

Implementasi *Deep Learning* Menggunakan *Vision Transformer* Untuk Klasifikasi Penyakit Daun Padi

Tri Febriyanto¹, Suzuki Syofian^{2*}

^{1,2} Program Studi Teknologi Informasi Fakultas Teknik, Universitas Darma Persada
Jl. Taman Malaka Selatan No.22, Pondok Kelapa, Duren Sawit, DKI Jakarta, Indonesia 13450

*suzukiunsada@gmail.com

Abstrak — Padi merupakan makanan pokok penting yang berperan signifikan dalam meningkatkan kesejahteraan masyarakat Indonesia. Namun, produksi padi sering terancam oleh berbagai penyakit daun yang disebabkan oleh patogen seperti jamur, hama, bakteri, dan virus. Beberapa penyakit daun padi yang umum termasuk Blas (*Blast*), Bercak Coklat (Brown Spot), Hawar Daun Bakteri (*Bacterial Leaf Blight*), dan Tungro. Kemajuan teknologi saat ini, khususnya dalam bidang *Deep Learning*, menawarkan solusi potensial untuk mengatasi tantangan tersebut. *Deep Learning*, sebagai sub-bidang *Machine Learning*, mengadopsi algoritma yang terinspirasi dari cara kerja otak manusia. Penelitian ini menggunakan metode *Vision Transformer* (ViT) dengan arsitektur ViT B16 untuk mengklasifikasikan penyakit daun padi. Dataset yang digunakan terdiri dari 1253 gambar, termasuk 419 gambar untuk penyakit Bercak Coklat, 355 gambar untuk penyakit Blast, 209 gambar untuk penyakit Hawar Daun Bakteri, dan 270 gambar untuk penyakit Tungro. Dataset dibagi menjadi 70% untuk pelatihan, 15% untuk validasi, dan 15% untuk pengujian. Pelatihan dilakukan dengan *batch size* 32, 50 *epoch*, dan menggunakan *optimizer* Adam. Hasil penelitian menunjukkan bahwa model yang dikembangkan mencapai akurasi, presisi, *recall*, dan *f1-score* sebesar 96%.

Kata kunci – Penyakit Daun Padi, *Vision Transformer*, *Precision*, *Recall*, *F1-Score*.

Copyright © 2024 JURNAL TIFDA
All rights reserved.

I. PENDAHULUAN

Pertanian adalah aktivitas yang memanfaatkan sumber daya alam yang dikelola manusia untuk menghasilkan bahan pangan. Di Indonesia, sektor ini memegang peranan penting dalam memajukan perekonomian nasional. Pertanian tidak hanya menjadi tulang punggung masyarakat dalam memenuhi kebutuhan sehari-hari, tetapi juga sebagai salah satu sektor utama yang menyediakan lapangan pekerjaan dan menyumbang devisa negara melalui ekspor[1].

Padi (*Oryza sativa L.*) sebagai makanan pokok penting dalam meningkatkan kesejahteraan masyarakat Indonesia. Produktivitas tanaman padi selalu menjadi fokus untuk mencapai kedaulatan pangan, yang mencerminkan kemampuan negara mengelola pangan secara mandiri dan harus didukung oleh ketahanan pangan yang kuat, yakni memenuhi kebutuhan pangan dari produksi dalam negeri. Dengan kedaulatan pangan, Indonesia dapat mengelola pangan tanpa bergantung pada impor. Ketahanan pangan yang kokoh memastikan masyarakat memiliki cukup makanan, memperkuat kesejahteraan dan stabilitas nasional. Inovasi dan upaya berkelanjutan akan

menjadikan padi sumber kekuatan dan kesejahteraan bagi seluruh rakyat Indonesia[2].

Berdasarkan data dari Badan Pusat Statistik (BPS), pada tahun 2023, luas panen padi di Indonesia diperkirakan mencapai sekitar 10,20 juta hektar, menghasilkan sekitar 53,63 juta ton gabah kering giling (GKG). Dari produksi tersebut, beras yang dihasilkan diperkirakan mencapai 30,90 juta ton, cukup untuk memenuhi kebutuhan konsumsi penduduk. Angka ini menunjukkan betapa vitalnya sektor pertanian padi dalam memastikan ketahanan pangan dan kesejahteraan masyarakat Indonesia[3].

Namun, berbagai penyakit sering mengancam produksi padi, salah satunya adalah penyakit daun padi. Penyakit pada tanaman padi disebabkan oleh berbagai patogen seperti jamur, hama, bakteri, dan virus. Beberapa penyakit yang sering menyerang daun padi antara lain Blas (*Blast*), Bercak Coklat (Brown Spot), Hawar Daun Bakteri (*Bacterial Leaf Blight*), dan Tungro.

Dengan kemajuan teknologi saat ini, *Deep Learning* menawarkan solusi yang potensial untuk mengatasi keterbatasan tersebut. *Deep Learning*

adalah cabang ilmu komputer yang menggunakan jaringan saraf tiruan untuk belajar dari data dan menyelesaikan berbagai tugas, termasuk klasifikasi gambar. *Vision Transformer* (ViT) adalah pendekatan baru dan inovatif dalam bidang pengolahan citra menggunakan jaringan saraf tiruan. Oleh karena itu, *Vision Transformer* (ViT) dapat diimplementasikan untuk klasifikasi penyakit daun padi.

II. METODOLOGI

Pengumpulan dataset yang diperlukan pada proses klasifikasi diperoleh melalui internet, pengamatan lapangan, dan studi pustaka. Setelah pengumpulan data, selanjutnya data ini kemudian dianalisis sesuai standar proses pada CRISP-DM. Sedangkan untuk mengembangkan sistem klasifikasi penyakit daun padi menggunakan *Vision Transformer* (ViT).

III. LANDASAN TEORI

3.1 Klasifikasi Penyakit Daun Padi

Penyakit daun padi adalah salah satu ancaman utama yang dapat menurunkan hasil panen secara signifikan. Berbagai penyakit ini disebabkan oleh patogen seperti jamur, hama, bakteri, dan virus. Pentingnya diagnosis dini dan akurat tidak bisa diabaikan, karena hal ini sangat penting untuk menjaga ketahanan pangan[4].

3.2 Deep Learning

Deep learning sebagai sub-bidang penting dari *machine learning*, menggabungkan konsep jaringan saraf tiruan (JST) yang berlapis-lapis untuk memahami dan memodelkan representasi data yang kompleks. Inspirasi dari struktur dan fungsi otak manusia membuatnya sangat kuat dalam memproses data dari berbagai domain[5].

3.3 Python

Python adalah bahasa pemrograman yang pertama kali muncul pada tahun 1991 oleh Guido van Rossum, terus berkembang di bawah naungan *Python Software Foundation* hingga saat ini. Keunikan *Python* terletak pada kemampuannya sebagai bahasa pemrograman interpretatif multiguna yang sangat mudah dibaca dan dipahami[6].

3.4 Vision Transformer

Vision Transformer (ViT) mewakili evolusi terbaru dalam *deep learning* dengan mengadopsi arsitektur Transformer, yang awalnya dikembangkan untuk pemrosesan bahasa alami. ViT menggunakan pendekatan *patch embeddings*, *multi-head attention*, dan *multi-layer perceptron* untuk mengubah cara kita memahami dan menganalisis data visual secara mendalam dan efisien. Gambar input dibagi menjadi beberapa patch berdasarkan ukuran arsitekturnya selama proses penerapan patch. Setelah itu, gambar diberi *position embeddings*, yang menunjukkan lokasi

patch pada gambar. Setelah itu, patch akan dimasukkan ke *multi-head attention*[7].

3.5 PyTorch

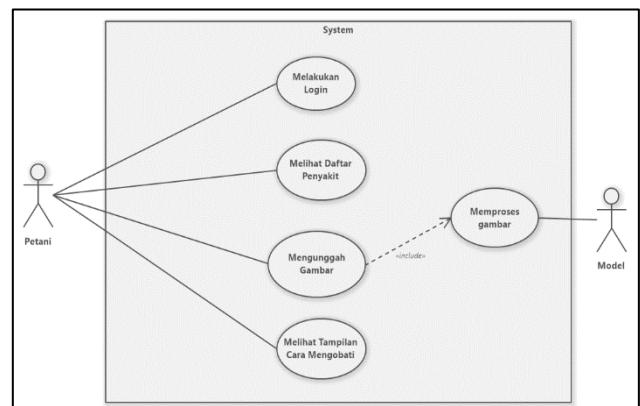
PyTorch adalah library program Python yang membantu membuat proyek *deep learning* dengan mudah. Hal ini menekankan fleksibilitas dan memungkinkan model *deep learning* diekspresikan dalam bahasa Python. Aksesibilitas dan kemudahan penggunaan ini segera diadopsi oleh komunitas riset, dan pada tahun-tahun sejak rilis pertamanya, ini telah menjadi salah satu alat *deep learning* yang paling penting dalam banyak aplikasi. Seperti halnya Python untuk pemrograman, *PyTorch* adalah pengantar yang bagus untuk *deep learning*[8].

3.6 CRISP-DM

CRISP-DM (*Cross-Industry Standard Process for Data Mining*) adalah model proses yang independen untuk data mining, terdiri dari enam langkah utama yaitu *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment*[9].

IV. HASIL DAN PEMBAHASAN

4.1 Sistem Yang Diusulkan



Gambar 4.1 Use Case Sistem Usulan

Sistem yang diusulkan adalah aplikasi implementasi *deep learning* menggunakan *vision transformer* untuk klasifikasi penyakit daun padi.

4.2 Penerapan Algoritma

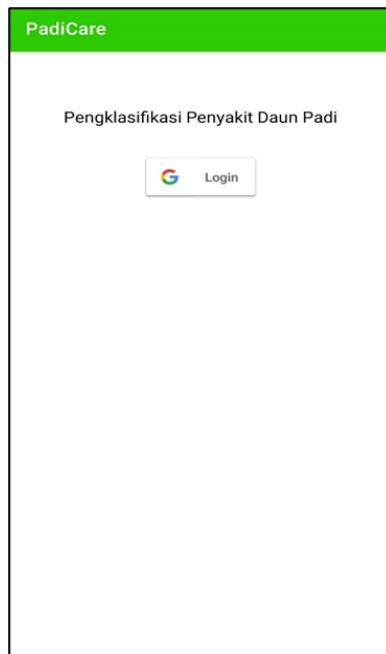
Tahapan penerapan algoritma yang dilaksanakan untuk menerapkan algoritma *Vision Transformer* dijelaskan dengan detail dibawah ini.

1. Persiapan dan Instalasi semua library yang diperlukan
2. Mengimpor dataset
3. Eksplorasi Data
4. Pembagian Data
5. Transformasi data menggunakan transforms dari torchvision.

6. Pembuatan dataset kustom dengan mengimplementasikan `__init__`, `__len__`, dan `__getitem__` untuk memproses gambar dan labelnya.
7. Membuat DataLoader untuk dataset pelatihan dan validasi
8. Augmentasi dan Normalisasi Data
9. Membuat model ViT
10. Pelatihan model ViT menjalankan proses pelatihan dan validasi.
11. Evaluasi model ViT menggunakan data pengujian.

4.3 Pembuatan User Interface

Tampilan *user interface* dari aplikasi yang sudah di deploy dalam bentuk aplikasi android. Gambar 4.2 merupakan halaman login pengguna aplikasi android dapat digunakan oleh pengguna untuk masuk ke dalam aplikasi untuk melakukan deteksi penyakit daun padi



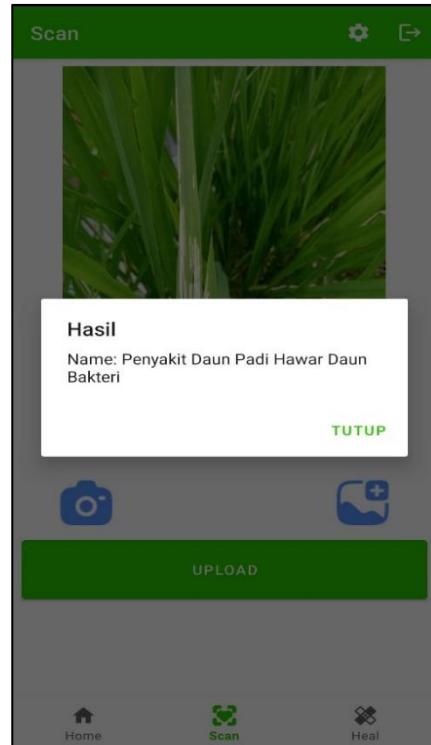
Gambar 4.2 Halaman Login

Gambar 4.3 dibawah ini merupakan tampilan halaman home pada aplikasi android yang berisi artikel tentang cara mendiagnosis penyakit daun padi secara dini, tanda-tanda penyakit daun padi hawar daun bakteri, tanda-tanda penyakit daun padi blast, tanda-tanda penyakit daun padi bercak coklat, dan tanda-tanda penyakit daun padi tungro dengan klik artikel tersebut maka akan menampilkan secara lengkap tanda-tanda penyakit daun padi.



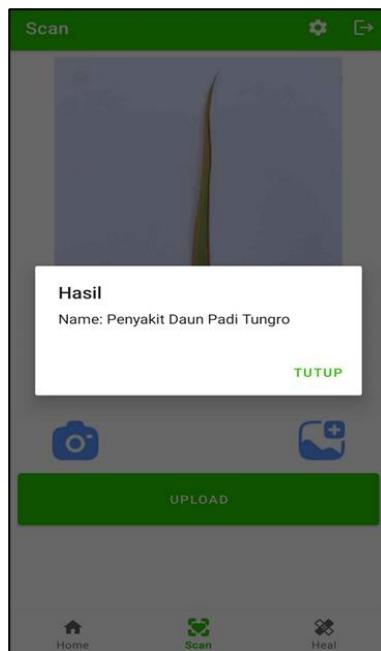
Gambar 4.3 Tampilan Home

Gambar 4.4 dibawah ini merupakan tampilan halaman scan pada aplikasi android yang digunakan untuk mendeteksi penyakit daun padi dengan hasil nama penyakit daun padi hawar daun bakteri.



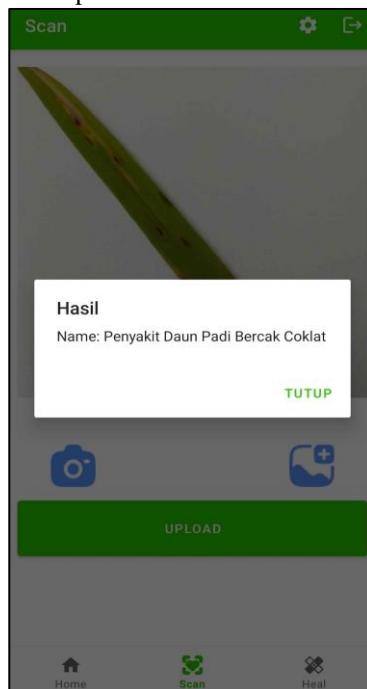
Gambar 4.4 Tampilan Halaman Scan Hasil Penyakit Hawar Daun Bakteri

Gambar 4.5 dibawah ini merupakan tampilan halaman scan pada aplikasi android yang digunakan untuk mendeteksi penyakit daun padi dengan hasil nama penyakit daun padi Tungro.



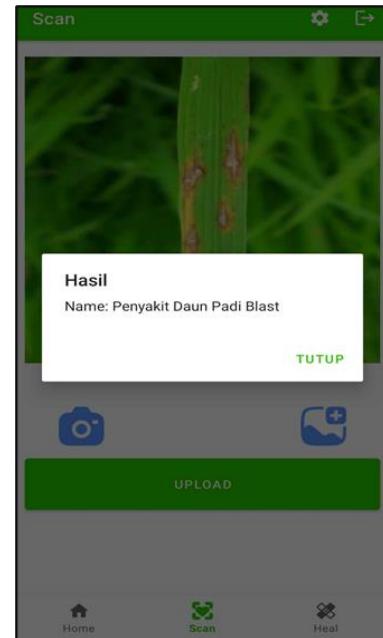
Gambar 4.5 Tampilan Halaman *Scan* Hasil Penyakit Tungro

Gambar 4.6 dibawah ini merupakan tampilan halaman scan pada aplikasi android yang digunakan untuk mendeteksi penyakit daun padi dengan hasil nama penyakit daun padi bercak coklat.



Gambar 4.6 Tampilan Halaman *Scan* Hasil Penyakit Bercak Coklat

Gambar 4.7 dibawah ini merupakan tampilan halaman scan pada aplikasi android yang digunakan untuk mendeteksi penyakit daun padi dengan hasilnama penyakit daun padi bercak blast.



Gambar 4.7 Tampilan Halaman *Scan* Hasil Penyakit Blast

Gambar 4.8 menampilkan halaman "Heal" pada aplikasi Android yang memberikan panduan tentang pengobatan penyakit pada daun padi, seperti hawar daun bakteri, blast, bercak coklat, dan tungro.



Gambar 4.8 Tampilan Halaman *Heal*

4.4 Implementasi CRISP-DM

4.4.1 Business Understanding

Penelitian ini bertujuan untuk mendeteksi dini penyakit pada daun padi masih terbatas menggunakan metode tradisional untuk mengidentifikasi penyakit, yang seringkali tidak akurat dan membutuhkan waktu yang cukup lama. Setelah memahami masalah, peneliti menentukan solusi bisnis dari jenis data mining yang akan berdasarkan masalah yang akan diselesaikan.

4.4.2 Data Understanding

Tahap pengumpulan dataset dengan mengambil dari internet dan kaggle, dataset yang dikumpulkan adalah empat kelas penyakit daun padi yaitu bercak coklat, blast, hawar daun bakteri, dan tungro dengan total keseluruhan sebesar 1.253 gambar.

4.4.3 Data Preparation

Pada tahap ini untuk memastikan bahwa model yang dilatih memiliki kemampuan generalisasi yang baik dan dapat berfungsi dengan baik, proses pengambilan data untuk validasi klasifikasi penyakit daun padi melibatkan beberapa langkah penting. Berikut adalah langkah-langkahnya:

a. Pengumpulan Data

Data yang dikumpulkan adalah empat kelas penyakit daun padi yaitu bercak coklat, blast, hawar daun bakteri, dan tungro dengan total keseluruhan sebesar 1.253 gambar.

b. Pelabelan Data

Gambar 4.9 merupakan pelabelan data setiap gambar daun padi diberi label sesuai dengan jenis penyakit yang terdeteksi.

```
[ ] # We have to define the mapping of the classes to convert the labels to numbers.
label_map = dict(zip(classes, range(len(classes))))
label_map

→ {'Penyakit Daun Padi Bercak Coklat': 0,
 'Penyakit Daun Padi Blast': 1,
 'Penyakit Daun Padi Hawar Daun Bakteri': 2,
 'Penyakit Daun Padi Tungro': 3}
```

Gambar 4.9 Pelabelan Data

c. Pembagian Dataset

Dataset yang telah dikumpulkan dibagi menjadi tiga bagian utama: data pelatihan, data validasi, dan data pengujian. Biasanya, pembagian ini dilakukan dengan rasio tertentu, seperti 70% untuk pelatihan, 15% untuk validasi, dan 15% untuk pengujian.

d. Preprocessing Data

Dataset memiliki gambar yang diproses lebih lanjut untuk memastikan konsistensinya. Gambar 4.10 merupakan augmentasi data termasuk perubahan ukuran gambar, normalisasi piksel, dan augmentasi data (seperti rotasi, flip, dan perubahan pencahayaan)

untuk menambah keragaman pelatihan dan validasi data.

```
[ ] # Augmentasi Data
from torchvision import transforms

# Augmentasi dan Normalisasi
data_transforms = transforms.Compose([
    transforms.RandomRotation(degrees=45), # Increase rotation angle
    transforms.RandomHorizontalFlip(),
    transforms.RandomVerticalFlip(),
    transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.2), # Increase magnitude
    transforms.RandomResizedCrop(224, scale=[0.7, 1.0]), # Wider range for crop scale
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])
```

Gambar 4.10 Augmentasi Data

4.4.4 Modelling

Modelling adalah fase yang berfokus pada pengembangan model data mining yang akan digunakan untuk memprediksi atau menjelaskan pola dalam data. Gambar 4.11 merupakan modelling ViT untuk membangun model menggunakan arsitektur *vision transformer* untuk mendeteksi penyakit daun padi.

```
# GPU
device = "cuda" if torch.cuda.is_available() else "cpu"

# ViT
model = vit_b_16(weights = weights)
output_shape = len(classes) # Total classes
model.heads = nn.Sequential(OrderedDict([('head', nn.Linear(in_features = 768, out_features = len(classes))))])

for param in model.parameters():
    param.requires_grad = False
for param in model.heads.parameters():
    param.requires_grad = True

# Check if the parameters were frozen
summary(model, input_size = [8,3,224,224],
         col_width = 15,
         col_names = ["Input_size", "Output_size", "Num_params", "Trainable"],
         row_settings = ["var_names"])

loss_fn = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.heads.parameters(), lr = 0.001)
```

Gambar 4.11 Modelling ViT

4.4.5 Evaluation

Evaluasi dilakukan dengan mengukur kinerja model melalui dua proses, yaitu *classification report* dan *confusion matrix*. Gambar 4.12 merupakan *classification report* dari *Vision transformer* (ViT). Gambar tersebut menunjukkan *classification report* yang mencakup metrik evaluasi seperti *precision*, *recall*, *f1-score*, dan *support* untuk setiap jenis penyakit daun padi. Model memiliki performa tinggi dengan akurasi keseluruhan 96%. Setiap kelas memiliki *f1-score* mendekati atau di atas 0,94, menunjukkan keseimbangan yang baik antara *precision* dan *recall*. Penyakit Tungro memiliki performa terbaik dengan precision dan recall 0,98, sementara Hawar Daun Bakteri juga sangat baik dengan precision 1,00 dan recall 0,97. Nilai *macro avg* dan *weighted avg* sebesar 0,96 mengindikasikan bahwa performa model konsisten di seluruh kelas, meskipun jumlah sampel per kelas berbeda.

Classification Report:				
	precision	recall	f1-score	support
Penyakit Daun Padi Bercak Coklat	0.92	0.97	0.95	63
Penyakit Daun Padi Blast	0.96	0.92	0.94	53
Penyakit Daun Padi Hawar Daun Bakteri	0.97	0.97	0.97	31
Penyakit Daun Padi Tungro	1.00	0.98	0.99	41
accuracy			0.96	188
macro avg	0.96	0.96	0.96	188
weighted avg	0.96	0.96	0.96	188

Gambar 4.12 Classification Report ViT

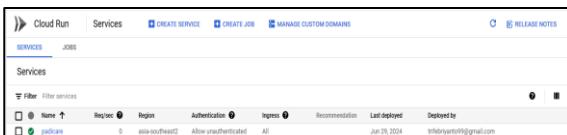
Gambar 4.13 merupakan *confusion matrix* dari *Vision transformer* (ViT). Gambar tersebut menunjukkan *confusion matrix* untuk pengujian klasifikasi penyakit daun padi, di mana model berhasil mengklasifikasikan sebagian besar sampel dengan benar, terlihat dari nilai tinggi pada diagonal utama (contoh: 58 untuk Bercak Coklat dan 45 untuk Blast). Namun, terdapat beberapa kesalahan prediksi, seperti 5 sampel Bercak Coklat diprediksi sebagai Blast dan 3 sampel Tungro diprediksi sebagai Hawar Daun Bakteri. Secara keseluruhan, model bekerja cukup baik, meskipun masih ada ruang untuk peningkatan dalam mengurangi kesalahan pada kelas yang serupa.

Confusion Matrix Test				
	Penyakit Daun Padi Bercak Coklat -	Penyakit Daun Padi Blast -	Penyakit Daun Padi Hawar Daun Bakteri -	Penyakit Daun Padi Tungro -
Penyakit Daun Padi Bercak Coklat -	58	3	2	0
Penyakit Daun Padi Blast -	5	45	2	1
Penyakit Daun Padi Hawar Daun Bakteri -	3	0	29	0
Penyakit Daun Padi Tungro -	1	0	0	39

Gambar 4.13 Confusion Matrix ViT

4.4.6 Deployment

Gambar 4.14 merupakan *deployment* model yang dihasilkan dengan menggunakan *google cloud run* dengan membuat *dockerfile*, membuat *backend* menggunakan *flask* dengan file *app.py* dan simpan model ViT yang sudah dihasilkan di dalam satu folder dengan nama padicare.



Gambar 4.14 Deployment Model ViT

Selanjutnya membuat *repository Artifact Registry* menggunakan *Cloud Build* setelah *repository* berhasil dibuat. Selanjutnya membuat *container image* baru setelah *container image* berhasil dibuat selanjutnya *deploy container image* yang terdapat folder padicare setelah berhasil di *deploy* maka akan mendapatkan URL untuk digunakan pada aplikasi android.

V. KESIMPULAN

Berdasarkan hasil penelitian, implementasi deep learning menggunakan arsitektur *Vision Transformer*

(ViT) untuk klasifikasi penyakit daun padi melibatkan beberapa langkah utama. Pertama, data gambar daun padi yang terinfeksi penyakit dikumpulkan dan diproses melalui tahap *pre-processing* seperti *resize* dan normalisasi. Kemudian, dataset yang telah diproses ini digunakan untuk melatih model *Vision Transformer*. Untuk melakukan ini, hyperparameter seperti jumlah *epoch*, *batch size*, *learning rate*, dan jenis *optimizer* yang digunakan. Proses pelatihan dilakukan hingga model dapat mengklasifikasikan dengan baik.

Faktor-faktor yang mempengaruhi kinerja model klasifikasi meliputi kualitas dan jumlah data yang digunakan, metode *pre-processing* data, dan penerapan *hyperparameter* yang tepat adalah beberapa faktor yang mempengaruhi kinerja model klasifikasi. Kualitas data yang baik dan representatif sangat penting untuk memastikan model dapat belajar dengan benar. *Pre-processing* data seperti augmentasi dapat membantu model menemukan variasi dalam data. Selain itu, pengaturan *hyperparameter* seperti *learning rate* sama dengan 0,001, *batch size* sama dengan 32, jumlah *epoch* sama dengan 50 dan menggunakan *optimizer adam*. Nilai akurasi yang didapatkan yaitu sebesar 96%, *precision* sebesar 96%, *recall* sebesar 96%, *f1-score* sebesar 96%.

DAFTAR PUSTAKA

- [1] S. A. M. Harun, M. I. Pradhita, and U. Achmad, “PERUBAHAN SOSIAL MASYARAKAT AKIBAT PENURUNAN KUALITAS PADI DI DESA WONOJATI KECAMATAN JENGGAWAH KABUPATEN JEMBER,” *SOCA: Jurnal Sosial, Ekonomi Pertanian*, p. 38, Feb. 2019, doi: 10.24843/soca.2019.v13.i01.p04.
- [2] C. Wati, S. Tinggi, P. Pertanian, (Stpp, and) Manokwari, “IDENTIFIKASI HAMA TANAMAN PADI (*Oriza Sativa L*) DENGAN PERANGKAP CAHAYA DI KAMPUNG DESAY DISTRIK PRAFI PROVINSI PAPUA BARAT IDENTIFICATION OF RICE PLANT (*Oriza Sativa L*) WITH LIGHT TRAP IN DESAY VILLAGE PRAFI DISTRICT WEST PAPUA PROVINCE,” 2017.
- [3] Badan Pusat Statistik, “Luas Panen dan Produksi Padi di Indonesia 2023,” Oct. 2023.
- [4] C. Irwan and S. S. Mu’min, *Buku Saku Penyakit Padi*. Karawang: BALAI BESAR PERAMALAN ORGANISME PENGGANGGU TUMBUHAN, 2020.
- [5] J. D. Kelleher, *DEEP LEARNING*. London: The Massachusetts Institute of Technology, 2019.
- [6] Mambang, F. D. Marleny, and M. Zulfadhilah, *ALGORITMA PEMROGRAMAN MENGGUNAKAN PYTHON*. Banyumas: CV. Pena Persada, 2022. [Online]. Available: <https://www.researchgate.net/publication/363769056Y>.
- [7] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” Oct. 2020, [Online]. Available: <http://arxiv.org/abs/2010.11929>.
- [8] E. Stevens, L. Antiga, and T. Viehmann, *Deep Learning with PyTorch*. New York: Manning Publications Co., 2020.
- [9] C. Schröer, F. Kruse, and J. M. Gómez, “A systematic literature review on applying CRISP-DM process model,” in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 526–534. doi: 10.1016/j.procs.2021.01.199.