PENERAPAN CONVOLUTIONAL NEURAL NETWORK DAN CAPSULE NETWORKS DALAM MENDETEKSI DEEPFAKE

Donni Suharyanto [1] Chairisni Lubis [2] Agus Budi Dharmawan [3]

[1) 2) 3)] Teknik Informatika, FTI, Universitas Tarumanaraga
Jl. Letjen S Parman no 1, Jakarta 11440 Indonesia
[1)] donni.535200042@stu.untar.ac.id [2)] chairisnil@fti.untar.ac.id [3)] agusd@fti.untar.ac.id

ABSTRAK

Pada tulisan ini mendemonstrasikan bagaimana perbandingan dalam pendeteksi *deepfake* antara tiga arsitektur *convolutional neural network* yaitu Resnet 50, VGG 19 dan Xception. Dalam percobaan ini, arsitektur-arsitektur tersebut akan dikombinasikan dengan model *capsule networks*, yang terdiri dari 2 lapisan konvolusi pada kapsul primer dan 2 kapsul keluaran yang akan diisi dengan label asli dan *deepfake*. Dibuat sebuah basis data yang terdiri dari 6000 data asli dan 4000 data *deepfake* dan dilatih melalui 100 *epochs* dan 10 *batch size*. Setelah itu, 20 gambar acak akan digunakan dalam proses pengujian pendeteksi *deepfake* berdasarkan label yang ada pada basis data. Hasilnya menunjukkan bahwa ketiga arsitektur tersebut mampu mendeteksi gambar *deepfake* dengan berbagai tingkat akurasi yaitu Resnet 50 dengan tingkat akurasi 64,133%, VGG 19 dengan tingkat akurasi 61,067% dan Xception dengan tingkat akurasi 64,067%.

Kata Kunci

Capsule Networks, Convolutional Neural Network, Resnet 50, VGG 19, Xception.

1. Pendahuluan

Deepfake adalah suatu Teknik manipulasi citra pada manusia yang menggunakan kecerdasan buatan untuk memodifikasi citra tersebut [1]. Deepfake pada awalnya memiliki tujuan untuk membantu dunia perfilman dimulai dari menciptakan sosok yang sudah tidak ada untuk kembali bermain pada sebuah film atau untuk mengubah wajah aktor untuk menyesuaikan perannya pada sebuah film. Namun kini *deepfake* sendiri sudah menjadi ancaman bagi Masyarakat dalam berinternet. Dalam hal ini, *deepfake* berpeluang menimbulkan banyak potensi yang dapat membahayakan pengguna internet lainnya seperti menyebabkan kesalahpahaman, pencemaran nama baik, pelecehan seksual secara digital, serta hal kriminal lainnya [2]. Oleh karena itu, diperlukan suatu sistem yang dapat membantu dalam melakukan pendeteksian *deepfake* pada media gambar.

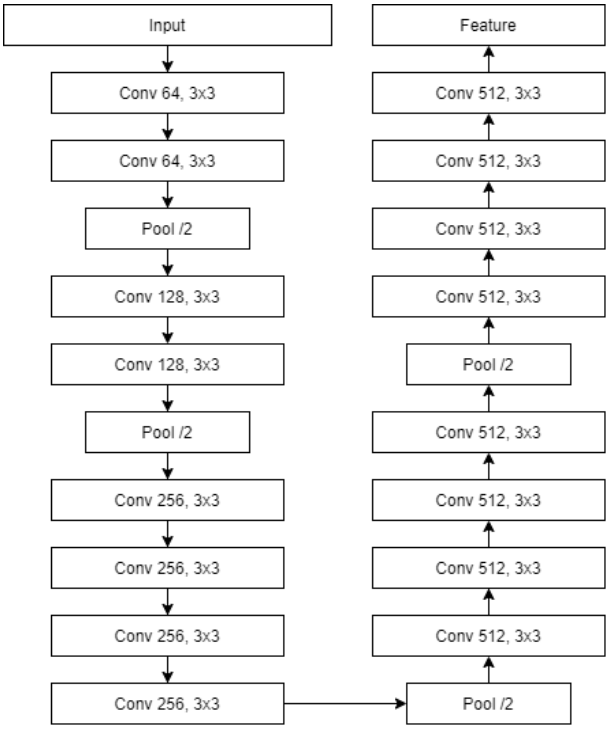
Beberapa penelitian telah dilakukan untuk mendapatkan sistem dengan arsitektur yang mampu membedakan citra asli dengan citra *deepfake*.

1. "USE OF A CAPSULE NETWORKS TO DETECTFAKE IMAGE AND VIDEOS" oleh Huy H. Nguyen, Junichi Yamagishi dan Isao Echizen [3]. Pada penelitian ini menggunakan *convolutional neural network* arsitektur VGG 19 dan c*apsule networks* dan mampu dengan baik memprediksi gambar *deepfake*. Namun membutuhkan data yang sangat banyak dan dengan ukuran yang lebih besar sebesar 300x300 px.
2. "Spasial Feature-based Fake Capsule Network Model for Deepfake Detection for Images and Video Data" oleh B. N. Karthik, Dr. P. Anbalagan dan Dr. G. Pradeep [4]. Pada penelitian ini menggunakan *fake capsule networks* dan dapat memprediksi gambar *deepfake* dengan baik.
3. "iCaps-Dfake: An Integrated Capsule-Based Model for Deepfake Image and Video Detection" oleh Samar Samir Khalil, Sherin M. Youssef dan Sherine Nagy Saleh [5]. Pada penelitian ini menggunakan iCaps-DFake yang digabungkan dengan *local binary pattern* (LBP) dan *convolutional neural network* dan dapat memprediksi gambar *deepfake* dengan baik.

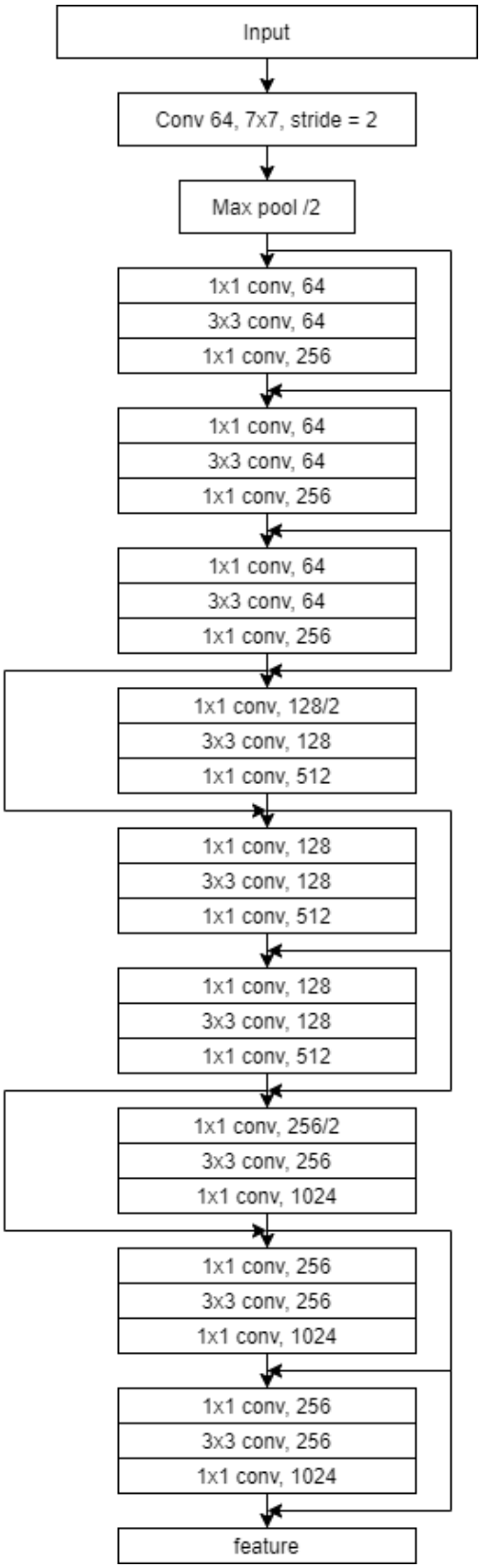
Pada perancangan penelitian ini bertujuan untuk mencari arsitektur *convolutional neural network* yang paling cocok jika digabungkan dengan model *capsule networks* jika digunakan untuk mendeteksi gambar asli dan *deepfake*. Ketiga arsitektur yang digunakan pada penelitian ini adalah VGG 19, Resnet 50 dan Xception. Pemilihan arsitektur ini didasarkan pada kemampuan ekstraksi fitur yang kuat pada VGG 19, kinerja efektif dalam klasifikasi gambar dan menangkap fitur tingkat rendah dan tinggi pada Resnet 50 dan ekstrasi fitur yang kuat serta representasi fitur yang abstrak sehingga mempermudah *capsule networks* untuk memahami pola dan hubungan yang lebih kompleks pada Xception.

2. Metode

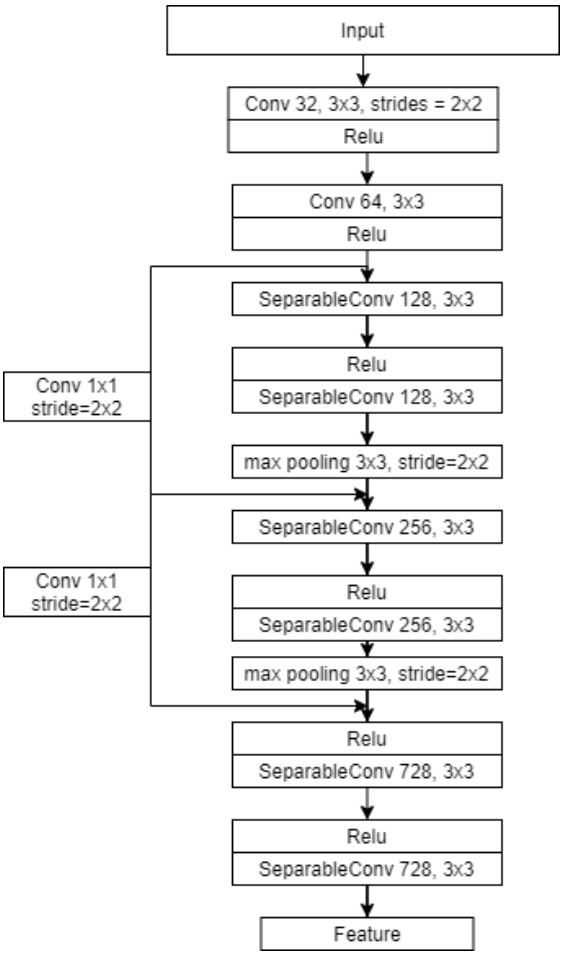
Pada sistem yang dirancang ini akan menggunakan *convolutional neural network* dengan arsitektur Resnet 50, VGG 19 dan Xception yang akan digabungkan dengan model *Capsule Networks* dengan *primary capsule* yang berisikan 2 lapisan konvolusi dan 2 *output capsule* berisikan label asli dan *deepfake* yang akan digunakan untuk mendeteksi apakah sebuah citra termasuk asli atau *Deepfake*. Dalam perancangan ini akan menggunakan *input* berupa citra dan menghasilkan *output* berupa label. Karena pada perancangan sistem menggunakan 3 arsitektur maka untuk masing-masing model dari tiap arsitektur dapat dilihat pada gambar 1, 2 dan 3.



Gambar 1. Arsitektur VGG 19 yang digunakan



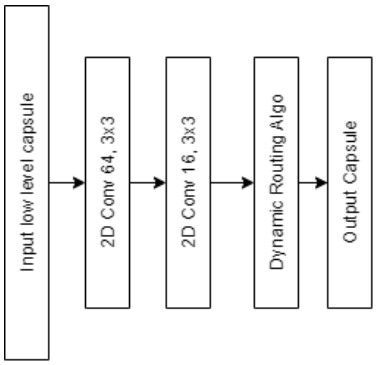
Gambar 2. Arsitektur Resnet 50 yang digunakan



Gambar 3. Arsitektur Xception yang digunakan

Penggunaan arsitektur *convolutional neural network* pada rancangan ini adalah untuk mengekstrak fitur pada suatu gambar dan pada penggunaannya tidak menggunakan semua lapisan pada arsitektur melainkan hanya menggunakan sebagian untuk mempertahankan ukuran fitur agar tidak terlalu kecil ketika di proses pada model *capsule networks*.

Pada model *capsule networks* sendiri akan tampak seperti gambar di bawah ini.

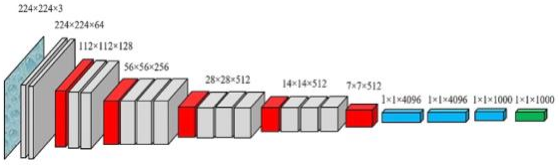


Gambar 4. Model *Capsule Networks* yang digunakan

Capsule networks di sini sendiri digunakan untuk melakukan prediksi sehingga dapat menghasilkan label berupa asli atau *deepfake*. Dalam proses untuk mencari labelnya sendiri terdapat sebuah algoritma yang disebut *dynamic routing algorithm* yang digunakan untuk menghitung *agreement between feature* pada fitur hasil dari ekstraksi *primary capsule* [3]. Sehingga dapat menghasilkan *agreement between feature* yang akan dipakai untuk mencari label yang sesuai dengan gambar tersebut.

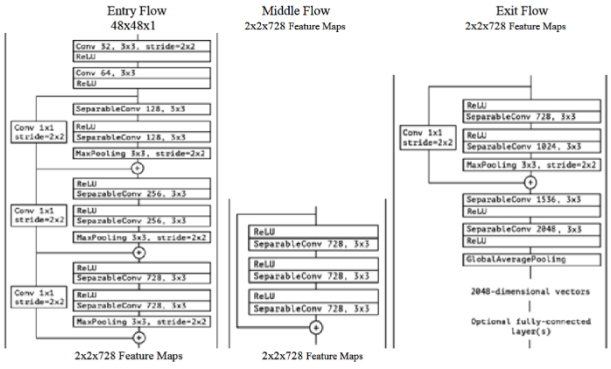
2.1. *Convolutional Neural Network*

Convolutional neural network atau yang sering disebut dengan CNN adalah salah satu metode *deep neural network* yang berfungsi untuk melakukan analisa pada suatu citra [6]. Metode ini sendiri sangat sering digunakan didalam deep learning khususnya dalam pendeteksi citra untuk mencari fitur-fitur dari sebuah citra gambar baik itu manusia, hewan, tanaman dan lainnya. pada metode ini sendiri pun terdapat beberapa arsitektur, termasuk arsitektur yang digunakan pada perancangan ini yaitu VGG 19, Resnet 50 dan Xception, setiap arsitektur tersebut memiliki pembedanya tersendiri seperti arsitektur VGG 19 yang memiliki 19 lapisan yang saling terhubung seperti pada gambar berikut.



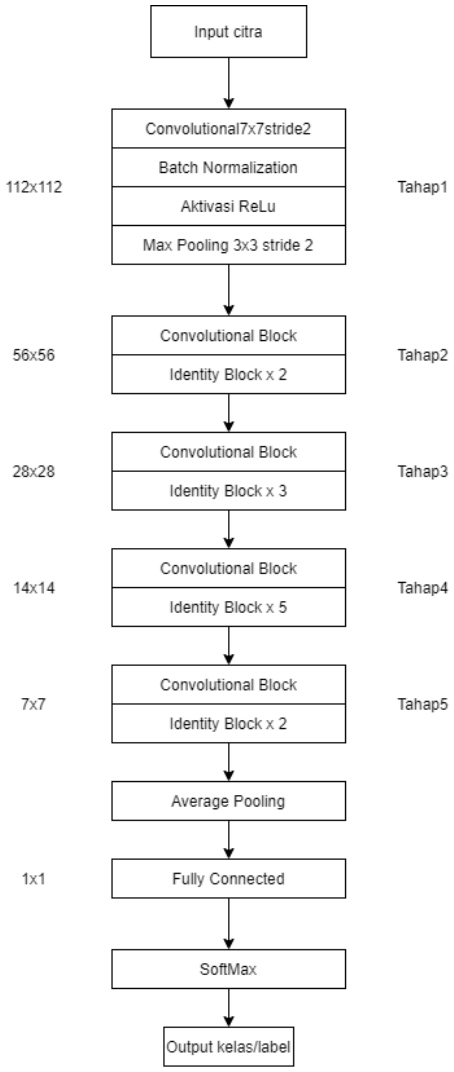
Gambar 5. Arsitektur VGG 19

Selain itu juga terdapat arsitektur Xception yang merupakan sebuah arsitektur dengan konsep *depthwise separable convolutional layers* yang membuat arsitektur ini tidak memiliki hubungan spasial peta fitur. Pada arsitektur ini sendiri ekstraksi fiturnya dilakukan dengan 38 lapisan konvolusi [7], serta untuk arsitektur Xception secara keseluruhan dapat dilihat pada gambar dibawah ini.



Gambar 6. Arsitektur Xception

Kemudian arsitektur Resnet 50 yang pada pembuatannya sendiri ditujukan untuk mengatasi kelemahan *convolutional neural network* pada umumnya terutama pada permasalahan lamanya proses latih dan kekurangan lapisan [8]. Kemudian untuk arsitekturnya secara keseluruhan sendiri dapat dilihat seperti pada gambar dibawah ini.



Gambar 7. Arsitektur Resnet 50

2.2. *Capsule Networks*

Capsule networks merupakan sebuah model jaringan yang digunakan untuk mengambil informasi dari fitur penting pada sebuah gambar. Pada *capsule networks* ini terdiri dari 2 *capsule* yaitu *primary capsule* dan *output capsule*. Pada *primary capsule* pada perancangan ini menggunakan 2 lapisan *convolution* dan pada *output capsule* menggunakan 2 *capsule* yang akan menampung label asli dan *deepfake*. Yang kemudian antara *primary capsule* dan *output capsule* akan dihubungkan menggunakan *dynamic routing algorithm*.

2.3 Dataset

Pada perancangan ini dataset yang digunakan adalah dataset yang berasal dari Kaggle yang berisikan Kumpulan gambar wajah manusia asli dan *deepfake.* Jumlah data secara keseluruhan adalah 95634 gambar dengan jumlah gambar *deepfake* sebanyak 83% dari keseluruhan data dan jumlah gambar real sebanyak 17% dari keseluruhan data. Namun data yang digunakan pada perancangan ini akan menggunakan 6000 data asli dan 4000 data *deepfake* untuk menghindari *error out of memory* yang akan terjadi jika model yang digunakan terlalu dalam dan data yang digunakan terlalu banyak yang menyebabkan penggunaan RAM menjadi melebihi batas. Berikut ini adalah contoh dari data yang digunakan.

Gambar 8. Contoh gambar asli

Gambar 9. Contoh gambar deepfake

3. Hasil Pengujian

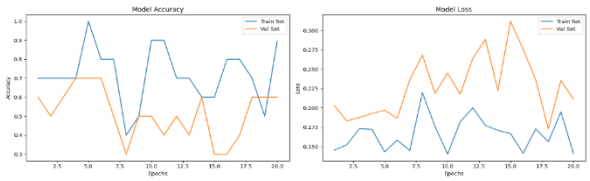
Hasil pengujian ini menggunakan 20 data acak yang kemudian dicari nilai *accuracy, Precision, Recall dan F1 Score*. Namun sebelum dilakukannya pengujian terlebih dahulu dilakukan uji coba untuk *train* dengan menggunakan 2000 data gambar dengan 1000 data asli dan 1000 data *deepfake*. Pada pengujian ini menggunakan arsitektur Resnet 50 dengan 200 data validasi dengan 100 data asli dan 100 data *deepfake* sedangkan untuk pengujian realnya menggunakan 20 data dengan 10 data asli dan 10 data *deepfake* serta menggunakan 100 *epoch*, 10 *batch size* dan *convolutional layer* yang digunakan berjumlah 2 dengan ukuran 64 dan 16 *filter* dengan *kernel* 3x3. Terdapat beberapa *experiment* yang dilakukan untuk mencari perbandingan jumlah data latih paling tepat seperti sebagai berikut.

1. Dengan menggunakan 1000 data asli dan 1000 data *deepfake*.
   Pada percobaan ini berhasil mendapatkan hasil seperti pada tabel berikut.

Tabel 1. Hasil percobaan pertama

Jenis pengukuran *performance*	**Hasil**
Accuracy	53%
Precision	0,5625
Recall	0,9
F1 Score	0,6923

Dengan grafik accuracy dan loss sebagai berikut.



Gambar 10. Grafik *accuracy* dan *loss* percobaan pertama

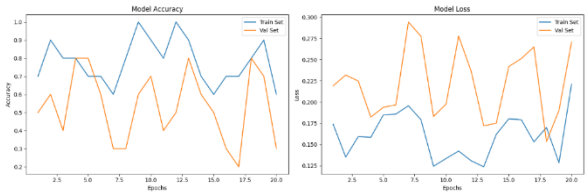
Serta pada percobaan ini mendapatkan hasil pengujian berupa 12 dari 20 gambar benar dan 7 gambar asli di prediksi sebagai *deepfake* dan 1 gambar *deepfake* diprediksi sebagai asli.

2. Dengan menggunakan 1000 data asli dan 800 data *deepfake*.
   Pada percobaan ini berhasil mendapatkan hasil seperti pada tabel berikut.

Tabel 2. Hasil percobaan kedua

Jenis pengukuran *performance*	**Hasil**
Accuracy	55,5%
Precision	0,6363
Recall	0,7
F1 Score	0,6667

Dengan grafik *accuracy* dan *loss* sebagai berikut.



Gambar 11. Grafik *accuracy* dan *loss* percobaan kedua

Serta pada percobaan ini mendapatkan hasil pengujian berupa 13 dari 20 gambar benar dan 4 gambar asli di prediksi sebagai *deepfake* dan 3 gambar *deepfake* diprediksi sebagai asli.

3. Dengan menggunakan 800 data asli dan 1000 data *deepfake*.
   Pada percobaan ini berhasil mendapatkan hasil seperti pada tabel berikut.

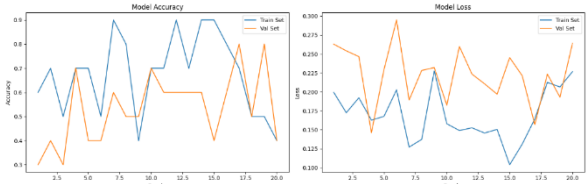
Tabel 3. Hasil percobaan ketiga

Jenis pengukuran *performance*	**Hasil**
Accuracy	56.5%

Tabel 4. Hasil percobaan ketiga (lanjutan)

Jenis pengukuran *performance*	**Hasil**
Precision	0,5334
Recall	0,8
F1 Score	0,64

Dengan grafik *accuracy* dan *loss* sebagai berikut.



Gambar 12. Grafik *accuracy* dan *loss* percobaan ketiga

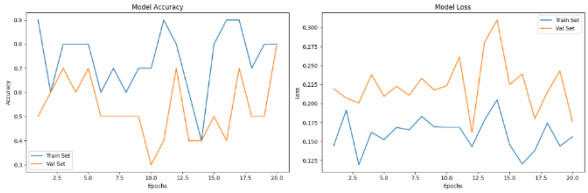
Serta pada percobaan ini mendapatkan hasil pengujian berupa 11 dari 20 gambar benar dan 7 gambar asli di prediksi sebagai *deepfake* dan 2 gambar *deepfake* diprediksi sebagai asli.

4. Dengan menggunakan 1000 data asli dan 800 data *deepfake* dan *dropout rate* 0,3.
   Pada percobaan ini berhasil mendapatkan hasil seperti pada tabel berikut.

Tabel 5. Hasil percobaan keempat

Jenis pengukuran *performance*	**Hasil**
Accuracy	56%
Precision	0,75
Recall	0,3
F1 Score	0,4285

Dengan grafik *accuracy* dan *loss* sebagai berikut.



Gambar 13. Grafik *accuracy* dan *loss* percobaan keempat

Serta pada percobaan ini mendapatkan hasil pengujian berupa 12 dari 20 gambar benar dan 1 gambar asli di prediksi sebagai *deepfake* dan 7 gambar *deepfake* diprediksi sebagai asli.

5. Dengan menggunakan 800 data asli dan 1000 data *deepfake* dan *dropout rate* 0,3.
   Pada percobaan ini berhasil mendapatkan hasil seperti pada tabel berikut.

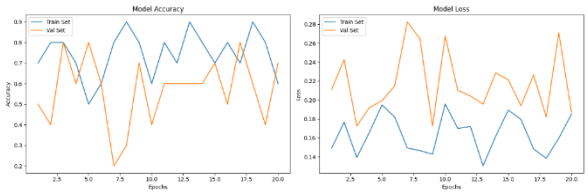
Tabel 6. Hasil Percobaan kelima

Jenis pengukuran *performance*	**Hasil**
Accuracy	54,5%

Tabel 7. Hasil Percobaan kelima (lanjutan)

Jenis pengukuran *performance*	**Hasil**
Precision	0,5714
Recall	0,8
F1 Score	0,6667

Dengan grafik *accuracy* dan *loss* sebagai berikut.



Gambar 14. Grafik *accuracy* dan *loss* percobaan kelima

Serta pada percobaan ini mendapatkan hasil pengujian berupa 12 dari 20 gambar benar dan 6 gambar asli di prediksi sebagai *deepfake* dan 2 gambar *deepfake* diprediksi sebagai asli.

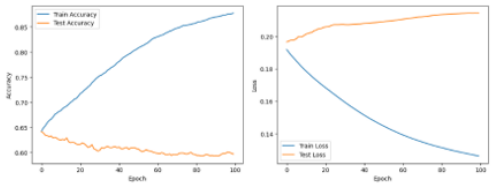
Berdasarkan hasil percobaan di atas maka digunakannya perbandingan di mana data asli lebih banyak dari pada data *deepfake*. Dalam proses *train* yang akan digunakan modelnya pada sistem ini menggunakan 10000 data yang terdiri dari 6000 data asli dan 4000 data *deepfake*. Yang kemudian dibagi lagi sebesar 7:3 di mana data *train* akan berjumlah 7000 dan data validasi berjumlah 3000 dan akan pengujian menggunakan 20 gambar yang ada pada data validasi. Berdasarkan jumlah data *train* dan data validasi di atas maka dilakukanlah proses *train* setiap arsitektur yang memberikan hasil sebagai berikut.

1. Resnet 50
   Pada arsitektur Resnet 50 memberikan hasil sebagai berikut.

Tabel 8. Hasil *train* Resnet 50

Jenis pengukuran *performance*	**Hasil**
Accuracy	64,133%
Precision	1,0
Recall	0,5714
F1 Score	0,7272

Dengan grafik *accuracy* dan *loss* sebagai berikut.



Gambar 15. Grafik *accuracy* dan *loss* Resnet 50

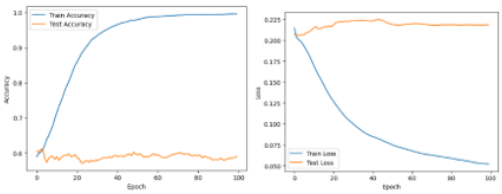
Pada arsitektur ini berhasil memprediksi 16 dari 20 gambar dengan benar.

2. VGG 19
   Pada arsitektur VGG 19 memberikan hasil sebagai berikut.

Tabel 9. Hasil *train* VGG 19

Jenis pengukuran *performance*	**Hasil**
Accuracy	61,067%
Precision	0,6
Recall	0,4285
F1 Score	0,5

Dengan grafik *accuracy* dan *loss* sebagai berikut.



Gambar 16. Grafik *accuracy* dan *loss* VGG 19

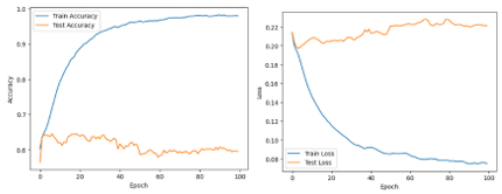
Pada arsitektur ini berhasil memprediksi 14 dari 20 gambar dengan benar.

3. Xception
   Pada arsitektur Xception memberikan hasil sebagai berikut.

Tabel 10. Hasil *train* Xception

Jenis pengukuran *performance*	**Hasil**
Accuracy	64,067%
Precision	0,5
Recall	0,5714
F1 Score	0,5334

Dengan grafik *accuracy* dan *loss* sebagai berikut.



Gambar 17. Grafik *accuracy* dan *loss* Xception

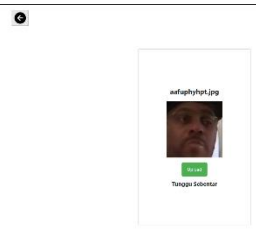
Pada arsitektur ini berhasil memprediksi 13 dari 20 gambar dengan benar.

Program pendeteksi *deepfake* ini sendiri dibuat dalam bentuk *website* dengan menggunakan Html, CSS dan JavaScript 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 *Home*
   Pada pengujian ini dilakukan dengan melihat apakah semua tombol dapat digunakan dan apakah setiap tombol dapat masuk ke dalam modul lain seperti modul *about*, prediksi dan *history*. Seperti pada gambar di bawah.

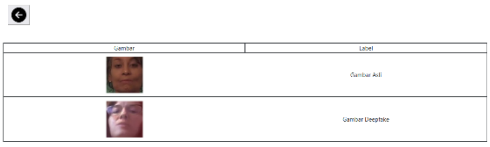
Gambar 18. Modul *Home*

2. Modul Prediksi
   Pada pengujian ini dilakukan dengan melihat apakah semua tombol berfungsi dengan baik dan apakah gambar yang dimasukkan dapat di *preview* serta apakah dengan menekan tombol *upload* dapat memberikan respon API dari *back-end* dan menghasilkan label yang akan ditampilkan pada modul ini. Seperti pada gambar di bawah.



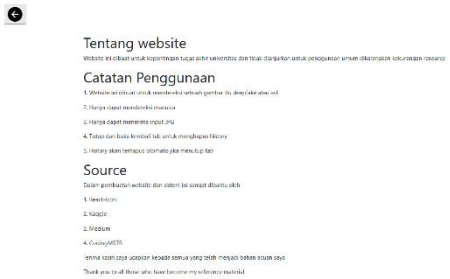
Gambar 19. Modul Prediksi

3. Modul *History*
   Pada pengujian ini dilakukan dengan melihat apakah semua tombol dapat berfungsi dengan baik dan apakah gambar yang sudah diproses pada modul prediksi dapat muncul pada *history*, seperti pada gambar di bawah.



Gambar 20. Modul *History*

4. Modul *About*
   Pada pengujian ini dilakukan dengan melihat apakah semua tombol dapat berfungsi dengan baik dan apakah konten tulisan pada modul ini dapat dilihat, seperti pada gambar di bawah.



Gambar 21. Modul *About*

4. Kesimpulan

Sistem ini melakukan percobaan dan pengujian pada program pendeteksi *deepfake* menggunakan arsitektur Resnet 50, VGG 19 dan Xception yang digabungkan dengan *Capsule Networks.* Dari hasil pengujian dapat disimpulkan bahwa Program ini hanya mampu mendapatkan akurasi 64,133% untuk akurasi tertingginya pada arsitektur Resnet 50 sedangkan 61,067% untuk arsitektur VGG 19 dan 64,067% untuk arsitektur Xception. Penggunaan perbandingan antara data asli dan *deepfake* harus dicari melalui *experiment* sesuai dengan model yang dirancang, karena penggunaan data secara berlebihan pada satu sisi dapat mempengaruhi hasil pengujian yang membuat program akan condong memprediksi semua data ke 1 label. Penggunaan *dropout* harus disesuaikan dengan model yang dibuat dan dilakukan pengujian apakah *dropout* efektif dalam meningkatkan akurasi sebelum digunakan. Berdasarkan pada pengujian ini maka arsitektur Resnet 50 menjadi arsitektur dengan tingkat akurasi pengujian tertinggi dengan dapat memprediksi 16 gambar dengan benar. Oleh karena itu, pengembangannya di masa mendatang akan sangat besar, terutama ketika *capsule networks* sudah banyak digunakan terutama dapat pendeteksi wajah manusia

REFERENSI

[1] R. A. S. Rahayu dan S. Handri, "ANALISIS GAMBAR WAJAH PALSU: MENDETEKSI KEASLIAN GAMBAR YANG DIMANIPULASI MENGGUNAKAN METODE VARIATIONAL AUTOENCODER DAN FORENSICS DEEP NEURAL NETWORK," *Sibatik Journal,* pp. 2701-2726, 2023.

[2] M. Napizahni, "Mengenal Deepfake dan Bahayanya yang MengintaiMengenal Deepfake dan Bahayanya yang Mengintai," Dewaweb, 20 Mei 2022. [Online]. Available: https://www.dewaweb.com/blog/apa-itu-deepfake/. [Diakses 8 November 2023].

[3] S. Singh, CAPSULE NETWORK TO DETECT FAKE IMAGES AND VIDEOS, Cham: Springer Nature, 2022.

[4] B. N. Karthik, D. p. Anbalangan dan D. G. Pradeep, "Spatial Feature-based Fake Capsule Network Model for Deep fake," *Mathematical Statistician and*

Engineering Applications, vol. 71, no. 4, pp. 1481-1489, 2022.

[5] S. S. Khalil, S. M. Youssef dan S. N. Saleh, "iCaps-Dfake: An Integrated Capsule-Based Model for," *future internet*, vol. 13, no. 4, pp. 1-19, 2021.

[6] P. A. Nugroho, I. Fenriana dan R. Arijanto, "IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA EKSPRESI MANUSIA," *ALGOR*, vol. 2, no. 1, pp. 12-21, 2020.

[7] N. Jaymon, S. Nagdeote, A. Yadav dan R. Rodrigues, "Real Time Emotion Detection Using Deep Learning," *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, pp. 1-7, 2021.

[8] B. M. Sujatmiko, E. Yudaningtyas dan P. M. Raharjo, "CONVOLUTION NEURAL NETWORK DENGAN DESAIN JARINGAN RESNET SEBAGAI METODE KLASIFIKASI TUMOR KULIT," *Jurnal SimanteC*, vol. 11, no. 1, pp. 53-64, 2022.

Donni Suharyanto, merupakan seorang mahasiswa Program Studi Teknik Informatika, Fakultas Teknologi Informasi Universitas Tarumanagara Angkatan 2020.

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 Electrical Engenering FH Darmstadt. Saat ini sebagai Dosen Program studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara.