

Transfer Learning untuk Klasifikasi Penyakit dan Hama Padi Menggunakan MobileNetV2

Transfer Learning for Rice Disease and Pest Classification using MobileNetV2

Oddy Virgantara Putra¹, Muhammad Zaim Mustaqim², Dihin Muriatmoko³

^{1,2,3}Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Darussalam Gontor, Ponorogo

E-mail: ¹oddy@unida.gontor.ac.id, ²mzaimmustaqim@unida.gontor.ac.id, ³dihin@unida.gontor.ac.id

Abstrak

Tanaman padi memegang peranan penting untuk ketersediaan pangan di Indonesia. Namun, banyak faktor yang dapat mempengaruhi panen pada tanaman padi, salah satunya adalah penyakit dan hama pada tanaman padi. Keterlambatan penanganan pada penyakit dan hama tanaman padi biasanya terjadi karena keterlambatan diagnosis penyakit dan hama tanaman padi, apalagi pada daerah terpencil yang memiliki keterbatasan akses internet dan komputasi. Oleh karena itu proses klasifikasi penyakit dan hama pada tanaman padi secara otomatis yang dapat di implementasikan pada perangkat dengan daya sumber yang terbatas, seperti perangkat seluler sangat dibutuhkan. Penelitian ini membandingkan lima arsitektur *Transfer Learning* yaitu MobileNet V2, NasNet Mobile, EfficientNet B7, Inception V3, VGG 16 dan model dengan simple CNN. Penelitian ini menggunakan dataset berisikan 5 penyakit, 3 hama dan 1 tanaman padi sehat. Setiap data akan melewati tahap preprocessing dan augmentasi. MobileNet V2 memiliki jumlah parameter dan performa yang cukup baik dalam mengklasifikasi penyakit tanaman padi dengan jumlah 2,27 juta parameter, *accuracy* sebesar 96%, *precision* sebesar 96%, *recall* sebesar 96%, *specificity* sebesar 99%, dan *f1-score* sebesar 96%.

Kata kunci: *Convolutional Neural Network, Transfer learning, Penyakit tanaman padi, Hama, Dataset*

Abstract

Rice plants play an important role for food availability in Indonesia. However, many factors can affect the yield of rice plants, one of which is disease and pests in rice plants. Delays in handling rice plant diseases and pests can occur due to delays in diagnosing rice plant diseases and pests, especially in remote areas that have limited internet access and repairs. Therefore an automatic process of classifying diseases and pests in rice plants that can be implemented on devices with limited resources, such as mobile devices is urgently needed. This study compares five Transfer Learning architectures, namely MobileNet V2, NasNet Mobile, EfficientNet B7, Inception V3, VGG 16 and the simple CNN model. This study used dataset which contained 5 diseases, 3 pests and 1 healthy rice plant. Each data will pass through the preprocessing and augmentation stages. MobileNet V2 has a fairly good number of parameters and performance in classifying rice plant diseases with a total of 2,27 million parameters, 96% accuracy, 96% precision, 96% recall, 99% specificity and 96% f1 score.

Keywords: Convolutional Neural Network, Transfer learning, Rice disease, Pest, Dataset

1. PENDAHULUAN

Salah satu bidang pertanian penting yang berkontribusi pada ekonomi Indonesia adalah pertanian padi [1]. Mengingat nasi adalah makanan pokok bagi mayoritas masyarakat di Indonesia. Tidak jarang sektor pertanian padi menjadi salah satu bidang pertanian yang sangat sering kita temui hampir di seluruh wilayah Indonesia. Namun pada tahun 2021 terjadi penurunan pada produksi hasil pertanian sebesar 0,3%, hal ini tentu berlawanan dengan kebutuhan konsumen yang semakin meningkat [2].

Banyak faktor yang dapat mempengaruhi penurunan luas panen pada tanaman padi salah satunya adalah penyakit dan hama pada tanaman padi [3][4]. Jamur, bakteri, virus dan hama adalah organisme yang dapat mengakibatkan penyakit pada tanaman padi [5]. Penyakit dan hama pada tanaman padi dapat mengganggu pertumbuhan tanaman padi, karna penyakit dan hama tersebut dapat mengganggu proses fotosintesis pada tanaman dan menimbulkan perubahan pada bentuk dan warna [6] pada tanaman padi yang diserang. Secara tradisional deteksi penyakit padi dilakukan dengan memberikan sampel tanaman yang terkena penyakit kepada tenaga ahli, dari sampel tersebut tenaga ahli akan mengidentifikasi penyakit tersebut, hal ini tentu akan memakan banyak waktu jika terimplementasikan pada pertanian besar [7]. ditambah dengan keterbatasan akses internet dan komputasi pada daerah terpencil, hal ini tentu membuat susah petani dalam menghubungi tenaga ahli [8]. Oleh karena itu proses klasifikasi penyakit dan hama pada tanaman padi secara otomatis yang dapat di implementasikan pada perangkat dengan daya sumber yang terbatas, seperti perangkat seluler sangat dibutuhkan.

E. Anggiratih mengklasifikasi penyakit padi menggunakan CNN dengan arsitektur *transfer learning* EfficientNet B3. Penelitian ini mengklasifikasi dua penyakit pada tanaman padi yaitu: *bacterial leaf* dan *brown spot* dengan jumlah dataset sebesar 875 citra [3]. Penelitian ini membandingkan dua arsitektur *Convolutional Neural Network* (CNN) yaitu EfficientNet B3 dan MobileNet V3 untuk melakukan pengklasifikasian. Didapatkan arsitektur EfficientNet B3 mendapatkan nilai akurasi pelatihan sebesar 99%. Nilai akurasi tersebut lebih tinggi dibandingkan dengan nilai yang didapatkan arsitektur MobileNet V2 yang mendapatkan nilai akurasi pelatihan sebesar 57%.

Ulfa Nur Oktaviana mengklasifikasi penyakit pada daun padi memakai arsitektur CNN ResNet101. Data citra melewati tahap augmentasi untuk memperbanyak varian padan citra [1]. Dataset yang digunakan ialah *Rice Leaves Diseases* dataset dengan total 120 sampel citra daun padi daun padi, dengan rincian 40 citra *bacterial leaf blast*, 40 citra *brown spot*, dan 40 citra *leaf smut*. Data dipecah menjadi data latih 90% dan sisanya data uji. Pengklasifikasian menggunakan arsitektur ResNet101 memperlihatkan *score* akurasi pengujian yang tinggi sebesar 100% dan *loss* pada pengujian sebesar 5,61%.

S. Ray mengklasifikasi penyakit daun padi menggunakan arsitektur VGG 16. Dataset dibuat dengan cara mengumpulkan data gambar penyakit tanaman padi dari ladang desa Madarat di Baripur, desa Dharinda di Tamluk dan Basirhat, negara bagian Benggala Barat, India di internet [8]. Dataset dengan total 2156 sampel citra daun padi, dengan rincian 1649 citra penyakit (*leaf blight*, *leaf blast*, dan *brown spot*) dan 507 citra padi sehat. Data di bagi menjadi data latih sebesar 70% dan 30% data uji. Arsitektur VGG 16 memperlihatkan nilai akurasi yang tinggi berupa akurasi pengujian sebesar 92,46%.

Syaikul Anam membuat model menggunakan *Convolutional Neural Network* (CNN). Data gambar akan melewati serangkaian proses *resizing* dan ekstraksi fitur. *RiceLeafs* dataset merupakan dataset yang digunakan pada penelitian ini, dataset terdiri dari 900 citra daun padi, pada penelitian ini data dibagi menjadi data latih sebesar 780 data dan data uji sebesar 180 data. Penelitian ini menggunakan empat *layer* konvolusi dan empat *layer max pooling* dengan *fully connected layer* pada arsitektur yang dibuat. Hasil pengujian padan model memiliki nilai akurasi pelatihan sebesar 92% dan akurasi pengujian sebesar 77% [9].

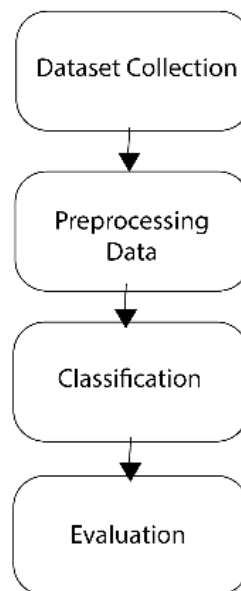
R. Rahman membandingkan efisiensi dari arsitektur VGG 16, Inception V3 MobileNet V2, NasNet Mobile, SqueezeNet, dan Simple CNN dalam mengklasifikasi penyakit dan hama tanaman padi. Penelitian menggunakan 10 *cross-validation* dikarenakan data yang digunakan

tidak seimbang. Hasil yang didapatkan adalah *simple* CNN memiliki parameter yang paling sedikit dan memiliki tingkat akurasi yang cukup baik yaitu 93% [10].

Fokus pada penelitian di atas adalah mengklasifikasi penyakit dan hama pada tanaman padi, beberapa penelitian berfokus pada performa klasifikasi pada model dan yang lain berfokus pada model yang ringan untuk melakukan klasifikasi. Oleh karena itu penelitian ini akan membandingkan arsitektur CNN seperti MobileNet V2 [11], NasNet Mobile [12], EfficientNet B7 [13], Inception V3 [14], VGG16 [15] yang di padukan dengan Global Average Pooling dan *simple* CNN untuk mencari model yang ringan dan memiliki performa yang baik dalam mengklasifikasi penyakit dan hama pada tanaman padi agar dapat digunakan pada daerah terpencil yang minim akses internet dan komputasi.

2. METODE PENELITIAN

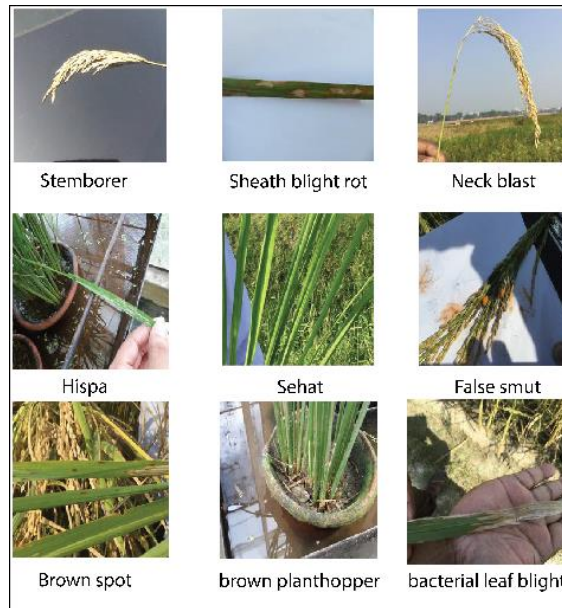
Langkah-langkah yang akan dilakukan dalam penelitian ini di antara lain *dataset collection*, melakukan *preprocessing data*, membangun model dengan beberapa arsitektur *transfer learning*, setelah itu mengevaluasi model, dapat dilihat pada Gambar 1.



Gambar 1 alur penelitian

2.1 Dataset Collection

Data gambar yang dipakai pada penelitian ini mencakup gambar penyakit dan hama padi yang berasal dari dataset Rahman 2020 yang terdiri 1426 citra. Dataset memiliki 9 jenis kelas dengan rincian data: *Stemborer* (hama) sebanyak 201 citra, *neck blast* (penyakit) sebanyak 286 citra, hispa (hama) sebanyak 73 citra, *false smut* (penyakit) sebanyak 93 citra, *brown planthopper* (hama) sebanyak 71 citra, *sheat blight rot* (penyakit) sebanyak 219 citra, *brown spot* (penyakit) sebanyak 111 citra dan citra padi sehat sebanyak 234 citra. Contoh gambar masing-masing kelas penyakit dan hama tanaman padi dapat dilihat pada Gambar 2.



Gambar 2 penyakit dan hama tanaman padi [10]

2.2 Preprocessing Data

Preprocessing gambar adalah langkah terpenting untuk mendapatkan fitur dan nilai akurasi klasifikasi yang baik [16]. Pada bagian ini ukuran gambar akan diubah menjadi 224x224 pixel untuk menyamakan dimensi setiap gambar di setiap kelas penyakit dan hama tanaman padi. Setelah itu dilakukan normalisasi pada data. Hal ini bertujuan untuk menyamakan rentan nilai variabel agar mempunyai rentan nilai yang serupa berkisar 0 sampai 1. Dikarenakan data yang digunakan tidak seimbang, penelitian ini menggunakan metode *oversampling* yang dimana gambar pada kelas minoritas akan diduplikat secara acak agar jumlahnya sama dengan jumlah gambar pada kelas mayoritas [17]. Setelah itu data akan dibagi menjadi 80% data latih dan 20% data uji. Beberapa teknik augmentasi pun digunakan seperti *horizontal flip*, *vertical flip*, dan *rotation*. Hal ini digunakan untuk menambahkan variasi pada data [18] dan mencegah terjadinya *overfitting* [19].

2.3 Membangun Model CNN dengan Transfer learning

Convolutional Neural Network (CNN) merupakan struktur modern dari *Neural Network* dan telah terbukti bekerja luar biasa dibidang visi computer [20]. Bagian *Convolutional Neural Network* (CNN) terdiri dari bagian konvolusi yang mengekstrak pola input yang relevan [21] dan bagian *fully connected layer* yang menghitung prediksi berdasarkan pola yang diekstraksi. *Transfer learning* merupakan proses pemakaian kembali jaringan yang telah dilatih untuk tujuan lain pada kumpulan data yang sangat besar untuk kita pakai sendiri [22]. Pada penelitian ini, peneliti menggunakan beberapa arsitektur *transfer learning* seperti MobileNet V2 [11], NasNet Mobile [12], EfficientNet B7 [13], Inception V3 [14], VGG16 [15] dengan Global Average Pooling untuk melakukan ekstraksi pada fitur pada model yang dibangun dan juga akan dibandingkan dengan arsitektur simple CNN.

2.3.1 MobileNetV2

MobileNet adalah kelas dari *Convolutional Neural Network* yang di design oleh peneliti Google sebagai “*mobile-fit*”. MobileNet ramah dengan sumber daya yang terbatas dan dapat berjalan langsung pada ponsel [23]. Pada jaringan MobileNet V2 terdapat 19 *layer*, yang dimana bagian tengah *layer* berfungsi untuk mengekstraksi fitur pada gambar dan bagian akhir *layer* berfungsi untuk mengklasifikasi gambar [24].

MobileNet didasarkan pada kedalaman konvolusi yang dapat dipisahkan, yang terdiri dari

dua lapisan inti di dalamnya: *depthwise convolution* dan *pointwise convolution*. *Depthwise convolution* adalah langkah memfilter *input* tanpa membuat fitur baru, dengan demikian *depthwise convolution* dapat menghasilkan keluaran yang sama dengan dengan kovolusi standart, hal ini tentu lebih efisien karna mengurangi jumlah parameter.

2.3.2 NasNet Mobile

Diperkenalkan oleh Google brain pada tahun 2018, gagasan *transferability* pertama kali di usulkan oleh mereka [22]. Lapisan konvolusi terbaik, dijuluki sebagai sel untuk memecahkan masalah klasifikasi pada dataset CIFAR10 dan beberapa tumpukan sel tersebut digunakan untuk memecahkan masalah klasifikasi ImageNet.

NasNet Mobile adalah CNN yang dilatih lebih dari satu juta gambar dari dataset ImageNet [12]. Arsitektur CNN NasNet dapat diskalakan (dibangun oleh pencarian Neural arsitektur) yang terdiri dari blok bangunan dasar (sel) yang dioptimalkan menggunakan *reinforcement learning*. Sebuah sel hanya terdiri dari beberapa operasi (beberapa dapat dipisahkan konvolusi dan pooling) dan di ulang beberapa kali sesuai dengan kapasitas jaringan yang dibutuhkan. Versi mobile yaitu NasNet Mobile terdiri dari 12 sel.

2.3.3 Inception V3

Inception V3 adalah salah satu arsitektur konvolusi yang dikembangkan oleh Google. Pada tahun 2014 penelitian Google menciptakan versi pertama dari arsitektur ini yaitu GoogleNet (Inception VI). GoogleNet adalah pemenang tantangan klasifikasi citra ILSVRC 2014. Teknik konvolusi 1 x 1 digunakan dalam arsitektur ini yang memungkinkan menciptakan kedalaman jaringan yang lebih dalam. *Global Average Pooling* juga diperkenalkan dalam arsitektur ini [14].

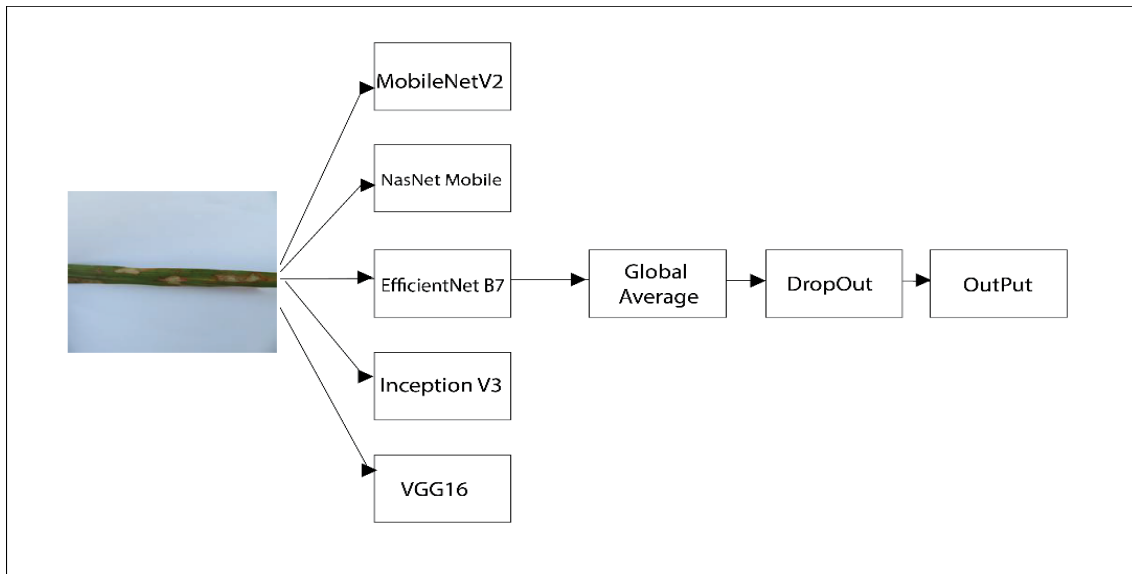
2.3.4 EfficientNet B7

EfficientNet B7 adalah model yang telah dilatih menggunakan dataset ImageNet, fundamental dari arsitektur EfficientNet adalah *Mobile Inverted Bottle Neck*, terdapat 8 model dari EfficientNet dan setiap model memiliki variasi dari lebar, kedalaman, resolusi dan ukuran yang berbeda-beda, EfficientNet B7 memiliki jumlah parameter yang lebih banyak dari varian lainnya. Hal ini memungkinkan EfficientNet B7 dapat mengenali pola yang lebih kompleks dan memiliki performa yang baik [13].

2.3.5 VGG16

VGG16 mengacu pada model VGG, yang bisa disebut juga VGGNet adalah model *Convolutional Neural Network* (CNN) yang mendukung 16 lapisan. Model ekstraksi fitur VGG16 dapat mengekstraksi data dengan jumlah besar dan menghasilkan akurasi yang baik [25] arsitektur VGG16 menggunakan 3 x 3 filter konvolusi. Setelah setiap lapisan *max pooling*, jumlah filter konvolusi menjadi dua kali lipat di VGG16. VGG16 adalah salah satu model teratas dari kompetisi ILSVRC pada 2014 [15].

Model *transfer learning* yang digunakan menggunakan bobot imagenet dan tidak memasukkan lapisan akhir pada setiap arsitektur *transfer learning* yang digunakan. penelitian ini menambahkan *Global average pooling2D*, *layer dropout* sebanyak 0,2 dan ditutup dengan *dense layer* dengan *output* 9 kelas dengan *activation softmax*. Alur model dapat dilihat pada Gambar 3, lalu model akan di-*compile* menggunakan optimasi adam, *loss categorical crossentropy* dan *learning rate* sebesar 0,001.



Gambar 3 Alur klasifikasi gambar

2.4 Evaluasi Model

Pada bagian ini model yang didapat dari beberapa arsitektur *transfer learning* dan model yang dibuat dengan simple CNN akan di evaluasi performanya. Hal ini bertujuan untuk mendapatkan model yang optimal diantara ketiga optimasi tersebut. Performa model di evaluasi menggunakan *precision, accuracy, f1-score, sensitivity, dan specificity* [26]. *Precision, accuracy, f1-score, sensitivity, specificity* dan *recall* di hitung berdasarkan *confusion matrix* [27]. Jika data positif diprediksi benar maka di sebut *true positive* (TP), jika data di prediksi salah disebut *false negative* (FN), jika data negatif di prediksi benar disebut *true negative* (TN), dan jika prediksi salah disebut *false positive* (FP).

Accuracy adalah proporsi prediksi yang benar (baik positif maupun negatif) dari semua data. *Accuracy* bisa dihitung dengan Persamaan (1). *Precision* adalah proporsi prediksi positif yang benar dari semua hasil positif yang diprediksi dan diperoleh dari Persamaan (2). *Sensitivity* atau *recall* adalah proporsi prediksi positif yang benar dari semua data positif yang sebenarnya. *Recall* bisa didapatkan dari Persamaan (3). *Specificity* adalah perbandingan prediksi negatif yang benar dengan semua data negatif sebagaimana diperoleh dari Persamaan (4). *F1-score* adalah rata-rata tertimbang dari *precision* dan *recall* dan dihitung dengan Persamaan (5).

$$Accuracy = \frac{Tp + Tn}{Tp + Fp + Fn + Tn} \times 100\% \quad (1)$$

$$Precision = \frac{Tp}{Tp + Fp} \times 100\% \quad (2)$$

$$Sensitivity = \frac{Tp}{Tp + Fn} \times 100\% \quad (3)$$

$$Specificity = \frac{TN}{Tn + Fp} \times 100\% \quad (4)$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% \quad (5)$$

3. HASIL DAN PEMBAHASAN

Data gambar yang telah melewati tahapan preprocessing akan berubah tinggi dan lebarnya menjadi 224 x 224 pixel dan memiliki rentang nilai antara 0 sampai 1 hal ini bertujuan dalam membantu proses pelatihan pada model. Setelah data disamaratakan rentan nilainya data

disamakan jumlah data citranya pada setiap kelas menjadi 286 citra pada setiap kelas dengan menggunakan metode *oversampling*. Setelah data melewati tahapan preprocessing, data dipecah menjadi dua bagian. yaitu data testing sebesar 20% yang berjumlah 515 data citra dan data training yaitu sebesar 80% yang berjumlah 2059 data citra. Setelah data dibagi menjadi data testing dan training, data melewati tahapan augmentasi agar data memiliki varian yang lebih banyak adapun teknik augmentasi yang digunakan adalah *horizontal flip*, *vertical flip*, dan *rotation*. Tahap selanjutnya adalah proses pelatihan, data-data yang telah diolah akan dimasukkan pada model-model yang telah dibuat dengan MobileNet V2, NasNet Mobile, InceptionV3, EfficientNet B7, simple CNN dan VGG 16 akan di compile menggunakan *learning rate* 0,001, *optimizer adam*, dan *loss categorical crossentropy* lalu dilatih dengan *epoch* sebanyak 100 di setiap model. Model-model tersebut akan dibandingkan performanya dengan nilai *accuracy*, *precision f1-score*, *sensitivity*, *specificity* dan jumlah parameter yang dihasilkan pada model

3.1 MobileNet V2

Model yang dibuat dengan MobilNet V2 menggunakan bobot imagenet dan tidak memasukkan lapisan akhir pada arsitektur yang digunakan. Kami menambahkan *Global average pooling2D*, *layer dropout* 0,2 kemudian ditutup dengan *Dense layer* dengan keluaran 9 kelas dengan *activation softmax*. Dari rangkaian di atas didapatkan jumlah parameter sebesar 2.269.513 pada model. Model yang telah dibuat di compile menggunakan *optimasi adam*, *learning rate* 0,001 dan *loss categorical crossentropy*. Arsitektur dapat dilihat pada Tabel 1, Hasil dari model yang menggunakan arisitektur MobileNetv2 menunjukkan hasil *accuracy training* sebesar 0,95 dan *validation accuracy* sebesar 0,96. Selain itu model mendapatkan *loss training* sebesar 0,19 dan *validation loss* sebanyak 0,18. Dari hasil *confution matrix* didapatkan score TN, TP, FN, dan FP dari seluruh kelas, dapat dilihat pada Tabel 2. Setelah mengetahui score TP, TN, FP, dan FN dari model, didapatkan *score accuracy* sebanyak 96%, *precision* sebesar 96%, *sensitivity* sebesar 96%, *specificity* sebesar 99%, dan *f1-score* sebesar 96%.

Tabel 1. Arsitektur MobileNet V2

| Layer | Output Shape | Param |
|------------------------|--------------|---------|
| Input | 224, 224, 3 | 0 |
| MobileNet V2 | 7, 7, 1280 | 2257984 |
| Global Average Pooling | 1280 | 0 |
| Dropout | 1280 | 0 |
| Dense | 9 | 11529 |

Tabel 2. Confution Matrix MobileNet V2

| Nama | TP | TN | FP | FN |
|----------------------|----|-----|----|----|
| False Smut | 48 | 447 | 3 | 1 |
| Hispa | 49 | 446 | 1 | 1 |
| Neck Blast | 55 | 440 | 5 | 1 |
| Brown Spot | 50 | 445 | 1 | 0 |
| Stemborer | 55 | 440 | 2 | 5 |
| Bacterial Leaf Blast | 75 | 420 | 2 | 1 |
| Brown Planthoper | 64 | 431 | 0 | 0 |
| Sheat blight Rot | 47 | 448 | 0 | 5 |
| Sehat | 52 | 443 | 6 | 6 |

3.2 NasNet Mobile

Model yang dibuat dengan NasNet Mobile menggunakan bobot imagenet dan tidak memasukkan lapisan akhir pada arsitektur yang digunakan. Kami menambahkan *Global average pooling2D*, *layer dropout* 0,2, ditutup dengan *Dense layer* dengan keluaran 9 kelas dengan *activation softmax*. Dari rangkaian di atas didapatkan jumlah parameter sebesar 4,279,229 pada model. Model yang telah dibuat di compile menggunakan *optimasi adam*, *learning rate* 0,001 dan *loss categorical crossentropy*. Arsitektur dapat dilihat pada Tabel 3, Hasil dari model yang menggunakan arsitektur NasNet, Mobile menunjukkan hasil *accuracy training* sebesar 0,95 dan *validation accuracy* sebesar 0,93. Selain itu model mendapatkan *loss training* sebesar 0,14 dan *validation loss* sebanyak 0,18 Dari hasil *confusion matrix* didapatkan score TN, TP, FN, dan FP dari seluruh kelas, dapat dilihat pada Tabel 4. Setelah mengetahui score TP, TN, FP, dan FN dari model, didapatkan *score accuracy* sebanyak 94%, *precision* sebanyak 94%, *sensitivity* sebanyak 94%, *specificity* sebanyak 99%, dan *f1-score* sebanyak 94%.

Tabel 3. Arsitektur NasNet Mobile

| Layer | Output Shape | Param |
|------------------------|--------------|---------|
| Input | 224, 224, 3 | 0 |
| MobileNet V2 | 7, 7, 1056 | 4269716 |
| Global Average Pooling | 1056 | 0 |
| Dropout | 1056 | 0 |
| Dense | 9 | 9513 |

Tabel 4. Confution Matrix NasNet Mobile

| Nama | TP | TN | FP | FN |
|----------------------|----|-----|----|----|
| False Smut | 46 | 437 | 2 | 3 |
| Hispa | 49 | 434 | 2 | 1 |
| Neck Blast | 52 | 431 | 2 | 4 |
| Brown Spot | 47 | 436 | 3 | 3 |
| Stemborer | 58 | 425 | 4 | 2 |
| Bacterial Leaf Blast | 70 | 413 | 3 | 6 |
| Brown Planthoper | 64 | 419 | 1 | 0 |
| Sheat blight Rot | 49 | 434 | 5 | 3 |
| Sehat | 48 | 435 | 10 | 10 |

3.3 Inception V3

Model yang dibuat dengan Inception V3 menggunakan bobot imagenet dan tidak memasukkan lapisan akhir pada arsitektur yang digunakan. Kami menambahkan *Global average pooling2D*, *layer dropout* 0,2, ditutup dengan *Dense layer* dengan keluaran 9 kelas dengan *activation softmax*. Dari rangkaian diatas didapatkan jumlah parameter sebesar 21,821,225 pada model. Arsitektur dapat dilihat pada Tabel 5, Model yang telah dibuat di compile menggunakan *optimasi adam*, *learning rate* 0,001 dan *loss categorical crossentropy*. Hasil dari model yang menggunakan arsitektur InceptionV3 menunjukkan hasil *accuracy training* sebesar 0,95 dan *validation accuracy* sebesar 0,96. Selain itu model mendapatkan *loss training* sebesar 0,12 dan *validation loss* sebanyak 0,12. Dari hasil *confusion matrix* didapatkan score TN, TP, FN, dan FP dari seluruh kelas, dapat dilihat pada Tabel 6. Setelah mengetahui score TP, TN, FP, dan FN dari model, didapatkan *score accuracy* sebanyak 97%, *precision* sebesar 97%, *sensitivity* sebesar 97%, *specificity* sebesar 99%, dan *f1-score* sebesar 96%.

Tabel 5. Arsitektur Inception V3

| Layer | Output Shape | Param |
|------------------------|--------------|----------|
| Input | 224, 224, 3 | 0 |
| MobileNet V2 | 7, 7, 2048 | 21802784 |
| Global Average Pooling | 2048 | 0 |
| Dropout | 2048 | 0 |
| Dense | 9 | 18441 |

Tabel 6. Confution Matrix Inception V3

| Nama | TP | TN | FP | FN |
|----------------------|----|-----|----|----|
| False Smut | 49 | 448 | 2 | 0 |
| Hispa | 46 | 451 | 1 | 4 |
| Neck Blast | 51 | 446 | 1 | 5 |
| Brown Spot | 50 | 447 | 0 | 0 |
| Stemborer | 58 | 439 | 4 | 2 |
| Bacterial Leaf Blast | 76 | 421 | 1 | 0 |
| Brown Planthoper | 64 | 433 | 3 | 0 |
| Sheat blight Rot | 48 | 449 | 2 | 4 |
| Sehat | 55 | 442 | 4 | 3 |

3.4 VGG 16

Model yang dibuat dengan VGG 16 menggunakan bobot imagenet dan tidak memasukan lapisan akhir pada arsitektur yang digunakan. Kami menambahkan *Global average pooling2D*, *layer dropout* 0,2, ditutup dengan *Dense layer* dengan keluaran 9 kelas dengan *activation softmax*. Dari rangkaian diatas didapatkan jumlah parameter sebesar 14,714,688 pada model. Arsitektur dapat dilihat pada Table 7, Model yang telah dibuat di compile menggunakan *optimasi adam*, *learning rate* 0,001 dan *loss categorical crossentropy*. Hasil dari model yang menggunakan arsitektur VGG16 menunjukkan hasil *accuracy training* sebesar 0,83 dan *validation accuracy* sebesar 0,84. Selain itu model mendapatkan *loss training* sebesar 0,52 dan *validation loss* sebanyak 0,48. Dari hasil *confusion matrix* didapatkan score TN, TP, FN, dan FP dari seluruh kelas, dapat dilihat pada Tabel 8. Setelah mengetahui score TP, TN, FP, dan FN dari model, didapatkan *score accuracy* sebanyak 84%, *precision* sebesar 85%, *sensitivity* sebesar 84%, *specificity* sebesar 98%, dan *f1-score* sebesar 84%.

Tabel 7. Arsitektur VGG16

| Layer | Output Shape | Param |
|------------------------|--------------|----------|
| Input | 224, 224, 3 | 0 |
| MobileNet V2 | 7, 7, 512 | 14714688 |
| Global Average Pooling | 512 | 0 |
| Dropout | 512 | 0 |
| Dense | 9 | 4617 |

Tabel 8. Confution Matrix VGG16

| Nama | TP | TN | FP | FN |
|------------|----|-----|----|----|
| False Smut | 42 | 393 | 10 | 10 |
| Hispa | 52 | 383 | 11 | 7 |
| Neck Blast | 38 | 397 | 11 | 2 |

| | | | | |
|----------------------|----|-----|----|----|
| Brown Spot | 44 | 319 | 7 | 10 |
| Stemborer | 54 | 381 | 11 | 4 |
| Bacterial Leaf Blast | 48 | 387 | 12 | 21 |
| Brown Planthoper | 41 | 394 | 13 | 14 |
| Sheat blight Rot | 60 | 375 | 4 | 8 |
| Sehat | 56 | 379 | 1 | 4 |

3.5 EfficientNet B7

Model yang dibuat dengan EfficientNet B7 menggunakan bobot imagenet dan tidak memasukkan lapisan akhir pada arsitektur yang digunakan. Kami menambahkan *Global average pooling2D*, *layer dropout* 0,2, ditutup dengan *Dense layer* dengan keluaran 9 kelas dengan *activation softmax*. Dari rangkaian diatas didapatkan jumlah parameter sebesar 64,120,736 pada model. Arsitektur dapat dilihat pada Tabel 9, Model yang telah dibuat di compile menggunakan *optimasi adam*, *learning rate* 0,001 dan *loss categorical crossentropy*. Hasil dari model yang menggunakan arsitektur EfficientNet B7 menunjukkan hasil *accuracy training* sebesar 0,19 dan *validation accuracy* sebesar 0,19. Selain itu model mendapatkan *loss training* sebesar 2,08 dan *validation loss* sebanyak 2,06. Dari hasil *confusion matrix* didapatkan score TN, TP, FN, dan FP dari seluruh kelas, dapat dilihat pada Tabel 10. Setelah mengetahui score TP, TN, FP, dan FN dari model, didapatkan *score accuracy* sebanyak 20%, *precision* sebesar 28%, *sensitivity* sebesar 20%, *specificity* sebesar 89%, dan *f1-score* sebesar 14%.

Tabel 9. Arsitektur EfficientNet B7

| Layer | Output Shape | Param |
|------------------------|--------------|----------|
| Input | 224, 224, 3 | 0 |
| MobileNet V2 | 7, 7, 2560 | 64097687 |
| Global Average Pooling | 2560 | 0 |
| Dropout | 2560 | 0 |
| Dense | 9 | 23049 |

Tabel 10. Confution Matrix EfficientNet B7

| Nama | TP | TN | FP | FN |
|----------------------|----|-----|-----|----|
| False Smut | 0 | 102 | 0 | 52 |
| Hispa | 0 | 102 | 0 | 59 |
| Neck Blast | 34 | 68 | 205 | 6 |
| Brown Spot | 19 | 83 | 66 | 35 |
| Stemborer | 1 | 101 | 0 | 57 |
| Bacterial Leaf Blast | 34 | 68 | 121 | 35 |
| Brown Planthoper | 4 | 98 | 15 | 51 |
| Sheat blight Rot | 10 | 92 | 6 | 58 |
| Sehat | 0 | 102 | 0 | 60 |

3.6 Simple CNN

Model yang dibuat dengan simple CNN menggunakan 5 layer konvolusi dengan *filter* 3 x 3 dengan masing-masing parameter berjumlah 16, 24, 32, 48, 64, *activation ReLu* dan di setiap *layer* terdapat *layer maxpooling* dan *batch normalization*. Pada bagian *fully connected layer*, kami menggunakan *flatten layer*, 2 *dense layer* berparameter 128 dengan *dropout* 0,3, diakhiri dengan *dense layer* dengan keluaran berupa 9 kelas dengan *activation softmax*. Dari rangkaian diatas didapatkan jumlah parameter sebesar 275.793 pada model. Arsitektur dapat dilihat pada Table 11, model yang telah dibuat di compile menggunakan *optimasi adam*, *learning rate* 0,001 dan *loss*

categorical_crossentropy. Hasil dari model yang menggunakan arsitektur simple CNN menunjukkan hasil yang baik, model mendapatkan *accuracy training* sebesar 0,97 dan *validation accuracy* sebesar 0,92. Selain itu model mendapatkan *loss training* sebesar 0,09 dan *validation loss* sebanyak 0,34. Dari hasil *confusion matrix* didapatkan score TN, TP, FN, dan FP dari seluruh kelas, dapat dilihat pada Tabel 12. Setelah mengetahui score TP, TN, FP, dan FN dari model, didapatkan *score accuracy* sebanyak 93%, *precision* sebesar 93%, *sensitivity* sebesar 93%, *specificity* sebesar 99% dan *f1-score* sebesar 93%.

Tabel 11. Arsitektur Simple CNN

| Layer | Output Shape | Param |
|-----------------------|--------------|--------|
| Input | 224, 224, 3 | 0 |
| Conv2d 1 | 222, 222, 16 | 448 |
| Max pooling 2D 1 | 111, 111, 16 | 0 |
| Batch normalization 1 | 111, 111, 16 | 64 |
| Conv2d 2 | 109, 109, 24 | 3480 |
| Max pooling 2D 2 | 54, 54, 24 | 0 |
| Batch normalization 2 | 54, 54, 24 | 96 |
| Conv2d 3 | 52, 52, 32 | 6944 |
| Max pooling 2D 3 | 26, 26, 32 | 0 |
| Batch normalization 3 | 26, 26, 32 | 128 |
| Conv2d 4 | 24, 24, 48 | 13872 |
| Max pooling 2D 4 | 12, 12, 48 | 0 |
| Batch normalization 4 | 12, 12, 48 | 192 |
| Conv2D 5 | 10, 10, 64 | 27712 |
| Max pooling 2D 5 | 5, 5, 64 | 0 |
| Batch_normalization 5 | 5, 5, 64 | 256 |
| Flatten | 1600 | 0 |
| Dense 1 | 128 | 204928 |
| Dropout | 128 | 0 |
| Dense 2 | 128 | 165112 |
| Dense 3 | 9 | 1161 |
| Input | 224, 224, 3 | 0 |

Tabel 12. Confution Matrix Simple CNN

| Nama | TP | TN | FP | FN |
|----------------------|----|-----|----|----|
| False Smut | 46 | 431 | 3 | 6 |
| Hispa | 57 | 420 | 1 | 2 |
| Neck Blast | 40 | 437 | 9 | 0 |
| Brown Spot | 54 | 423 | 0 | 0 |
| Stemborer | 45 | 432 | 0 | 13 |
| Bacterial Leaf Blast | 58 | 419 | 3 | 11 |
| Brown Planthoper | 55 | 422 | 10 | 0 |
| Sheat blight Rot | 65 | 412 | 2 | 3 |
| Sehat | 57 | 420 | 10 | 3 |

3.7 Perbandingan

Dari hasil yang didapat dari setiap model, model-model yang telah dibuat memiliki jumlah parameter yang beragam dan tentunya memiliki performa yang beragam pula dalam mengklasifikasi penyakit dan hama pada tanaman padi. Jumlah parameter yang sedikit sangatlah penting pada pembuatan model, apalagi jika ingin diimplementasikan pada perangkat seluler. Agar model yang dibuat tidak terlalu berat dan dapat berjalan dengan baik pada perangkat seluler. Jumlah parameter yang telah dibuat dengan arsitektur *transfer learning* MobileNetV2 sebesar 2.269.513, NasNet mobile sebesar 4.279.229, InceptionV3 sebesar 21.812.225, EfficientNetB7 sebesar 64.120.736, VGG16 sebesar 14.714.688 dan model dengan simple CNN sebesar 275.793 dapat dilihat pada Table 13. dengan melihat Table 13, beberapa arsitektur dengan jumlah parameter paling sedikit ialah Simple CNN sebesar 275.793 dan MobileNetV2 sebesar 2,269,513 dan beberapa arsitektur dengan parameter terbanyak ialah InceptionV3 dengan 21,812,225 dan EfficientNet B7 sebesar 64.120.736.

Setelah melihat jumlah parameter dari masing-masing arsitektur, perbandingan dilanjutkan dengan membandingkan performa model dalam pengklasifikasian penyakit pada tanaman padi. Adapun hal-hal yang dibandingkan adalah *accuracy*, *specificity*, *sensitivity*, *precision*, dan *f1-score*. Dari Tabel 13 model dengan arsitektur *transfer learning* InceptionV3 memiliki hasil yang tinggi dibandingkan model dengan arsitektur lainnya. dengan *score accuracy* sebesar 97%, *precision* sebesar 97%, *sensitivity* sebesar 97%, *specificity* sebesar 99%, dan *f1 score* sebesar 96% dan disusul dengan MobileNetV2 dengan *score accuracy* sebesar 96%, *precision* sebesar 96%, *sensitivity* sebesar 96%, *specificity* sebesar 99%, dan *f1-score* sebesar 96% dapat dilihat pada Tabel 13. Dengan mempertimbangkan performa dan jumlah parameter, dapat disimpulkan bahwa model dengan arsitektur MobileNet V2 lebih unggul dengan jumlah parameter yang cukup sedikit dan performa klasifikasi yang cukup tinggi untuk mendeteksi penyakit pada tanaman padi pada penelitian ini.

Tabel 13. Perbandingan performa dan parameter

| Nama | Acc (%) | Sen (%) | Spe (%) | Pre (%) | F1 (%) | Param (dalam juta) |
|----------------------|-----------|-----------|-----------|-----------|-----------|--------------------|
| Mobile Net V2 | 96 | 96 | 99 | 96 | 96 | 2,27 |
| NasNet Mobile | 94 | 94 | 99 | 94 | 94 | 4,28 |
| Inception V3 | 97 | 97 | 99 | 97 | 96 | 21,8 |
| VGG 16 | 84 | 84 | 98 | 85 | 84 | 14,7 |
| EfficientNet B7 | 20 | 20 | 89 | 28 | 14 | 64,12 |
| Simple CNN | 93 | 93 | 99 | 93 | 93 | 0,27 |

4. KESIMPULAN DAN SARAN

Penelitian ini bertujuan untuk mengklasifikasi penyakit pada tanaman padi dengan total 9 kelas dengan rincian: *Stemborer*(hama) sebanyak 201 citra, *neck blast* (penyakit) sebanyak 286 citra, hispa (hama) sebanyak 73 citra, *false smut* (penyakit) sebanyak 93 citra, *brown planthopper* (hama) sebanyak 71 citra, *sheat blight rot* (penyakit) sebanyak 219 citra, *brown spot* (penyakit) sebanyak 111 citra dan citra padi sehat sebanyak 234 citra. Beberapa arsitektur *transfer learning* seperti MobilNet V2, Inception V3, Nasnet Mobile, VGG 16, EfficientNet B7 dan model simple CNN digunakan untuk melakukan klasifikasi. Inception V3 mendapatkan nilai yang sangat tinggi dengan *accuracy* sebesar 97%, *precision* sebesar 97%, *sensitivity* sebesar 97%, *specificity* sebesar 99%, dan *f1-score* sebesar 96%. Namun model dengan arsitektur Inception V3 memiliki jumlah parameter yang sangat banyak, tentu hal ini membuat berat model. Dengan membandingkan performa dan jumlah parameter pada model, didapatkan nilai yang cukup baik dari model yang dibuat dengan arsitektur MobileNet V2 dengan *accuracy* sebesar 96%, *precision* sebesar 96%, *sensitivity* sebesar 96%, *specificity* sebesar 99%, dan *f1-score* sebesar 96%. Selain nilai pengujian

model yang cukup baik, model yang dibuat dengan MobilenetV2 juga memiliki jumlah parameter yang sedikit yaitu sebesar 2,27, hal ini tentu menjadikan model ringan jika ingin di aplikasikan pada perangkat seluler. Dari hasil tersebut menunjukkan model CNN dengan *transfer learning* MobileNet V2 cukup baik untuk mendeteksi penyakit pada tanaman padi penelitian ini. Penggunaan model yang lebih ringan lagi tetapi tetap mendapatkan performa klasifikasi yang tinggi, diharapkan untuk penelitian selanjutnya.

DAFTAR PUSTAKA

- [1] Ulfah Nur Oktaviana, Ricky Hendrawan, Alfian Dwi Khoirul Annas, and Galih Wasis Wicaksono, "Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 6, pp. 1216–1222, Dec. 2021, doi: 10.29207/resti.v5i6.3607.
- [2] O. V. Putra, M. N. Annafii, T. Harmini, and N. Trisnaningrum, "Semantic Segmentation of Rice Leaf Blast Disease using Optimized U-Net," *Proceeding Int. Conf. Comput. Eng. Netw. Intell. Multimedia, CENIM 2022*, pp. 43–48, 2022, doi: 10.1109/CENIM56801.2022.10037550.
- [3] E. Anggiratih, S. Siswanti, S. K. Octaviani, and A. Sari, "Klasifikasi Penyakit Tanaman Padi Menggunakan Model Deep Learning Efficientnet B3 dengan Transfer Learning," *J. Ilm. SINUS*, vol. 19, no. 1, p. 75, 2021, doi: 10.30646/sinus.v19i1.526.
- [4] R. R. Patil and S. Kumar, "Rice-Fusion: A Multimodality Data Fusion Framework for Rice Disease Diagnosis," *IEEE Access*, vol. 10, pp. 5207–5222, 2022, doi: 10.1109/ACCESS.2022.3140815.
- [5] M. J. Hasan, S. Mahbub, M. S. Alom, and M. Abu Nasim, "Rice Disease Identification and Classification by Integrating Support Vector Machine with Deep Convolutional Neural Network," *1st Int. Conf. Adv. Sci. Eng. Robot. Technol. 2019, ICASERT 2019*, vol. 2019, no. Icasert, pp. 1–6, 2019, doi: 10.1109/ICASERT.2019.8934568.
- [6] O. Virgantara, N. Trisnaningrum, N. Sylvia, and A. Toto, "HiT-LIDIA : A Framework for Rice Leaf Disease Classification using Ensemble and Hierarchical Transfer Learning," vol. 13, no. 3, pp. 196–207, 2022.
- [7] R. and I. T. Association of Knowledge, Jāmi‘at Ibn Zuhr. École nationale des sciences appliquées d’Agadir, and Institute of Electrical and Electronics Engineers, *Proceedings of 2019 International Conference of Computer Science and Renewable Energies (ICCSRE) : 2019 July 22-24. .*
- [8] S. Ray, A. Kundu, T. Na, Institute of Electrical and Electronics Engineers. Kolkata Section, and Institute of Electrical and Electronics Engineers, *CALCON 2020 : 2020 IEEE Calcutta Conference : proceedings : 28-29 February, 2020, Kolkata, India. .*
- [9] A. Syaikhul, M. Aziz, and O. Virgantara, "Disease Detection in Rice Leaf Plants Using Convolutional Neural Network (Cnn) Method," 2021.
- [10] C. R. Rahman *et al.*, "Identification and recognition of rice diseases and pests using convolutional neural networks," *Biosyst. Eng.*, vol. 194, pp. 112–120, 2020, doi: 10.1016/j.biosystemseng.2020.03.020.
- [11] R. Indraswari, R. Rokhana, and W. Herulambang, "Melanoma image classification based on MobileNetV2 network," in *Procedia Computer Science*, 2021, vol. 197, pp. 198–207, doi: 10.1016/j.procs.2021.12.132.
- [12] K. Rahouma and A. Salama, "Knee Images Classification using Transfer Learning," in *Procedia Computer Science*, 2021, vol. 194, pp. 9–21, doi: 10.1016/j.procs.2021.10.055.
- [13] T. Selim, I. Elkabani, and M. A. Abdou, "Students Engagement Level Detection in Online e-Learning Using Hybrid EfficientNetB7 Together With TCN, LSTM, and Bi-LSTM," *IEEE Access*, vol. 10, no. August, pp. 99573–99583, 2022, doi: 10.1109/ACCESS.2022.3206779.
- [14] A. Bari, T. Saini, and A. Kumar, "Fire detection using deep transfer learning on surveillance videos," *Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mob.*

- Networks, ICICV 2021*, no. Icicv, pp. 1061–1067, 2021, doi: 10.1109/ICICV50876.2021.9388485.
- [15] S. Sharma and K. Guleria, “A Deep Learning based model for the Detection of Pneumonia from Chest X-Ray Images using VGG-16 and Neural Networks,” *Procedia Comput. Sci.*, vol. 218, pp. 357–366, 2023, doi: 10.1016/j.procs.2023.01.018.
- [16] D. Saranyaraj, M. Manikandan, and S. Maheswari, “A deep convolutional neural network for the early detection of breast carcinoma with respect to hyper- parameter tuning,” *Multimed. Tools Appl.*, vol. 79, no. 15–16, pp. 11013–11038, 2020, doi: 10.1007/s11042-018-6560-x.
- [17] W. Sae-Lim, W. Wettayaprasit, and P. Aiyarak, “Convolutional Neural Networks Using MobileNet for Skin Lesion Classification,” *JCSSE 2019 - 16th Int. Jt. Conf. Comput. Sci. Softw. Eng. Knowl. Evol. Toward. Singul. Man-Machine Intell.*, pp. 242–247, 2019, doi: 10.1109/JCSSE.2019.8864155.
- [18] M. Fikri, N. Syahbani, and N. G. Ramadhan, “Klasifikasi Gerakan Yoga dengan Model Convolutional Neural Network Menggunakan Framework Streamlit,” vol. 7, pp. 509–519, 2023, doi: 10.30865/mib.v7i1.5520.
- [19] M. Arbane, R. Benlamri, Y. Brik, and M. Djerioui, “Transfer Learning for Automatic Brain Tumor Classification Using MRI Images,” *2020 2nd Int. Work. Human-Centric Smart Environ. Heal. Well-Being, IHSH 2020*, pp. 210–214, 2021, doi: 10.1109/IHSH51661.2021.9378739.
- [20] Universiti Teknologi PETRONAS. Computer and Information Sciences Department and Institute of Electrical and Electronics Engineers, *2020 International Conference on Computational Intelligence (ICCI) : proceedings : first virtual conference by Computer and Information Sciences Department (CISD), Universiti Teknologi PETRONAS (UTP), 8th-9th October 2020*. .
- [21] E. Villain, G. M. Mattia, F. Nemmi, P. Peran, X. Franceries, and M. V. Le Lann, “Visual interpretation of CNN decision-making process using Simulated Brain MRI,” in *Proceedings - IEEE Symposium on Computer-Based Medical Systems*, Jun. 2021, vol. 2021-June, pp. 515–520, doi: 10.1109/CBMS52027.2021.00102.
- [22] S. D. Deb, R. K. Jha, K. Jha, and P. S. Tripathi, “A multi model ensemble based deep convolution neural network structure for detection of COVID19,” *Biomed. Signal Process. Control*, vol. 71, Jan. 2022, doi: 10.1016/j.bspc.2021.103126.
- [23] Institute of Electrical and Electronics Engineers, *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. .
- [24] Institute of Electrical and Electronics Engineers. Turkey Section. and Institute of Electrical and Electronics Engineers, *HORA 2020 : 2nd International Congress on Human-Computer Interaction, Optimization and Robotic Applications : proceedings : June 26-27, 2020, Turkey*. .
- [25] S. Sharma, K. Guleria, S. Tiwari, and S. Kumar, “A deep learning based convolutional neural network model with VGG16 feature extractor for the detection of Alzheimer Disease using MRI scans,” *Meas. Sensors*, vol. 24, Dec. 2022, doi: 10.1016/j.measen.2022.100506.
- [26] K. Shankar, Y. Zhang, Y. Liu, L. Wu, and C. H. Chen, “Hyperparameter Tuning Deep Learning for Diabetic Retinopathy Fundus Image Classification,” *IEEE Access*, vol. 8, pp. 118164–118173, 2020, doi: 10.1109/ACCESS.2020.3005152.
- [27] F. F. Firdaus, H. A. Nugroho, and I. Soesanti, “Deep Neural Network with Hyperparameter Tuning for Detection of Heart Disease,” in *Proceedings - 2021 IEEE Asia Pacific Conference on Wireless and Mobile, APWiMob 2021*, Apr. 2021, pp. 59–65, doi: 10.1109/APWiMob51111.2021.9435250.