, menunjukkan domain data sintesis x. Tanda panah yang mengubungkan garis z dengan garis x merupakan pemetaan $x = G(z)$.

Pada tahap awal, (a) D yang merupakan diskriminator yang akurat, dapat membedakan antara data asli dan data sintesis dengan baik. (b) Saat proses *training* model D dilakukan, D membedakan data asli dengan sintesis. (c) Supaya G dapat mengelabui D dalam klasifikasi, maka G menghasilkan data dengan distribusi yang semakin mirip dengan p_{data}. (d) Kondisi yang tercapai ketika $p_{data} \approx p_g$, yang berarti data sintesis dari G menyerupai distribusi data asli. Kondisi ini adalah

Gambar 2.10 Hasil dari GAN dengan data *training* TFD. Kolom paling kanan menunjukkan hasil model G setelah melalui proses *training* (I. J. Goodfellow, dkk., 2014)

kondisi ideal yang diharapkan, yaitu $D(x) = \frac{1}{2}$. Hasil dari GAN (I. J. Goodfellow, dkk., 2014) diilustrasikan pada Gambar 2.10.

2.5 *Curriculum Learning*

Ketika menyelesaikan masalah atau tugas yang kompleks, pengajar cenderung membuat sebuah *curriculum*. Pelajar, secara perlahan menerima pengetahuan dan mempelajari dari hal-hal yang sederhana hingga tugas semakin kompleks. Dengan metode seperti ini, diharapkan pelajar dapat menggunakan pengetahuan sederhana yang telah diperoleh sebelumnya untuk menyelesaikan masalah dengan tingkat kesulitan semakin tinggi atau semakin kompleks. Konsep ini menjadi dasar dalam *curriculum learning*. Salah satu contoh penerapan di masalah klasifikasi, proses pembelajaran diawali dengan memberi contoh-contoh yang jelas di awal proses. Lama-kelamaan, proses pembelajaran dilakukan dengan menggunakan contoh-contoh yang semakin ambigu serta semakin sulit untuk diklasifikasikan (Hacohen & Weinshall, 2019). *Curriculum learning* merupakan metode *learning* yang dilakukan secara bertahap dengan meningkatkan tingkat kesulitan. *Curriculum learning* dapat disebut metode berkelanjutan (*continuation*)

Misal diberikan sekumpulan data $\mathbb{X} = \{X_i\}_{i=1}^{N} = \{(x_i, y_i)\}_{i=1}^{N}$, dengan $x_i \in \mathbb{R}^d$ menyatakan satu titik data, dan y_i merupakan label yang berkaitan dengan data x_i. Didefinisikan sebuah *network* f_θ yang melakukan pemetaan dari $x_i \rightarrow y_i$ dengan parameter *network* $\boldsymbol{\theta}$, serta *cost function* $J(\boldsymbol{\theta})$. *Cost function* baru dibentuk, yaitu $\{J^{(0)}, J^{(1)}, \dots, J^{(n)}\}$, diatur sedemikian rupa sehingga tingkat kesulitan meningkat secara perlahan. $J^{(0)}$ merupakan *cost function* dengan tingkat kesulitan terendah, sehingga mudah untuk menemukan titik minimum, kemudian $J^{(n)}$ merupakan *cost function* dengan tingkat kesulitan tertinggi yang menjadi $J(\boldsymbol{\theta})$. $J^{(i)}$ dinyatakan lebih mudah dari $J^{(i+1)}$ ketika $J^{(i)}$ menghasilkan nilai yang lebih baik dari $J^{(i+1)}$ dengan parameter $\boldsymbol{\theta}$. Tujuan dari metode berkelanjutan ini adalah untuk mengatasi masalah optimasi ketika bertemu dengan minimum lokal, sehingga titik optimum global tetap dapat ditemukan meskipun ditemukan banyak minimum lokal (I. Goodfellow, dkk., 2016).

2.6 Metode Evaluasi

Terdapat beberapa metrik yang dapat digunakan untuk memeriksa kualitas dari citra hasil yang diperoleh. Metrik sederhana yang umum digunakan seperti *Peak Signal to Noise Rasio* (PSNR). Metrik ini digunakan untuk membandingkan citra hasil suatu *network* dengan *ground truth.* Misal didefinisikan citra hasil *network* sebagai I_o, serta citra *ground truth* I_{GT}, W dan H menyatakan lebar dan tinggi (ukuran) dari gambar, PSNR didefinisikan dalam persamaan (2.7). *Mean Square Error* (MSE) merupakan jarak *euclidean* antara citra *ground truth* dengan citra *output network,* dinyatakan dalam persamaan (2.8). Nilai PSNR yang kecil menyatakan perbedaan yang tinggi antara I_o dan I_{GT}, secara numerik.

$$PSNR(I_o,\ I_{GT}) = 10\log_{10}\left(\frac{255^2}{MSE(I_o,\ I_{GT})}\right) \tag{2.7}$$

$$MSE(I_o,\ I_{GT}) = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}\left(I_o(i,j) - I_{GT}(i,j)\right)^2 \tag{2.8}$$

Namun, PSNR dan MSE tidak dapat digunakan sebagai pengukur utama tingkat keberhasilan hasil *inpainting*, karena tidak memiliki korelasi kuat dengan kualitas perseptual dari citra. Sementara, tujuan dari *inpainting* adalah melakukan restorasi pada bagian yang hilang sedemikian sehingga bagian yang direstorasi terlihat layak dengan bagian sekitarnya serta tidak terdapat artefak yang terlihat (Qureshi, dkk., 2017). Untuk kasus *inpainting*, dapat digunakan metrik lain seperti *Similarity Indeks Measure* (SSIM). Metrik ini diajukan oleh Zhou, dkk. (Zhou, dkk., 2004) untuk menyatakan jarak kemiripan antara dua citra. Penentuan nilai SSIM dinyatakan pada persamaan (2.9) dan (2.10). Notasi μ_I menyatakan rata-rata nilai piksel pada citra I, sementara σ_I menyatakan simpangan baku dari nilai piksel-piksel pada citra. Terdapat 3 komponen utama, yaitu $l(I_o,\ I_{GT}), c(I_o,\ I_{GT}), s(I_o,\ I_{GT})$, yang menyatakan *luminance*, contrast, serta structure. Masing-masing komponen menyatakan perbandingan antara kedua citra pada domain *luminance*, kontras, serta struktur. Konstanta C_1, C_2, C_3 merupakan konstanta yang mencegah agar pembagi tidak sama dengan 0. Nilai SSIM terdapat pada range [0,1], nilai 0 menyatakan tidak ada korelasi antara citra I_o dengan I_{GT}, sementara nilai 1 menyatakan sebaliknya.

$$SSIM(I_o, I_{GT}) = l(I_o, I_{GT})c(I_o, I_{GT})s(I_o, I_{GT}) \tag{2.9}$$

$$\begin{cases} l(I_o, I_{GT}) & = \dfrac{2\mu_{I_O}\mu_{I_{GT}} + C_1}{\mu_{I_O}^2 + \mu_{I_{GT}}^2 + C_1} \\ c(I_o, I_{GT}) & = \dfrac{2\sigma_{I_O}\sigma_{I_{GT}} + C_2}{\sigma_{I_O}^2 + \sigma_{I_{GT}}^2 + C_2} \\ s(I_o, I_{GT}) & = \dfrac{\sigma_{I_O I_{GT}} + C_3}{\sigma_{I_O}\sigma_{I_{GT}} + C_3} \end{cases} \tag{2.10}$$

Selain evaluasi secara kuantitatif, evaluasi secara kualitatif juga dilakukan pada penentuan evaluasi performa algoritma *inpainting*. Hal ini karena nilai kuantitatif berupa PSNR dan SSIM tidak cukup digunakan sebagai kriteria evaluasi (Yijun dkk., 2017). Evaluasi kualitatif dilakukan dengan cara melakukan observasi manual oleh beberapa pengamat, kemudian dihitung rerata dari beberapa penilaian yang telah diperoleh. Metode evaluasi seperti ini dilakukan oleh Hays dan Efros. Hasil *inpainting* ditunjukkan ke beberapa pengamat kemudian setiap pengamat memberikan penilaian secara subjektif untuk mengidentifikasi hasil *inpainting* (Hays & Efros, 2007).

[*Halaman ini sengaja dikosongkan*]

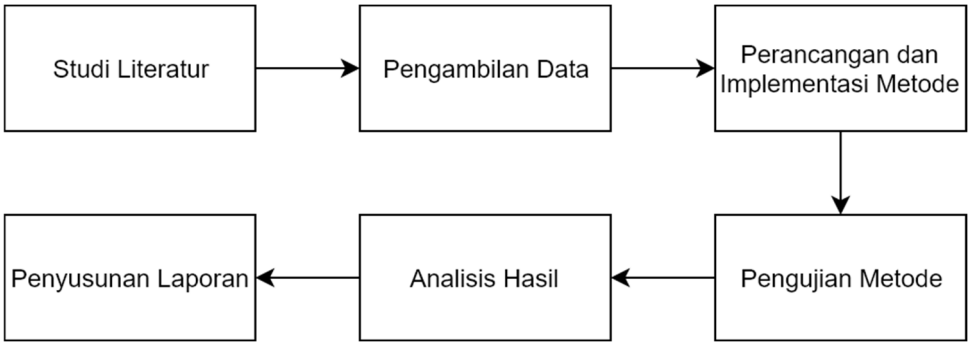
BAB 3
METODOLOGI PENELITIAN

Pada Bab ini akan dijelaskan metode pada penelitian yang akan dilakukan untuk mencapai tujuan dari penelitian yang sebelumnya telah didefinisikan dari perumusan masalah. Tahapan yang akan dilalui pada penelitian ini terdiri dari 1) studi literatur, 2) pengambilan data, 3) perancangan dan implementasi metode, 4) pengujian metode, 5) analisis hasil yang diperoleh, 6) penyusunan laporan. Tahapan ini disajikan dalam diagram alir pada Gambar 3.1.

3.1 Studi Literatur

Studi literatur dilakukan pada domain penelitian *deep learning*, *image processing, inpainting*, serta komputer visi untuk mengetahui perkembangan saat ini di bidang tersebut. Sumber literatur yang dimaksud berupa artikel dari jurnal ilmiah, prosiding internasional, buku, atau berupa situs daring. Berdasarkan studi literatur yang telah dilakukan, diperoleh beberapa informasi yang mendasari penelitian ini akan dilakukan.

Salah satu masalah yang menarik di bidang *image processing*, yaitu *image inpainting*. Seiring dengan berkembangnya perkembangan teknologi dan ketersediaan data, masalah *inpainting* ini memungkinkan untuk diselesaikan dengan menggunakan konsep *deep learning*. Salah satu konsep *deep learning* yang dapat digunakan adalah *Generative Adversarial Network* (GAN). GAN dipilih karena hasil dari metode yang baik untuk masalah *generative* seperti *inpainting*, serta metode ini merupakan metode yang baru di dunia penelitian, ditemukan pada



Gambar 3.1 Diagram alir penelitian

tahun 2014 (I. J. Goodfellow, dkk., 2014). Hal yang mungkin dikembangkan dari penerapan GAN pada masalah *inpainting* citra wajah terletak pada kekurangan yang ditemui ketika proses *inpainting* dihadapkan pada citra wajah yang *unaligned* atau bagian yang masih memiliki keterkaitan spasial dengan bagian sekitar yang hilang (Yijun, dkk., 2017), sesuai yang diilustrasikan pada Gambar 1.2 dan Gambar 1.3. Masalah keterkaitan spasial dan citra wajah *unaligned* ini dapat diatasi dengan penggunaan *feature reconstruction loss* untuk mempertahankan *hidden-feature* pada citra (Hou, dkk., 2019, 2017), *semantic parsing* dan *landmark* dapat digunakan untuk meningkatkan kualitas perseptual citra wajah yang direkonstruksi (Liao dkk., 2018).

3.2 Deskripsi *Dataset*

Data yang digunakan pada penelitian ini merupakan data sekunder dari *CelebFaces Attributes Dataset* (CelebA), diambil dari penelitian Ziwei, dkk. (Ziwei, dkk., 2015). *Dataset* CelebA terdiri dari 10,177 identitas, dengan setiap identitas memiliki 20 gambar. Sehingga, total gambar pada CelebA adalah 202,599 gambar. Setiap gambar pada CelebA memiliki 40 anotasi atribut, seperti bentuk wajah oval, menggunakan kacamata, senyum, rambut bergelombang, dan lain-lain, dinyatakan dalam bilangan biner. Selain anotasi atribut, data CelebA memiliki anotasi lokasi lima *landmark* dari wajah, yaitu lokasi mata kiri, mata kanan, sudut kiri mulut, sudut kanan mulut, dan hidung. Contoh gambar dari CelebA disajikan pada Gambar 3.2.

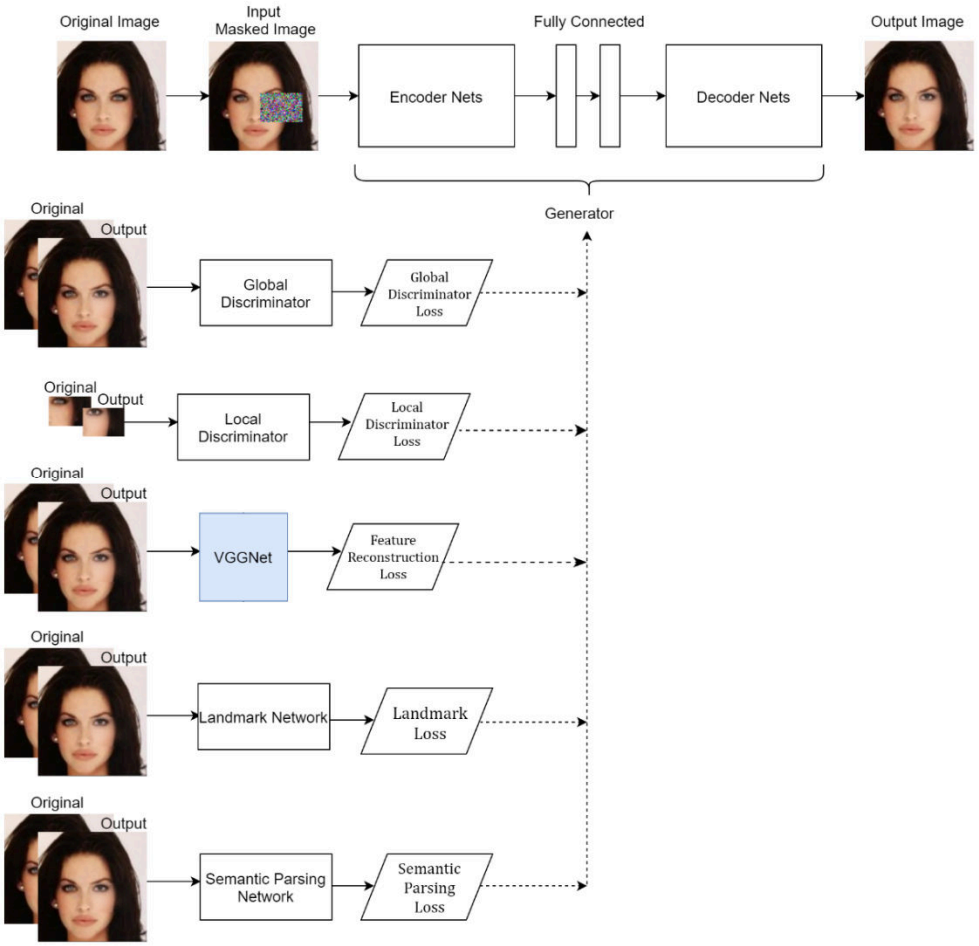
Sesuai yang diilustrasikan pada Gambar 3.2, gambar pada CelebA memuat bagian selain kepala, sehingga proses *cropping* perlu dilakukan agar *region of*

Gambar 3.2 Contoh gambar pada CelebA

interest (ROI) dibatasi pada bagian wajah saja. Proses *alignment* yang dilakukan sebelum *training network* dapat mengakibatkan *network* gagal menghasilkan citra yang sesuai ketika menghadapi citra *unaligned* sehingga proses *cropping* dilakukan cukup dengan ketentuan memuat *landmark location* dari wajah saja, tanpa melakukan *alignment* agar model dapat menangkap karakteristik data secara *general* tanpa *overfitting* (Liao dkk., 2018). Augmentasi data juga dilakukan dengan *random shifting, scaling*, serta *rotasi*. Proses *resize* citra dilakukan hingga citra berukuran $128 \times 128 \times 3$ piksel (Yijun, dkk., 2017).

3.3 Perancangan dan Implementasi Metode

Arsitektur yang diajukan secara umum diilustrasikan pada Gambar 3.3. Sebelum *training* dimulai, *preprocessing* dilakukan pada CelebA dengan operasi



Gambar 3.3 Arsitektur *network* yang diajukan

cropping, dan *resize*. Kemudian, pemberian *masking* pada *input* citra asli (*original image*) yang telah dilakukan pada data citra hasil *preprocess*, I. Citra input yang telah diberi *masking*, I_m, menjadi *input* pada *network*. Berikutnya, akan dilakukan *training* dengan menggunakan lima *network*, yaitu: 1) *generator*, 2) VGG-Net*,* 3) *discriminator,* 4) *landmark network,* dan 5) *semantic parsing network*. Masing-masing *network* ini akan menghasilkan *loss* yang akan digunakan untuk melakukan *update* bobot pada *network generator*. Strategi *learning network* dilakukan dengan metode *curriculum learning*, sehingga secara keseluruhan proses *training* dibagi menjadi beberapa tahap dengan ukuran *network* dan jumlah parameter yang berbeda pada tiap tahapannya. Pada penelitian ini, proses *learning* terbagi menjadi tiga tahap.

Pada tahap pertama, *update* bobot pada *generator* dilakukan dengan dua kriteria *loss* saja, yaitu *K-L divergence loss* ($\mathcal{L}_{KL}$) yang diperoleh di dalam *network generator,* dan *feature reconstruction loss* ($\mathcal{L}_f$).Pada penelitian ini *feature reconstruction loss* tidak menggunakan jarak *euclid* antara citra hasil rekonstruksi dengan citra hasil *inpainting* pada domain spasial RGB, melainkan jarak *euclid* antara kedua citra tersebut pada domain fitur VGG-Net.

Tahap kedua, dua *discriminator network* digunakan untuk pendefinisian *loss*, *loss discriminator* $\mathcal{L}_d$ dinyatakan dalam penjumlahan berbobot dari *loss discriminator local* dan *discriminator global*, yaitu L_{dl} dan L_{dg}. Penggunaan konsep *adversarial* bertujuan supaya *network* dapat mengenali pola distribusi data citra masukan secara implisit, memberikan detail yang lebih bagus untuk mengatasi kekurangan VAE yang cenderung menghasilkan citra buram/*noise* dan tidak tajam.

Tahap ketiga, *loss landmark network* dan *semantic parsing network* digunakan pada *network* utama yang telah dilakukan *training* sebelumnya, lalu dilakukan *training* kembali secara *fine-tuning*. Sehingga, kriteria *loss* total ditambahkan dengan *semantic parsing loss* ($\mathcal{L}_s$) dan *landmark loss* ($\mathcal{L}_h$). Kemudian, proses *training* dilakukan sampai mencapai kondisi yang diinginkan. Dengan demikian, secara keseluruhan fungsi *loss* ($\mathcal{L}$) untuk *training network* dapat dinyatakan dalam penjumlahan berbobot dari 6 *loss*, dinyatakan pada persamaan

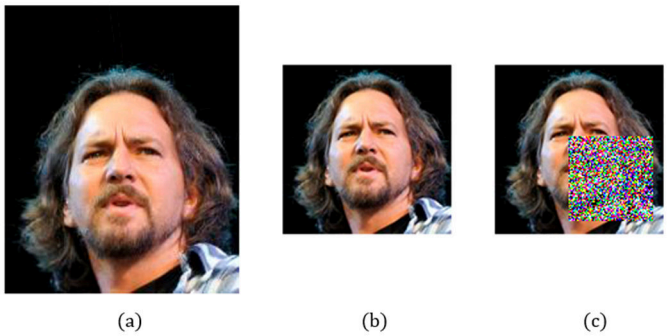
(3.1), dengan λ_i merupakan bobot untuk masing-masing *loss*. Pada saat proses pengujian, hanya *network generator* yang digunakan untuk *testing*.

$$\mathcal{L} = \lambda_1 \mathcal{L}_{KL} + \lambda_2 \mathcal{L}_f + \left(\lambda_3 \mathcal{L}_{dl} + \lambda_4 \mathcal{L}_{dg}\right) + \lambda_5 \mathcal{L}_s + \lambda_6 \mathcal{L}_h \quad (3.1)$$

3.3.1 Praproses

Tahap praproses data dilakukan dengan melakukan operasi pengirisan (*cropping*), dan *resize,* pada data CelebA. *Cropping* dilakukan dengan ketentuan memuat *landmark location* dari wajah, tanpa proses *alignment*. *Resize* dilakukan hingga citra berukuran 128×128. Pemilihan ukuran ini agar percobaan dapat dibandingkan dengan metode lain (Liao dkk., 2018). Proses ini diilustrasikan pada Gambar 3.4, (a) merupakan citra asli dari *dataset* CelebA, dengan ukuran 178×218, (b) adalah citra asli setelah dilakukan *cropping* dan *resize* hingga ukurannya menjadi 128×128, (c) merupakan citra setelah melalui tahap *preprocessing* dan diberi *masking* dengan ukuran 64×64. *Masking* yang diberikan pada citra merupakan *random pixel noise*, dan penentuan posisi *masking* ditentukan secara acak.

Dataset CelebA yang disediakan oleh Ziwei, dkk. (Ziwei, dkk., 2015) merupakan dataset citra wajah yang sebelumnya telah dilakukan proses *adjustment* dengan metode *similarity transformation* terhadap lokasi dua mata, sehingga data pada CelebA dipastikan posisi mata terletak di tengah citra. Proses *cropping* dilakukan secara otomatis menggunakan skema yang diajukan oleh Radford, dkk. pada DCGAN (Radford, dkk., 2016). Skema dimulai dengan memilih titik awal



Gambar 3.4 Citra *input*. (a) Citra asli CelebA, (b) setelah dilakukan proses *cropping* dan *resize,* (c) Citra *I* stelah diberi *masking.*

untuk *cropping* citra CelebA, yaitu $(x_1,\ y_1) = (30,\ 40)$. Kemudian, seluruh piksel yang terletak pada (x, y) akan diambil, dimana $x \in [x_1,\ x_1 + 138]$ dan $y \in [y_1,\ y + 138]$. Sebagai catatan, skema *cropping* ini dapat digunakan pada data CelebA saja, karena skema dibuat sesuai dengan karakteristik CelebA.

3.3.2 *Splitting* Data

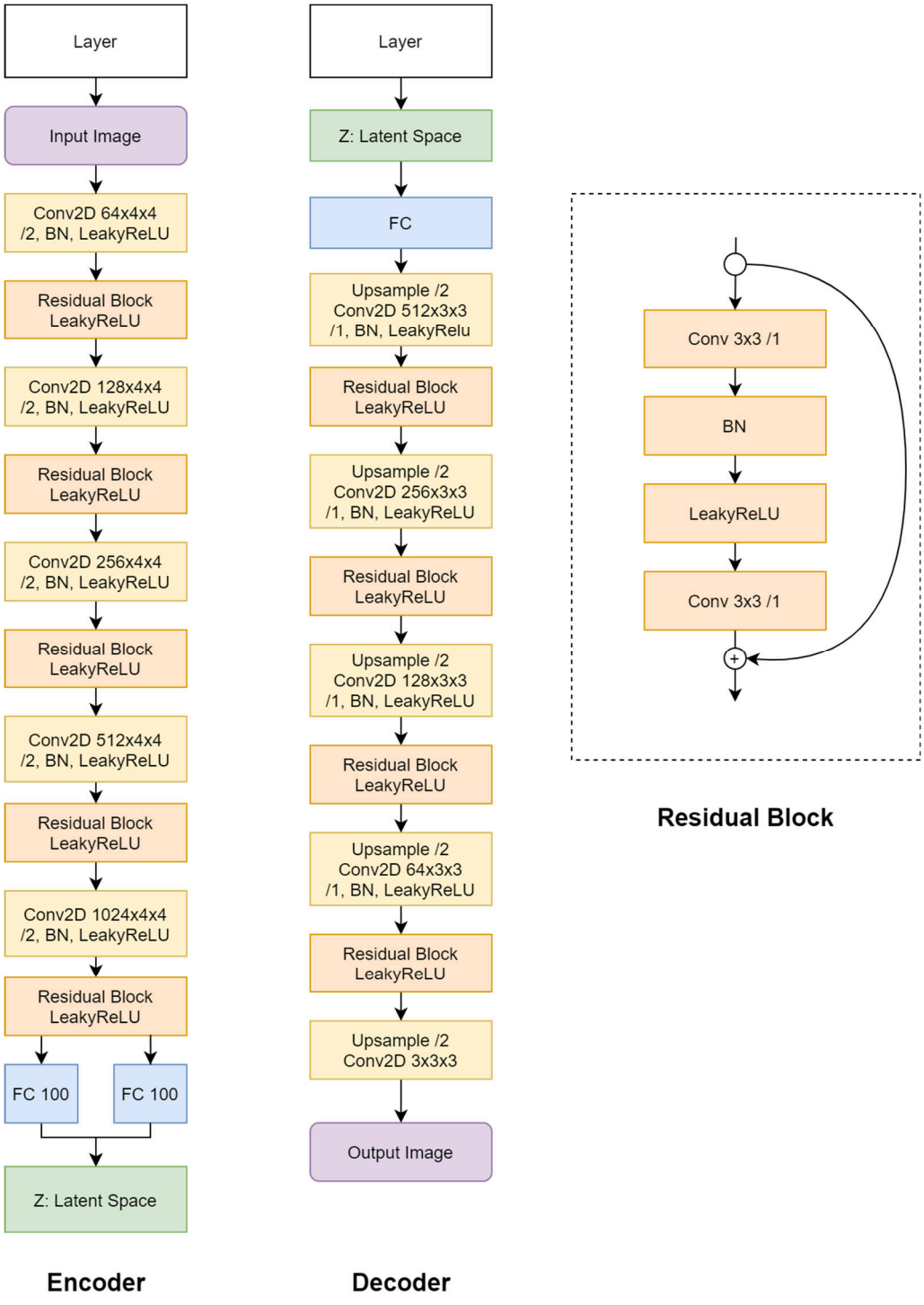
Splitting data untuk *training*, *validation*, dan *testing* mengikuti aturan *splitting* yang disajikan oleh penyedia data (Ziwei, dkk., 2015). Dari total 202,599 citra, 162,770 untuk *training*, 19,867 untuk *validation*, dan 19,962 untuk *testing*.

3.3.3 Augmentasi Data

Augmentasi pada dataset dilakukan sebagai salah satu langkah mengurangi kemungkinan terjadinya *overfitting,* terutama ketika jumlah data yang dimiliki untuk *training* dalam jumlah terbatas. Pada penelitian ini, dilakukan beberapa teknik augmentasi data, yaitu *random shifting, scaling, rotating,* serta *flipping*. *Random shifting* dilakukan dengan cara melakukan penggeseran (translasi) citra dengan arah acak. *Scaling* merupakan proses memperbesar atau memperkecil gambar. *Rotating* dilakukan dengan melakukan rotasi terhadap sebesar 1° sampai dengan 359°. *Flipping* dapat dilakukan secara vertikal serta horizontal. *Flipping vertical* akan menghasilkan *output* yang sama dengan *rotating* sebesar 180°, sementara *flipping horizontal* menghasilkan gambar hasil pencerminan relative terhadap sumbu vertikal. Pada proses augmentasi, bagian yang hilang karena proses augmentasi akan diberi suatu nilai konstan. Proses augmentasi data tidak menyebabkan perubahan dimensi citra.

3.3.4 *Generator Network*

Network Generator, dinotasikan dengan G, merupakan jenis *network* VAE. *Network* G terdiri menjadi 2 bagian utama, yaitu *encoder*, dan *decoder*.



Gambar 3.5 Susunan *layer network Generator G,* terdiri dari *encoder* dan *decoder*

Susunan *layer* dari *generator* yang digunakan mengikuti arsitektur dari DFC-VAE (Hou, dkk., 2017), yang mengadopsi dari AlexNet dan VGG-Net, ditambah dengan konsep *residual learning* yang diajukan pada ResNet (He, dkk., 2016). Susunan *layer* dari *generator* diilustrasikan pada Gambar 3.5. Bagian e*ncoder* terdiri dari 5 *layer* konvolusi, dan 2 *layer fully-connected* pararel. Setiap *layer* konvolusi yang digunakan merupakan konvolusi 2 dimendi dengan ukuran kernel 4×4, *stride* 2. Pemilihan *stride* 2 bertujuan untuk melakukan *down-sampling* tanpa menggunakan fungsi deterministik spasial seperti *maxpool*. Kemudian, melalui proses *Batch-Normalization* (BN) dan fungsi aktivasi *LeakyReLU*. Fungsi *LeakyReLU* didefinisikan pada persamaan (3.2). Setiap layer konvolusi disertai dengan *residual block* (He, dkk., 2016), dalam satu blok *residual* terdiri dari operasi konvolusi dengan ukuran *filter* 3×3, BN, dengan fungsi aktivasi *LeakyReLU* kemudian dilakukan konvolusi kembali dengan ukuran *filter* yang sama. Pada blok residual tidak dilakukan operasi *downsampling* sehingga ukuran *stride* yang digunakan adalah 1. Keluaran dari setiap blok *residual* diterapkan fungsi aktivasi *LeakyReLU*. Layer *fully-connected* akan melakukan pemetaan dari *input* ke dalam nilai z_μ dan z_σ sebelum menjadi variabel z. *Encoder* ini akan menerima masukan berupa citra wajah yang diberi *masking*, lalu dilakukan proses *encoding* ke domain laten z.

$$f(x) = \begin{cases} x & \text{jika } x \geq 0 \\ 0{,}02x & \text{selainnya} \end{cases} \tag{3.2}$$

Bagian *decoder* berbentuk simetris dengan *encoder*. Terdiri dari 5 *layer* konvolusi, dengan ukuran kernel 3×3, dan besar *stride* 1. Sebelum *konvolusi*, *upsampling* pada setiap *layer* konvolusi dilakukan dengan menggunakan metode *nearest neighbor* dengan besar skala 2. Penggunaan *upsampling* menggunakan konsep *nearest neighbor* dapat membantu kestabilan dari GAN, seperti yang diajukan Hou, dkk. pada DFC-VAE (Hou, dkk., 2017). Setiap layer *konvolusi* diikuti oleh proses *Batch-Normalization* (BN) dan fungsi aktivasi *LeakyReLU,* kemudian blok residual seperti pada *encoder*, kecuali *layer* konvolusi terakhir yang hanya terdiri dari *upsampling* dan operasi konvolusi. Penggunaan *Batch-Normalization* bertujuan untuk meningkatkan stabilitas pada saat proses *training* dilakukan. Fungsi aktivasi yang digunakan pada keluaran *network* G merupakan fungsi $tanh$, untuk menjaga domain hasil tetap di $[-1, 1]$. Pada *network G,*

diperoleh nilai *loss* yang pertama dan kedua, yaitu $KL - Divergence\ Loss$ ($\mathcal{L}_{KL}$) dan *feature reconstruction loss* ($\mathcal{L}_f$). $KL - Divergence\ Loss$ dinyatakan pada persamaan (2.5). $\mathcal{L}_{KL}$ digunakan untuk melakukan *update* pada *generator*.

Metode GFC yang diajukan Yijun (Yijun, dkk., 2017) hanya menggunakan jarak *euclid* citra masukan dengan citra keluaran pada domain RGB sebagai kriteria *feature reconstruction loss*, dan tanpa penggunaan $KL - Divergence\ Loss$. Pada domain RGB, jarak *euclid* atau L2 dari citra masukan dengan keluaran dapat didefinisikan seperti pada persamaan (3.3), dengan W, H, C menyatakan ukuran lebar, tinggi, dan *channel* citra. Pada domain RGB, nilai $C = 3$.

$$\mathcal{L} = \frac{1}{6WH} \sum_{c=1}^{C=3} \sum_{i=1}^{W} \sum_{j=1}^{H} \left(I_{i,j,c} - \hat{I}_{i,j,c}\right)^2 \tag{3.3}$$

Namun, pada metode yang diajukan, p*re-trained network* VGG-Net akan digunakan sebagai *feature reconstruction loss* dari *generator.* Penggunaan *pre-trained network* seperti VGGNet untuk penentuan kriteria *loss* dapat membantu mempertahankan fitur-fitur penting pada citra (Hou, dkk., 2019, 2017). Sehingga, kriteria *loss* bukan $L2$ *distance* antara citra I dengan $\hat{I}$ pada domain RGB, melainkan $L2$ *distance* antara I dengan $\hat{I}$ pada domain fitur yang diekstrak oleh VGGNet. Hal ini bertujuan agar fitur-fitur penting dari citra dapat dipertahankan, untuk mengatasi masalah korelasi spasial yang muncul pada penelitian GFC (Yijun, dkk., 2017) seperti keterkaitan spasial dan masalah *inpainting* pada *unaligned face*.

Pada penelitian ini akan digunakan tiga *layer* awal VGG-Net 16. Penelitian sebelumnya (Hou dkk., 2017, 2019) menggunakan tiga *layer* dan lima *layer* awal VGG-Net. Penggunaan tiga layer awal VGG-Net telah dapat mempertahankan *deep-feature* dari citra masukan. Meskipun hasil yang lebih baik dan konsisten secara visual diperoleh pada penggunaan lima *layer*, namun penggunaan lima *layer* membuat ukuran matriks yang diproses menjadi semakin besar, sehingga pada penelitian ini digunakan tiga *layer* awal saja.

Didefinisikan *pre-trained network* VGGNet sebagai pemetaan terhadap citra asli I pada domain fitur VGGNet, dinotasikan dengan $\psi_{i,j}(I)$. Sehingga, *feature reconstruction loss* memiliki bentuk yang lebih umum dari persamaan (3.3),

disajikan pada persamaan (3.4). Persamaan (3.4) menyatakan nilai *loss* untuk satu *feature map*, pada *layer* l, dinotasikan dengan $\mathcal{L}_l$. Total dari *loss* pada *feature reconstruction loss* (L_f), merupakan penjumlahan dari $\mathcal{L}_l$ untuk seluruh *layer* atau *feature map* yang digunakan, dinyatakan pada persamaan (3.5). C_l, W_l, H_l menyatakan jumlah filter, lebar, dan tinggi pada domain fitur VGGNet pada *layer* ke-l.

$$\mathcal{L}_l = \frac{1}{2C_l W_l H_l} \sum_{c=1}^{C_l} \sum_{i=1}^{W_l} \sum_{j=1}^{H_l} \left(\psi_{c,i,j}(I) - \psi_{c,i,j}\left(\hat{I}\right) \right)^2 \tag{3.4}$$

$$\mathcal{L}_f = \sum_{l=1}^{L} \frac{100}{C_l^2} \mathcal{L}_l \tag{3.5}$$

Setelah proses *training* selesai dilakukan, hanya bagian *generator* yang digunakan untuk melakukan *testing*. *Input* dari *network* G adalah citra asli I dengan *masking*, yaitu I_m. *Output* dari *network* G merupakan citra hasil *inpainting,* dinotasikan dengan $\hat{I}$, dengan ukuran yang sama dengan I_m.

3.3.5 *Discriminator Network*

Metode yang diajukan menggunakan dua jenis *discriminator*, yaitu *local* dan *global*, seperti yang digunakan pada GFC (Yijun, dkk., 2017). Penggunaan dua jenis *discriminator* bertujuan agar bagian *missing region* yang disintesis oleh *network* adalah bagian yang detail (lokal) namun tetap realistis secara keseleruhan (global). Pada penelitian sebelumnya (Yijun, dkk., 2017), *loss* pada *discriminator* merupakan *Sigmoid Cross Entropy* antara keluaran dari *discriminator* dengan masukan berupa citra asli dengan keluaran dari *discriminator* dengan masukan berupa citra hasil *inpainting*.

Pada penelitian ini, *discriminator* yang digunakan bukan *discriminator* dengan basis GAN standar seperti pada GFC, melainkan *discriminator critic* seperti pada Wasserstein GAN (Arjovsky, dkk., 2017). Pemilihan jenis WGAN dibanding GAN standar karena dalam beberapa kasus GAN standar sangat sulit untuk mencapai kestabilan, sehinggai dilakukan beberapa pengembangan untuk mengatasi masalah ini, salah satunya WGAN. Perbedaan antara WGAN dengan GAN standar terletak pada fungsi tujuan yang digunakan, yang berarti krtieria *loss*

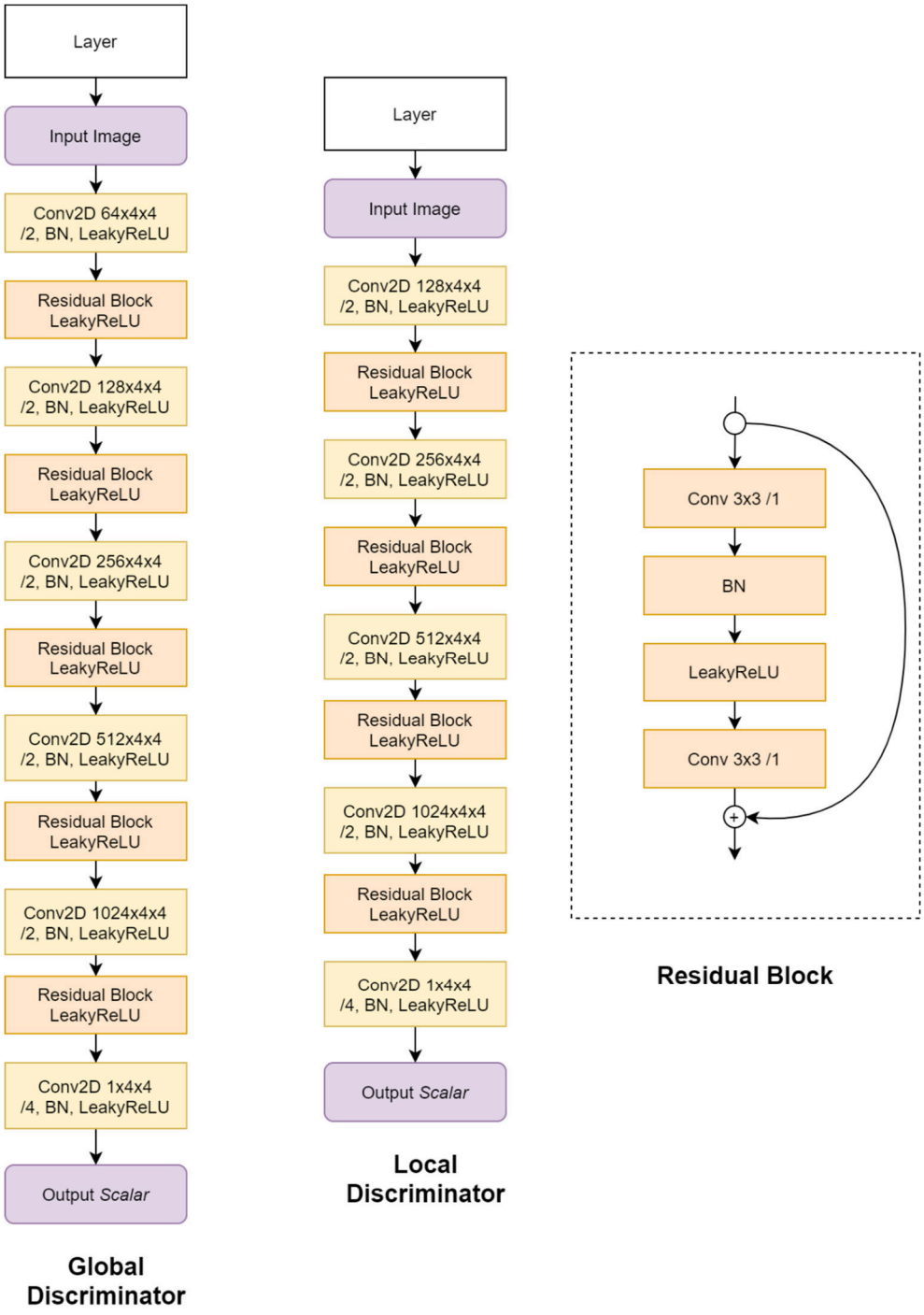
Tabel 3.1 Perbedaan fungsi *loss* GAN dan WGAN

	Discriminator	***Generator***
GAN	$\frac{1}{m}\sum_{i=1}^{m}[\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))]$	$\frac{1}{m}\sum_{i=1}^{m}[\log(D(G(z^{(i)})))]$
WGAN	$\frac{1}{m}\sum_{i=1}^{m}[D(x^{(i)}) - D(G(z^{(i)}))]$	$\frac{1}{m}\sum_{i=1}^{m}[D(G(z^{(i)}))]$

adversarial dan *loss* yang digunakan untuk *update generator*. Perbedaan GAN dengan WGAN disajikan pada Tabel 3.1. Pada GAN biasa, fungsi *loss* yang digunakan adalah rata-rata penjumlahan dari log dari keluaran *discriminator* dengan masukan citra asli, dan $1 -$ keluaran dari *discriminator* dengan masukan berupa citra sintetis. Hal ini dapat menyebabkan masalah *vanished gradient* karena nilai *loss* yang akan semakin mendekati nol saat *discriminator* semakin unggul, sehingga tidak ada *gradient* yang digunakan untuk melakukan pembaruan bobot pada *generator*. Sementara pada WGAN, fungsi yang digunakan bukan fungsi logaritmik melainkan keluaran dari *discriminator* langsung.

Susunan *layer network discriminator* lokal dan global yang akan dipakai pada penelitian disajikan pada Gambar 3.6. Skema ini identik dengan skema yang digunakan pada *encoder*. Pembeda terletak pada lapis terakhir yang digunakan sebagai keluaran. Pada *encoder*, keluaran berupa variabel z yang merupakan hasil *encoding* citra masukan pada domain laten z, sementara pada *discriminator*, hasil keluaran berupa hasil klasifikasi kelas dari masukan, termasuk kelas citra asli atau citra sintesis. Sebagai catatan, pada *network* ini tidak digunakan fungsi aktivasi logartimik seperti *sigmoid,* atau fungsi hiperbolik seperti $tanh$. Fungsi yang digunakan adalah aktivasi LeakyReLU dengan konvolusi berukuran $1 \times 4 \times 4$ agar hasil keluaran yang diperoleh berupa skalar dengan dimensi 1-d.

Perbedaan antara *local* dan *global discriminator* terletak pada jumlah lapis yang digunakan. Jika pada *global discriminator* terdapat 5 lapis konvolusi ditambah 1 lapis konvolusi untuk menghasilkan keluaran skalar, *local discriminator* hanya terdapat 4 lapis konvolusi ditambah 1 lapis konvolusi akhir untuk menghasilkan



Gambar 3.6 Susuan *layer* pada *network local* dan *global discriminator*

keluaran skalar. Hal ini dikarenakan ukuran masukan pada *global discriminator* adalah $128 \times 128 \times 3$, sementara ukuran masukan pada *local discriminator* adalah ukuran masking, yaitu $64 \times 64 \times 3$.

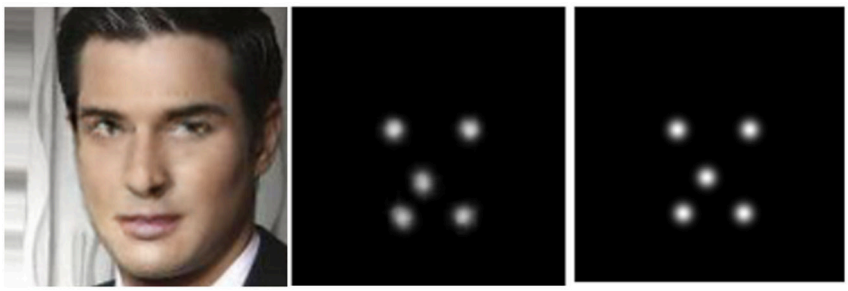
3.3.6 Facial Landmark Network

Network ini bertugas untuk mengekstrak *landmark* dari wajah. *Loss* yang dihasilkan pada *network* ini merupakan *landmark loss* $\mathcal{L}_h$, dan digunakan untuk mengembangkan hasil sintesis bagian *missing region*. Nilai $\mathcal{L}_{lm}$ merupakan *Sigmoid Cross Entropy* antara *landmark* asli dengan *landmark* hasil *network*, dinyatakan dalam persamaan (3.7). LH_h merupakan citra *landmark* hasil *network*, dan LH_{GT} menyatakan *landmark* asli (*ground truth*). SCE merupakan hasil *network* dengan fungsi aktivasi *sigmoid* kemudian dihitung *Binary Cross Entropy*. Persamaan (3.6) menyatakan persamaan *Binary Cross Entropy* yang digunakan, dengan z_i menyatakan label asli/target, sedangkan x_i menyatakan *logits* yang dihasilkan *network* dengan fungsi aktivasi *sigmoid*.

$$BCE = -\sum_{i=1}^{C=2} z_i \log(x_i) = -z_1 \log(x_1) - (1 - z_1) \log(1 - x_1) \tag{3.6}$$

$$\mathcal{L}_{lm} = SCE(label = LH_{GT}, logits = LH_h) \tag{3.7}$$

Contoh hasil *heat-map landmark network* yang digunakan pada metode Liao disajikan pada Gambar 3.7, (a) merupakan citra asli yang menjadi *input*, hasil ekstraksi *heat-map landmark* dari (a) diilustrasikan pada (b), dan *ground truth* yang menjadi pembanding diilustrasikan pada (c). Namun pada penelitian yang diajukan, *landmark network* yang digunakan akan berbeda dengan *landmark* yang digunakan di metode sebelumya. *Landmark network* yang digunakan akan mengekstrak 68



(a) (b) (c)

Gambar 3.7 *Heat-map landmark* pada metode Liao (Liao dkk., 2018). (a) Citra input asli, (b) citra hasil *network landmark* dari (a), (c) citra *ground truth heat-map landmark* dari (a)

titik *landmark* wajah, mengikuti skema yang diajukan oleh Jimei, dkk. (Jimei dkk., 2016). *Network* ini terlebih dulu dilatih menggunakan data wajah dengan informasi *landmark* (*ground truth*), kemudian digunakan pada metode yang diajukan untuk mendapatkan *loss landmark*.

3.3.7 *Semantic Parsing Network*

Semantic Parsing Network merupakan *network pre-trained* yang digunakan untuk membantu hasil sintesis data pada *network* G. Penggunaan *semantic parsing* pada *facial inpainting* dapat membantu meningkatkan kualitas perseptual hasil *inpainting* (I. J. Goodfellow dkk., 2014; Liao dkk., 2018; R. Ma dkk., 2019; Pathak dkk., 2016). Pada *network* ini, dilakukan proses segmentasi pada wajah, lalu dihasilkan nilai *semantic parsing loss* ($\mathcal{L}_s$) yang digunakan untuk membantu proses *update* bobot pada *generator*. Nilai $\mathcal{L}_s$ merupakan *cross entropy* antara *output* hasil segmentasi dengan *ground truth* (GT) dari citra *input*. Hasil perhitungan $\mathcal{L}_s$ didefinisikan pada persamaan (3.8) dimana I_S merupakan hasil *output* segmentasi pada *network*, S_{GT} merupakan *ground truth* segmentasi citra wajah *input*.

$$\mathcal{L}_s = SCE(label = S_{GT}, logits = I_S) \tag{3.8}$$

Contoh hasil *semantic parsing* yang digunakan pada metode GFC diilustrasikan pada Gambar 3.8, (a) merupakan citra *input* asli, (b) merupakan hasil *network semantic parsing*. Hasil dari *network semantic parsing* akan digunakan

(a) (b)

Gambar 3.8 Contoh s*emantic parsing* pada GFC. (a) Citra input asli, (b) citra hasil *network semantic parsing*. Nilai L_s merupakan L_2 dari citra hasil *network semantic parsing* dengan *ground truth* (Yijun, dkk., 2017).

untuk menghitung $\mathcal{L}_s$ dengan membandingkan nilai setiap pikselnya dengan hasil segmentasi asli (*ground truth*). Namun pada penelitian yang diajukan, *semantic parsing network* yang digunakan akan berbeda dengan *semantic parsing network* yang digunakan di metode sebelumya (Yijun dkk., 2017). *Semantic parsing network* yang digunakan akan mengekstrak 19 label wajah, mengikuti skema pada *network* BiSeNet (Yu dkk., 2018).. *Network* ini dilatih terlebih dahulu dengan menggunakan *dataset* CelebA agar dapat melakukan tugas segmentasi semantik pada wajah. Kemudian, *semantic parsing network* yang sudah dilatih digunakan pada *network* yang diajukan untuk memperoleh *loss semantic parsing*.

3.3.8 Skenario Uji Coba

Uji coba dilakukan untuk menganalisis performa dari metode yang diajukan dalam menyelesaikan masalah yang telah dirumuskan, yaitu masalah yang timbul pada *inpainting* pada citra wajah dengan mempertahankan keterkaitan spasial. Pada saat proses pengujian, hanya *network generator* yang digunakan. Data yang digunakan sebanyak 19,962 dari 202,599 data pada dataset CelebA. Pada saat *training,* ukuran *masking* konstan 64×64, sementara pada saat *testing* dilakukan, skenario *masking* tidak konstan pada satu ukuran, namun terbagi menjadi beberapa skenario. Skenario pertama dengan bentuk persegi dengan ukuran 32×32 sampai dengan 64×64 (kelipatan 8) dengan penempatan acak, kemudian dengan bentuk *masking* acak. Metode yang diajukan menggunakan skema *curriculum learning*, sehingga hasil dari setiap tahap *learning* akan didokumentasikan lalu dilakukan evaluasi untuk melihat perkembangan hasil yang diperoleh di tiap tahap *learning*. Selain penempatan *masking* secara acak, *testing* juga dilakukan dengan melakukan penempatan *masking* pada posisi tertentu yang mengharuskan *network* mempertimbangkan keterkaitan spasial, seperti separuh bagian mulut, atau salah satu mata.

Ukuran citra yang digunakan untuk *testing* adalah 128×128. Model tidak dapat menerima masukan citra dengan ukuran selain ukuran yang telah ditentukan, sehingga harus dilakukan *resize* ukuran citra masukan jika citra masukan memiliki ukuran yang berbeda. Model *generator* yang digunakan melakukan proses *encoding* dengan masukan citra berukuran 128×128 ke dalam domain *latent*,

kemudian proses *decoding* dari domain *latent* dilakukan dengan hasil *decoding* berupa citra berukuran 128×128. Proses *decoding* akan menghasilkan citra dengan ukuran 128×128.

Terdapat dua jenis penilaian yang akan dilakukan terhadap hasil yang diperoleh, yaitu penilaian secara kualitatif dan kuantitatif. Penilaian secara kualitatif dilakukan dengan menunjukkan hasil *inpainting* ke beberapa pengamat kemudian setiap pengamat memberikan penilaian secara subjektif untuk mengidentifikasi hasil *inpainting* atau bukan. Secara kuantitatif, digunakan 2 jenis metrik perhitungan, yaitu *Peak Signal to Noise Rasio* (PSNR), *Structure Similarity Index* (SSIM) yang digunakan untuk memeriksa kemiripan dari dua buah gambar. Hasil yang diperoleh dibandingkan dengan penelitian terdahulu, yaitu metode *Generative Face Completion* (GFC) (Yijun, dkk., 2017) pada *citra unaligned face* serta memuat masalah keterkaitan spasial.

BAB 4
HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan hasil yang diperoleh dari penelitian yang telah dilakukan dengan skenario implementasi yang ditentukan sebelumnya. Analisis terhadap hasil yang didapat juga dilakukan sebagai bentuk evaluasi terhadap hasil yang diperoleh. Analisis dilakukan dengan berdasarkan penilaian kualitatif dan kuantitatif berdasarkan beberapa skenario uji coba dan hasil yang diperoleh.

4.1 Hasil Penelitian

Hasil-hasil dari percobaan yang dilakukan, keluaran *network* pada saat proses *training*, hasil dari beberapa skenario *testing generator* setelah proses *training* telah selesai dilaksanakan, lingkungan uji coba dan skenario untuk memperoleh hasil yang diperoleh berikutnya akan dijelaskan.

4.1.1 Lingkungan Uji Coba

Lingkungan perangkat lunak merupakan lingkunan yang digunakan untuk menjalankan rancangan penelitian, baik dari strategi maupun uji coba, Lingkungan perangkat keras merupakaan fasilitas yang digunakan untuk mengimplementasikan rancangan tersebut ke dalam perangkat lunak. Lingkungan uji coba yang digunakan disajikan pada Tabel 4.1

Tabel 4.1 Lingkungan Uji Coba

Lingkungan	**Spesifikasi**	
Perangkat Keras	Prosesor	Intel Core i7 – 8750H
	VGA	GTX 1050 Ti 4GB
	RAM	8GB DDR4
Perangkat Lunak	Sistem Operasi	Ubuntu 18.04
	Bahasa	Python 2.7

4.1.2 Skenario Uji Coba

Strategi yang digunakan untuk *training* pada *network* yang diajukan mengikuti strategi *curriculum learning* (Bengio, dkk., 2009), yang berarti kriteria *loss* akan semakin sulit seiring bertambahnya kemampuan *network*. Pada proses

Tabel 4.2 Konfigurasi parameter

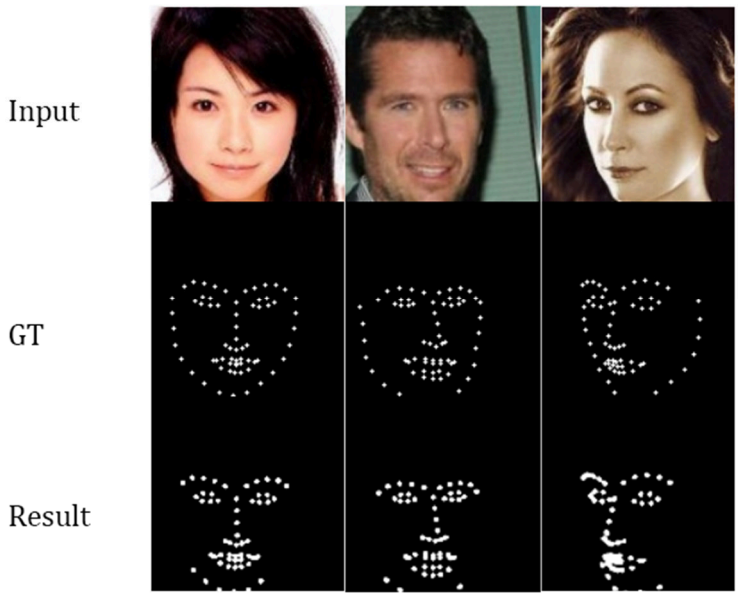
Parameter	**Notasi**	**Nilai**	***Parameter***	**Notasi**	**Nilai**
Learning Rate Generator	L_r	1e-4	Bobot K-L *Loss*	λ_1	1e-1
Learning Rate Discriminator	L_d	5e-4	Bobot *Feature Reconstruction Loss*	λ_2	1e-4
Batas *Step* Tahap 1	$step\_train1$	15000	Bobot *Local Discriminator*	λ_3	1e-1
Batas *Step* Tahap 2	$step\_train2$	25000	Bobot *Global Discriminator*	λ_4	1e-1
Step Maksimum	max_iter	45000	Bobot *Landmark Loss*	λ_5	1e-1
Epoch maksimum	max_epoch	10	Bobot *Parsing Loss*	λ_6	1e-1
Momentum ADAM	β_1	0.5			

training, metode optimasi yang digunakan untuk meminimasi nilai *loss generator* dan *discriminator* adalah metode ADAM dan RMSProp. *Masking* yang diberikan pada citra merupakan *noise* berupa nilai acak berdistribusi *normal* dengan ukuran konstan, yaitu persegi sebesar 64×64 piksel. Namun pada saat proses pengujian (*testing*), skenario pengujian dilakukan dengan *masking noise* berbentuk persegi dengan ukuran bervariasi, yaitu 64×64 piksel, 32×32 piksel, kemudian 16×16 piksel, dan *noise* dengan bentuk acak terutama pada bagian yang membutuhkan informasi keterkaitan spasial seperti separuh bibir. Citra masukan yang digunakan tetap merupakan data dari CelebA dataset, yang tidak digunakan pada saat *training* sebelumnya karena telah dilakukan pembagian dataset untuk *training* dan *testing*. Beberapa citra wajah yang *unaligned* dipilih untuk dilakukan analisis lebih lanjut.

Hyper-parameter yang digunakan pada saat proses *training* disajikan pada Tabel 4.2. Pemberian bobot *loss* dilakukan untuk meregulasi efek dari *loss* terhadap *network*. Penentuan nilai *learning rate* berdasarkan studi literatur penelitian terdahulu pada domain WGAN. Penentuan nilai batas *step* dilakukan berdasarkan studi literatur penelitian terdahulu. Pada saat *training* dilakukan, *network* selain *Generator* dan *Discriminator* tidak dilakukan operasi pembaharuan bobot *network*, yang berarti parameter pada *network parsing, landmark,* dan VGG tetap konstan.

4.1.3 Proses *Training Network*

Sebelum proses *training* untuk *network* GAN dilakukan, dilakukan *training* untuk *network* yang akan digunakan untuk kriteria *loss landmark* dan *loss semantic parsing*. *Landmark network* terlebih dulu dilatih dengan menggunakan data CelebA. Karena pada dataset CelebA tidak tersedia *ground-truth* untuk *landmark* yang diinginkan, maka pembuatan *network landmark* dilakukan dengan menggunakan tambahan *library* lain, yaitu DLIB C++ Library dan OpenCV untuk *landmark*. Sementara untuk *semantic face parsing* menggunakan *pre-trained face-parser network* berdasarkan CelebA Mask HQ (Ziwei, dkk., 2015).



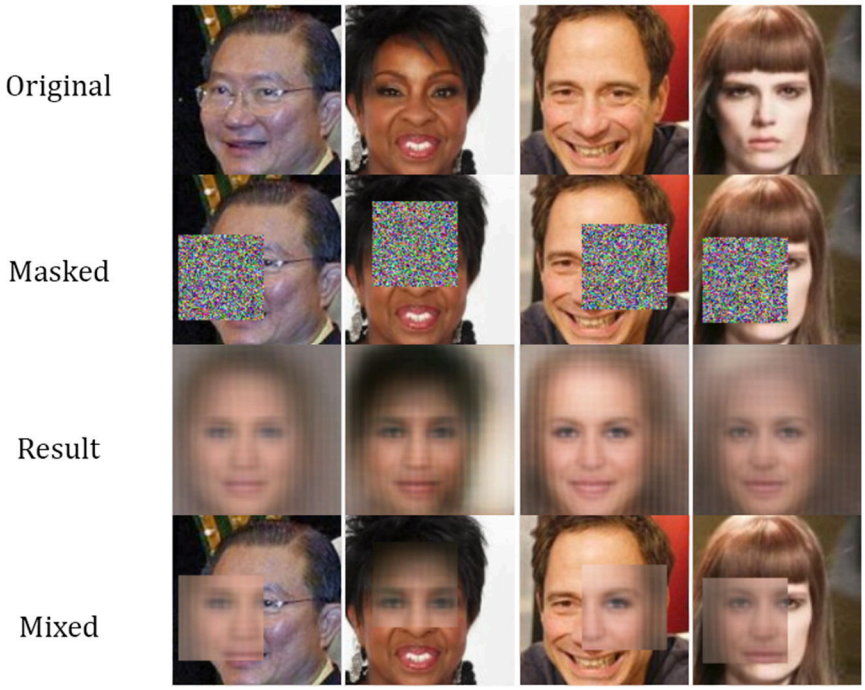
Gambar 4.1 *Landmark* wajah. Baris pertama merupakan citra masukan, baris kedua adalah hasil *landmark* menggunakan DLIB, baris ketiga merupakan hasil *network landmark* yang dibuat

(a) (b)

Gambar 4.2 Hasil *semantic parsing* dengan BiSeNet. Data CelebA, dengan jumlah kelas 19

DLIB digunakan untuk melakukan prediksi bentuk wajah, sementara OpenCV digunakan untuk mendeteksi *bounding box* wajah. Pemilihan kombinasi OpenCV dan DLIB berdasarkan pertimbangan waktu yang dibutuhkan untuk proses penentuan *landmark* wajah. Untuk melakukan prediksi *bounding box* wajah pada dataset CelebA, DLIB dapat melakukan deteksi wajah dalam waktu $24.58\,ms$, sementara OpenCV memerlukan waktu yang lebih lama: $32.95\,ms$. Namun, tingkat keberhasilan mendeteksi OpenCV lebih tinggi dari DLIB, maka dipilih OpenCV untuk menentukan *bounding box* wajah. Untuk melakukan prediksi posisi *landmark* wajah, DLIB memiliki fasilitas *shape-predictor*, maka DLIB digunakan. Pemilihan DLIB untuk melakukan prediksi terhadap *shape* wajah karena DLIB memiliki tingkat keberhasilan untuk menemukan *shape* wajah lebih tinggi dari OpenCV. Contoh hasil ekstraksi *fitur landmark* pada wajah disajikan pada Gambar 4.1. Sementara hasil dari *semantic parsing network* menggunakan BiSeNet (Yu dkk., 2018) disajikan pada Gambar 4.2. Hasil *semantic parsing* terdiri dari 19 kelas label. Kedua *network* ini, *landmark* dan *semantic parsing network,* berikutnya akan digunakan sebagai *loss* tambahan untuk GAN. Tujuan dari penambahan dua *network* ini beserta VGG-Net adalah untuk meningkatkan kualitas perseptual yang diperoleh, serta mengatasi dua masalah yang ditemukan pada *inpainting* citra wajah, yaitu masalah keterkaitan spasial dan masalah pada citra wajah *unaligned*.

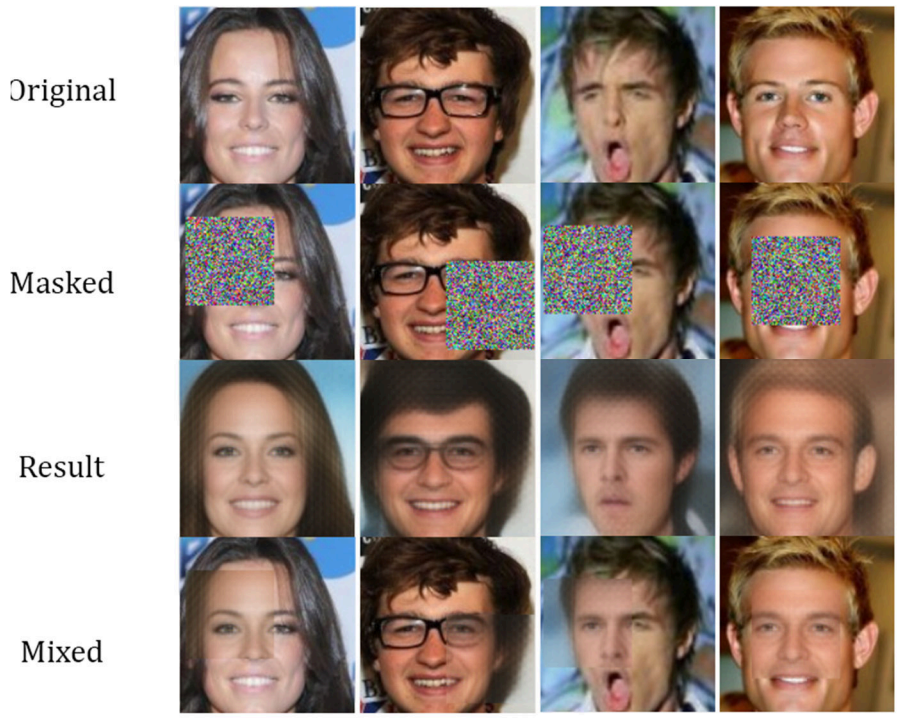
Pada saat proses *training* GAN dilakukan, strategi *curriculum learning* berarti fungsi *loss* yang digunakan akan dikembangkan secara bertahap dari *loss* yang sederhana hingga *loss* yang kompleks, seiring dengan ukuran *network* yang



Gambar 4.3 Hasil *Network Generator* saat *training* dimulai

semakin besar dengan bertambahnya parameter yang akan diperbarui. Tahap *training* dibagi menjadi 3 tahapan, ketika satu tahap selesai, hasil *network* yang telah dilatih pada tahap tersebut akan dipakai untuk tahap latihan berikutnya yang lebih kompleks. Karena nilai *loss* yang digunakan memiliki domain yang bervariasi, maka penjumlahan dari 6 *loss* dilakukan dengan menggunakan bobot (λ_i) pada masing-masing *loss* untuk meregulasi efek dari *loss* yang digunakan. Pemberian bobot ini berguna agar efek setiap *loss* yang digunakan tidak mendominasi keseluruhan *network*.

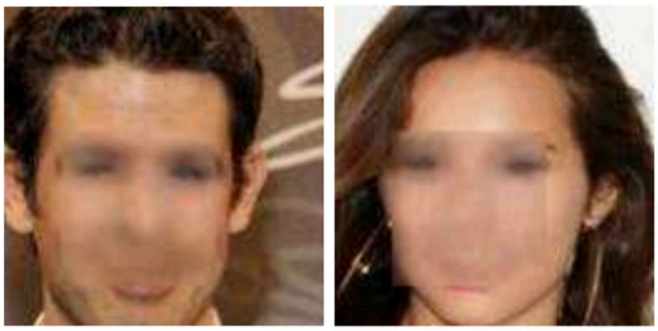
Tahap pertama pelatihan *network* menggunakan dua jenis *loss*, yaitu *KL-divergence loss* ($\mathcal{L}_{KL}$) dan *Feature Reconstruction Loss* pada domain VGG-Net ($\mathcal{L}_f$). Tahap ini dilakukan sebanyak 15000 *step.* Hasil keluaran *network* pada awal *training* dimulai disajikan pada Gambar 4.3. Baris pertama merupakan citra asli (*ground truth*) sebelum diberi *masking*, baris kedua menujukkan citra asli setelah diberi *noise* dengan ukuran 64×64 piksel. Ukuran 64×64 piksel merupakan setengah dari ukuran citra masukan, yaitu 128×128 piksel. Pemilihan ukuran *masking* ini bertujuan agar terjamin terdapat satu bagian inti wajah yang tertutupi, seperti mata, mulut, atau hidung. Baris ketiga menunjukkan hasil keluaran dari



Gambar 4.4 Hasil *Network Generator* menggunakan dua *loss*: *KL-Divergence* dan *Feature Reconstruction Loss*

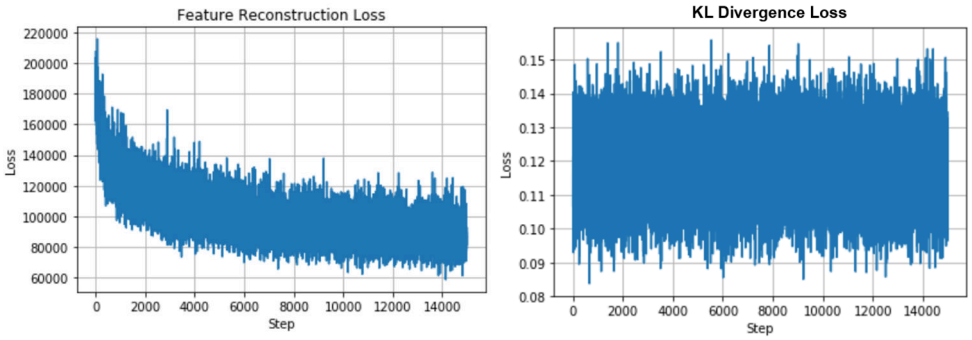
network. Keluaran dari *network* merupakan keseluruhan citra dengan ukuran 128×128 piksel. Baris keempat merupakan gabungan antara citra *ground truth* dengan keluaran dari *network*.

Pada tahap ini, keluaran yang dihasilkan oleh *network generator* merupakan hasil dari proses *update* menggunakan dua kriteria *loss* saja, yaitu *KL-Divergence Loss* dan *Feature Reconstruction Loss* berdasarkan *pretrained network* VGG-Net. Hasil *training* pada tahap pertama disajikan pada Gambar 4.4. Penelitian sebelumnya yang dilakukan oleh Yijun, dkk. pada metode GFC (Yijun, dkk., 2017), tidak menggunakan *KL-Divergence loss*, dan hanya menggunakan jarak *euclid* antara citra masukan dengan citra keluaran pada domain RGB. Hasil *network generator* pada metode GFC dengan tahap pertama *loss* yang digunakan pada metode tesrsebut disajikan pada Gambar 4.5. Pada tahap ini, terlihat bahwa penambahan penggunaan kriteria *loss KL-Divergence* $(\mathcal{L}_{KL})$ beserta *Feature Reconstruction Loss* $(\mathcal{L}_f)$ menggunakan VGG-Net dapat membantu menghasilkan citra *inpainting* dengan kualitas perseptual yang lebih baik dibanding penggunaan



Gambar 4.5 Hasil *Network Generator* tahap pertama pada metode GFC (Yijun dkk., 2017)

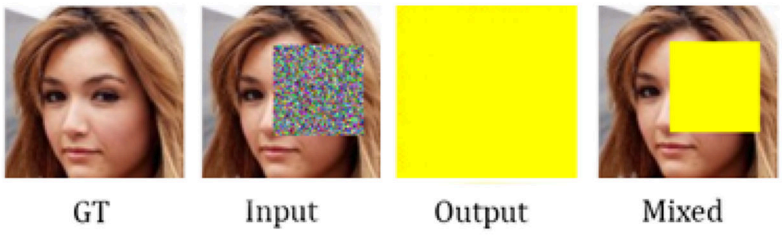
loss yang hanya berdasarkan jarak *euclid* antara citra masukan dengan keluaran pada domain warna RGB, seperti pada GFC. *Feature reconstruction loss* berhasil menangkap pola bentuk wajah yang lebih detail dan spesifik. Namun, meskipun hasil yang dikeluarkan oleh *network generator* sudah cukup baik dari segi perseptual, perbedaan warna masih terlihat antara bagian citra yang disintesis dengan sekitarnya, serta detail yang dihasilkan masih belum cukup bagus. Hal ini yang akan diperbaiki dengan penggunaan *loss* berikutnya. *Loss* dari *generator* yang diperoleh pada tahap pertama *training* disajikan pada Gambar 4.6. Terlihat pada *feature reconstruction loss,* proses optimasi nilai *loss* berjalan dan nilai *loss* semakin rendah. Hal ini menunjukkan bahwa citra keluaran yang dihasilkan semakin mirip dengan citra *ground truth*. Namun, nilai *loss* yang rendah tidak dapat menjadi jaminan kualitas perseptual citra yang dihasilkan terlihat realistis.



Gambar 4.6 Nilai loss dari *network generator* pada 15000 *step* awal, kiri menunjukkan *feature reconstruction loss,* kanan menunjukkan *KL-Divergence Loss*

Setelah *training* dilakukan menggunakan dua *loss*, *loss* pertama yang ditambahkan pada *network* merupakan *loss adversarial* dari dua buah *network discriminator*, yaitu *loss discriminator local* dan *discriminator global*, $\mathcal{L}_{dl}$ dan $\mathcal{L}_{gl}$. Kedua *network* ini bertugas melakukan klasifikasi dari citra masukan, dengan label kelas *real* dan sintesis. Nilai *loss* dari klasifikasi yang dilakukan akan diteruskan ke seluruh *network* untuk proses pembaruan bobot. Didefinisikan oleh Goodfellow (I. J. Goodfellow dkk., 2014), bahwa optimasi yang dilakukan pada jenis *network* GAN merupakan konsep *mini-max* antara *discriminator* dengan *generator*. Sehingga untuk mencapai tingkat kesetimbangan yang diinginkan, *discriminator* dan *generator* harus memiliki kekuatan yang berimbang. Ketika salah satu sub-*network* bagian dari GAN terlalu mendominasi/unggul, baik *generator* atau-pun *discriminator*, maka kondisi kesetimbangan yang diinginkan tidak dapat tercapai, menyebabkan sub-*network* dari GAN menghasilkan nilai *loss* yang cukup besar dan menyebabkan citra keluaran *generator* menjadi rusak/tidak sesuai harapan, atau nilai *loss* semakin mendekati nol sehingga konsep adversarial tidak berpengaruh pada *generator*.

Jika *network discriminator* yang digunakan terlalu lemah dalam membedakan citra *real* dan sintesis, maka yang terjadi adalah *loss* besar yang dihasilkan oleh *discriminator* akan diteruskan ke seluruh *network generator*, sehingga *network generator* menghasilkan keluaran seperti ditunjukkan pada Gambar 4.7. Sehingga, untuk mengatasi masalah ini, salah satu strategi yang dicoba adalah melakukan *warming-up* pada *discriminator*. Pada saat tahap pelatihan dari *step* 15000 sampai dengan 25000, proses *feeding* data latih tetap dilakukan terhadap kedua *discriminator,* baik *local* maupun *global*, hanya saja nilai *loss adversarial* yang dihasilkan oleh *discriminator* digunakan hanya untuk pembaruan bobot *discriminator* saja, bukan *generator* secara keseluruhan. Beberapa strategi lain yang juga digunakan:



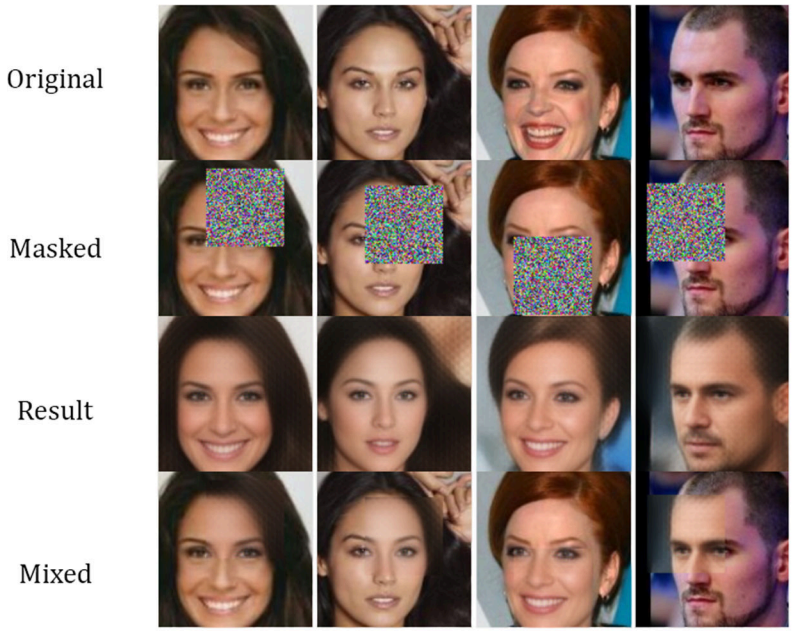
Gambar 4.7 Hasil *network generator* pada GAN standar ketika *discriminator* tidak seimbang, terlalu lemah atau terlalu kuat dari *generator*

1) Penggunaan *noisy label*, dengan probabilitas 5%. *Noisy label* yang dimaksud adalah melakukan *flipping* label *ground truth* untuk *network discriminator*. Ketika nilai *ground truth* bernilai 1 untuk citra *real* dan 0 untuk citra sintesis, dengan peluang kejadian sebesar 5% dilakukan penukaran untuk kedua nilai ini, nilai 0 untuk citra *real* dan 1 untuk citra sintesis.
2) *Smooth labeling*. Penggunaan label dengan nilai 1 dan 0 untuk kelas *real* dan sintesis dinilai dapat menyebabkan *network discriminator over-confidence* dalam menentukan kelas dari citra masukan. Sehingga digunakan nilai 0.9 untuk kelas *real* dan 0.1 untuk kelas sintesis. Selain itu, dilakukan pemilihan nilai *random* dengan rentang $0.8 - 1.2$ untuk kelas *real* dan $0. -0.3$ untuk kelas sintesis. Hal ini bertujuan untuk menyulitkan *network discriminator*.
3) Penambahan *noise* untuk input *discriminator.* Citra masukan pada *network discriminator* merupakan citra *ground truth* dan citra hasil sintesis, ditambahkan dengan nilai *random normal* untuk menjaga agar *discriminator* dapat menangkap distribusi masukan dengan lebih stabil.
4) Pemilihan beberapa kombinasi *hyper-parameter*

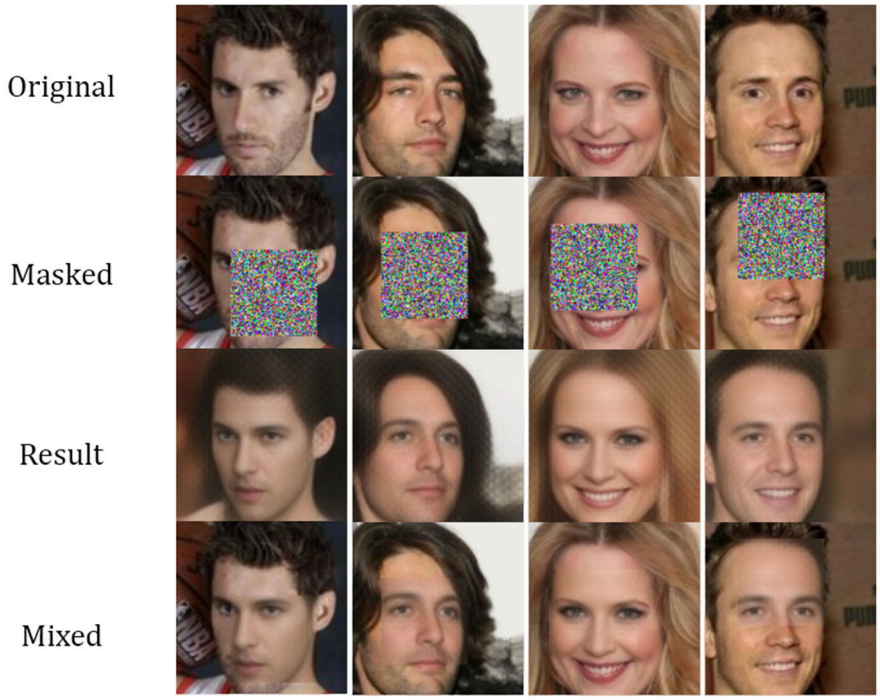
Namun, beberapa strategi yang telah digunakan ini tidak dapat mengantar *network* yang diajukan pada kondisi kesetimbangan antara *generator* dan *discriminator*. Hal ini mengartikan bahwa proses *training* GAN standar cukup sulit untuk dilakukan dengan bentuk *network* yang diajukan. Sehingga, WGAN digunakan untuk

mengatasi masalah ketidakstabilan dan kesulitan ini. Pada saat penggunaan WGAN, kondisi kesetimbangan antara *generator* dan *discriminator* dapat tercapai dengan mudah, strategi yang dilakukan untuk membantu kestabilan WGAN cukup dengan penambahan *noise* untuk masukan dari *discriminator*.

Pada tahap kedua ini, penggunaan *loss discriminator* dilakukan secara simultan dengan *training discriminator* dan *training generator*, dengan rasio 1:5. Pemilihan rasio 1:5 berdasarkan *trend* yang ada di kalangan peneliti. Penambahan *discriminator local* bertujuan memberi detail yang lebih spesifik, sementara *global* bertujuan untuk memberi kualitas perseptual yang lebih baik secara keseluruhan pada citra keluaran. Sesuai dengan algoritma WGAN, sebelum gradien dari *loss* diteruskan ke seluruh *network* untuk pembaharuan bobot, operasi *clipping* gradien dilakukan dengan batas *clipping* yaitu $[-c, c]$, dimana diberikan nilai $c = 0.01$. *Clipping* ini berguna sebagair regulator nilai gradien. Bobot yang diberikan untuk *loss adversarial discriminator local* dan *global* terhadap *loss* total adalah 0.1 dan 0.1. Hasil *inpainting* dari *network generator* setelah 10,000 *step update* bobot dilakukan dengan menggunakan tambahan *loss adversarial* dari *local* dan *global discriminator* disajikan pada Gambar 4.8. Pada Gambar 4.8 terlihat bahwa hasil



Gambar 4.8 Hasil *network generator* step 25,000, tahap kedua dengan tambahan *loss adversarial* dari *local* dan *global discriminator*



Gambar 4.9 Hasil *network generator* pada step 40,000, tahap ketiga dengan tambahan *loss landmark* dan *semantic parsing*

inpainting dengan penambahan *loss adversarial* menunjukkan hasil yang lebih realistis secara perseptual dibanding hasil sebelumnya yang diperoleh tanpa penggunaan *loss adversarial*, yaitu $KL - Divergence$ dan *feature reconstruction loss* saja.

Tahap terakhir, nilai *loss landmark* beserta *semantic parsing* ditambahkan pada *network*, yaitu ($\mathcal{L}_h$) dan ($\mathcal{L}_s$), dengan hasil disajikan pada Gambar 4.9. Penambahan *loss landmark* dan *semantic parsing* ini memiliki tujuan yang sama, yaitu meningkatkan hasil citra sintetis denga kualitas perseptual yang lebih baik sehingga citra yang dihasilkan semakint terlihat realistis. Dari Gambar 4.9 secara observasi citra yang dihasilkan terlihat bahwa penambahan dua kriteria *loss* ini dapat membantu meningkatkan kualitas perseptual dari citra yang dihasilkan sehingga citra yang dihasilkan terlihat lebih realistis.

4.1.4 Hasil *Inpainting Testing Network* Berdasarkan Kualitas Gambar

Pengujian *network* yang telah dilatih dilakukan dengan menggunakan data *testing* (bukan data *training*). Metrik yang digunakan untuk penilaian *testing* adalah SSIM dan PSNR kemudian dibandingkan dengan metode terdahulu seperti CE

Tabel 4.3 Hasil SSIM antara citra keluaran dengan masukan (SSIM semakin besar semakin baik)

Ukuran *Masking*	Minimum	Maksimum	Rerata	Standar Deviasi
32	0.0702	0.899	0.665	$\pm$0.0814
40	0.0735	0.906	0.662	$\pm$0.0815
48	0.0715	0.896	0.659	$\pm$0.0820
56	0.0710	0.898	0.654	$\pm$0.0824
64	0.071	0.908	0.646	$\pm$ 0.083

(Pathak, dkk., 2016), serta GFC (Yijun, dkk., 2017). Penilaian secara kualitatif dilakukan dengan melakukan observasi manual oleh beberapa responden.

Beberapa ukuran *masking* yang dipakai untuk *testing* adalah 32, 40, 48, 56, 64. Rata-rata, maksimum, minimum, serta standar deviasi dari SSIM yang diperoleh untuk tiap skenario *masking* disajikan pada Tabel 4.3, sementara untuk PSNR disajikan pada Tabel 4.4. Komparasi hasil yang diperoleh dengan metode terdahulu disajikan pada Tabel 4.5. Metode sebelumnya, GFC (Yijun, dkk., 2017) memperoleh nilai SSIM rata-rata 0.841 sementara PSNR pada 20.0. Di sisi lain, *Context Encoder* (CE) (Pathak, dkk., 2016) memperoleh skor SSIM dan PSNR sebesar 0.818 dan 19.3 pada konteks *inpainting* citra wajah. Dilihat dari faktor kuantitatif berupa PSNR, metode yang diajukan berhasil mendapat nilai PSNR rerata yang lebih tinggi dibanding dua metode sebelumnya (GFC dan CE) dengan

Tabel 4.4 Hasil PSNR antara citra keluaran dengan masukan (PSNR semakin besar semakin baik)

Ukuran *Masking*	Minimum	Maksimum	Rerata	Standar Deviasi
32	11.789	29.922	21.528	$\pm$1.958
40	11.890	29.690	21.459	$\pm$1.962
48	11.847	29.775	21.378	$\pm$1.976
56	11.831	28.989	21.248	$\pm$1.991
64	12.098	29.448	21.016	$\pm$ 1.997

Tabel 4.5 Perbandingan hasil yang diperoleh dengan metode terdahulu

Metrik Penilaian	CE	GFC	Metode yang Diajukan	
			Rata-rata	Maksimum
PSNR	19.3	20.0	21.528	29.922
SSIM	0.818	0.843	0.665	0.908

nilai PSNR sebesar 21.528, sementara nilai rerata SSIM yang diperoleh lebih rendah dibanding metode terdahulu dengan nilai sebesar 0.665. Sementara dengan menggunakan skor SSIM, keunggulan metode yang diajukan terletak pada SSIM maksimum yang mungkin diperoleh. Namun kembali lagi, karena metrik penilaian PSNR dan SSIM tidak dapat menjamin kualitas perseptual dari citra sintesis (Yijun dkk., 2017), nilai ini tidak dapat dijadikan pedoman utama sebagai penentu kualitas perseptual/tingkat kerealistisan citra yang dihasilkan. Nilai PSNR menyatakan keberhasilan *generator* dalam menghilangkan *noise* dalam merubah bagian *masking* (berupa *noise*) hingga menyerupai citra asli, sedangkan SSIM menyatakan kemiripan dari citra hasil *inpainting* dengan citra asli. Semakin mirip citra hasil *inpainting* dengan citra asli berarti tingkat *noise* yang dihasilkan proses *inpainting* juga semakin rendah. Sehingga, ketika proses *inpainting* menunjukkan hasil yang baik dengan meningkatnya SSIM, nilai PSNR juga ikut meningkat.

4.1.5 Hasil *Inpainting Testing Network* Secara Visual

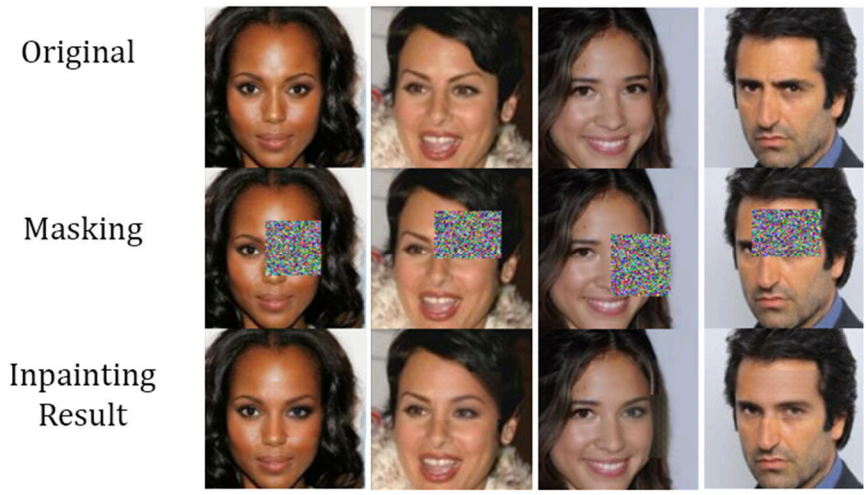
Selain penilaian kuantitatif dengan PSNR dan SSIM, penilaian secara subjektif juga dilakukan dengan observasi secara manual oleh beberapa pengamat. Penilaian ini dilakukan dengan menyajikan beberapa citra hasil rekonstruksi bersamaan dengan citra asli terhadap tujuh responden, kemudian responden diminta untuk menentukan keaslian citra yang diberikan. Pada proses penilaian ini, disajikan sepuluh citra asli dan sepuluh citra hasil rekonstruksi dalam urutan yang acak, dengan jumlah responden sebanyak tujuh orang. Contoh *form* yang digunakan disajikan pada Lampiran 4.1. Hasil penilaian dari responden yang diperoleh disajikan pada Lampiran 4.2, dengan *confusion matrix* disajikan dalam bentuk tabel pada Tabel 4.6. Dari Tabel 4.6, dari sepuluh citra hasil *inpainting* yang diberikan, dapat disimpulkan bahwa tingkat keberhasilan *inpainting* sehingga pengamat tidak

Tabel 4.6 Hasil penilaian responden

		Penilaian responden	
		Original	Hasil *inpainting*
Ground truth	*Original*	52	18
	Hasil *inpainting*	33	37

dapat membedakan *hasil inpainting* terhadap citra *original* adalah sebesar $\frac{33}{(33+37)} = \frac{33}{70} = 47.1\%$.

Beberapa hasil yang diperoleh pada data *test* disajikan pada Gambar 4.10. *Network generator* berhasil melakukan *inpainting* dengan kualitas perseptual yang baik. Pada beberapa kondisi bagian hasil rekonstruksi masih menunjukkan sedikit perbedaan warna dengan piksel sekitarnya. Namun dari hasil yang ditunjukkan, ketika *network generator* mampu merekonstruksi bagian hilang pada data uji, hal ini mengartikan bahwa *network* telah berhasil melakukan *learning* dan kondisi *overfitting* tidak terjadi pada *network* yang dilatih. Proses *inpainting* tetap dapat dilakukan pada citra wajah yang *unaligned*, atau ketika bagian yang diberi *masking* merupakan bagian yang memerhatikan keterkaitan spasial.



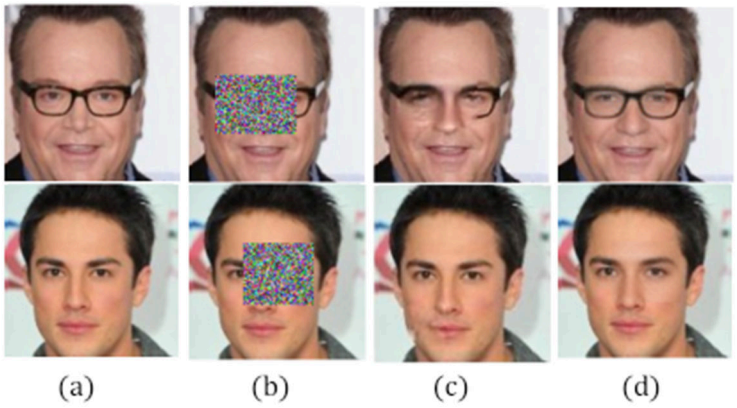
Gambar 4.10 Hasil *network generator* pada beberapa masukan data *test*

4.1.6 Pembahasan Masalah Keterkaitan Spasial

Masalah keterkaitan spasial didefinisikan sebagai masalah yang timbul saat *inpainting* ketika bagian *missing region* yang akan direkonstruksi memerlukan pengetahuan semantik terkait bagian yang hilang dengan bagian sekitarnya. Seperti sebagian bibir, atau bagian kacamata. Jika bagian yang hilang merupakan setengah bagian bibir berwarna merah, maka bagian yang direkontruksi harus menghasilkan warna merah yang serupa, agar tidak terjadi ketimpangan tingkat saturasi atau warna yang berbeda. Demikian pula, jika bagian yang hilang merupakan bagian kacamata, maka bagian yang direkontruksi juga harus menghasilkan bagian kacamata yang sesuai.

Masalah keterkaitan spasial ini timbul pada metode yang diajukan oleh Yijun, dkk pada metode GFC, seperti disajikan pada Gambar 4.11. Kolom (a) menunjukkan citra asli sebelum diberi *masking*, kolom (b) merupakan citra asli (a) setelah diberi *masking*, kolom (c) menunjukkan hasil *inpainting* metode GFC, sementara (d) merupakan hasil *inpainting* metode yang diajukan. Terlihat pada Gambar 4.11 kolom (c), metode GFC gagal merekonstruksi bagian yang hilang, hasil *inpainting* terlihat tidak realistis karena tidak konsisten dengan piksel sekitarnya.

Masalah ini diselesaikan dengan penambahan *feature reconstruction loss*. *Feature reconstruction loss* berdasarkan VGG-Net layer pertama, kedua, dan ketiga dapat membantu mempertahankan korelasi spasial, seperti yang disampaikan oleh

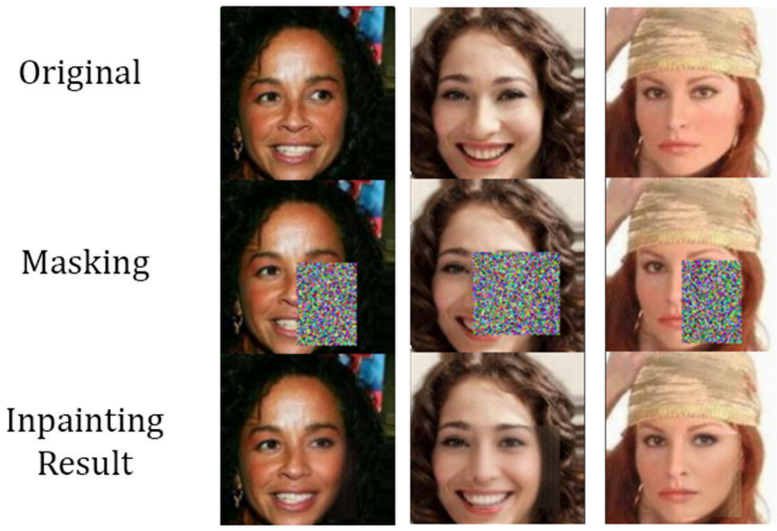


Gambar 4.11 Perbandingan hasil yang diperoleh dalam mempertahankan keterkaitan spasial

Hou, dkk. (Hou dkk., 2019). Berdasarkan percobaan yang telah dilakukan, terllihat pada Gambar 4.11 kolom (d), hasil *inpainting* terlihat lebih realistis dan konsisten dengan piksel sekitarnya. Penggunaan *feature reconstruction loss* dapat membantu mempertahankan keterkaitan spasial bahkan pada kasus *inpainting*.

Masalah ini diselesaikan dengan penambahan *feature reconstruction loss*. *Feature reconstruction loss* berdasarkan VGG-Net layer pertama, kedua, dan ketiga dapat membantu mempertahankan korelasi spasial, seperti yang disampaikan oleh Hou, dkk. (Hou dkk., 2019). Berdasarkan percobaan yang telah dilakukan, terllihat pada Gambar 4.11 kolom (d), hasil *inpainting* terlihat lebih realistis dan konsisten dengan piksel sekitarnya. Penggunaan *feature reconstruction loss* dapat membantu mempertahankan keterkaitan spasial bahkan pada kasus *inpainting*.

Beberapa hasil *inpainting* dari metode yang diajukan disajikan pada Gambar 4.12. Hasil keluaran *network generator* dengan citra masukan terdapat *missing region* di sebagian bibir. Warna bibir yang dihasilkan secara kualitatif atau observasi terlihat lebih senada dengan warna sekitarnya. Hal ini menunjukkan bahwa penambahan *feature reconstruction loss* dapat meningkatkan kualitas perseptual citra yang dihasilkan, beserta membantu mempertahankan masalah keterkaitan spasial pada kasus *inpainting*.

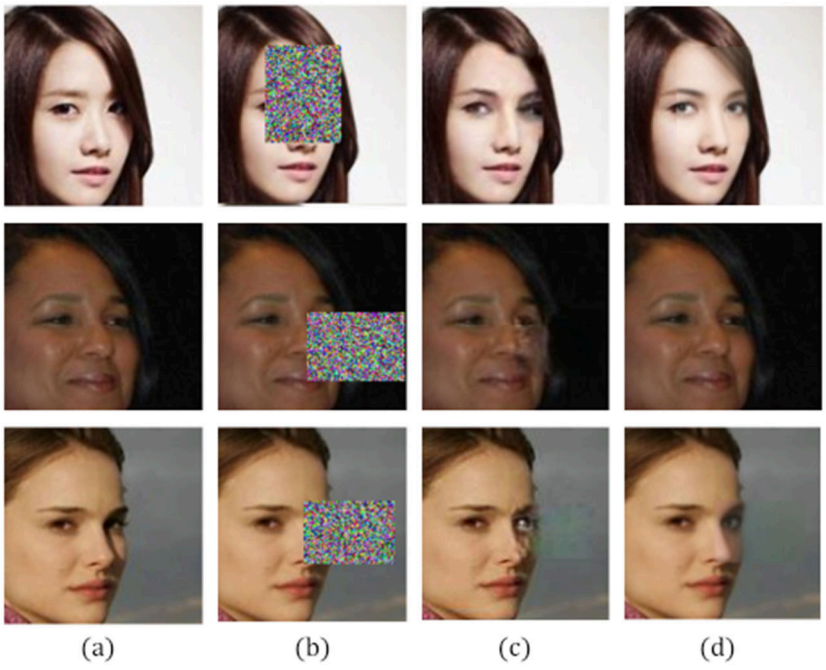


Gambar 4.12 Hasil *network generator* pada kasus *inpainting* yang memperhatikan keterkaitan spasial

4.1.7 Pembahasan Masalah Citra Wajah *Unaligned*

Citra wajah *unaligned* yang dimaksud adalah citra wajah dengan kondisi objek wajah tidak lurus berhadapan dengan arah pengambil gambar wajah, atau posisi wajah tidak tepat tegak lurus terhadap bidang horizontal sumbu gambar. Masalah *inpainting* muncul ketika citra wajah yang akan dilakukan proses *inpainting* merupakan citra wajah yang *unaligned*. Contoh citra wajah *unaligned* seperti disajikan pada Gambar 4.13. Kolom (a) menunjukkan citra asli sebelum masking, (b) menunjukkan citra (a) setelah diberi masking, (c) menunjukkan hasil *inpainting* metode GFC, (d) merupakan hasil *inpainting* dengan metode yang diajukan. Hasil *inpainting* pada metode GFC menunjukkan bahwa metode GFC kurang berhasil melakukan sintetis bagian yang hilang saat citra masukan berupa wajah *unaligned.* Metode GFC menggunakan *loss semantic parsing* dan jarak *euclid* pada domain RGB untuk kriteria *loss*nya.

Masalah ini diselesaikan dengan penggunaan *loss* tambahan, yaitu *landmark face*, yang berfungsi untuk memastikan *landmark* wajah tetap berada sesuai orientasinya, beserta *pre-trained* VGG-Net untuk menangkap *hidden feature* wajah sehingga proses *inpainting* dapat tetap dilakukan ketika citra wajah yang



Gambar 4.13 Perbandingan hasil yang diperoleh pada kasus citra wajah *unalgined*

Gambar 4.14 Hasil *network generator* pada *masukan* citra wajah *unaligned*

menjadi masukan berupa citra *unaligned face*. Hasil yang diperoleh metode yang diajukan disajikan pada Gambar 4.13 kolom (d). Terlihat pada gambar yang disajikan bahwa *inpainting* tetap dapat dilakukan pada citra wajah yang *unaligned.* Metode *inpainting* yang diajukan berhasil merekonstruksi bagian yang hilang dengan tetap terlihat realistis.

Beberapa hasil *inpainting* pada citra wajah *unaligned* pada *network generator* yang diajukan pada penelitian ini disajikan pada Gambar 4.14. Baris pertama menunjukkan citra wajah *unaligned* yang asli sebelum diberi *masking*, baris kedua menunjukkan citra yang telah diberi *masking*, baris ketiga menunjukkan hasil *network generator* dalam mengembalikan *missing region*. Terlihat pada Gambar 4.14, ketika citra dengan *masking* yang masuk berupa citra *unaligned*, *generator* tetap mampu melakukakn proses restorasi terlepas dari kondisi citra masukan merupakan citra *unaligned face* atau tidak.

BAB 5
KESIMPULAN

5.1 Kesimpulan

Kesimpulan yang dapat ditarik dari penelitian yang telah dilakukan adalah sebagai berikut:

1) Metode *inpainting* pada citra wajah *unaligned* dapat dilakukan dengan menggunakan *Generative Adversarial Network* (GAN) dengan tambahan *loss network* berupa *feature reconstruction loss* menggunakan *pretrained network* VGGNet, *landmark loss* dari *facial landmark network,* serta *semantic parsing loss* dari *semantic parsing network*.
2) Selain mengatasi masalah ketika citra wajah *unaligned*, penambahan *feature reconstruction loss* dengan VGG-Net beserta *landmark loss* berhasil membantu mempertahankan keterkaitan spasial dari citra keluaran, terlihat dari kualitas perseptual citra keluaran yang lebih baik dan realistis.
3) Berdasarkan beberapa skenario *masking yang* dilakukan saat percobaan dengan data *testing*, penerapan inpainting menggunakan GAN dengan tambahan *loss* ini memungkinkan untuk mendapatkan hasil PSNR yang lebih baik, yaitu 21.528, dengan nilai SSIM maksimum yang dapat diperoleh yaitu 0.908. Masalah *inpainting* yang timbul pada penelitian sebelumnya ketika citra wajah *unaligned* berhasil teratasi. Terlihat bahwa metode yang diajukan berhasil melakukan sintesis bagian wajah pada citra wajah yang *unaligned* ataupun memerhatikan keterkaitan spasial.

5.2 Saran

Berdasarkan hasil yang telah diperoleh, saran yang dapat diajukan untuk penelitian *inpainting* berbasi GAN berikutnya:

1) menggunakan jenis GAN yang lain dengan fungsi *loss* yang berbeda, seperti *Least Square* GAN (LS-GAN), WGAN dengan *Gradient*

Penalty (WGAN-GP) yang merupakan pengembangan dari WGAN, atau jenis GAN lain sesuai dengan tren terbaru,

2) kombinasi bobot *weight* yang berbeda untuk melakukan analisis lebih lanjut terhadap efek masing-masing *bobot* terhadap hasil *inpainting* secara keseluruhan,
3) penggunaan *pre-trained network* lain seperti VGGNet-19, serta jumlah *layer* yang digunakan untuk *feature reconstruction loss*,
4) pengembangan metode untuk metrik penilaian kualitas perseptual dari hasil *inpainting*,
5) penggunaan metode *inpainting* untuk kasus yang lebih khusus, seperti estimasi wajah pengguna masker dengan asusmi bagian masker adalah bagian *masking* yang akan dilakukan proses *inpainting*,
6) *inpainting* pada bagian dengan tingkat variasi yang besar: rambut,
7) metode *post-processing* untuk meningkatkan kualitas citra keluaran, seperti *image blending* berbasis Jacobi-Poisson,
8) pengembangan metode *inpainting* pada kasus yang lebih umum, tidak terbatas pada citra wajah saja.

DAFTAR PUSTAKA

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. *ArXiv*, *abs/1701.0*. http://arxiv.org/abs/1701.07875

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum Learning. *International Conference on Machine Learning*, 41–48. https://doi.org/10.1017/S1047951100000925

Chollet, F. (2016). *Building Autoencoders in Keras*. https://blog.keras.io/building-autoencoders-in-keras.html

Frans, K. (2016). *Variational Autoencoders Explained*. http://kvfrans.com/variational-autoencoders-explained/

Furht, B. (Ed.). (2008). Image Inpainting. In *Encyclopedia of Multimedia* (p. 329). Springer US. https://doi.org/10.1007/978-0-387-78414-4_344

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org

Goodfellow, I. J., Puget-Abadie, J., Mirza, M., Bing, X., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, *27*, 4089–4099.

Hacohen, G., & Weinshall, D. (2019). *On The Power of Curriculum Learning in Training Deep Networks*. http://arxiv.org/abs/1904.03626

Hays, J., & Efros, A. A. (2007). Scene completion using millions of photographs. *ACM Transactions on Graphics*, *26*(3), 1–7. https://doi.org/10.1145/1276377.1276382

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hou, X., Shen, L., Sun, K., & Qiu, G. (2017). Deep Feature Consistent Variational Autoencoder. *IEEE Winter Conference on Applications of Computer Vision*, 1133–1141. https://doi.org/10.1109/WACV.2017.131

Hou, X., Sun, K., Shen, L., & Qiu, G. (2019). Improving variational autoencoder

with deep feature consistent and generative adversarial training. *Neurocomputing*, *341*, 183–194. https://doi.org/10.1016/j.neucom.2019.03.013

Jimei, Y., Price, B., Cohen, S., Lee, H., & Ming-Hsuan, Y. (2016). Object contour detection with a fully convolutional encoder-decoder network. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 193–202. https://doi.org/10.1109/CVPR.2016.28

Kawai, N., & Yokoya, N. (2012). Image Inpainting Considering Symmetric Patterns. *International Conference on Pattern Recognition*, 2744–2747.

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, *Ml*, 1–14.

Krizhevsky, A., Sitskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *International Conference on Neural Information Processing Systems*, *1*, 1097–1105. https://doi.org/10.1201/9781420010749

Liao, H., Funka-Lea, G., Yefeng, Z., Jiebo, L., & Zhou, S. K. (2018). Face Completion with Semantic Knowledge and Collaborative Adversarial Learning. *Asian Conference on Computer Vision*, 382–397. https://doi.org/10.1007/978-3-030-20887-5_24

Ma, R., Hu, H., Wang, W., Xu, J., & Li, Z. (2019). Photorealistic face completion with semantic parsing and face identity-preserving features. *ACM Transactions on Multimedia Computing, Communications and Applications*, *15*(1), 1–18. https://doi.org/10.1145/3300940

Ma, T., Wang, F., Cheng, J., Yu, Y., & Chen, X. (2016). A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. *Sensors (Switzerland)*, *16*(10). https://doi.org/10.3390/s16101701

Ng, A., Ngiam, J., Foo, C. Y., Mai, Y., Suen, C., Coates, A., Maas, A., Hannun, A., Huval, B., Wang, T., & Tandon, S. (n.d.). *Autoencoders*. Retrieved December 4, 2019, from http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context Encoders: Feature Learning by Inpainting. *IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition, 2536–2544. https://doi.org/10.1109/CVPR.2016.278

Qureshi, M. A., Deriche, M., Beghdadi, A., & Amin, A. (2017). A critical survey of state-of-the-art image inpainting quality assessment metrics. *Journal of Visual Communication and Image Representation*, *49*, 177–191. https://doi.org/10.1016/j.jvcir.2017.09.006

Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 1–16.

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*, 1–14.

Stoll, S. (2017). *Convolutional Netural Networks and Caffe*. https://hackaday.io/project/26979-vision-based-grasp-learning-for-prosthetics/log/65975-convolutional-neural-networks-and-caffe

Tanaka, T., Kawai, N., Nakashima, Y., Sato, T., & Yokoya, N. (2018). Iterative applications of image completion with CNN-based failure detection. *Journal of Visual Communication and Image Representation*, *55*(May), 56–66. https://doi.org/10.1016/j.jvcir.2018.05.015

Yijun, L., Sifei, L., Jimei, Y., & Ming-Hsuan, Y. (2017). Generative Face Completion. *IEEE Conference on Computer Vision and Pattern Recognition*, 5892–5900. https://doi.org/10.1109/CVPR.2017.624

Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., & Sang, N. (2018). BiSeNet: Bilateral segmentation network for real-time semantic segmentation. *European Conference on Computer Vision*, *11217 LNCS*, 334–349. https://doi.org/10.1007/978-3-030-01261-8_20

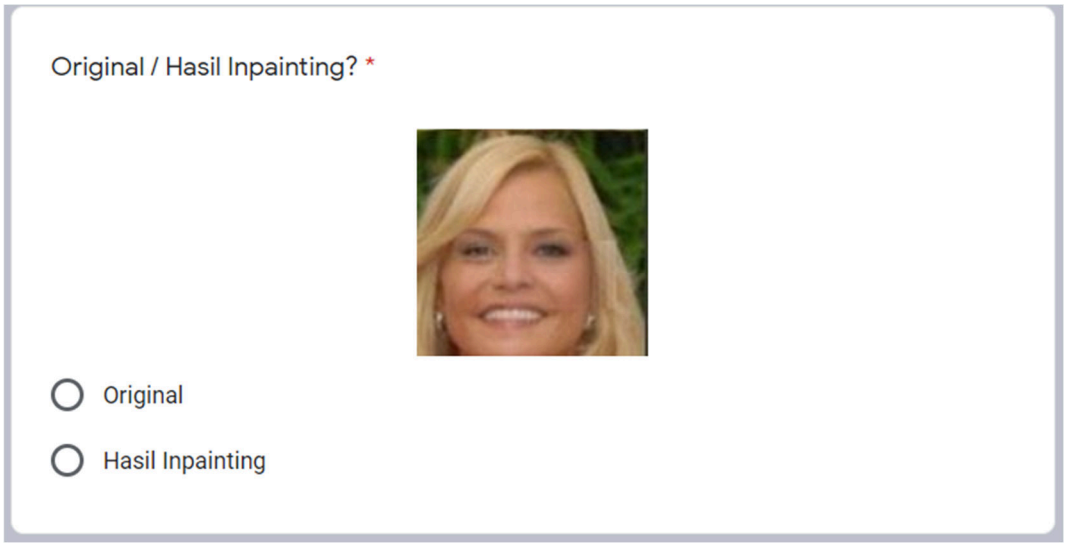
Zhou, W., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, *13*(4), 600.

Ziwei, L., Ping, L., Xiaogang, W., & Xiaoou, T. (2015). Deep learning face attributes in the wild. *IEEE International Conference on Computer Vision*,

3730–3738. https://doi.org/10.1109/ICCV.2015.425

LAMPIRAN

Lampiran 4.1. **Contoh tampilan *form* penilaian untuk responden**



Lampiran 4.2. **Tabel hasil observasi manual dengan responden**

Nomor	***Ground Truth***	**Penilaian Responden**	
		Original	**Hasil *Inpainting***
1	Original	2	5
2	Hasil *Inpainting*	1	6
3	Original	5	2
4	Original	6	1
5	Hasil *Inpainting*	6	1
6	Hasil *Inpainting*	3	4
7	Hasil *Inpainting*	5	2
8	Hasil *Inpainting*	5	2
9	Original	7	0
10	Original	5	2
11	Original	4	3
12	Hasil *Inpainting*	5	2
13	Original	5	2
14	Hasil *Inpainting*	1	6
15	Original	5	2
16	Hasil *Inpainting*	2	5
17	Hasil *Inpainting*	2	5
18	Original	7	0
19	Original	6	1
20	Hasil *Inpainting*	3	4

BIODATA PENULIS

Avin Maulana, merupakan anak pertama dari empat bersaudara. Putra dari Ahmadi dan Mulyani, merantau dari Madura hingga akhirnya berdomisili di Bali. Penulis menempuh pendidikan di MI Raudlatul Mustarsyidin, SMP Harapan Mulia, MA Al Ma'ruf Denpasar, dan pada tahun 2014 sebagai mahasiswa program studi S1 Matematika di Universitas Brawijaya Malang hingga lulus dengan predikat cumlaude pada tahun 2018. Tertarik dengan bidang komputer khusunya *computational intelligence*, penulis memilih bidang Sains Komputasi sebagai tugas akhir pada masa studi S1 dengan topik penelitian Optimasi Berbasis Algoritma Genetika, hingga akhirnya melanjutkan ke Pascasarjana Teknik Informatika iTS pada Tahun 2018. Penulis dapat dihubungi melalui email di afin.maulana@gmail.com