PENGGUNAAN AUTOENCODER DAN GENERATIVE ADVERSARIAL NETWORKS DALAM MENDETEKSI DEEPFAKE

[1] Kelvin Samuel [2] Chairisni Lubis [3] Agus Budi Dharmawan

[1] [2] [3] Teknik Informatika, FTI, Universitas Tarumanagara
Jl. Letjen S Parman no 1, Jakarta 11440 Indonesia
[1] kelvin.535200008@stu.untar.ac.id [2] chairisnil@fti.untar.ac.id [3] agusd@fti.untar.ac.id

ABSTRAK

Artikel ini dimaksudkan untuk membandingkan hasil deteksi *Deepfake* antara 2 metode *neural network* dan menemukan tingkat akurasi terbaik. Pada artikel ini arsitektur *neural network* yang akan digunakan adalah *Encoder, Decoder, Generator* dan *Diskriminator* yang masing – masing merupakan arsitektur dari masing – masing metode untuk melakukan deteksi..

Hasil perbandingan 2 metode *neural network* yang masing – masing berisi label *real* dan *deepfake* menunjukkan bahwa metode *Autoencoder* memiliki akurasi *training* tertinggi yaitu 99.74% dan akurasi *validation* tertinggi yaitu 73%, sedangkan untuk dan *Generative Adversarial Network* memiliki akurasi *training* tertinggi yaitu 96.82% dan akurasi *validation* tertinggi yaitu 84% dengan spesifikasi pelatihan menggunakan 50 epoch, 128 *batch size*, dengan 32.000 data. Berisi 16.000 data *real* dan 16.000 data *deepfake* namun dibagi lagi menjadi 5.120 untuk data *validation* dan 6.400 untuk data *testing*, namun dalam proses pengujiannya hanya menggunakan 10 data gambar acak yang menghasilkan proses yang dapat diprediksi oleh metode *Autoencoder* hingga 8 gambar diprediksi dengan benar sedangkan *Generative Adversarial Network* hanya dapat memprediksi hingga 7 gambar dengan benar.

Key words

Autoencoder, Generative Adversarial Network, Encoder, Decoder, Generator, Discriminator

1. Pendahuluan

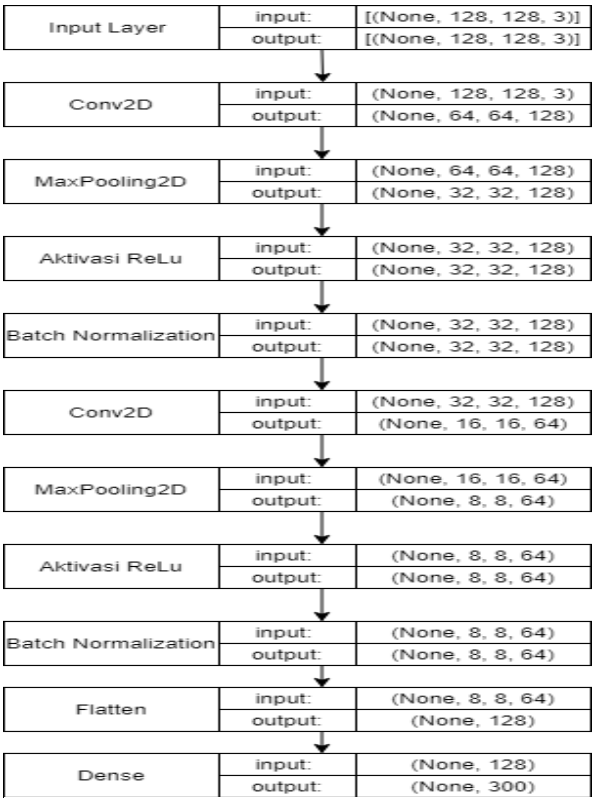
Deepfake merupakan teknologi yang digunakan untuk manipulasi gambar. Teknik manipulasi yang digunakan dalam *Deepfake* digunakan untuk merubah gambar baik dari bentuk wajah, tempat, objek atau bahkan suara.[1]. Penggunaan *Deepfake* yang tidak tepat juga dapat mengancam banyak hal seperti merusak reputasi, pencurian identitas dan juga penyebaran hoax dan kesalahpahaman di Masyarakat. Tujuan dari pembuatan sistem pendeteksi *Deepfake* pada skripsi ini adalah untuk membuat sistem yang efektif untuk mengidentifikasi apakah sebuah citra image tersebut *real* atau *fake* menggunakan metode *Autoencoder* dan *Generative Adversarial Networks* serta mengidentifikasi dari kedua metode tersebut mana yang lebih baik dalam segi akurasi untuk mendeteksi *Deepfake.* Sehingga diharapkan sistem ini dapat digunakan untuk mencocokan apakah citra image yang ingin diuji tersebut *real* atau *fake* ketika citra tersebut sudah dicurigai dan terindikasi *Deepfake.*

Pada perancangan penelitian ini dibuat dengan beberapa sumber penelitian sebelumnya yang menggunakan sebuah konsep yang sama dan dengan tujuan yang sama untuk mendeteksi *Deepfake,* sehingga diharapkan dapat memberikan hasil yang dapat membantu penelitian dalam bidang ini. Berikut ini beberapa penelitian – penelitian relevan yang pernah dibuat sebelumnya:

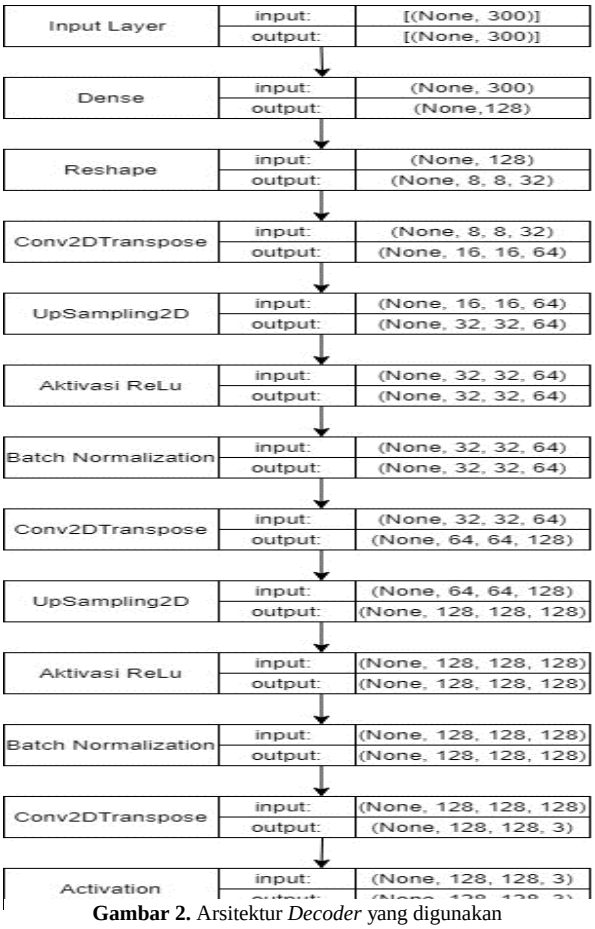
1. "OC-FakeDect: Classifying Deepfakes Using One-class Variational Autoencoder" oleh H. Khalid and S. Woo. [2]
2. "MRI-GAN: A Generalized Approach to Detect DeepFakes using Perceptual Image Assessment," oleh P. Prajapati and C. Pollett [3]
3. "A GAN-Based Model of *Deepfake* Detection In Social Media" oleh Preeti, Kumar M, Sharma M [4].

2. Metode

Pada sistem yang dirancang ini akan menggunakan *Autoencoder* dengan arsitektur *Encoder* dan *Decoder*, dan juga menggunakan *Generative Adversarial Network* dengan arsitektur *Generator* dan *Diskriminator*, Dalam perancangan ini menggunakan *input* berupa citra dan menghasilkan *output* berupa label. Karena pada perancangan sistem menggunakan 4 arsitektur maka untuk masing – masing model dari tiap arsitektur dapat dilihat pada gambar berikut.

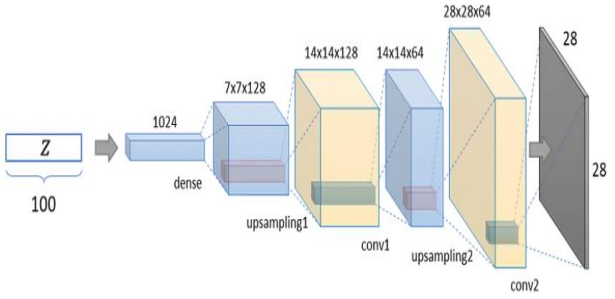


Gambar 1. Arsitektur *Encoder* yang digunakan

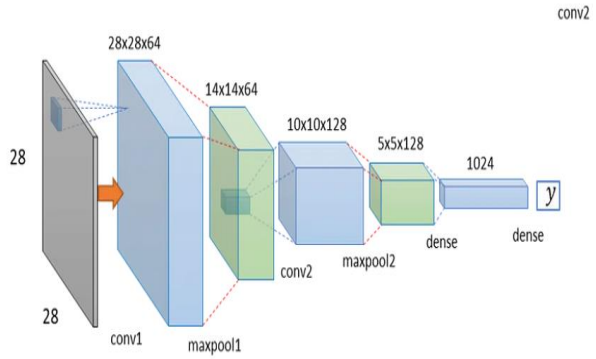


Gambar 2. Arsitektur *Decoder* yang digunakan

Penggunaan arsitektur *Encoder* dan *Decoder* pada *Autoencoder* pada rancangan ini adalah untuk merekonstruksi fitur pada sebuah gambar, yang bertujuan agar model pada *Autoencoder* mempelajari fitur - fitur penting pada sebuah gambar wajah seseorang dan kemudian merekonstruksi wajah tersebut. Serta mendeteksi sebuah anomaly atau kekurangan pada daerah tertentu dari sebuah citra.



Gambar 3. Arsitektur *Generator* yang digunakan



Gambar 4. Arsitektur *Diskriminator* yang digunakan

Penggunaan arsitektur *Generator* dan *Diskriminator* pada *Generative Adversarial Network* pada rancangan ini adalah untuk membuat citra yang serupa dengan citra asli nya, dan kemudian tugas *Diskriminator* berfungsi untuk mendeteksi antara citra image yang di generate *Generator* dan citra image asli manakah yang *Deepfake* sehingga model *Diskriminator* dapat terlatih untuk mendeteksi sebuah image citra wajah yang *real* dan *fake*

2.1. Dataset

Pada perancangan ini dataset yang digunakan adalah dataset yang berasal dari Kaggle yang berisikan sebuah kumpulan gambar wajah manusia *real* dan *deepfake.* Jumlah data secara keseluruhan adalah 95.634 gambar dengan jumlah gambar *deepfake* sebesar 83% lebih banyak secara keseluruhan dari jumlah total dataset dan, dataset *real* nya sebesar 17% dari keseluruhan total dataset. Namun data yang digunakan pada perancangan

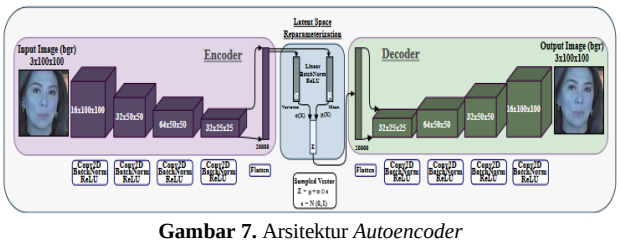
ini menggunakan 32.000 data yang di bagi menjadi 2 bagian yaitu *real* dan *fake* dengan masing – masing 16.000, untuk menghindari *error out of memory* yang akan terjadi jika model yang digunakan terlalu kompleks dan data yang digunakan terlalu banyak yang menyebabkan penggunaan RAM menjadi melebihi batas. Berikut ini adalah contoh dari data yang digunakan.

Gambar 5. Contoh gambar dataset asli

Gambar 6. Contoh gambar dataset *deepfake*

2.2. *Autoencoder*

Autoencoder pada *deepfake* adalah salah satu metode yang digunakan untuk mendeteksi *deepfake. Autoencoder* merupakan model *deep learning* yang digunakan untuk melakukan reduksi dimensi pada image dengan cara mempelajari representasi data image.serta digunakan untuk mempelajari fitur – fitur penting pada wajah seseorang dan kemudian merekonstruksi wajah tersebut dalam berbagai bentuk [5]. Pada metode ini terdapat beberapa arsitektur seperti *Encoder* dan *Decoder*

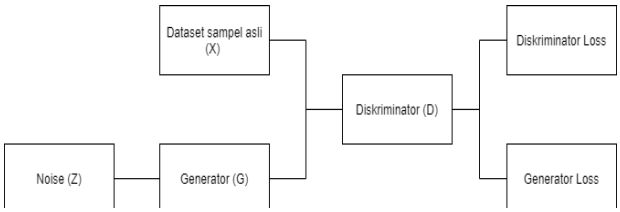


Gambar 7. Arsitektur *Autoencoder*

2.3. *Generative Adversarial Network*

Generative Adversarial Network atau disingkat dengan GAN adalah sebuah algoritma untuk pembuatan dan pendeteksi *deepfake* dengan menggabungkan dua jaringan saraf yang mampu menghasilkan gambar secara realistis dan mendeteksi suatu gambar apakah asli atau *deepfake.* Jaringan permusuhan *generative* menggabungkan dua jaringan saraf yang disebut *generator* dan *diskriminator*. *Generator* adalah jaringan saraf yang mencoba untuk menghasilkan gambar tiruan dari kumpulan data gambar yang diberikan dan mencoba untuk menghasilkan sebuah gambar nyata, dan *diskriminator* adalah jaringan saraf yang berfungsi untuk membedakan antara data citra asli dengan tiruan yang dihasilkan oleh *generator* [6]. Untuk proses cara kerja *Generative Adversarial Network* dapat dijabarkan sebagai berikut :

1. Data Sampel (X) akan masuk ke *diskriminator* untuk memberikan hasil berupa label – label klasifikasi.
2. Data palsu atau sampel acak (*noise*) masuk ke *generator*.
3. *Generator* akan menerima data palsu dan memberikan gambaran yang sangat mendekati data sampel untuk mengelabui *diskriminator.*
4. *Diskriminator* menerima data sampel asli (X) dan data palsu yang dihasilkan *generator* untuk menentukan *result*.
5. *Result* atau hasil akhir dari klasifikasi adalah nilai *real* atau *fake* dari prediksi *disrkiminator* terhadap data citra gambar.



Gambar 8. Arsitektur *Generative Adversarial Network*

3. Hasil Pengujian

Hasil pengujian ini menggunakan 10 data acak yang kemudian dicari nilai *Accuracy*, *Precision*, *Recall* dan *F1 Score.* Namun sebelum dilakukannya pengujian terlebih dahulu dilakukan uji coba untuk *training* dengan menggunakan 32.000 data gambar dengan 16.000 data *real* dan 16.000 gambar dengan data *fake.* Pada pengujian ini menggunakan 2 metode yaitu *Autoencoder* dan *Generative Adversarial Network* dengan menggunakan data sebanyak 20.480 untuk data *training* yang dibagi menjadi 2 class, data validasi sebanyak 5.120 yang dibagi menjadi 2 class dan data *testing* sebanyak 20 yang dibagi menjadi 2 class, serta menggunakan 50 *epoch*, 128 *batchsize.*

Berikut ini merupakan hasil dari rekonstruksi *Autoencoder* menggunakan arsitektur *Encoder* dan *Decoder*. Untuk gambar rekonstruksi hasil dari *Autoencoder* dapat dilihat pada gambar di bawah ini :

Gambar 9. Hasil rekonstruksi *Autoencoder*

Terdapat beberapa scenario *experiment* yang dilakukan menggunakan metode *Autoencoder* untuk mencari perbandingan jumlah data *testing* paling tepat seperti berikut ini :

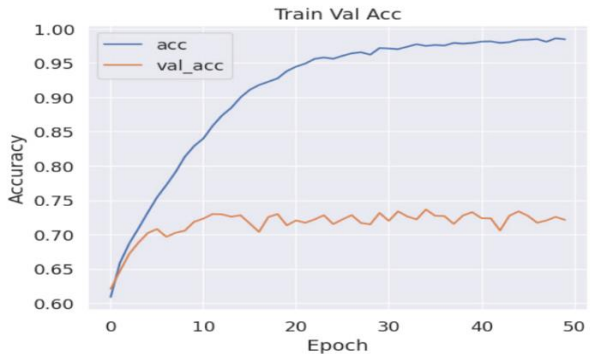
1. Dengan menggunakan data *training* sebanyak 20.480 data dan data validasi 5.120 dan *Epoch* 50 pada *Autoencoder*

Pada percobaan ini berhasil mendapatkan hasil seperti pada tabel berikut.

Tabel 1. Hasil akurasi metode Autoencoder *Epoch* 50

Jenis pengukuran *performance*	**Hasil**
Epoch	50
Accuracy	98,44%
Validation Accuracy	73%
Precision	0,72
Recall	0,72
F1 Score	0,72

Dengan grafik *accuracy* dan *loss* percobaan pertama



Gambar 10. Grafik *accuracy* dan *loss* percobaan *Epoch* 50

Serta pada percobaan ini mendapatkan hasil pengujian berupa 15 dari 20 gambar benar dan 5 gambar asli diprediksi sebagai *deepfake.*

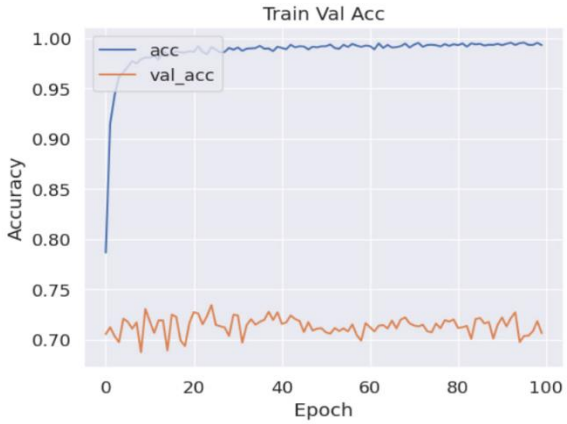
2. Dengan menggunakan data *training* sebanyak 20.480 data dan data validasi 5.120 dan *Epoch* 100 pada *Autoencoder*

Pada percobaan ini berhasil mendapatkan hasil seperti pada tabel berikut.

Tabel 2. Hasil akurasi metode Autoencoder *Epoch* 100

Jenis pengukuran *performance*	**Hasil**
Epoch	100
Accuracy	99,31%
Validation Accuracy	73%
Precision	0,72
Recall	0,71
F1 Score	0,71

Dengan grafik *accuracy* dan *loss* percobaan kedua



Gambar 11. Grafik *accuracy* dan *loss* percobaan *Epoch* 100

Serta pada percobaan ini mendapatkan hasil pengujian berupa 16 dari 20 gambar benar dan 4 gambar asli diprediksi sebagai *deepfake.*

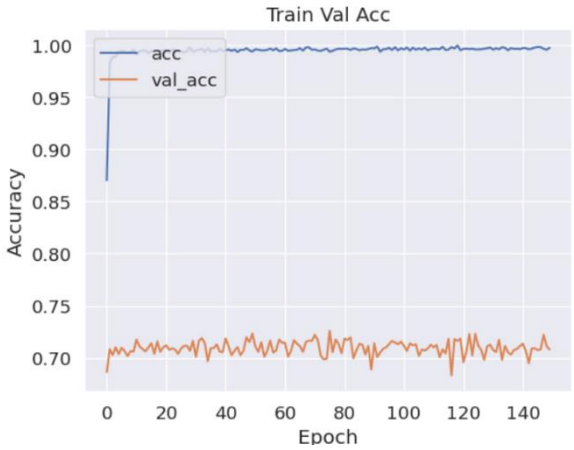
3. Dengan menggunakan data *training* sebanyak 20.480 data dan data validasi 5.120 dan *Epoch* 150 pada *Autoencoder*

Pada percobaan ini berhasil mendapatkan hasil seperti pada tabel

Tabel 3. Hasil akurasi metode Autoencoder *Epoch* 150

Jenis pengukuran *performance*	**Hasil**
Epoch	150
Accuracy	99,74%
Validation Accuracy	72%
Precision	0,72
Recall	0,71
F1 Score	0,71

Dengan grafik *accuracy* dan *loss* percobaan ketiga



Gambar 12. Grafik *accuracy* dan *loss* percobaan Epoch 150

Serta pada percobaan ini mendapatkan hasil pengujian berupa 17 dari 20 gambar benar dan 3 gambar asli diprediksi sebagai *deepfake.*

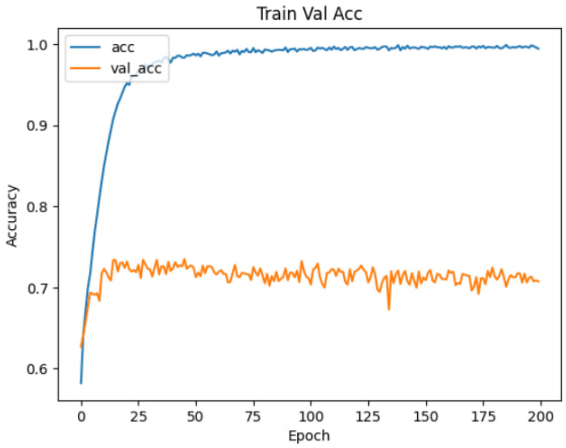
4. Dengan menggunakan data *training* sebanyak 20.480 data dan data validasi 5.120 dan *Epoch* 200 pada *Autoencoder*

Pada percobaan ini berhasil mendapatkan hasil seperti pada tabel berikut

Tabel 4. Hasil akurasi metode Autoencoder *Epoch* 200

Jenis pengukuran *performance*	**Hasil**
Epoch	200
Accuracy	99,42%
Validation Accuracy	73%
Precision	0,70
Recall	0,70
F1 Score	0,70

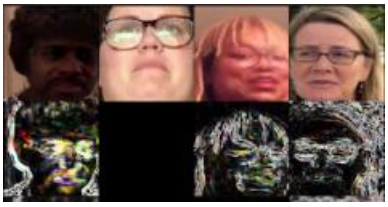
Dengan grafik *accuracy* dan *loss* percobaan keempat



Gambar 13. Grafik *accuracy* dan *loss* percobaan Epoch 200

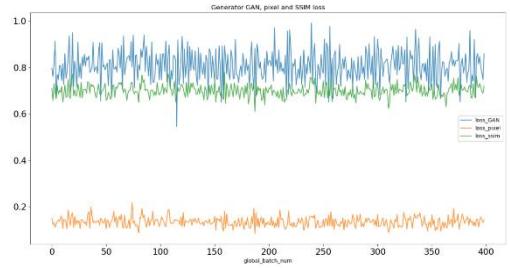
Serta pada percobaan ini mendapatkan hasil pengujian berupa 18 dari 20 gambar benar dan 2 gambar asli diprediksi sebagai *deepfake.*

Dalam pengujian metode *Generative Adversarial Network*, tahap pertama *Generator* mengenerate MRI image dari dataset citra dengan ciri jika image *real* maka image tersebut ketika di *generate* dalam bentuk MRI maka dia akan tampil image blank hitam, jika image *fake* maka image tersebut akan muncul beberapa *pattern* pada wajah yang menandakan bahwa image tersebut *fake.* Untuk contoh gambar image MRI dapat dilihat pada gambar berikut ini.

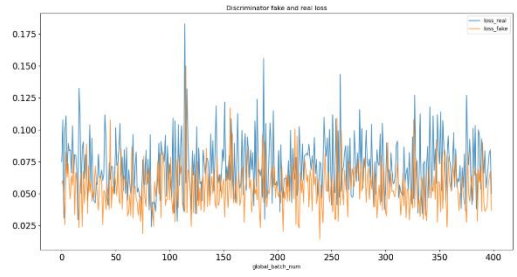


Gambar 14. *Generate* MRI Image

Dan untuk grafik *loss Generator* dan grafik *loss Diskriminator* dapat dilihat pada gambar berikut ini :



Gambar 15. *Generator Loss*



Gambar 16. *Discriminator Loss*

Terdapat beberapa scenario *experiment* yang dilakukan menggunakan metode *Generative Adversarial Network* untuk mencari perbandingan jumlah data *testing* paling tepat seperti berikut ini :

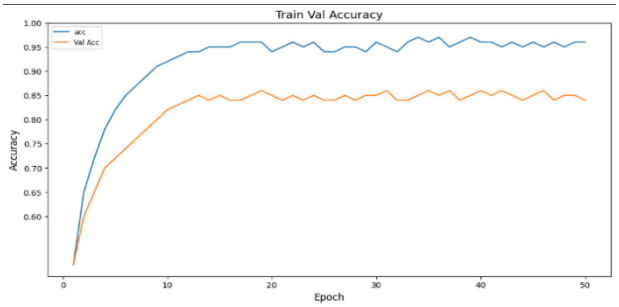
1. Dengan menggunakan data *training* sebanyak 20.480 data dan data validasi 5.120 dan *Epoch* 50 pada *Generative Adversarial Network*

Pada percobaan ini berhasil mendapatkan hasil seperti pada tabel berikut.

Tabel 5. Hasil akurasi metode Autoencoder *Epoch* 50

Jenis pengukuran *performance*	**Hasil**
Epoch	50
Accuracy	96,73%
Validation Accuracy	81%
Precision	0,79
Recall	0,79
F1 Score	0,79

Dengan grafik *accuracy* dan *loss* percobaan pertama



Gambar 17. Grafik *accuracy* dan *loss* percobaan *Epoch* 50

Serta pada percobaan ini mendapatkan hasil pengujian berupa 14 dari 20 gambar benar dan 6 gambar asli diprediksi sebagai *deepfake.*

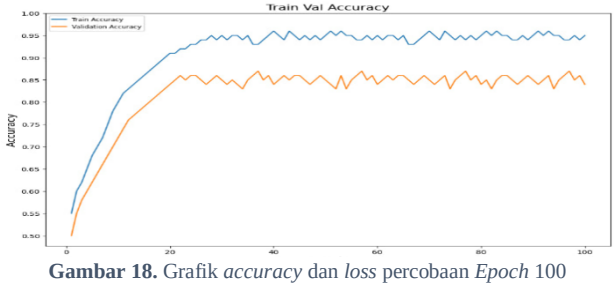
2. Dengan menggunakan data *training* sebanyak 20.480 data dan data validasi 5.120 dan *Epoch* 100 pada *Generative Adversarial Network*

Pada percobaan ini berhasil mendapatkan hasil seperti pada tabel berikut.

Tabel 6. Hasil akurasi metode Autoencoder *Epoch* 100

Jenis pengukuran *performance*	**Hasil**
Epoch	100
Accuracy	95,35%
Validation Accuracy	82%
Precision	0,77
Recall	0,77
F1 Score	0,77

Dengan grafik *accuracy* dan *loss* percobaan kedua



Gambar 18. Grafik *accuracy* dan *loss* percobaan *Epoch* 100

Serta pada percobaan ini mendapatkan hasil pengujian berupa 15 dari 20 gambar benar dan 5 gambar asli diprediksi sebagai *deepfake.*

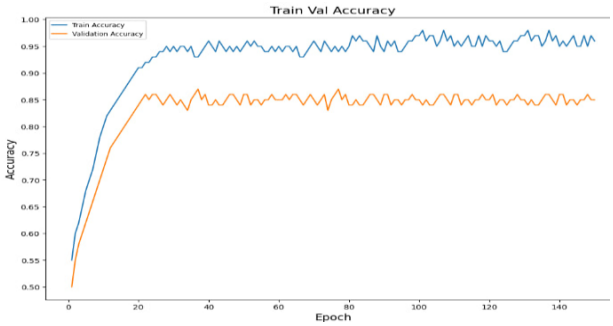
3. Dengan menggunakan data *training* sebanyak 20.480 data dan data validasi 5.120 dan *Epoch* 150 pada *Generative Adversarial Network*

Pada percobaan in berhasil mendapatkan hasil seperti pada tabel berikut.

Tabel 7. Hasil akurasi metode Autoencoder *Epoch* 150

Jenis pengukuran *performance*	**Hasil**
Epoch	150
Accuracy	95,41%
Validation Accuracy	83%
Precision	0,79
Recall	0,79
F1 Score	0,79

Dengan grafik *accuracy* dan *loss* percobaan ketiga



Gambar 19. Grafik *accuracy* dan *loss* percobaan *Epoch* 150

Serta pada percobaan ini mendapatkan hasil pengujian berupa 16 dari 20 gambar benar dan 4 gambar asli diprediksi sebagai *deepfake.*

4. Dengan menggunakan data *training* sebanyak 20.480 data dan data validasi 5.120 dan *Epoch* 200 pada *Generative Adversarial Network*

Pada percobaan in berhasil mendapatkan hasil seperti pada tabel berikut.

Tabel 8. Hasil akurasi metode Autoencoder *Epoch* 200

Jenis pengukuran *performance*	**Hasil**
Epoch	200
Accuracy	96,82%
Validation Accuracy	84%
Precision	0,82
Recall	0,82
F1 Score	0,82

Dengan grafik *accuracy* dan *loss* percobaan keempat

Gambar 20. Grafik *accuracy* dan *loss* percobaan *Epoch* 200

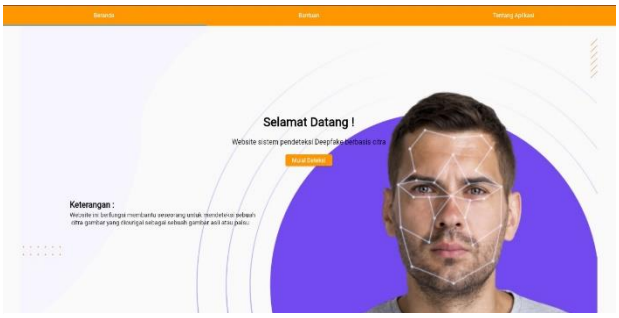
Serta pada percobaan ini mendapatkan hasil pengujian berupa 17 dari 20 gambar benar dan 3 gambar asli diprediksi sebagai *deepfake.*

Berdasarkan hasil percobaan diatas maka digunakannya perbandingan dimana epoch lebih banyak maka tingkat *accuracy* semakin bagus dan tingkat *accuracy validation* semakin mengecil, serta *layer, learning rate, batch size* sangat berpengaruh terhadap hasil *training* nanti. Dalam proses *training* yang akan digunakan oleh modelnya pada sistem ini menggunakan 32.000 data yang terdiri dari 16.000 data *real* dan 16.000 data *fake.* Yang kemudian dibagi lagi sebesar 80:20 di mana data *training* akan berjumlah 20.480 data dan data *validation* 5.120 dan akan pengujian menggunakan 20 gambar pada data validasi.

Program pendeteksi *deepfake* ini sendiri dibuat dengan bentuk *website* dengan menggunakan Flutter untuk *front-end* dan Python dengan FastApi untuk *back-end. Website* ini juga akan diuji untuk melihat apakah sebuah bagian pada *website* tersebut dapat berjalan dengan baik.

1. Modul Beranda

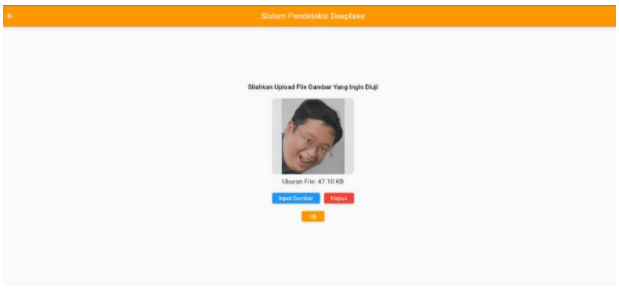
Pada pengujian ini dilakukan dengan melihat apakah semua tombol dapat digunakan dan apakah setiap tombol dapat masuk ke modul lain seperti modul uji, modul hasil, modul bantuan dan modul tentang aplikasi. Seperti pada gambar dibawah.



Gambar 21. Modul Beranda

2. Modul Uji

Pada pengujian ini dilakukan dengan melihat apakah semua tombol berfungsi dengan baik dan apakah gambar yang dimasukkan dapat di *preview* dan juga apakah tombol uji dapat memberikan respon API dan menghasilkan label yang akan ditampilkan pada modul ini. Seperti pada gambar di bawah.



Gambar 22. Modul Uji

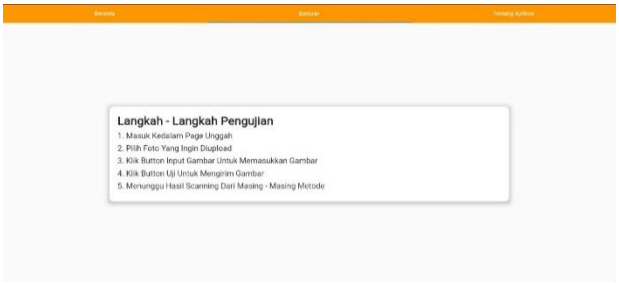
3. Modul Hasil

Pada pengujian ini dilakukan dengan melihat apakah semua tombol berfungsi dengan baik dan apakah hasil label dari proses pendeteksi *deepfake* menampilkan hasilnya dari *response* Api dan apakah button back nya berfungsi dengan baik untuk kembali ke modul beranda. Seperti pada gambar di bawah.

Gambar 23. Modul Hasil

4. Modul Bantuan

Pada pengujian ini dilakukan dengan melihat apakah semua tombol berfungsi dengan baik dan apakah tulisan petunjuk penggunaan aplikasi dapat muncul dan berfungsi dengan baik, Seperti pada gambar dibawah.



Gambar 24. Modul Bantuan

5. Modul Tentang Aplikasi

Pada pengujian ini dilakukan dengan melihat apakah semua tombol berfungsi dengan baik dan apakah tulisan informasi deskripsi aplikasi dan deskripsi tentang pembuat aplikasi dapat muncul dan berfungsi dengan baik, Seperti pada gambar dibawah.

Gambar 25. Modul Tentang Aplikasi

4. Kesimpulan

Berdasarkan hasil dari percobaan dan pengujian yang sudah dilakukan pada hasil sistem pendeteksi *deepfake* menggunakan metode *Autoencoder* dan *Generative Adversarial Network,* dapat disimpulkan bahwa :

1. Program yang dirancang ini masih sangat jauh dari kata sempurna sehingga pengembangannya masih akan sangat besar untuk di kembangkan di masa yang akan datang, Ketika *Autoencoder* dan *Generative Adversarial Network,* karena kedua metode tersebut sudah banyak digunakan terutama untuk mendeteksi *deepfake.*
2. Penggunaan perbandingan antara data *real* dan data *fake* harus sesuai perbandingannya 50:50, layer, *optimizer, learning rate, batch size* dan *epoch* sangat mempengaruhi hasil pengujian yang membuat program akan lebih condong memprediksi semua data ke 1 label saja.
3. Penggunaan *dropout* harus disesuaikan dengan model yang dibuat dan dilakukan pengujian jika sebuah model sudah kompleks dan hasil nya sudah menunjukkan gejala nya *overfitting* sehingga ditambahkan *dropout* untuk mengurangi *overfitting.*
4. Berdasarkan pada pengujian ini maka model *Autoencoder* menjadi model dengan tingkat *accuracy* yang lebih baik dibandingkan dengan metode *Generative Adversarial Network* karena dapat memprediksi 18 gambar dengan benar.

REFERENSI

[1] Angelika, R., & Santoso, H. 2023. "Analisis Gambar Wajah Palsu: Mendeteksi Keaslian Gambar Yang Dimanipulasi Menggunakan Metode Varitional Autoencoder Dan Forensics Deep Neural Network". *Sibatik Journal*
[2] H. Khalid and S. Woo Li, Shujun., Zheng, Xuan. "OC-FakeDect: Classifying Deepfakes Using One-class Variational Autoencoder"
[3] P. Prajapati and C. Pollett. "MRI-GAN: A Generalized Approach to Detect DeepFakes using Perceptual Image Assessm\ent,"
[4] Preeti, Kumar M, Sharma M. "A GAN-Based Model of *Deepfake* Detection In Social Media".
[5] P. Charitidis, G. Kordopatis-Zilos, S. Papadopoulos, and I. Kompatsiaris, "Investigating the Impact of Pre-processing and Prediction Aggregation on the DeepFake Detection Task," .
[6] Sio jurnalis pipin, "Deteksi Video Deepfake Menggunakan Spatiotemporal Convolutional Network dan Photo-Response Non-Uniformity - Mikroskil repository,"

Kelvin Samuel merupakan seorang mahasiswa Angkatan 2020 dan seorang mahasiswa semester akhir yang berasal dari Universitas Tarumanagara.

Chairisnis Lubis dra., M.Kom. memperoleh gelar Dra dari Universitas Indonesia. Kemudian memperoleh gelar M.Kom dari Universitas Indonesia. Saat ini sebagai Dosen Program studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara.

Agus Budi Dharmawan S.kom, M.T., M.sc. memperoleh gelar M.T dari ITS Surabaya. Kemudian memperoleh gelar M.Sc dari Electronic Engineering FH Darmstadt. Saat ini sebagai Dosen Program studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara.