

Klasifikasi Hama Tanaman Padi Menggunakan Metode *Convolutional Neural Network* (CNN)

Melki Garonga^{1*}, Yusni salissing²

^{1*}Program Studi Teknik Informatika, Universitas Kristen Indonesia Toraja, Tana Toraja, Sulawesi Selatan

Email: [1*melkigaronga@ukitoraja.ac.id](mailto:melkigaronga@ukitoraja.ac.id), [2yusnisalissing16@gmail.com](mailto:yusnisalissing16@gmail.com)

Abstrak

Padi merupakan tanaman pangan yang banyak ditanam oleh para petani karena padi salah satu makanan pokok masyarakat Indonesia. Kualitas padi dipengaruhi oleh berbagai faktor, salah satunya adalah serangan hama. Dalam sistem penilaian hama tanaman padi, semakin banyak tanaman padi yang diserang oleh hama maka semakin menurun pula produksi dari tanaman padi. Oleh karena itu, untuk membantu para petani dalam klasifikasi hama tanaman padi diperlukan pemodelan yang dapat diimplementasikan pada mesin berbasis kecerdasan buatan. Salah satu metode yang dapat digunakan adalah metode *Convolutional Neural Network* (CNN), dengan model *MobilNetv2* yang digunakan dalam proses *training* dengan *input image* sebesar 224 x 224 piksel dengan data *training* yang digunakan sebanyak 960 citra, data validasi sebanyak 120 dan data testing sebanyak 120 citra dari tiga kelas menggunakan ruang warna RGB, yang diuji menggunakan *Confusion Matrix*. Adapun nilai akurasi yang diperoleh pada ruang warna RGB menggunakan *epoch* 140 dan *batch_size* 64 adalah 99% dengan *loss* sebesar 0.0140.

Kata Kunci : Padi, Klasifikasi, CNN, *Confusion Matrix*.

Classification Of Rice Plant Pests Using The Convolutional Neural Network (CNN) Method

Abstract

Rice is a food crop that is widely grown by farmers because rice is one of the staple foods of Indonesian society. Paddy quality is influenced by various factors, one of which is pest attack. In the rice plant pest assessment system, the more rice plants that are attacked by pests, the lower the production of rice plants yesterday. Therefore, to assist farmers in classifying rice plant pests, modeling is needed that can be implemented on artificial intelligence-based machines. One method that can be used is the *Convolutional Neural Network* (CNN) method, with the *MobilNetv2* model used in the training process with an input image of 224 x 224 pixels with 960 images of training data, 120 validation data and 120 testing data. of the three classes using the RGB color space, which was tested using the *Confusion Matrix*. The accuracy value obtained in the RGB color space using epoch 140 and batch_size 64 is 99% with a loss of 0.0140.

Keywords : Rice, classification, CNN, *Confusion Matrix*.

I. PENDAHULUAN

Makanan merupakan hasil dari pertanian, dimana setiap tahun kebutuhan akan pangan semakin meningkat karena populasi manusia yang terus bertambah. Padi merupakan tanaman pangan yang paling banyak ditanam oleh petani karena merupakan salah satu makanan pokok masyarakat Indonesia. Beras merupakan hasil dari tanaman padi yang digunakan sebagai makanan pokok dari manusia. Untuk memperoleh hasil dari tanaman padi, petani harus menunggu sampai masa panen tiba. Berdasarkan umur tanaman padi dikategorikan dalam umur genjah (sekitar 110 hari) dan lebih dari 120 hari.

Dalam proses bertani petani sering mengalami gagal panen, sehingga hasil panen padi menjadi menurun. Penyebab hasil panen menjadi menurun biasanya disebabkan oleh hama yang menyerang tanaman padi. Hama yang menyerang tanaman padi pada umumnya akan menunjukkan gejala-gejala serangan yang masih ringan. Namun petani sering mengabaikan hal ini dan karena keterlambatan menangani masalah ini sehingga serangan hama dapat mencapai pada tahap yang meluas. Hama adalah salah satu bentuk kendala yang dihadapi oleh para petani dalam berproduksi. Hama pada tanaman padi dapat berupa belalang, ulat dan hama putih palsu. Banyaknya serangan hama tanaman padi terutama terjadi selama tahap

pertumbuhan vegetatif, dimana daun dan pembentukan rumpun terjadi pada padi, sehingga jika ada serangan hama terutama pada tahap ini, maka perkembangan tanaman padi akan terhambat. Hama pada tanaman padi dapat diidentifikasi dengan memahami gejala yang muncul khususnya pada bagian daun, sehingga serangan hama dapat ditanggulangi sedini mungkin[1]. memudahkan dalam pendeteksian dini terhadap hama tanaman padi khususnya bagi para petani dan bagi kepentingan dalam industri pertanian agar serangan hama tidak mencapai tahap yang lebih parah dan meluas, apalagi kemajuan teknologi saat ini *artificial intelligence* atau kecerdasan buatan sering diperbincangkan dalam beberapa tahun belakangan ini. Sektor industri adalah salah satu sektor yang dapat mengimplementasikan kecerdasan buatan. Sehingga dengan menggunakan teknologi ini dapat diketahui hama tanaman padi melalui citra daun dengan membaginya kedalam beberapa kategori. Sehingga dibutuhkan kemampuan dalam pengolahan klasifikasi citra, yang memiliki kemampuan dalam pengolahan citra ini salah satunya yaitu teknik *deep learning* dengan metode *Convolutional Neural Network* (CNN). Metode CNN mampu menghasilkan tingkat akurasi yang signifikan karena memiliki kedalaman jaringan dan telah banyak diaplikasikan pada data gambar.

Penelitian ini bertujuan untuk mendapatkan informasi terkait dengan jenis-jenis hama yang dapat menghambat pertumbuhan dan perkembangan tanaman padi. Dengan adanya penelitian ini diharapkan serangan hama pada tanaman padi dapat dapat ditanggulangi sedini mungkin. Maka, pada penelitian ini diusulkan topik “Klasifikasi Hama Tanaman Padi Menggunakan Metode *Convolutional Neural Network* (CNN)”.

A. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka terdapat rumusan masalah yaitu:

1. Bagaimana model *Convolutional Neural Network* dalam klasifikasi hama tanaman padi
2. Bagaimana tingkat akurasi algoritma *Convolution Neural Network* (CNN) dalam melakukan klasifikasi hama tanaman padi.

B. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

1. Memodelkan algoritma CNN dalam mengklasifikasikan hama tanaman padi.
2. Mengetahui tingkat akurasi algoritma CNN dalam klasifikasi hama tanaman padi.

II. TINJAUAN PUSTAKA

A. Landasan Teori

1). Tanaman Padi

Tanaman padi adalah salah satu jenis tanaman yang menghasilkan beras yang merupakan sumber karbohidrat bagi sebagian besar penduduk Indonesia. Sebagian besar penduduk Indonesia mengkonsumsi beras sebagai makanan pokok. Hal ini mengakibatkan setiap tahunnya

permintaan terhadap kebutuhan beras terus meningkat seiring dengan bertambahnya jumlah penduduk. Menurut Poedjiadi (1994), kandungan karbohidrat padi giling sebesar 78,9%, protein 6,8%, lemak 0,7% dan lain-lain 0,6

2). Jenis-jenis Hama Tanaman Padi

a. Belalang

Belalang pengembara (*Locusta migratoria*, L) merupakan salah satu hama penting di Indonesia karena hama ini merupakan salah satu faktor penghambat dalam rencana peningkatan hasil panen

b. Ulat

Hama tanaman padi selanjutnya adalah ulat. Biasanya ulat memakan daun dan batang tanaman.

c. Hama Putih Palsu

Hama ini dianggap sebagai hama putih palsu lantaran tanda-tanda serangannya hampir menyerupai tanda-tanda agresi hama putih. Walaupun hama putih palsu (*Cnaphalocrosis medinalis*) bukan hama primer dan hama yang membahayakan bagi tumbuhan padi akan namun agresi hama putih palsu permanen akan berdampak merugikan bagi petani. Dari pengalaman maspariy agresi hama putih palsu terjadi dalam ketika tumbuhan masih pada fase vegetatif (tumbuhan muda) walaupun tidak menutup kemungkinan juga kadang terjadi ketika tumbuhan telah keluar malai. Dan biasanya sebagai agresi yg berarti apabila kerusakan dalam daun terjadi ketika padi memasuki fase anakan maksimum dan fase pematangan mencapai > 50%.

3). Deep Learning

Deep Learning yaitu bagian dari kecerdasan buatan atau Machine Learning yang menggunakan jaringan saraf tiruan (*artificial neural network*) untuk mengimplementasikan deretan data yang akbar, teknik *deep learning* sangat populer di kalangan praktisi data serta menarik perhatian banyak pihak. Hal ini disebabkan karena teknologi deep learning telah diterapkan dalam aneka macam produk berteknologi tinggi mirip *self-driving car*.

4). Convolutional Neural Network

Convolutional Neural network (CNN) adalah salah satu jenis *neural community* yang biasa digunakan pada *facts photograph*. CNN bisa digunakan untuk mendeteksi dan mengenali *object* pada sebuah *photograph*. Secara garis besar *Convolutional Neural community* (CNN) tidak jauh beda dengan *neural network* biasanya. CNN terdiri dari neuron yang memiliki *weight*, *bias* dan *activation characteristic*. *Convolutional layer* juga terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah *filter out* dengan panjang dan tinggi (*pixels*)

5). Pengolahan Citra

Pengolahan citra adalah pengolahan citra dengan menggunakan komputer sesuai keinginan pemakainya, dengan tujuan untuk meningkatkan kualitas citra agar mudah diinterpretasikan oleh manusia atau mesin. Kualitas gambar yang Anda miliki sering menurun. Penurunan kualitas yang sering terjadi adalah adanya noise atau ketidaksempurnaan pada gambar, seperti bintik-bintik, kurang fokus, gambar buram, dan warna yang tidak rata.

6). Python

Python adalah bahasa pemrograman yang sifatnya open source. Bahasa pemrograman ini dioptimalisasikan untuk *component integration developer productivity, software quality* dan program *portability*. Python sudah banyak digunakan dalam pengembangan berbagai perangkat lunak, seperti *user interface, numeric programing, product customization, system programming* dan *internet scripting*.

7). Confusion Matrix

Penelitian ini menggunakan *Confusion Matrix* yang disebut *error matrix* dalam pengujian tingkat akurasi. *Confusion matrix* adalah sebuah tabel yang menyatakan jumlah data uji benar dan salah di klasifikasi. Dalam *Confusion matrix* terdapat 4 istilah yang digunakan sebagai representasi hasil proses klasifikasi antara lain: *True Negatif* (TN) yaitu data negatif yang diprediksi benar. *True Positif* (TP) yaitu data positif yang diprediksi benar. *False Negatif* (FN) yaitu positif namun diprediksi sebagai data negatif, yang kemungkinan seluruh kejadian sebenarnya positif (P) dan kemungkinan seluruh kejadian sebenarnya negatif (N). *False Positif* (FP) yaitu data negatif tetapi diprediksi sebagai data positif.

B. Penelitian Terkait

Kusumaningrum (2018), melakukan penelitian dengan judul implementasi *Convolutional Neural Network* (CNN) untuk klasifikasi jamur konsumsi di Indonesia menggunakan keras. Penelitian ini melakukan klasifikasi terhadap jamur konsumsi dengan mengklasifikasikan 3 jenis jamur dengan menggunakan metode CNN dan memperoleh hasil akurasi 100%[2].

Hidayat dan Hermawan (2018), melakukan penelitian dengan judul deteksi hama pada daun teh dengan metode *Convolutional Neural Network* (CNN). Penelitian ini melakukan klasifikasi berdasarkan 2 jenis hama yaitu *helopeltis* dan *blister blight* pada citra daun teh[3].

Wicaksono, Benrahman dan Andrayana (2020), melakukan penelitian dengan judul aplikasi pendeteksi penyakit daun tanaman apel dengan menggunakan metode *Convolutional Neural Network* (CNN). Dalam penelitian ini mengimplementasikan teknik *deep learning* dengan metode CNN untuk mendeteksi penyakit pada tanaman daun apel [4].

Hengki dan Tamando Sihotang, H. (2018), melakukan penelitian dengan judul diagnosa hama dan penyakit tanaman padi menggunakan metode bayes. Dengan menggunakan *teorema bayes* dapat mengidentifikasi hama dan penyakit tanaman padi dengan menggunakan persentase gejala tanaman [5].

Husrina, Ismail, Musyriah (2021), melakukan penelitian dengan judul sistem pakar mendiagnosis penyakit dan hama tanaman padi menggunakan *Theorema Bayes* [6].

Siti rahmah (2020), melakukan penelitian dengan judul implementasi algoritma *Convolutional Neural Network* dalam klasifikasi penyakit padi melalui citra daun [7].

III. METODE PENELITIAN

A. Waktu dan Lokasi Penelitian

Adapun Waktu penelitian ini dimulai dari bulan Maret sampai bulan Mei 2023 dengan lokasi di Lembang Issong Kalua' Kecamatan Buntao' Kabupaten Toraja Utara

B. Alat dan Bahan

Bahan dan alat pada penelitian ini antara lain:

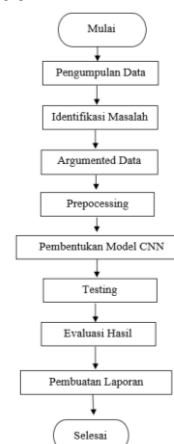
1). Alat

Alat yang digunakan yaitu : Laptop dengan sistem operasi *Windows 10, Software Python* dan Kamera.

2). Bahan

Adapun bahan yang akan digunakan dalam penelitian ini adalah citra daun tanaman padi.

C. Tahapan Penelitian



Gambar 1. Tahapan Penelitian

IV. HASIL DAN PEMBAHASAN

A. Pengumpulan Dataset

Pengumpulan dataset dilakukan secara langsung di Lembang Issong Kalua', Kecamatan Buntao', Kabupaten Toraja Utara. Dataset digunakan sebagai pengujian terhadap data *training*, yaitu pengambilan citra tanaman padi secara langsung. Setelah daun padi yang terkena hama terkumpul, selanjutnya dilakukan proses pengambilan citra dengan dua kondisi, yaitu kondisi dalam ruangan dan luar ruangan menggunakan kamera *smartphone android* dengan resolusi kamera 48MP dan rasio 1:1 serta menggunakan alas berwarna putih. Total dataset yaitu sebanyak 1.200

citra yang terdiri dari 300 citra hama putih palsu, 300 citra belalang, 300 citra hama ulat, dan 300 citra tanpa hama.



Gambar 2. Hama Putih Palsu



Gambar 3. Hama Ulat



Gambar 4. Hama Belalang



Gambar 5. Tanpa Hama

Setelah proses pengambilan citra selesai maka citra akan dipisahkan berdasarkan kelasnya, kemudian mengubah nama file citra yang telah dipisahkan. Setelah nama semua file diubah maka akan dilakukan pembagian dataset yang akan dibagi menjadi 2 bagian data, yang terdiri dari data *training* dan data *testing*. berikut ini dapat dilihat pembagian data *training* dan data *testing*.

Kelas	Training	Testing	Validasi
Hama putih palsu	240	30	30
Hama ulat	240	30	30
Hama belalang	240	30	30
Tanpa hama	240	30	30
Total	960	120	120

Setelah proses pembagian citra dataset selesai, selanjutnya akan dipanggil ke halaman page jupyter dengan nama subfolder “dataset” untuk diproses ke tahap selanjutnya dengan bantuan *Jupyter*.

C. Preprocessing

Dalam tahapan ini tahapan pertama yg dilakukan yaitu dataset citra dipanggil ke page *jupyter* selanjutnya akan masuk kedalam tahapan awal yaitu tahapan *preprocessing*, tahapan awal yaitu menghubungkan *jupyter* dengan dataset yang ada di komputer dan membagi ke dalam beberapa *class*

```
In [213]: os.listdir('E:\Dataset')
data all dir - os.path.join('E:\Dataset\belalang', 'data_all')
print ('Jumlah total data tiap kelas : ')
print (df['labels'].value_counts())

Jumlah total data tiap kelas :
belalang      300
hama putih palsu 300
hama ulat      300
tanpa hama     300
Name: labels, dtype: int64
```

Gambar 6. Proses pemanggilan data set dari komputer ke *Jupyter*

Setelah dataset sudah dipanggil ke *Jupyter*, selanjutnya dalam tahapan *preprocessing* data yang ada akan dilakukan proses pembagian data yaitu data *train*, data *valid* dan data *test*.

```
In [214]: train_split=0.8
test_split=0.1
dummy_split=test_split/(1-train_split)
train_df, dummy_df=train_test_split(df, train_size=train_split, shuffle=True, random_state=123)
test_df, valid_df=train_test_split(dummy_df, train_size=dummy_split, shuffle=True, random_state=123)
print ('data train : ', len(train_df)),
print ('data test : ', len(test_df)),
print ('data valid : ', len(valid_df))

data train: 960
data test : 120
data valid: 120
```

Gambar 7 Proses Pembagian data *train*, data *valid* dan data *test*

Setelah proses pembagian data selesai selanjutnya dalam tahapan *preprocessing* data yang ada akan dilakukan proses penyamaan ukuran dan ruang warna atau mengubah komponen atau susunan warna menjadi RGB, setelah itu mengubah ukuran atau *resize dataset* citra yaitu 224 x 224 piksel sesuai ukuran *inputan layer* CNN *mobilenetv2* serta normalisasi dilakukan dengan melakukan *rescale* pada semua citra yang ada dengan membagi nilai piksel yang ada pada semua citra dengan 255.

```
In [319]: height=224
width=224
channels=3
batch_size=64

img_shape=(height, width, channels)
img_size=(height, width)
length=len(test_df)
test_batch_size=sorted([int(length/n) for n in range(1,length+1) if length % n ==0 and length/n<80,reverse=True])[0]
test_steps=int(length/test_batch_size)
print ('test batch size: ', test_batch_size, ' test steps: ', test_steps)

test batch size: 60 test steps: 2
```

Gambar 8 Proses *resize* normalisasi

Setelah melakukan *resize* normalisasi akan dilanjutkan ke tahapan normalisasi RGB yang ada dalam dataset.

```
In [320]: gen_image_datagenerator(
rescale=1./255,
train_gen=gen_from_dataframes(train_df, x_col='filepath', y_col='labels', target_size=img_size, class_mode='categorical',
color_mode='rgb', shuffle=True, batch_size=batch_size),
valid_gen=gen_image_datagenerator(rescale=1./255,
valid_df, x_col='filepath', y_col='labels', target_size=img_size, class_mode='categorical',
color_mode='rgb', shuffle=False, batch_size=batch_size),
test_gen=gen_image_datagenerator(rescale=1./255,
test_df, x_col='filepath', y_col='labels', target_size=img_size, class_mode='categorical',
color_mode='rgb', shuffle=False, batch_size=batch_size))

classes=list(train_gen.class_indices.keys())
print (classes)
class_count=len(classes)

Found 960 validated image filenames belonging to 4 classes.
Found 120 validated image filenames belonging to 4 classes.
Found 120 validated image filenames belonging to 4 classes.
['belalang', 'hama putih palsu', 'hama ulat', 'tanpa hama']
```

Gambar 9. Proses normalisasi RGB

Dalam *preprocessing* setelah normalisasi akan dilanjutkan ke tahapan berikutnya yaitu menampilkan image yang ada dalam dataset.

```
In [6]: def show_image_samples(gen):
test_dict=test_gen.class_indices
classes=list(test_dict.keys())
images, labels=next(gen) # get a sample batch from the generator
plt.figure(figsize=(20, 20))
length=len(labels)
if length>25: #show maximum of 25 images
r=length
else:
r=25
for i in range(r):
plt.subplot(5, 5, i + 1)
images=images[i]
plt.imshow(images)
index=np.argmax(labels[i])
class_name=classes[index]
plt.title(class_name, color='blue', fontsize=10)
plt.axis('off')
plt.show()
```

Gambar 10. Tahapan untuk menampilkan image dalam dataset



Gambar 11. Hama Tanaman Padi Dan Tanpa Hama RGB

D. Pre-Training Model MobileNetv2

Dalam penelitian ini setelah beberapa tahap yang dilakukan mulai dari pengumpulan data, preprocessing maka tahapan selanjutnya yaitu *pre-training* model MobileNetv2.

```
In [12]: base_model=tf.keras.applications.MobileNetV2(include_top=False, weights='imagenet', input_tensor=Input(shape=(224,224,3)))
base_model.summary()

WARNING:tensorflow: Input_shape is undefined or non-square, or 'rows' is not in [96, 128, 160, 192, 224]. Weights for Input shape (224, 224) will be loaded as the default.
Model: "mobilenetv2_1.00_224"
Layer (type) Output Shape Param # Connected to
-----
input_3 (InputLayer) [(None, 224, 224, 3)] 0 []
conv1 (Conv2D) (None, 112, 112, 32) 864 ['input_3[0][0]']
bn_conv1 (BatchNormalization) (None, 112, 112, 32) 128 ['conv1[0][0]']
conv1_relu (ReLU) (None, 112, 112, 32) 0 ['bn_conv1[0][0]']
expanded_conv_depthwise (Depthwise Conv2D) (None, 112, 112, 32) 288 ['conv1_relu[0][0]']
```

Gambar 12. Pembuatan Model MobileNetv2

```
In [12]: model_name='Cnn'
print('Building model with', base_model)
model = tf.keras.Sequential([
# Note the input shape is the desired size of the image 224x224 with 3 bytes color
# This is the first convolution
base_model,
tf.keras.layers.Conv2D(filters=32, padding='same', kernel_size=3, activation='relu', strides=1),
tf.keras.layers.MaxPool2D(pool_size=2, strides=2),
tf.keras.layers.Dropout(rate=0.5),

tf.keras.layers.Flatten(),
tf.keras.layers.Dense(4, activation='softmax')
])

model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics='accuracy')
```

Gambar 13 Proses Penambahan Model

Layer (type)	Output Shape	Param #
mobilenetv2_1.00_224 (Functional)	(None, 7, 7, 1280)	2257984
conv2d_6 (Conv2D)	(None, 7, 7, 32)	368672
max_pooling2d_2 (MaxPooling2D)	(None, 3, 3, 32)	0
dropout_2 (Dropout)	(None, 3, 3, 32)	0
flatten_2 (Flatten)	(None, 288)	0
dense_2 (Dense)	(None, 4)	1156
Total params: 2,627,812		
Trainable params: 369,828		
Non-trainable params: 2,257,984		

Gambar 14. Output Model

Jumlah keseluruhan parameter yang didapat pada proses penambahan model sebanyak 2,627,812 dengan jumlah parameter yang tidak dapat dilatih sebanyak 2,257,984 dan jumlah parameter yang dapat dilatih sebanyak 369,228.

E. Proses Training

Dalam proses *training* parameter yang dipakai terdiri dari *validation split* sebesar 0.2, sebanyak 240 citra 20% data *training* yang akan digunakan untuk menguji performa model. Selanjutnya *epoch* yang digunakan *epoch* 140. Dalam hal ini *epoch* adalah banyaknya proses pengulangan atau iterasi dalam *training*. *Batch_size* yang digunakan dalam penelitian ini *batch_size* 64 dengan *verbose* 1 untuk melihat progres sudah epoch berapa dan step berapa. *Batch_size* adalah jumlah *training* citra yang digunakan dalam proses *training*. Proses *training* dapat dilihat pada gambar 15.

```
In [13]: %time
epochs =140

history=model.fit(x=train_gen, epochs=epochs, validation_data=valid_gen)
```

Gambar 15 Proses Training

Hasil proses *training* menggunakan *batch_size* 64 *epoch* 140 pada gambar berikut:

```
Epoch 135/140
15/15 [=====] - 34s 2s/step - loss: 0.0025 - accuracy: 0.9990 - val_loss: 0.1548 - val_accuracy: 0.9833
Epoch 136/140
15/15 [=====] - 34s 2s/step - loss: 0.0031 - accuracy: 0.9970 - val_loss: 0.0350 - val_accuracy: 0.9917
Epoch 137/140
15/15 [=====] - 34s 2s/step - loss: 0.0201 - accuracy: 0.9958 - val_loss: 0.0798 - val_accuracy: 0.9833
Epoch 138/140
15/15 [=====] - 34s 2s/step - loss: 0.0067 - accuracy: 0.9980 - val_loss: 0.1154 - val_accuracy: 0.9917
Epoch 139/140
15/15 [=====] - 35s 2s/step - loss: 0.0132 - accuracy: 0.9940 - val_loss: 0.2051 - val_accuracy: 0.9667
Epoch 140/140
15/15 [=====] - 34s 2s/step - loss: 0.0463 - accuracy: 0.9917 - val_loss: 0.2223 - val_accuracy: 0.9758
Wall time: 1h 45min 52s
```

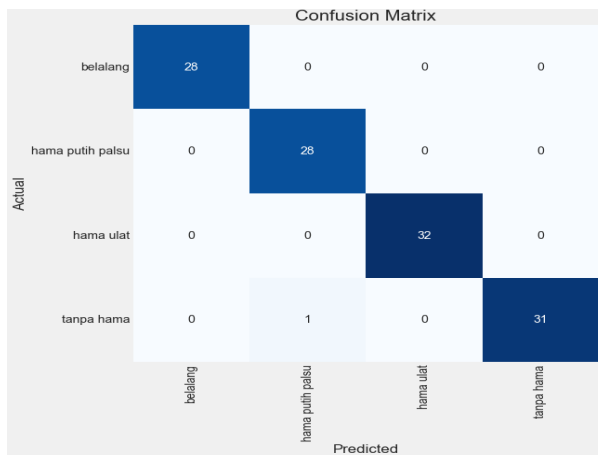
Gambar 16. Hasil Proses Training Ruang Warna RGB

Setelah proses training selesai dengan algoritma *Convolutional Neural Network* (CNN) menggunakan epoch 140. Grafik hasil *training* menggunakan *matplotlib* dapat dilihat pada gambar 4.16.



Gambar 17. Grafik Pelatihan Ruang Warna RGB Epoch 140

Pada penelitian ini setelah proses *training* dilakukan, maka tahapan selanjutnya yaitu menyimpan model pada format “h5”. Model yang telah disimpan tersebut akan digunakan untuk menguji data *testing*. Setelah penyimpanan model selesai selanjutnya melakukan proses *testing* menggunakan *confusion matrix*. Hasil proses *testing* dapat dilihat sebagai berikut.



Gambar 18. *Confusion Matrix* MobilNetv2 Epoch 140 Batch_size 64

hasil klasifikasi:

	precision	recall	f1-score	support
belalang	1.00	1.00	1.00	28
hama putih palsu	0.97	1.00	0.98	28
hama ulat	1.00	1.00	1.00	32
tanpa hama	1.00	0.97	0.98	32
accuracy			0.99	120
macro avg	0.99	0.99	0.99	120
weighted avg	0.99	0.99	0.99	120

Berdasarkan gambar 17 diatas perhitungan akurasi *matrix*, *presisi*, *recall* dan *F1 score* dapat dihitung seperti :

a). Hama Belalang

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} = \frac{119}{120} = 0,99$$

$$Presisi = \frac{TP}{TP + FP} = \frac{28}{28} = 1.00$$

$$Recall = \frac{TP}{TP + FN} = \frac{28}{28} = 1.00$$

$$F1_Score = 2 \times \frac{presisi \times Recall}{Presisi + Recall} = 2 \times \frac{1.00 \times 1.00}{1.00 + 1.00} = \frac{1}{2}$$

b). Hama Putih Palsu

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} = \frac{119}{120} = 0,99$$

$$Presisi = \frac{TP}{TP + FP} = \frac{28}{29} = 0.97$$

$$Recall = \frac{TP}{TP + FN} = \frac{28}{28} = 1.00$$

$$F1_Score = 2 \times \frac{presisi \times Recall}{Presisi + Recall} = 2 \times \frac{0.97 \times 1.00}{0.97 + 1.00} = \frac{1.97}{1.97} = 0.98$$

c). Hama Ulat

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} = \frac{119}{120} = 0,99$$

$$Presisi = \frac{TP}{TP + FP} = \frac{32}{32} = 1.00$$

$$Recall = \frac{TP}{TP + FN} = \frac{32}{32} = 1.00$$

$$F1_Score = 2 \times \frac{presisi \times Recall}{Presisi + Recall} = 2 \times \frac{1.00 \times 1.00}{1.00 + 1.00} = \frac{1}{2}$$

d). Tanpa Hama

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} = \frac{119}{120} = 0,99$$

$$Presisi = \frac{TP}{TP + FP} = \frac{31}{31} = 1.00$$

$$Recall = \frac{TP}{TP + FN} = \frac{31}{32} = 0.97$$

$$F1_Score = 2 \times \frac{presisi \times Recall}{Presisi + Recall} = 2 \times \frac{0.97 \times 1.00}{0.97 + 1.00} = \frac{1.97}{1.97} = 0.98$$

F. Pembahasan

Dalam penelitian ini untuk mengklasifikasikan jenis hama tanaman padi dilakukan beberapa proses yaitu proses *Training* (Pelatihan) dan proses *Testing* (Pengujian). Pada proses training parameter yang digunakan terdiri dari *Learning Rate* 0,001, dimana laju pembelajaran ini merupakan salah satu parameter pelatihan yang digunakan untuk menghitung nilai koneksi bobot pada saat proses pelatihan dilakukan. Semakin kecil laju pembelajarannya maka ketelitian jaringan akan semakin besar, begitupun sebaliknya semakin besar laju pembelajarannya maka ketelitian jaringannya semakin kecil. Kemudian *epoch* yang digunakan dalam penelitian ini adalah *epoch* 140, dimana *epoch* ini adalah banyaknya iterasi atau pengulangan dalam proses *Training* serta menggunakan *batch_size* 15, 25, 35, 45, 64. *Batch_size* adalah banyaknya citra yang akan digunakan dalam satu kali proses iterasi serta *verbose_1* yang berfungsi untuk menampilkan setiap proses dan hasil. Dalam penelitian ini, pelatihan dilakukan sebanyak 5 kali percobaan dengan mengubah nilai *batch_size* dan *epochnya*. Dari tabel 4.2 diatas Proses Pelatihan dilakukan selama 5 kali percobaan untuk model *MobileNetv2* dengan menggunakan parameter laju pembelajaran 0,001. Dari hasil proses *training* menggunakan *batch_size* 64, *epoch* 140 maka didapatkan akurasi sebesar 99% dan loss 0,0150.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Pada penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) dengan model *MobilNetv2* yang digunakan dalam proses *training* dengan *input image* sebesar 224 x 224 piksel dengan data *training* yang digunakan sebanyak 960 citra, data validasi sebanyak 120 dan data testing sebanyak 120 citra dari tiga kelas menggunakan ruang warna RGB, yang diuji menggunakan *confusion matrix*.
2. Nilai akurasi diperoleh pada ruang warna RGB dengan *epoch* 140 *batch_size* 64 diperoleh akurasi sebanyak 99% dengan *loss* 0.0140.

B. Saran

Berdasarkan penelitian yang dilakukan dalam pembuatan model ini perlu diadakan pengembangan, karena pada penelitian ini masi banyak kekurangan. Adapun saran yang penulis berikan yaitu:

1. Bisa merancang dengan menggunakan model berbeda
2. Dalam pengambilan citra masih terbatas diperlukan penambahan variasi dataset yang lebih banyak lagi.

REFERENSI

- [1] E. Antika and R. Jannah Desy Sabatini, "Klasifikasi Hama Tanaman Padi Berdasarkan Gejala Yang Timbul Pada Fase Vegetatif," *Jurnal Teknologi Informasi Politeknik Negeri Jember*, vol. 1, no. 1. pp. 116–120, 2015.
- [2] H. Shim *et al.*, "Implementasi Convolution Neural Network (Cnn) Untuk Klasifikasi Jamur Konsumsi Di Indonesia Menggunakan Keras," *Advanced Optical Materials*, vol. 10, no. 1. pp. 1–9, 2018. [Online]. Available: <https://doi.org/10.1103/PhysRevB.101.089902>0Ahttp://dx.doi.org/10.1016/j.nantod.2015.04.009%0Ahttp://dx.doi.org/10.1038/s41467-018-05514-9%0Ahttp://dx.doi.org/10.1038/s41467-019-13856-1%0Ahttp://dx.doi.org/10.1038/s41467-020-14365-2%0Ahttp://dx.doi.org/1
- [3] "Deteksi Hama Pada Daun Teh Dengan Metode Convolutional Neural Network (Cnn) - Repository UNIKOM."
- [4] G. Wicaksono, S. Andryana, and B. -, "Aplikasi Pendeteksi Penyakit Pada Daun Tanaman Apel Dengan Metode Convolutional Neural Network," *JOINTECS (Journal of Information Technology and Computer Science)*, vol. 5, no. 1. p. 9, 2020. doi: 10.31328/jointecs.v5i1.1221.
- [5] Husrina, Ismail, Musyriifah , "D iagnosa Hama dan Penyakit Tanaman Padi Menggunakan M etode Bayes," no. November, 2021.
- [6] Hengki dan Tamando Sihotang, H. , "diagnosa hama dan penyakit tanaman padi menggunakan metode bayes " 2018.
- [7] 2020 Siti,tugas akhir, B. L. Blight, B. Spot, and L. Spot, "implementasi algoritma convolutional neural network pada klasifikasi penyakit padi melalui citra daun," 2020.