Implementasi Algoritma *Convolutional Neural Network* Arsitektur Model *MobileNetV2* dalam Klasifikasi Penyakit Tumor Otak *Glioma, Pituitary dan Meningioma*

Anti Nada Nafisa[1], Erika Nia Devina Br Purba[2], Fahri Aulia Alfarisi Harahap[3]], Nurul Adawiyah Putri[4]
[1,2,3,4]Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Medan
Jl. Williem Iskandar/Pasar V, Medan, Sumatera Utara
Email: antinada7@gmail.com, niapurbaa@gmail.com, fahriaulialfarisiharahap07@gmail.com, nuruladawiyahp@gmail.com

Abstract

Tumor otak merupakan penyakit yang ditandai dengan pertumbuhan sel yang tidak normal pada jaringan otak. Salah satu cara yang dapat dilakukan dokter dalam pendeteksian tumor otak yaitu pengamatan langsung dengan diagnosis secara manual yang memiliki resiko terjadinya kesalahan. Beberapa jenis tumor otak antara lain Glioma, Pituitary dan Meningioma. 3 jenis tumor otak ini jika dilihat dari citranya ketiganya hampir mirip. Tetapi para ahli radiologi dan juga dokter spesialis bedah berhasil menemukan bahwa ada perbedaan antara citra Glioma, Pituitary dan Meningioma. Penelitian ini menggunakan algoritma deep learning yaitu Convolutional Neural Network (CNN) yang ditambahkan dengan model arsitektur model MobileNetV2 untuk melakukan klasifikasi penyakit tumor otak. Data citra yang digunakan adalah data citra penyakit otak yang diperoleh dari Kaggle dengan total 186 data citra yang terbagi atas 3 kategori, yaitu penyakit otak Glioma, Meningioma, dan Pituitary. Dari hasil pengujian dengan data testing didapat nilai evaluasi akurasi sebesar 0.7833 atau 78% dan hasil pengujian dengan data validasi didapat nilai evaluasi akurasi sebesar 0.83 atau 83% dalam melakukan klasifikasi penyakit tumor otak Glioma, Pituitary dan Meningioma.

Keywords: *Klasifikasi, Convolutional Neural Network, Tumor Otak, MobileNetV2*

1. PENDAHULUAN

Penyakit tumor otak adalah pertumbuhan sel otak yang abnormal di dalam atau di sekitar otak secara tidak wajar dan tidak terkendali [1]. Beberapa jenis tumor otak antara lain *Glioma, Pituitary dan Meningioma.* 3 jenis tumor otak ini jika dilihat dari citranya ketiganya hampir mirip. Tetapi para ahli radiologi dan juga dokter spesialis bedah berhasil menemukan bahwa ada perbedaan antara citra *Glioma, Pituitary dan Meningioma*.

Kasus tumor otak di dunia semakin meningkat setiap tahunnya. Di Indonesia, terhitung ada 300 pasien setiap tahunnya yang terdiagnosis tumor otak. Bukan hanya orang dewasa, tetapi tumor otak juga menyerang anak-anak dengan usia yang tergolong muda. Banyak orang mengabaikan gejala yang disebabkan oleh tumor otak [2].

Cara yang biasa digunakan dokter dalam mengklasifikasi jenis tumor otak yang diderita pasien adalah dengan biopsi dan pengamatan langsung. Biopsi membutuhkan waktu yang lama sekitar 10 - 15 hari untuk uji laboratorium, sedangkan pengamatan langsung oleh dokter beresiko terjadi kesalahan. Oleh sebab itu deep learning dengan metode *Convolutional Neural Network* (CNN) menjadi salah satu solusi yang dapat membantu seorang dokter dalam mengklasifikasikan dan mendiagnosa jenis tumor otak yang diderita pasien dan menghasilkan tingkat kesalahan yang rendah [1].

Convolutional Neural Network (CNN) merupakan metode *deep learning* yang populer terhadap pengenalan pola citra. CNN sangat baik dalam mengekstraksi fitur yang kompleks secara otomatis dan efisien untuk klasifikasi citra dengan skala yang besar. Metode *deep learning* dapat dengan jelas dalam membedakan citra dengan karakteristik serupa yang sulit dikenali oleh metode tradisional machine learning. Bahkan deep learning dapat mengekstraksi fitur secara objektif dengan sendirinya dan dapat langsung memproses data gambar dalam dua dimensi [2].

MobileNet merupakan salah satu dari sekian banyak arsitektur CNN yang dapat digunakan pada aplikasi mobile. Beberapa kelebihan dari arsitektur CNN ini yaitu ketebalan dari filter konvolusi yang sesuai dengan gambar. Sehingga bisa menghemat ukuran dari model yang dibuat [2]. Pada penelitian ini digunakan arsitektur CNN *MobileNetV2*.

Penelitian mengenai klasifikasi tumor otak dengan judul "Klasifikasi Tumor Otak Menggunakan

Convolutional Neural Network Dengan Arsitektur *EfficientNet-B3*" sudah dilakukan oleh Andre R, dkk [2] Pada penelitian ini menggunakan metode *Convolutional Neural Network* untuk melakukan klasifikasi citra tumor otak menggunakan deep learning. Dari hasil pengujian menghasilkan akurasi mencapai 99.7% dan mendapatkan nilai F1-Score tertinggi mencapai 99.6%.

Penelitian lain dilakukan oleh Hanin, dkk [3] menggunakan *Convolutional Neural Network* (CNN) dalam mengklasifikasi penyakit kulit, pada penelitian ini dikembangkan sistem indentifikasi penyakit kulit yang dapat mengklasifikasikan kondisi cacar air, campak, skabies, dan jerawat menggunakan metode *Convolutional Neural Network* (CNN) dan memperoleh nilai akurasi sebesar 96,53%.

Berdasarkan uraian latar belakang diatas, maka penelitian ini bertujuan untuk mengklasifikasi tumor otak menggunakan algoritma *Convolutional Neural Network* (CNN) dengan arsitektur model *MobileNetV2.*

2. TINJAUAN PUSTAKA

2.1. Tumor Otak

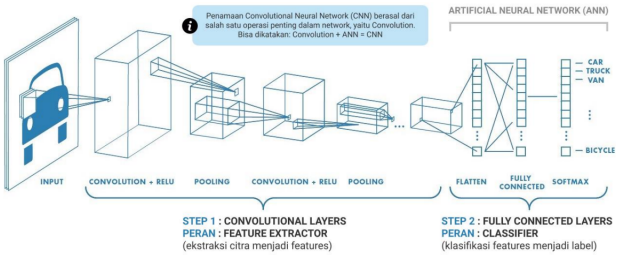
Tumor otak adalah pertumbuhan sel otak yang abnormal di dalam atau di sekitar otak secara tidak wajar dan tidak terkendali [1]. Beberapa jenis tumor otak antara lain *Glioma, Pituitary dan Meningioma*. 3 jenis tumor otak ini jika dilihat dari citranya ketiganya hampir mirip. Tetapi para ahli radiologi dan juga dokter spesialis bedah berhasil menemukan bahwa ada perbedaan antara citra *Glioma, Pituitary dan Meningioma.*

Glioma adalah jenis tumor otak orang dewasa yang paling umum, terhitung 78% dari tumor otak ganas. Tumor tersebut muncul dari sel pendukung otak, yang disebut *glia*. Sel-sel ini dibagi lagi menjadi *astrosit*, sel *ependymal*, dan sel *oligodendroglial* (atau *oligo*). *Meningiomas* adalah tumor intrakranial jinak yang paling umum, terdiri dari 10 hingga 15% dari semua neoplasma otak, meskipun sebagian kecil merupakan tumor ganas. Tumor ini berasal dari *meninges*, yaitu struktur mirip membran yang mengelilingi otak dan sumsum tulang belakang. *Pituitary* adalah tumor intrakranial yang paling umum setelah *glioma, meningioma, dan schwannoma.* Sebagian besar pituitary adenoma merupakan tumor jinak dan tumbuh cukup lambat. Bahkan tumor ganas *pituitary* jarang menyebar ke bagian tubuh yang lain. Adenoma sejauh ini merupakan penyakit paling umum yang menyerang jaringan *pituitary.* Tumor tersebut biasanya menyerang orang-orang berusia 30-an atau 40-an bahkan orang dewasa. Sebagian besar tumor ini dapat diobati sampai hilang [4].

2.2. *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) adalah salah satu metode klasifikasi yang termasuk ke dalam kelompok *deep learning* yang menggunakan layer konvolusi untuk mengonvolusi suatu input dengan filter. CNN memiliki kemampuan untuk mempelajari fitur secara *unsupervised* yang membedakan metode ini dengan metode machine learning lainnya yang membutuhkan fitur yang sebelumnya harus ditentukan [5].

Secara umum tahapan klasifikasi citra di CNN dibagi menjadi dua bagian besar yaitu *feature extractor* dan *classification/fully connected* (ANN). Dimana tahap *feature extractor* berperan melakukan ekstraksi dari sebuah citra (*image*) menjadi sebuah features berupa angka-angka yang merepresentasikan citra tersebut, atau dengan kata lain input berupa citra dan *output* berupa *features*. Selanjutnya *features* yang dihasilkan dari tahap *feature extractor* ini masih berbentuk *array multidimensi*, sehingga sebelum masuk sebagai input ke tahap *classification/fully connected* (ANN) untuk dilakukan klasifikasi, perlu di-*flatten* terlebih dahulu yaitu mengubah bentuk *array multidimensi* tersebut kedalam sebuah *vector* (array satu dimensi) [6].



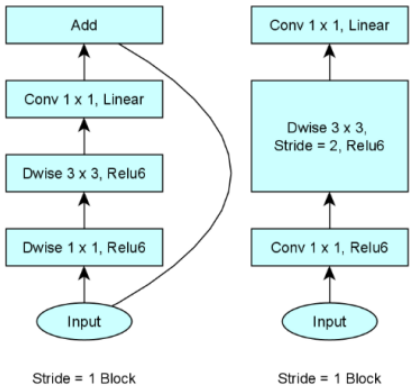
Gambar 1. Tahapan klasifikasi citra di CNN (Arsitektur CNN)

2.3. *Transfer Learning*

Transfer learning adalah metode yang bekerja dengan memanfaatkan arsitektur *network* yang telah ada. *Transfer learning* melakukan modifikasi dan mengupdate parameter-parameter pada *network* tersebut. *Transfer learning* menjadikan *network* yang telah termodifikasi sebagai pembelajaran dengan tugas berbeda. Arsitektur CNN yang digunakan untuk transfer learning telah melakukan pembelajaran terhadapt data-data lain, sehingga tidak diperlukan pembelajaran dari awal. Arsitektur *network* telah mengenali fitur-fitur berupa tekstur, bentuk dan warna sebagai hasil dari pembelajaran yang telah dilakukan sebelumnya [7].

2.4. *MobileNetV2*

MobileNetV2 yang dirilis pada tahun 2018 masih menggunakan *depth wise* dan *point wise konvolusi* seperti pada versi sebelumnya, namun *MobileNetV2* menambahkan dua fitur baru yaitu *linear bottleneck* dan *shortcut connections antar layer*. Struktur dasar dari arsitektur *MobileNetV2* dapat di lihat pada Gambar 2 [8].



Gambar 2. Arsitektur *MobileNetV2*

2.5. Optimasi *Adaptive Moment Estimation* (Adam) *Learning*

*Adaptive Moment Estimation (*Adam) adalah salah satu jenis fungsi optimasi yang merupakan n kombinasi dari RMSprop dan momentum. Adam adalah hasil penurunan metode SGD yang didasarkan pada estimasi adaptif momen orde pertama dan kedua. algoritma optimasi yang menghitung tingkat pembelajaran secara adaptif untuk setiap parameter. Adam menyimpan rata-rata gradien proses sebelumnya secara eksponensial sama seperti RMSprop. Nilai learning rate standar pada Adam adalah 0,001. Rumus perhitungan optimasi Adam ditunjukkan pada persamaan (1).

$$\theta_{t+1} = \theta_t - \frac{\partial}{\sqrt{\hat{v}_t + \varepsilon}} . \hat{m}_t \quad (1)$$

dengan θ_{t+1} adalah parameter hasil pembaruan, θ_t adalah parameter hasil pembaruan sebelumnya, η adalah *learning rate*, m_t adalah gradien kuadrat momen orde pertama, v_t gradien kuadrat momen orde kedua, dan ε merupakan scalar kecil untuk mencegah pembagian dengan nol. Persamaan (1) menunjukkan perhitungan optimasi Adam dalam memperbarui nilai error dalam proses pelatihan dengan memanfaatkan nilai gradien pada momen orde pertama dan orde kedua [9].

2.6. Normalisasi Data

Normalisasi adalah proses penskalaan citra, yaitu dengan mengurangi resolusi citra yang berguna saat proses pengenalan citra dan juga meningkatkan akurasi pengenalan. Proses yang digunakan pada tahap normalisasi ini adalah proses penskalaan citra [10].

2.7. *One Hot Encoding*

One Hot Encoding adalah cara merepresentasikan data yang digunakan agar dapat dimenegerti oleh komputer. Cara kerja *One Hot* adalah dengan membuat sebuah tensor atau array 1 dimensi dengan panjang sebanyak jenis class yang ada dan mempunyai isi biner antara 0 atau 1. Ilustrasi *One Hot Encoding* dapat dilihat pada Gambar 3 [11].

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

→

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

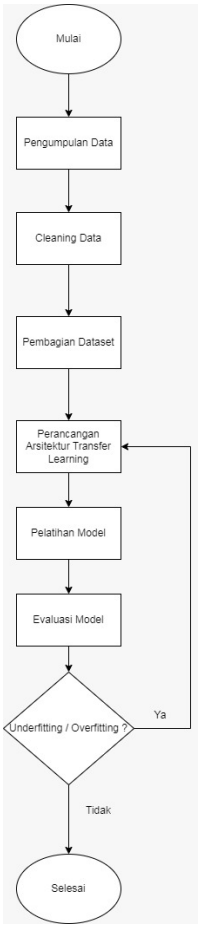
Gambar 3. Ilustasi *One Hot Encoding*

2.8. *Konversi Color Space RGB*

Metode color space merupakan salah satu metode untuk membantu pengelompokkan warna disamping berbagai besaran warna yang banyak diamati seperti RGB, YCbCr, HIS maupun CMYK. Teknik color space telah banyak dilakukan para periset untuk melakukan pengolahan citra berwarna dan mereka mencoba untuk menguraikan menjadi beberapa elemen bagian-bagian penyusun sebuah obyek atau benda. Bermacam metode segmentasi dalam beberapa rangkaian penelitian untuk memperoleh hasil maksimal telah diupayakan dan hal ini merupakan salah satu bagian proses tugas yang paling sulit dalam pengolahan citra digital. Prosedur segmentasi yang dilakukan secara sembarangan lebih sering membawa proses jauh dari sebuah solusi yang diinginkan dan berhasil terhadap fokus masalah gambar yang memerlukan objek yang akan diidentifikasi secara individual[12].

3. METODE PENELITIAN

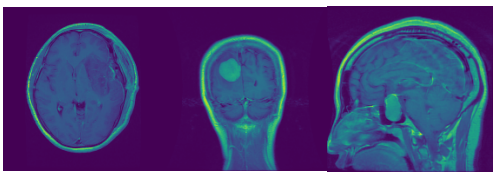
Metodologi penelitian pada penelitian ini terdiri dari pengumpulan data, *cleaning* data, pembagian dataset, perancangan arsitektur transfer learning, pelatihan model dan evaluasi model. Alur metodologi dapat dilihat pada Gambar 4.



Gambar 4. Alur Metodologi

3.1. Pengumpulan Data

Gambar yang digunakan pada penelitian ini adalah data citra yang bersifat publik. Data citra yang digunakan adalah data citra penyakit otak yang diperoleh dari Kaggle https://www.kaggle.com/datasets/denizkavi1/brain-tumor?select=1 dengan total 186 data citra yang terbagi atas 3 kategori, yaitu penyakit otak *Giloma, Meningioma,* dan *Pituitary*. Persebaran data pada tiap kategori adalah 62 data citra untuk kategori *Giloma*, 62 data citra untuk kategori *Meningioma* dan 62 data citra untuk kategori *Pituitary*. Gambar data citra kategori penyakit *Giloma, Meningioma,* dan *Pituitary* dapat ditunjukkan pada Gambar 2.



(a) (b) (c)

Gambar 5. Gambar Penyakit Otak
a) Giloma, b) Meningioma, c) Pituitary

3.2. *Cleaning Data*

Persebaran data yang tidak merata pada setiap kategori dapat menimbulkan terjadinya kesalahan dalam melakukan klasifikasi objek. Hal tersebut dapat diatasi dengan mengurangi data yang berlebih hingga setiap kategori memiliki jumlah yang sama atau yang under sampling [13]. *Cleaning* data citra dilakukan dengan cara me*resize* semua ukuran data citra ke dalam ukuran 128 x 128 piksel untuk menjalankan *pre-trained Transfer Learning* Arsitektur *MobileNetV2*. Pada arsitektur ini, diperlukan inputan data citra optimal dengan ukuran 128 x 128 piksel, setelah itu semua data citra dilakukan konversi warna *color space* untuk mengubah mode warna dari BGR ke RGB

3.3. Pembagian Dataset

Dalam membuat model deep learning diperlukan data untuk melatih model tersebut. Pertimbangan pertama dalam pembangunan dataset adalah untuk menentukan ukuran untuk setiap set [13]. Dalam pembuatan model *deep learning Convolutional Neural Network*, yang harus diperhatikan adalah pembagian dataset yang akan digunakan. Pada penelitian ini, dataset akan dibagi dalam 3 kelas, yaitu *data train*, data *test*, dan data validasi.

TABEL I. TABEL PEMBAGIAN DATA

	Data Train	Data Test	Data Validasi
Giloma	32	20	10
Meningioma	32	20	10
Pituitary	32	20	10

3.4. Perancangan Arsitektur *Transfer Learning*

Penelitian ini menggunakan model Arsitektur *Transfer Learning MobileNetV2*, dimana *MobileNetV2* adalah salah satu arsitektur *Convolutional Neural Network* (CNN) berbasis ponsel yang dapat digunakan untuk mengatasi kebutuhan akan computing resource berlebih. *MobileNetV2* merupakan penyempuranaan dari arsistektur *MobileNet*. Aristektur *MobileNet* dan arsitektur CNN pada umumnya memiliki perbedaan pada penggunaan lapisan atau *convolution layer* [14].

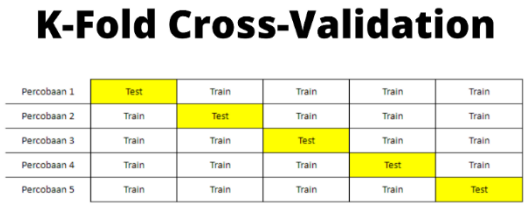
Convolution layer pada *MobileNetV2* menggunakan ketebalan filter yang sesuai dengan ketebalan dari *input image*. *MobileNetV2* menggunakan *depthwise convolution, pointwise convolution, linear bottleneck* dan *shortcut connections antar bottlenecks* [15]. Diagram Arsitektur *MobileNetV2* dapat dilihat pada Gambar 6.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Gambar 6. Diagram Arsitektur *Transfer Learning Pretrained MobileNetV2*

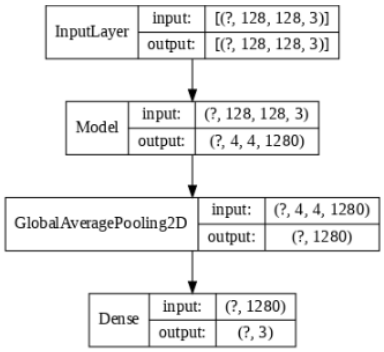
3.5. Pelatihan Model

Pada penelitian ini, proses pelatihan data menggunakan metode K-Fold Cross Validation. *K fold cross validation* digunakan untuk mengestimasi kesalahan prediksi dalam mengevaluasi kinerja model. Data dibagi menjadi himpunan bagian k berjumlah hampir sama. Model dalam klasifikasi dilatih dan diuji sebanyak k. Disetiap pengulangan, salah satu himpunan bagian akan digunakan sebagai data training dan data *testing*. Tujuan dari *K-Fold Cross Validation* adalah untuk mendefenisikan dataset untuk "menguji" model dalam tahap pelatihan (validasi data) dalam rangka membatasi masalah seperti terjadinya overfitting, memberikan wawasan tentang bagaimana model akan menggeneralisasi independent dataset [16]. Visualisasi data *K-Fold Cross Validation* dapat dilihat pada Gambar 7.



Gambar 7. Visualisasi *K-Fold Cross Validation*

Penelitian yang dilakukan disini menggunakan *K-Fold Validation* dengan nilai fold sebanyak 4 dan *batch size* yang akan digunakan adalah 32. Pada penelitian ini model yang akan digunakan dalam pelatihan data dapat dilihat pada gambar 8.



Gambar 8. Arsitektur Model Yang Digunakan Dalam Pelatihan Data

3.6. Perancangan Arsitektur *Transfer Learning*

Dalam penelitian ini, evaluasi model akan dilakukan dengan mempertimbangkan beberapa hal, seperti: nilai akurasi, nilai *precision*, nilai *recall*, *F-1 Score*, dan juga *confusion matrix*. *Confusion matrix* adalah alat evaluasi visual yang digunakan dalam *machine learning*. Kolom pada *confusion matrix* mewakili hasil kelas prediksi dan baris mewakili hasil kelas sebenarnya, sehingga dapat menghitung semua kemungkinan kasus masalah klasifikasi [17]. Akurasi adalah rasio prediksi benar terhadap keseluruhan data. *Precision* adalah perbandingan nilai prediksi benar positif dibandingkan dengan total hasil dengan prediksi positif. *Recall* adalah perbandingan nilai prediksi benar posotif dengan seluruh data yang benar positif. *F-1 score* merupakan perbandingan rata-rata nilai presisi dan *recall* yang dibebankan. Akurasi, *Precision*, *Recall* dan *F-1 Score* dapat dilihat pada persamaan berikut [13]:

$$Akurasi = \frac{(True\ Positive + True\ Negatif)}{(True\ Positive + True\ Negative + +True\ Positive + True\ Negative\ )}$$

$$= \frac{(True\ Positive + True\ Negatif)}{(2\ x\ (True\ Positive + True\ Negative\ ))} \quad (2)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (3)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (4)$$

$$F1\ Score = \frac{\left(2\ x\ \frac{True\ Positive}{True\ Positif + False\ Negative}\ x\ \frac{True\ Positive}{True\ Positive + False\ Posirive}\right)}{\left(\frac{True\ Positive}{True\ Positive + False\ Negative} + \frac{True\ Positive}{True\ Positive + False\ Posirive}\right)}$$

$$= \frac{2\ x\ Recall\ x\ Precision}{Recall + Precision} \quad (5)$$

4. HASIL DAN PEMBAHASAN

Pada penelitian ini, digunakan metode *k-fold cross validation* sebagai metode dalam melakukan latih data, digunakan 4 nilai *fold* dengan masing-masing *epoch* tiap fold adalah 30 *epoch* dan ukuran *batch size* yang digunakan adalah 32. Pengujian dilakukan dengan

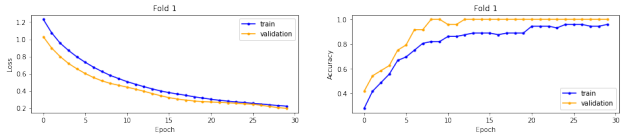
menggunakan dataset penyakit otak sebagai *input* ke dalam arsitektur *MobileNetV2*.

Pada Fold Ke-1 ditampilkan hasil latih pada tabel berikut ini :

TABEL II. Hasil Latih *Fold* Ke-1

Iterasi	loss	accuracy	val_loss	val_accuracy
1	1.23232305049896	0.277777791023254	1.03027725219726	0.416666656732559
2	1.07534623146057	0.416666656732559	0.898437440395355	0.541666686534881
3	0.956758201122283	0.486111104488372	0.800231717710968	0.583333313465118
4	0.871022403240203	0.555555582046508	0.719477713108062	0.625
5	0.79670113325119	0.666666686534881	0.657289206981658	0.75
6	0.732640326023101	0.694444417953491	0.602085947990417	0.791666686534881
7	0.675692975521087	0.75	0.554965555667877	0.916666686534881
8	0.624960124492645	0.805555582046508	0.516579329967498	0.916666686534881
9	0.579456686973571	0.819444417953491	0.487001687288284	1
10	0.542797803878784	0.819444417953491	0.46514555811882	1
11	0.507085084915161	0.861111104488372	0.443331629037857	0.958333313465118
12	0.477264612913131	0.861111104488372	0.420025825500488	0.958333313465118
13	0.449195653200149	0.875	0.395354121923446	1
14	0.423007816076278	0.888888895511627	0.368582457304	1
15	0.399037510156631	0.888888895511627	0.343409448862075	1
16	0.379870176315307	0.888888895511627	0.321086436510086	1
17	0.36321410536766	0.875	0.304841369390487	1
18	0.348116129636764	0.888888895511627	0.292471140623092	1
19	0.330941647291183	0.888888895511627	0.281673520803451	1
20	0.316401690244674	0.888888895511627	0.275407999753952	1
21	0.302527427673339	0.944444417953491	0.270485550165176	1
22	0.290309309959411	0.944444417953491	0.266042947769165	1
23	0.280838817358016	0.944444417953491	0.261144548654556	1
24	0.271317154169082	0.930555582046508	0.255905896425247	1
25	0.263074159622192	0.958333313465118	0.249269172549247	1
26	0.255690693855285	0.958333313465118	0.244403406977653	1
27	0.246936455368995	0.958333313465118	0.232238098978996	1
28	0.237897098064422	0.944444417953491	0.218559190630912	1
29	0.229804813861846	0.944444417953491	0.205246582627296	1
30	0.222347557544708	0.958333313465118	0.196752369403839	1

Plot pada *Fold* Ke-1 dapat dilihat pada gambar berikut :



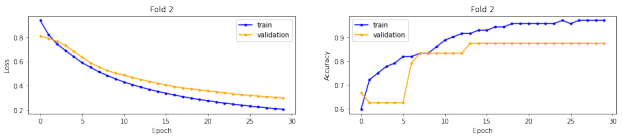
Gambar 9. Plot *Fold* Ke-1

Pada *Fold* Ke-2 ditampilkan hasil latih pada tabel berikut ini :

TABEL III. Hasil Latih *Fold* Ke-2

Iterasi	loss	accuracy	val_loss	val_accuracy
1	0.93932557106018	0.597222208976745	0.809324741363525	0.666666686534881
2	0.820712447166442	0.722222208976745	0.791143357753753	0.625
3	0.744698703289031	0.75	0.767561733722686	0.625
4	0.690609097480773	0.777777791023254	0.733533799648284	0.625
5	0.639750897884368	0.791666686534881	0.684288084506988	0.625
6	0.58995121717453	0.819444417953491	0.635332524776458	0.625
7	0.552065432071685	0.819444417953491	0.589196383953094	0.791666686534881
8	0.514636754989624	0.833333313465118	0.555041015148162	0.833333313465118
9	0.48484194278717	0.833333313465118	0.526075780391693	0.833333313465118
10	0.456053107976913	0.861111104488372	0.503770768642425	0.833333313465118
11	0.431110739707946	0.888888895511627	0.486634641885757	0.833333313465118
12	0.409059166908264	0.902777791023254	0.469315141439437	0.833333313465118
13	0.388272583484649	0.916666686534881	0.451447874307632	0.833333313465118
14	0.370192736387252	0.916666686534881	0.435544610023498	0.875
15	0.353159368038177	0.930555582046508	0.420433729887008	0.875
16	0.338263422250747	0.930555582046508	0.405872583389282	0.875
17	0.323977261781692	0.944444417953491	0.393174678087234	0.875
18	0.310084670782089	0.944444417953491	0.382162898778915	0.875
19	0.297720521688461	0.958333313465118	0.373701900243759	0.875
20	0.28616651892662	0.958333313465118	0.366103410720825	0.875
21	0.276808619499206	0.958333313465118	0.358221143484115	0.875
22	0.266056358814239	0.958333313465118	0.34986326098442	0.875
23	0.256761670112609	0.958333313465118	0.341685742139816	0.875
24	0.248317182064056	0.958333313465118	0.333688497543334	0.875
25	0.240006864070892	0.972222208976745	0.326424151659011	0.875
26	0.232780009508132	0.958333313465118	0.32045304775238	0.875
27	0.225010558962821	0.972222208976745	0.315195173025131	0.875
28	0.21856252849102	0.972222208976745	0.309683173894882	0.875
29	0.211575672030448	0.972222208976745	0.304661691188812	0.875
30	0.205538645386695	0.972222208976745	0.30072471499443	0.875

Plot pada *Fold* Ke-2 dapat dilihat pada gambar berikut :



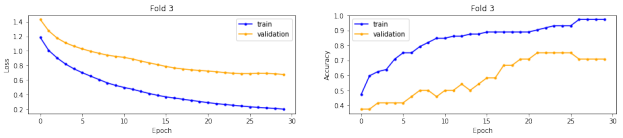
Gambar 10. Plot *Fold* Ke-2

Pada Fold Ke-3 ditampilkan hasil latih pada tabel berikut ini :

TABEL IV. Hasil Latih *Fold* Ke-3

Iterasi	loss	accuracy	val_loss	val_accuracy
1	1.183565	0.472222	1.429921	0.375
2	1.006546	0.597222	1.274725	0.375
3	0.903953	0.625	1.176166	0.416667
4	0.820264	0.638889	1.11096	0.416667
5	0.755162	0.708333	1.064877	0.416667
6	0.70191	0.75	1.02738	0.416667
7	0.653829	0.75	0.995176	0.458333
8	0.607434	0.791667	0.967011	0.5
9	0.561684	0.819444	0.941804	0.5
10	0.526393	0.847222	0.92435	0.458333
11	0.498867	0.847222	0.910807	0.5
12	0.474429	0.861111	0.889101	0.5
13	0.44442	0.861111	0.860976	0.541667
14	0.416084	0.875	0.835774	0.5
15	0.390837	0.875	0.81159	0.541667
16	0.369989	0.888889	0.78658	0.583333
17	0.354114	0.888889	0.766256	0.583333
18	0.337716	0.888889	0.751859	0.666667
19	0.322037	0.888889	0.74038	0.666667
20	0.306462	0.888889	0.731689	0.708333
21	0.291266	0.888889	0.723661	0.708333
22	0.277598	0.902778	0.714548	0.75
23	0.266292	0.916667	0.702371	0.75
24	0.255199	0.930556	0.692795	0.75
25	0.244589	0.930556	0.689194	0.75
26	0.233863	0.930556	0.689804	0.75
27	0.22599	0.972222	0.692368	0.708333
28	0.218805	0.972222	0.691894	0.708333
29	0.211177	0.972222	0.686859	0.708333
30	0.20237	0.972222	0.676432	0.708333

Plot pada *Fold* Ke-3 dapat dilihat pada gambar berikut :



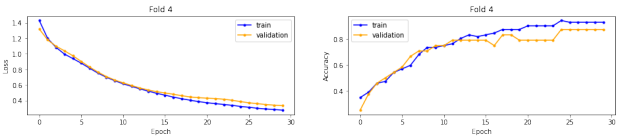
Gambar 11. Plot *Fold* Ke-3

Pada *Fold* Ke-4 ditampilkan hasil latih pada tabel berikut ini :

TABEL V. Hasil Latih *Fold* Ke-4

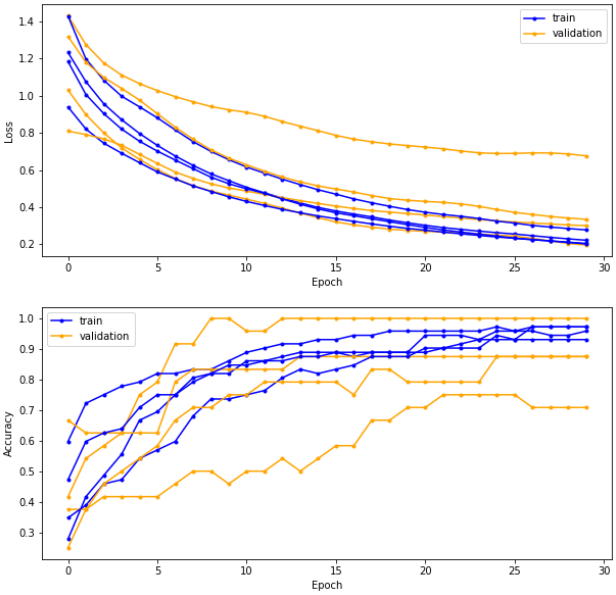
Iterasi	loss	accuracy	val_loss	val_accuracy
1	1.42853903770446	0.347222208976745	1.31891345977783	0.25
2	1.19767904281616	0.388888895511627	1.17928302288055	0.375
3	1.08343362808227	0.45833334326744	1.09813201427459	0.45833334326744
4	0.997298121452331	0.472222208976745	1.03919792175292	0.5
5	0.940138518810272	0.541666686534881	0.976241767406463	0.541666686534881
6	0.880979418754577	0.569444417953491	0.903235018253326	0.583333313465118
7	0.816111326217651	0.597222208976745	0.830689430236816	0.666666686534881
8	0.7527996301651	0.680555582046508	0.765829980373382	0.708333313465118
9	0.70030665397644	0.736111104488372	0.709108829498291	0.708333313465118
10	0.655715644359588	0.736111104488372	0.66418844461441	0.75
11	0.615988075733184	0.75	0.62775832414627	0.75
12	0.583221733570098	0.763888895511627	0.593752264976501	0.791666686534881
13	0.550853490829467	0.805555582046508	0.562182426452636	0.791666686534881
14	0.521527528762817	0.833333313465118	0.536598443984985	0.791666686534881
15	0.493954181671142	0.819444417953491	0.514276266098022	0.791666686534881
16	0.470160007476806	0.833333313465118	0.49811640381813	0.791666686534881
17	0.444856226444244	0.847222208976745	0.481451272964477	0.75
18	0.423894464969635	0.875	0.46283820271492	0.833333313465118
19	0.404965758323669	0.875	0.446694701910018	0.833333313465118
20	0.387419819831848	0.875	0.437585085630416	0.791666686534881
21	0.372984766960144	0.902777791023254	0.431391090154647	0.791666686534881
22	0.360870480537414	0.902777791023254	0.426433444023132	0.791666686534881
23	0.351105481386184	0.902777791023254	0.417899399995803	0.791666686534881
24	0.33934861421585	0.902777791023254	0.404681891202926	0.791666686534881
25	0.325399905443191	0.944444417953491	0.387500166893005	0.875
26	0.314682602882385	0.930555582046508	0.372059941291809	0.875
27	0.301937967538833	0.930555582046508	0.361088514328002	0.875
28	0.293232560157775	0.930555582046508	0.350176483392715	0.875
29	0.285319119691848	0.930555582046508	0.341893047094345	0.875
30	0.278054684400558	0.930555582046508	0.334800332784652	0.875

Plot pada *Fold* Ke-4 dapat dilihat pada gambar berikut :



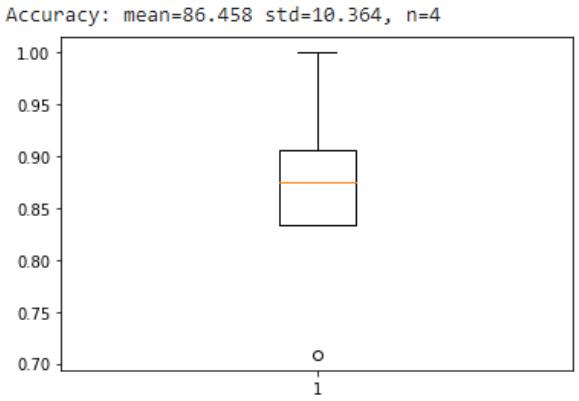
Gambar 12. Plot *Fold* Ke-4

Hasil dari Plot gabungan ditampilkan pada grafik berikut :



Gambar 13. Hasil dari Plot Gabungan

Hasil data latih juga diplotting menggunakan metode *Boxplot*



Gambar 14. *Plotting* Hasil Data Latih

Didapat nilai rata-rata dari *train* data latih ke-4 fold adalah 86.458 dengam standart deviasinya adalah 10.364.

Dari hasil latih data tersebut, didapat perbandingan pada tiap *fold* sebagai berikut

TABEL VI. PERBANDINGAN TIAP *FOLD*

Fold	Akurasi *Fold* Ke -	*Loss Fold* Ke -
1	100.000	0.197
2	87.500	0.301
3	70.833	0.676
4	87.500	0.3335

Dari tabel diatas, model yang akan diambil dan diuji adalah model dengan *fold* ke-1, dikarenakan model *fold* ke-4 memiliki akurasi tertinggi yaitu sebesar 100.000 dan *loss* yang paling kecil yaitu 0.197.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
mobilenetv2_1.00_128 (Model) (None, 4, 4, 1280)        2257984
_________________________________________________________________
global_average_pooling2d_1 ( (None, 1280)              0
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 3843
=================================================================
Total params: 2,261,827
Trainable params: 3,843
Non-trainable params: 2,257,984
_________________________________________________________________
```

Gambar 15. Model arsitektur *fold* ke-1 yang dipilih

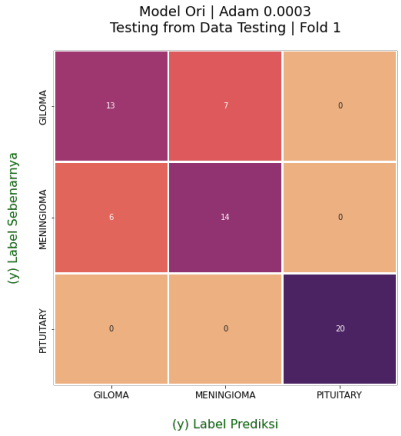
Dari hasil pengujian dengan data *testing* sebesar 60 buah, didapat nilai evaluasi akurasi sebesar 0.7833 dengan *loss* sebesar 0.4511.

Hasil dari *Classification Report* akan ditampilkan pada tabel dibawah ini :

TABEL VII. TABEL HASIL *CLASSIFICATION REPORT*

	Precision	Recall	F1-Score	Support
Glioma	0.68	0.65	0.67	20
Meningioma	0.67	0.70	0.68	20
Pituitary	1.00	1.00	1.00	20
Akurasi	-	-	0.78	60
Macro Average	0.78	0.78	0.78	60
Weighted Average	0.78	0.78	0.78	60

Confusion Matrix hasil pengujian dengan data testing akan ditampilkan pada gambar berikut ini



Gambar 16. Hasil Pengujian Data *Testing*

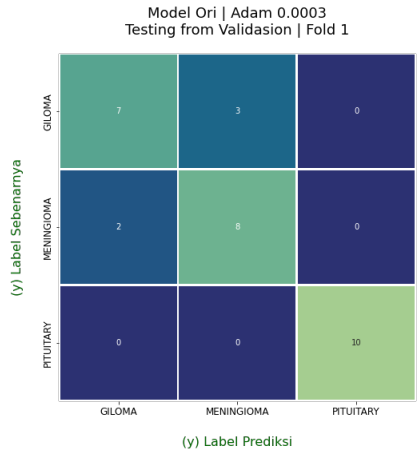
Dari hasil pengujian dengan data validasi sebesar 30 buah, didapat nilai evaluasi akurasi sebesar 0.83 dengan loss sebesar 0.4511.

Hasil dari *Classification Report* akan ditampilkan pada tabel dibawah ini

TABEL VIII. TABEL HASIL *CLASSIFICATION REPORT*

	Precision	Recall	F1-Score	Support
Glioma	0.78	0.70	0.74	10
Meningioma	0.73	0.80	0.76	10
Pituitary	1.00	1.00	1.00	10
Akurasi	-	-	0.83	30
Macro Average	0.84	0.84	0.83	30
Weighted Average	0.84	0.83	0.83	30

Confusion Matrix hasil pengujian dengan data validasi akan ditampilkan pada gambar berikut ini :



Gambar 17. Hasil Pengujian Data Validasi

5. KESIMPULAN DAN SARAN

Arsitektur Transfer Learning Model *MobileNetV2* dapat digunakan sebagai pretrained dalam pembuatan arsitektur model *Convolutional Neural Network* dengan data citra yang kecil dan data citra yang imbalance. Berdasarkan percobaan menggunakan 4 *cross validation* dan 30 *epoch*, *MobileNetV2* yang dikombinasikan dengan transfer learning dari database ImageNet dapat melakukan klasifikasi penyakit otak dengan rata-rata akurasi sebesar 86.458. Model *Convolutional Neural Network Transfer Learning MobileNetV2* juga mampu melakukan klasifikasi penyakit otak pituitary dengan sangat baik, pada nilai classification report, klasifikasi penyakit otak pituitary mendapatkan nilai *precision, recall* dan *f-1 score* sebesar 100%, akan tetapi model belum bisa sepenuhnya melakukan klasifikasi penyakit otak *glioma* dengan baik, dikarenakan terkadang model mendeteksi penyakit otak *glioma* terdeteksi sebagai penyakit *meningioma*

Klasifikasi penyakit otak *glioma* mendapat nilai *precision* sebesar 0.78, nilai *recall* sebesar 0.70 dan *f-1 score* sebesar 0.74 pada uji data dengan data validasi. Klasifikasi penyakit otak *meningioma* mendapat nilai *precision* sebesar 0.73, nilai recall sebesar 0.80 dan f-1 score sebesar 0.76 pada uji data dengan data validasi.

Pengembangan model lebih lanjut dapat dilakukan dengan menggunakan *Transfer Learning* yang lain seperti *InceptionV3, InceptionV4* atau *GoogleNet*. Selain itu pengembangan model lebih lanjut juga bisa dengan dilakukan dengan penambahan dataset atau dengan melakukan penambahan kategori penyakit otak.

UCAPAN TERIMA KASIH

Penelitani ini dapat dilaksanakan dengan lancar, atas bantuan beberapa pihak. Untuk itu, peneliti mengucapkan terima kasih untuk rekan tim penelitian dan Ibu Dr. Arnita, M.Si selaku dosen pembimbing dalam penelitian ini yang telah memberikan bimbingan dan dorongan kepada tim peneliti dalam menyelesaikan penelitian ini.

DAFTAR PUSTAKA

[1] I. B. L. M. Suta, R. S. Hartati, and Y. Divayana, "Diagnosa Tumor Otak Berdasarkan Citra MRI (Magnetic Resonance Imaging)," *Maj. Ilm. Teknol. Elektro*, vol. 18, no. 2, 2019, doi: 10.24843/mite.2019.v18i02.p01.

[2] R. Andre R, B. Wahyu P, and R. Purbaningtyas, "Klasifikasi Tumor Otak Menggunakan Convolutional Neural Network Dengan Arsitektur Efficientnet-B3," *JUST IT J. Sist. Informasi, Teknol. Inf. dan Komput.*, vol. 11, no. 3, pp. 55–59, 2021.

[3] M. A. Hanin, R. Patmasari, and R. Y. Nur, "Sistem Klasifikasi Penyakit Kulit Menggunakan Convolutional Neural Network (CNN)," *e-Proceeding Eng.*, vol. 8, no. 1, pp. 273–281, 2021.

[4] M. Ghozali and H. Sumarti, "Pengobatan Klinis Tumor Otak pada Orang Dewasa," *J. Pendidik. Fis. dan Fis. Terap.*, vol. 2, no. 1, pp. 1–14, 2021.

[5] Y. Achmad, R. C. Wihandika, and C. Dewi, "Klasifikasi Emosi Berdasarkan Ciri Wajah Menggunakan Convolutional Neural Network," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 11, pp. 10595–10604, 2019.

[6] H. F. Jessar, A. T. Wibowo, and E. Rachmawati, "Klasifikasi Genus Tanaman Sukulen Menggunakan Convolutional Neural Network (CNN)," *e-Proceeding Eng.*, vol. 8, no. 2, pp. 1153–1189, 2021, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/14709/14486.

[7] M. Ramadhan, D. I. Mulyana, and M. B. Yel, "Optimasi Algoritma Cnn Menggunakan Metode Transfer Learning Untuk Klasifikasi Citra X-Ray Paru-Paru Pneumonia Dan Non-Pneunomia," *J. Tek. Inform. Kaputama*, vol. 6, no. 2, pp. 670–679, 2022.

[8] A. Dharmaputra, M. Cahyanti, M. R. D. Septian, and E. R. Swedia, "Aplikasi Face Mask Detection Menggunakan Neural Network Mobilenetv2 Berbasis Android," *Sebatik*, vol. 25, no. 2, pp. 382–389, 2021, doi: 10.46984/sebatik.v25i2.1503.

[9] N. D. Miranda, L. Novamizanti, and S. Rizal, "Convolutional Neural Network Pada Klasifikasi Sidik Jari Menggunakan Resnet-50," *J. Tek. Inform.*, vol. 1, no. 2, pp. 61–68, 2020, doi: 10.20884/1.jutif.2020.1.2.18.

[10] I. Supiyani and N. Arifin, "Identifikasi Nomor Rumah Pada Citra Digital Menggunakan Neural Network," *J. Method.*, vol. 8, no. 1, pp. 18–21, 2022.

[11] K. A. Shianto, K. Gunadi, and E. Setyati, "Deteksi Jenis Mobil Menggunakan Metode YOLO Dan Faster," *J. Infra*, vol. 7, no. 1, pp. 157–163, 2019, [Online]. Available: http://publication.petra.ac.id/index.php/teknik-informatika/article/view/8065.

[12] E. Nurraharjo and D. B. Santoso, "Teknik Pengambilan Warna Terpilih pada Citra Motif Batik dengan Metode Color Space Adaptif," *Pros. SENDI_U*, pp. 213–219, 2018, [Online]. Available: https://unisbank.ac.id/ojs/index.php/sendi_u/article/view/5984.

[13] A. Ridhovan and A. Suharso, "Penerapan Metode Residual Network (RESNET) Dalam Klasifikasi Penyakit Pada Daun Gandum," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 7, no. 1, pp. 58–65, Feb. 2022, [Online]. Available: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi/article/view/2410.

[14] N. Hikmatia A.E and M. Ihsan Zul, "Aplikasi Penerjemah Bahasa Isyarat Indonesia menjadi Suara berbasis Android menggunakan Tensorflow," *J. Komput. Terap.*, vol. 7, no. 1, pp. 74–83, 2021.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.

[16] L. Mardiana, D. Kusnandar, and N. Satyahadewi, "Analisis Diskriminan Dengan K Fold Cross Validation Untuk Klasifikasi Kualitas Air Di Kota Pontianak," *Bimaster Bul. Ilm. Mat. Stat. dan Ter.*, vol. 11, no. 1, pp. 97–102, 2022, [Online]. Available: https://jurnal.untan.ac.id/index.php/jbmstr/article/view/51608.

[17] J. Xu, Y. Zhang, and D. Miao, "Three-way confusion matrix for classification: A measure driven view," *Inf. Sci. (Ny).*, vol. 507, pp. 772–794, 2020, doi: 10.1016/j.ins.2019.06.064.