FACIAL INPAINTING PADA CITRA WAJAH *UNALIGNED* MENGGUNAKAN *GENERATIVE ADVERSARIAL NETWORK* DENGAN *FEATURE RECONSTRUCTION LOSS*

Avin Maulana[1)], Chastine Fatichah[2)], dan Nanik Suciati[3)]
[1, 2, 3)] Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia 60111
e-mail: afin.maulana@gmail.com[1)], chastine@if.its.ac.id[2)], nanik@if.its.ac.id[3)]

ABSTRAK

Facial inpainting atau restorasi citra wajah merupakan proeses merekonstruksi kembali bagian yang hilang pada citra wajah sedemikian sehingga citra hasil rekonstruksi dapat tetap terlihat realistis, sehingga pihak pengamat tidak dapat mengenali bagian yang merupakan hasil rekontsruksi. Beberapa penelitian sebelumnya melakukan inpainting menggunakan Generative Adversarial Network (GAN). Namun terdapat masalah yang timbul pada hasil inpainting ketika proses inpainting dilakukan pada citra wajah yang miring (unaligned). Hasil rekonstruksi menunjukkan hasil yang terlihat tidak realistis. Metode inpainting gagal untuk merekonstruksi ulang bagian wajah yang hilang ketika citra masukan berupa citra unaligned. Sehingga, diajukan metode pengembangan untuk melakukan facial inpainting menggunakan GAN dengan feature reconstruction loss dan dua jenis discriminator untuk mengatasi masalah yang timbul tersebut. Feature reconstruction loss adalah loss yang diperoleh dari penggunaan VGG-Net. Hasil yang diperoleh menunjukkan bahwa penambahan loss dan dua jenis discriminator dapat meningkatkan kualitas visual citra yang diperoleh dengan nilai PSNR dan SSIM yang lebih baik dari metode pendahulu.

***Kata kunci**: Adversarial, GAN, inpainting, restorasi, unaligned.*

FACIAL INPAINTING IN UNALIGNED FACE IMAGES USING GENERATIVE ADVERSARIAL NETWORK WITH FEATURE RECONSTRUCTION LOSS

Avin Maulana[1)], Chastine Fatichah[2)], and Nanik Suciati[3)]
[1, 2, 3)] Department of Informatics, Institut Sepuluh Nopember
Surabaya, Indonesia 60111
e-mail: afin.maulana@gmail.com[1)], chastine@if.its.ac.id[2)], nanik@if.its.ac.id[3)]

ABSTRACT

Facial inpainting or face restoration is a process to reconstruct some missing region on face images such that the inpainting results still can be seen as a realistic and original image without any missing region, in such a way that the observer could not realize whether the inpainting result is a generated or original image. Some of previous researches have done inpainting using generative network, such as Generative Adversarial Network. However, some problems may arise when inpainting algorithm have been done on unaligned face. The inpainting result show spatial inconsistency between the reconstructed region and its adjacent pixel, and the algorithm fail to reconstruct some area of face. Therefore, an improvement method in facial inpainting based on deep-learning is proposed to reduce the effect of the stated problem before, using GAN with additional loss from feature reconstruction and two discriminators. Feature reconstruction loss is a loss obtained by using pretrained network VGG-Net, Evaluation of the result shows that additional loss from feature reconstruction loss and two type of discriminators may help to increase visual quality of inpainting result, with higher PSNR and SSIM than previous result.

***Keywords**: Adversarial, GAN, inpainting, restoration, unaligned.*

I. PENDAHULUAN

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 [1]. 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, seperti logo pada suatu citra, atau restorasi citra yang mengalami kerusakan [2], seperti kerusakan fisik atau kerusakan yang terjadi pada saat proses transmisi data. Metode untuk melakukan *image inpainting* dapat dibagi menjadi 4 bagian, yaitu: 1) e*xemplar-based*; 2) *sparsity-based*; 3) *PDE-based;* 4) *Hybrid* [2]. Seiring dengan

DOI: http://dx.doi.org/10.12962/j24068535.v18i2.a1004

Gambar 1. Hasil metode GFC pada citra wajah *unaligned.*

perkembangan teknologi dan ketersediaan data, *inpainting* dapat dilakukan dengan menggunakan metode yang berbasis konsep *learning*, seperti *artificial neural network*. Konsep *learning* berarti sistem akan mempelajari/mengekstrak pola dari data, kemudian menyelesaikan masalah yang melibatkan pengetahuan berdasarkan pengetahuan yang diekstrak [3]. Jenis *network* yang dapat digunakan pada *inpainting* seperti *Variational Auto-Encoder* (VAE) atau *Generative Adversarial Network* (GAN).

Salah satu metode berbabis GAN untuk melakukan *inpainting* diajukan oleh Yijun, dkk. [4]. Yijun memanfaatkan proses *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. 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 yang diajukan Yijun diberi nama metode GFC (*Generative Face Completion*). Dengan metode GFC ini, diharapkan mendapat citra hasil *inpainting* yang detail, namun tetap realistis. Hanya saja, masalah timbul ketika pendekatan yang dipakai metode GFC diterapkan pada citra *unaligned face*, yaitu citra wajah yang miring, atau orientasi wajah tidak tegak lurus terhadap sumbu horizontal. Hasil *inpainting* metode GFC pada citra *unaligned face* disajikan pada Gambar 1. Bagian (a) menunjukkan citra asli, bagian (b) menunjukkan citra yang diberi *masking*, bagian (c) hasil metode GFC. Terlihat pada bagian (c) metode GFC gagal merekonstruksi bagian mata yang hilang.

Masalah yang timbul pada proses *inpainting* ini akan diselesaikan dengan menggunakan kriteria *loss* tambahan, yaitu *feature reconstruction loss* berdasarkan *pre-trained network* dari *Visual Geometry Group* (VGGNet) [5]. Berdasarkan penelitian sebelumnya [6], VGGNet dapat digunakan untuk membantu mempertahankan *deep-feature* dari citra pada *network* VAE yang digunakan. Sehingga pada penelitian ini, diajukan pengembangan metode *inpainting* berbasis GFC dengan tambahan penggunaan *network* VGGNet sebagai kriteria *loss* dalam proses *inpainting* untuk membantu menyelesaikan masalah yang timbul pada citra keluaran hasil *inpainting* pada wajah *unaligned*. Proses *training* dilakukan dengan skema *curriculum learning* dengan dua tahapan, dengan dataset yang digunakan merupakan data CelebA [7].

II. KAJIAN PUSTAKA

Tanaka, dkk. [8] 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 [9]. GAN menggunakan model *generative* (G) dan *discriminative* (D), dan 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. [10], 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) [5] sebagai pembanding antara citra *input* dengan citra *output*. Hal ini dilakukan karena *hidden-representation* dapat menangkap fitur penting yang berhubungan dengan kualitas perseptual, seperti korelasi spasial pada citra. Namun, metode yang diajukan oleh Hou, dkk. tidak menjelaskan tentang kemampuan metode yang diajukan ketika digunakan untuk masalah *inpainting*.

Penelitian yang menggunakan konsep GAN pada masalah *facial inpainting* juga dilakukan oleh Haofu, dkk. [1], dengan konsep *multi-task learning*. Proses rekonstruksi citra bagian wajah dilakukan bersamaan sekaligus dengan mengekstrak posisi *heat-map* dan *semantic segmentation* dari citra. Ketiga hasil dari model digunakan langsung sebagai pembanding yang berikutnya akan digunakan untuk memperbarui bobot pada *network*. Haofu juga mengajukan skema *concentrated inpainting*, dengan harapan komputasi yang dilakukan oleh model terfokus pada rekonstruksi region yang rusak atau hilang saja, bukan proses sintesis citra secara keseluruhan. Pendekatan Haofu berhasil menghasilkan citra rekonstruksi dengan baik terutama pada masalah *facial inpainting*, namun penggunaan *pre-trained network* VGG-Net digunakan secara *fine-tuning* sebagai generator saja. VGG-Net dapat dimanfaatkan lebih lanjut untuk menentukan nilai *loss* dari citra *input* dan *output* yang dihasilkan sehingga dapat meningkatkan kualitas perseptual citra yang direkonstruksi.

III. METODE YANG DIAJUKAN

Sebelum *training* dimulai, tahap praproses dilakukan pada dataset CelebA dengan operasi *cropping,* dan *resize* menjadi ukuran 128×128. Kemudian, pemberian *masking* pada masukan citra asli (*original image*) yang telah dilakukan pada data citra hasil praproses, I, dengan ukuran *masking* 64×64. Citra input yang telah diberi *masking*, I_m, menjadi *input* pada *network*. Berikutnya, akan dilakukan *training* dengan menggunakan tiga jenis *network*, yaitu: 1) *generator*, 2) VGG-Net, 3) *discriminator* (*local* dan *global*). Skema *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 dua tahap.

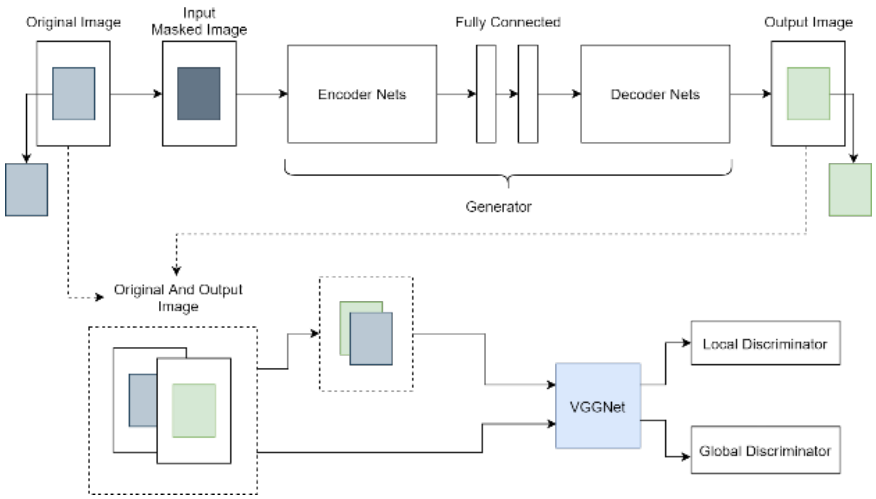
Tahap pertama, *training network* terdiri dari *training generator*. Pada tahap pertama ini, *update* bobot pada *generator* dilakukan dengan dua kriteria *loss* saja, yaitu *K-L divergence loss* ($\mathcal{L}_{KL}$), dan *feature reconstruction loss* ($\mathcal{L}_f$). Namun, pada metode yang diajukan, *feature reconstruction loss* bukan menggunakan jarak *euclid* antara citra hasil rekonstruksi dengan citra hasil *inpainting* pada domain RGB, melainkan jarak *euclid* kedua citra tersebut pada domain fitur VGGNet.

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 persamaan yang disajikan pada persamaan (1). Persamaan (1) 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 (2). Pada penelitian ini, jumlah *layer* yang digunakan pada domain VGGNet adalah 3. $C_l,\ W_l,\ H_l$ menyatakan jumlah *channel*, lebar, dan tinggi pada domain fitur VGGNet pada *layer* ke-l.

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

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

Tahap berikutnya, proses *training* dilakukan dengan tambahan *loss adversarial* dari dua buah *network discriminator*, yaitu *local* dan *global discriminator*. Pada penelitian ini, *discriminator* yang digunakan bukan *discriminator* GAN standar seperti pada GFC, melainkan *discriminator critic* seperti pada Wasserstein GAN [11]. Pemilihan jenis WGAN dibanding GAN standar karena dalam beberapa kasus GAN standar sangat sulit untuk mencapai kestabilan, sehingga dilakukan beberapa pengembangan untuk mengatasi masalah ini, salah satunya WGAN. Pada tahap ini, dua *discriminator network* digunakan, *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



Gambar 2. Metode inpainting dengan VGGNet dan dua discriminator yang diajukan.

/ *noise* dan tidak tajam. Secara keseluruhan, fungsi objektif dari *network* disajikan dalam persamaan (3), dengan λ_i menyatakan bobot dari masing-masing *loss* untuk meregulasi efek *loss* yang diberikan.

$$\mathcal{L} = \lambda_1 \mathcal{L}_{KL} + \lambda_2 \mathcal{L}_f + \left(\lambda_3 \mathcal{L}_{dl} + \lambda_4 \mathcal{L}_{dg}\right) \quad (3)$$

A. Arsitektur Generator

Arsitektur yang diajukan secara keseluruhan diilustrasikan pada Gambar 2. *Network generator* merupakan *network* berbasis VAE yang terdiri dari *encoder* dan *decoder,* mengikuti arsitektur dari DFC-VAE [7,10]. *Encoder* 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*. Setiap layer konvolusi disertai dengan *residual block* [12], 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. 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. Citra masukan berada pada domain nilai $[-1, 1]$, fungsi aktivasi yang digunakan pada keluaran *network* G menggunakan fungsi $tanh$, untuk menjaga domain hasil tetap di $[-1, 1]$.

B. Arsitektur Discriminator

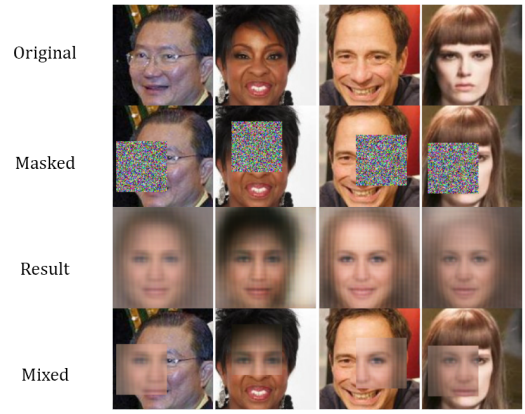
Arsitektur *discriminator* yang digunakan identik dengan bentuk *encoder*. Pembeda terletak pada *layer* terakhir yang digunakan sebagai keluaran. Pada *encoder*, keluaran berupa variabel z yang merupakan hasil *encoding* citra masukan pada domain laten z, sementara pada *discriminator*, hasil keluaran 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 *layer* yang digunakan. Jika pada *global discriminator* terdapat 5 *layer* konvolusi ditambah 1 *layer* konvolusi untuk menghasilkan keluaran skalar, *local discriminator* hanya terdapat 4 *layer* konvolusi ditambah 1 *laye* konvolusi akhir untuk menghasilkan 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 $64 \times 64 \times 3$.

ALGORITMA 1: PROSES *TRAINING*

Input: Citra dengan *masking* (I_m) dan tanpa *masking* (I), *learning rate*
Output: Citra hasil *inpainting* (I_r)

Initialization: $\boldsymbol{W}_{generator}, \boldsymbol{W}_{local}, \boldsymbol{W}_{global}$
***Repeat*:**
$Z = Encoder(I_m)$
$I_r = Decoder(Z)$
$\mathcal{L}_{KL} = \frac{1}{2}\left[\sum_{i=1}^{n} Z_{\mu,i}^2 + \sum_{i=1}^{n} Z_{\sigma,i}^2 - \sum_{i=1}^{n} \log\left(Z_{\sigma,i}^2 + 1\right)\right]$
for l = 1 to 3 do:
$\mathcal{L}_l = \frac{1}{2C_l W_l H_l} \sum_{c=1}^{C_l} \sum_{i=1}^{W_l} \sum_{j=1}^{H_l} \left(\psi_{c,i,j}(I) - \psi_{c,i,j}(\hat{I})\right)^2$
$\mathcal{L}_f = \sum_{l=1}^{3} \frac{100}{C_l^2} \mathcal{L}_l$
for l = 1 to 3 do:
$L_{gd} = Discriminator(I) - Discriminator(I_r)$
$L_{ld} = Discriminator(I) - Discriminator(I_r)$
$\boldsymbol{W}_{global} = \boldsymbol{W}_{global} - \nabla \boldsymbol{W}_{global}\, \mathcal{L}_{gd}$
$\boldsymbol{W}_{global} = clip(\boldsymbol{W}_{global}, -0.01, 0.01)$
$\boldsymbol{W}_{local} = \boldsymbol{W}_{local} - \nabla \boldsymbol{W}_{local}\, \mathcal{L}_{gd}$
$\boldsymbol{W}_{local} = clip(\boldsymbol{W}_{local}, -0.01, 0.01)$
$\boldsymbol{W}_{generator} = \boldsymbol{W}_{generator} - \nabla \boldsymbol{W}_{generator}\, (\mathcal{L}_{KL} + \mathcal{L}_f - \mathcal{L}_{gd} - \mathcal{L}_{ld})$
***Until* convergence**



Gambar 3. Hasil *Network Generator* saat *training* dimulai.

C. *Skema training*

Algoritma dari proses *training* yang dilakukan disajikan pada *listing* algoritma *training* berikut. *Learning rate* yang digunakan untuk *generator* adalah 0.0001, sementara untuk *discriminator* baik *local* maupun *global* digunakan nilai 0.0002. Rasio *training* antara *generator* dengan *discriminator* adalah 1:5.

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. Momentum pada ADAM bernilai 0.5, dengan batas tahap pertama *training* adalah 15000 *step*, ukuran *batch* yang digunakan pada saat *training* adalah 16.

IV. HASIL DAN PEMBAHASAN

Pada bab ini akan dipaparkan hasil beserta pembahasan baik pada saat proses *training* dilakukan atau setelah proses *training* selesai dilakukan.

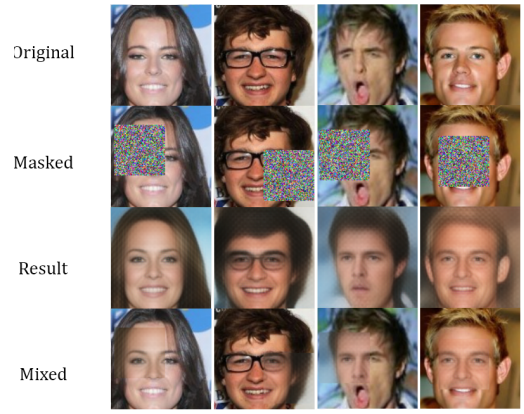
A. *Proses training*

Tahap pertama pelatihan *network* menggunakan dua jenis *loss*, yaitu K-*L 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 3, sementara saat *training* tahap pertama telah selesai dilalui disajikan pada Gambar 5. 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 *network*. Keluaran dari *network* merupakan keseluruhan citra dengan ukuran 128×128 piksel. Baris keempat merupakan gabungan antara citra *ground truth* dengan keluaran dari *network*.

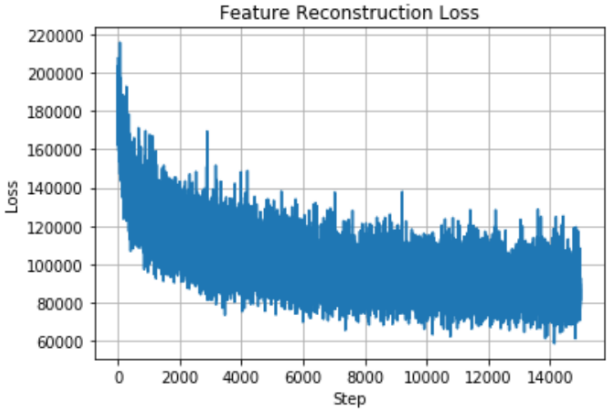
Penelitian sebelumnya pada metode GFC, tidak menggunakan *KL-Divergence loss*, dan hanya menggunakan jarak *euclid* antara citra masukan dengan citra keluaran pada domain RGB. Hasil *network generator* pada metode GFC dengan tahap pertama *loss* yang digunakan pada metode tesrsebut disajikan pada Gambar 5. Pada tahap ini, terlihat bahwa penambahan penggunaan kriteria *loss KL-Divergence* $(\mathcal{L}_{KL})$ beserta *Feature Reconstruction Loss* $(\mathcal{L}_f)$ menggunakan VGGNet dapat membantu menghasilkan citra *inpainting* dengan kualitas perseptual yang lebih baik dibanding penggunaan *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 7. 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 adalah layak / realistis.

Loss pertama yang ditambahkan pada *network* merupakan *loss adversarial* dari dua buah *network discriminator*, yaitu *loss discriminator local* dan *discriminator global*, $\mathcal{L}_{ld}$ dan $\mathcal{L}_{lg}$. 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 [9], 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 7. Sehingga, untuk mengatasi masalah ini, perlu dilakukan beberapa strategi untuk mencegah ketidakstabilan ini terjadi.

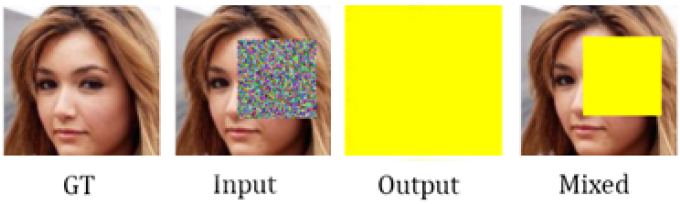


Gambar 4. Hasil *Network Generator* menggunakan dua *loss*: *KL-Divergence* dan *Feature Reconstruction Loss.*

Gambar 5. Hasil *Network Generator* tahap pertama pada metode sebelumnya, GFC. Tanpa *feature reconstruction loss* VGGNet dan *KL-Divergence Loss.*



Gambar 6. Nilai *loss* dari *network generator* pada 15000 *step* awal.



Gambar 7. Hasil *network generator* pada GAN standar ketika *discriminator* tidak seimbang, terlalu lemah atau terlalu kuat dari *generator*.

TABEL I
HASIL PSNR DAN SSIM YANG DIPEROLEH.

Metrik	Kuantitas	Metode yang Diajukan	Context Encoder (CE) [13]	Generative Face Completion (GFC) [4]
SSIM	Minimum	0.068	0.818	0.841
	Maksimum	0.890		
	Rata-rata	0.651		
	Standar deviasi	0.083		
PSNR	Minimum	11.657	19.3	20.2
	Maksimum	27.996		
	Rata-rata	21.067		
	Standar deviasi	1.907		

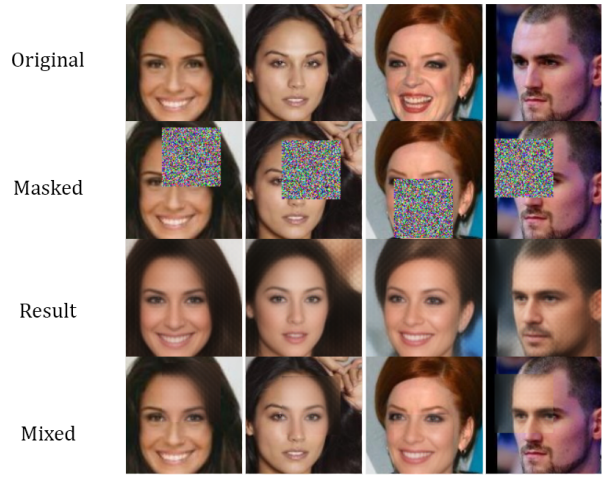
B. Hasil testing

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 [13], serta GFC [4]. Penilaian secara kualitatif dilakukan dengan melakukan observasi visual hasil yang diperoleh. Hasil PSNR dan SSIM yang diperoleh oleh metode yang diajukan disajikan pada Tabel I. Dari hasil PSNR yang diperoleh, metode yang diajukan berhasil mendapat nila rata-rata (*mean*) yang lebih baik dari metode terdahulu, yaitu 21,066. Sementara metode terdahulu mendapat nilai 19,3 dan 20,2, untuk metode CE [13] dan GFC [4]. PSNR yang lebih tinggi menandakan hasil yang diperoleh memiliki tingkat *noise* yang lebih rendah daripada PSNR yang lebih rendah. Sementara dari SSIM yang diperoleh, hasil metode yang diajukan mendapatkan hasil yang lebih variatif, rentang nilai lebih besar. Dengan nilai maksimum yang diperoleh untuk SSIM adalah 0.890.

Beberapa hasil *inpainting* pada citra wajah *unaligned* disajikan pada Gambar 9. Terlihat pada gambar yang disajikan, metode yang diajukan untuk melakukan *inpainting* berhasil melakukan rekonstruksi pada bagian wajah yang diberi *masking*, meskipun citra masukan yang diberikan merupakan citra wajah *unaligned*.

V. KESIMPULAN

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. Masalah yang timbul pada saat *inpainting* dilakukan pada citra wajah *unaligned* dapat teratasi dengan penambahan *loss* dengan *pretrained network* VGGNet, terlihat dari kualitas perseptual citra keluaran yang lebih baik dan realistis penerapan *inpainting,* serta metode yang diajukan berhasil melakukan sintesis bagian wajah



Gambar 8. Hasil *network generator* step 25,000, tahap kedua dengan tambahan *loss adversarial* dari *local* dan *global.*

Gambar 9. Hasil *network generator* pada *masukan* citra wajah *unaligned.*

pada citra wajah yang *unaligned*. Penggunaan GAN dengan tambahan *loss* ini memungkinkan untuk mendapatkan hasil PSNR yang lebih baik, yaitu 21.067, dengan nilai SSIM maksimum yang dapat diperoleh yaitu 0.890.

DAFTAR PUSTAKA

[1] L. Haofu, G. Funka-Lea, Z. Yefeng, L. Jiebo, dan S. K. Zhou, "Face Completion with Semantic Knowledge and Collaborative Adversarial Learning," dalam *Proc. Asian Conference on Computer Vision*, Dec. 2018, hal. 382–397, doi: 10.1007/978-3-030-20887-5_24.
[2] M. A. Qureshi, M. Deriche, A. Beghdadi, dan A. Amin, "A critical survey of state-of-the-art image inpainting quality assessment metrics," *J. Vis. Commun. Image Represent.*, vol. 49, hal. 177–191, 2017, doi: 10.1016/j.jvcir.2017.09.006.
[3] I. Goodfellow, Y. Bengio, dan A. Courville, *Deep Learning*. MIT Press, 2016.
[4] Y. Li, S. Liu, J. Yang, dan M.-H. Yang, "Generative Face Completion," dalam *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, hal. 5892–5900, doi: 10.1109/CVPR.2017.624.
[5] K. Simonyan dan A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," dalam *Proc. International Conference on Learning Representations*, 2015, hal. 1–14.
[6] X. Hou, L. Shen, K. Sun, dan G. Qiu, "Deep Feature Consistent Variational Autoencoder," dalam *Proc. IEEE Winter Conference on Applications of Computer Vision*, 2017, hal. 1133–1141, doi: 10.1109/WACV.2017.131.
[7] L. Ziwei, L. Ping, W. Xiaogang, dan T. Xiaoou, "Deep learning face attributes in the wild," dalam *Proc. IEEE International Conference on Computer Vision*, 2015, hal. 3730–3738, doi: 10.1109/ICCV.2015.425.
[8] T. Tanaka, N. Kawai, Y. Nakashima, T. Sato, dan N. Yokoya, "Iterative applications of image completion with CNN-based failure detection," *J. Vis. Commun. Image Represent.*, vol. 55, hal. 56–66, 2018, doi: 10.1016/j.jvcir.2018.05.015.
[9] I. J. Goodfellow *et al.*, "Generative Adversarial Nets," *Adv. Neural Inf. Process. Syst.*, vol. 27, hal. 4089–4099, 2014.
[10] X. Hou, K. Sun, L. Shen, dan G. Qiu, "Improving variational autoencoder with deep feature consistent and generative adversarial training," *Neurocomputing*, vol. 341, hal. 183–194, 2019, doi: 10.1016/j.neucom.2019.03.013.
[11] M. Arjovsky, S. Chintala, dan L. Bottou, "Wasserstein GAN," *arXiv*, vol. abs/1701.0, 2017, [Online]. Available: http://arxiv.org/abs/1701.07875.
[12] K. He, X. Zhang, S. Ren, dan J. Sun, "Deep residual learning for image recognition," dalam *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, hal. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
[13] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, dan A. A. Efros, "Context Encoders: Feature Learning by Inpainting," dalam *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, hal. 2536–2544, 2016, doi: 10.1109/CVPR.2016.278.