

**IMPLEMENTASI *SWIN TRANSFORMER* UNTUK IDENTIFIKASI JENIS
REMPAH BERBENTUK RIMPANG BERBASIS WEB MENGGUNAKAN
FRAMEWORK STREAMLIT**

(Skripsi)

**Oleh
M. ARKAN NIBRASTAMA
2117051079**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

**IMPLEMENTASI *SWIN TRANSFORMER* UNTUK IDENTIFIKASI JENIS
REMPAH BERBENTUK RIMPANG BERBASIS WEB MENGGUNAKAN
FRAMEWORK STREAMLIT**

Oleh

M. ARKAN NIBRASTAMA

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

ABSTRAK

IMPLEMENTASI *SWIN TRANSFORMER* UNTUK IDENTIFIKASI JENIS REMPAH BERBENTUK RIMPANG BERBASIS WEB MENGGUNAKAN *FRAMEWORK* STREAMLIT

Oleh

M. ARKAN NIBRASTAMA

Indonesia merupakan negara yang kaya akan sumber daya alam, terutama dalam hal keanekaragaman rempah-rempah. Salah satu kelompok rempah yang banyak digunakan adalah rempah berbentuk rimpang seperti jahe, kunyit, lengkuas, dan kencur. Namun, kemiripan visual antar jenis rimpang ini seringkali menimbulkan tantangan dalam proses identifikasi dan klasifikasi secara otomatis. Penelitian ini bertujuan untuk mengembangkan sistem identifikasi berbasis citra dengan memanfaatkan arsitektur Swin Transformer, sebuah model Vision Transformer mutakhir yang telah terbukti efektif dalam berbagai tugas klasifikasi gambar. Model Swin Transformer dalam penelitian ini berhasil mencapai performa tinggi dengan akurasi sebesar 99,11%, presisi 98,24%, recall 98,21%, dan F1-score 98,21%, jauh mengungguli model Xception yang sebelumnya dianggap sebagai baseline terbaik dengan akurasi 95,00%, presisi 90,14%, recall 90,00%, dan F1-score 90,01%. Untuk memastikan sistem ini dapat digunakan secara praktis, model akhir diimplementasikan dalam bentuk aplikasi web menggunakan framework Streamlit. Aplikasi ini memungkinkan pengguna untuk mengidentifikasi jenis rempah rimpang melalui unggahan gambar. Temuan ini menunjukkan bahwa Swin Transformer tidak hanya unggul dari segi performa, tetapi juga memiliki potensi tinggi untuk diterapkan secara langsung dalam konteks klasifikasi citra berbasis web, khususnya dalam bidang pertanian dan industri rempah-rempah.

Kata kunci: Vision Transformer (ViT), Swin Transformer, Rempah Berbentuk Rimpang, Streamlit.

ABSTRACT

IMPLEMENTATION OF SWIN TRANSFORMER FOR WEB-BASED IDENTIFICATION OF RHIZOME SHAPED SPICES USING STREAMLIT FRAMEWORK

By

M. ARKAN NIBRASTAMA

Indonesia is a country rich in natural resources, particularly in various types of spices. Among these, rhizome-shaped spices such as ginger, turmeric, galangal, and aromatic ginger often present classification challenges due to their similar visual appearances. This study aims to address this issue by developing a web-based identification system utilizing the Swin Transformer—an advanced Vision Transformer architecture known for its effectiveness in image classification tasks. The Swin Transformer model demonstrated superior performance, achieving an accuracy of 99.11%, precision of 98.24%, recall of 98.21%, and F1-score of 98.21%. These results significantly outperform the Xception convolutional neural network (CNN) model, which was previously considered state-of-the-art, with 95.00% accuracy, 90.14% precision, 90.00% recall, and 90.01% F1-score. To ensure practical usability, the final Swin Transformer model was deployed as a web application using the Streamlit framework, allowing users to classify rhizome spices through image uploads. These findings highlight the effectiveness of Swin Transformer for practical image-based spice classification.

Keywords: Vision Transformer (ViT), Swin Transformer, Rhizome Spices, Streamlit.

Judul Skripsi : **IMPLEMENTASI *SWIN TRANSFORMER*
UNTUK IDENTIFIKASI JENIS REMPAH
BERBENTUK RIMPANG BERBASIS WEB
MENGUNAKAN *FRAMEWORK*
STREAMLIT**

Nama Mahasiswa : **M. Arkan Nibrastama**

Nomor Pokok Mahasiswa : 2117051079

Program Studi : Ilmu Komputer

Fakultas : Matematika dan Pengetahuan Alam



1. Komisi Pembimbing

Didik Kurniawan, S. Si., M. T.
NIP. 19800419 200501 1 004

2. Ketua Jurusan Ilmu Komputer

Dwi Sakethi, S.Si., M.Kom.
NIP. 19680611 199802 1 001

MENGESAHKAN

1. Tim Penguji

Ketua

: **Didik Kurniawan, S.Si., M.T.**



Sekretaris

: **Rico Andrian, S.Si., M.Kom.**



Penguji Utama

: **Prof. Admi Syarif, Ph.D.**



2. Dekan Fakultas Matematika dan Pengetahuan Alam



Dr. Eng. Heri Satria, S.Si., M.Si.

NIP. 19711001 200501 1 002



Tanggal Lulus Ujian Skripsi: **19 Juni 2025**

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : M. Arkan Nibrastama

NPM : 2117051079

Dengan ini menyatakan bahwa skripsi saya yang berjudul “**Implementasi *Swin Transformer* Untuk Identifikasi Jenis Rempah Berbentuk Rimpang Berbasis Web Menggunakan *Framework Streamlit***” merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya tulis ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil jiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang saya terima.

Bandar Lampung, 19 Juni 2025

Penulis,



M. Arkan Nibrastama
NPM. 2117051079

RIWAYAT HIDUP



Penulis dilahirkan di Bandar Lampung pada tanggal 06 April 2003 sebagai anak pertama dari dua bersaudara dari pasangan Bapak Yudistira Erik Hendarsyah, S.T., M.T. dan Ibu Yurita Sari, S.Sos. serta menjadi kakak dari Muhammad Hilmy Antartama. Penulis menyelesaikan pendidikan dasar di SDIT Permata Bunda 1 Bandar Lampung pada tahun 2015. Penulis menyelesaikan pendidikan menengah pertama di SMPQ Darul Fattah Bandar Lampung pada tahun 2018. Penulis melanjutkan pendidikan menengah atas di MAN 1 Bandar Lampung dan telah menyelesaikan pendidikan tersebut pada tahun 2021.

Penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung pada tahun 2021 melalui jalur SBMPTN. Pada Januari-Februari 2024 penulis melaksanakan Kuliah Kerja Nyata (KKN) di Desa Batu Menyan, Kecamatan Teluk, Kabupaten Pesawaran. Penulis melaksanakan Magang MBKM di Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi sebagai Database Engineer pada Februari - Juli 2024.

Penulis tergabung dalam Organisasi Himpunan Mahasiswa Ilmu Komputer sebagai anggota biasa. Selain itu penulis aktif sebagai asisten dosen di Jurusan Ilmu Komputer untuk mata kuliah Basis Data semester ganjil TA 2023/2024. Penulis juga aktif dalam perlombaan, seperti menjadi finalis pada lomba Turnamen Sains Data Nasional (TSDN) 2024.

PERSEMBAHAN

Segala puji saya panjatkan kepada Allah Subhanahu wa Ta'ala atas segala rahmat-Nya. Shalawat dan salam semoga senantiasa tercurah kepada Nabi Muhammad Shallallahu 'Alaihi wa Sallam, sehingga saya dapat menyelesaikan skripsi ini.

Dengan segala ketulusan serta kerendahan hati, sebuah karya sederhana ini
kupersembahkan kepada:

Ayah dan Bunda tercinta yang telah membesarkan,
mendidik dan menyayangiku, serta selalu berdoa
untuk keberhasilanku.

Seluruh keluarga dan para sahabat yang senantiasa mengiringi langkahku dengan
doa dan dukungan

Serta

Institusi yang turut membentuk diriku menjadi lebih dewasa dalam berpikir dan
bertindak.

Almamater tercinta

UNIVERSITAS LAMPUNG

MOTTO HIDUP

“Maka sesungguhnya sesudah kesulitan ada kemudahan, sesungguhnya sesudah
kesulitan ada kemudahan.”

(Qs. Al-Insyirah: Ayat 5-6)

“Tetaplah berjuang walaupun badai menghadang.”

(Muhammad Arkan Nibrastama)

SANWACANA

Puji syukur penulis ucapkan kehadiran Allah SWT karena atas limpahan rahmat dan karunia-Nya sehingga skripsi ini dapat diselesaikan. Skripsi dengan judul **“Implementasi *Swin Transformer* Untuk Identifikasi Jenis Rempah Berbentuk Rimpang Berbasis Web Menggunakan *Framework Streamlit*”** adalah salah satu syarat untuk memperoleh gelar sarjana Jurusan Ilmu Komputer di Universitas Lampung.

Pada kesempatan ini dengan ketulusan hati penulis menyampaikan ucapan terima kasih kepada:

1. Allah SWT, yang telah memberikan nikmat, rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan lancar hingga akhir.
2. Kedua orang tua penulis, Ayah dan Bunda tercinta atas segala doa, semangat, pengorbanan, dan kasih sayang yang tulus ikhlas dan senantiasa berjuang untuk keberhasilanku.
3. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung atas izin yang diberikan.
4. Bapak Dwi Sakethi, S.Kom., M.Kom. selaku ketua Jurusan Ilmu Komputer Universitas Lampung atas izin yang diberikan.
5. Bapak Didik Kurniawan, S.Si., M.Kom. selaku pembimbing utama atas saran, motivasi, arahan, nasihat, ilmu, dan bimbingan serta segala bantuan selama masa studi dan penulisan skripsi.
6. Bapak Prof. Admi Syarif, Ph.D. selaku Pembimbing akademik dan pembahas satu atas bimbingan, motivasi, arahan, serta kritik dan saran yang diberikan kepada penulis dan segala bantuan yang diberikan selama masa studi dan penyusunan skripsi.

7. Bapak Rico Andrian, S.Si. M.Kom. selaku pembahas satu atas bimbingan, motivasi, arahan, serta kritik dan saran yang diberikan kepada penulis dan segala bantuan yang diberikan selama masa studi dan penyusunan skripsi.
8. Bapak dan ibu dosen Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung atas bimbingan, nasehat, dan ilmu yang diberikan selama masa studi.
9. Rekan seperjuangan sekaligus tempat bertukar pikir penulis, Salsabila Juandira, atas segala bantuan, semangat, dukungan, dan motivasi yang telah diberikan selama masa kuliah hingga proses penyusunan skripsi ini.
10. Sahabat-sahabat penyemangat sejak awal perkuliahan Roy Rafles Matorang Pasaribu, Radiya Adhirajasa, dan Alfa Rizki Fadlillah atas segala dukungan, semangat dan suasana canda tawa yang diberikan kepada penulis selama masa perkuliahan.
11. Keluarga besar Jurusan Ilmu Komputer angkatan 2021 atas suasana kekeluargaan dan kenangan indah selama masa studi serta motivasi yang diberikan.

Semoga semua bantuan dan jasa baik yang telah diberikan kepada penulis mendapat pahala dari Allah SWT, dan semoga skripsi ini dapat bermanfaat bagi kita semua. Amin.

Bandar Lampung, Juni 2025

Penulis,

M. Arkan Nibrastama
NPM. 2117051079

DAFTAR ISI

DAFTAR ISI	i
DAFTAR TABEL	iii
DAFTAR GAMBAR	iv
I. PENDAHULUAN	1
1.1 Latar Belakang dan Masalah	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	5
II. TINJAUAN PUSTAKA	6
2.1 Penelitian Terdahulu	6
2.2 Rempah.....	7
2.3 Rimpang.....	8
2.3.1 Jahe.....	8
2.3.2 Kunyit.....	9
2.3.3 Kencur	10
2.3.4 Lengkuas	11
2.4 Machine Learning.....	12
2.4.1 <i>Supervised Learning</i>	13
2.4.2 <i>Unsupervised Learning</i>	13
2.4.3 <i>Reinforcement Learning</i>	14
2.5 Vision Transformer (ViT)	14
2.5.1 <i>Swin Transformer</i>	16
2.6 Transfer Learning	20
2.7 Preprocessing.....	20

2.8	Confusion Matrix.....	20
2.8.1	Akurasi	21
2.8.2	Presisi	22
2.8.3	<i>Recall</i>	22
2.8.4	<i>F1-Score</i>	22
2.9	Streamlit.....	23
2.10	Black Box Testing.....	23
2.11	Kurva ROC	24
III.	METODOLOGI PENELITIAN.....	26
3.1	Waktu dan Tempat Penelitian	26
3.2	Perangkat Penelitian	26
3.2.1	Perangkat Keras (<i>Hardware</i>).....	26
3.3	<i>Dataset</i>	27
3.4	Tahap Penelitian	27
3.4.1	Pengumpulan <i>Dataset</i>	29
3.4.2	<i>Preprocessing Dataset</i>	29
3.4.3	Pembuatan Model.....	33
3.4.4	Pembuatan Web.....	37
3.4.5	Pengujian.....	38
IV.	HASIL DAN PEMBAHASAN.....	40
4.1	Pengumpulan <i>Dataset</i>	40
4.2	<i>Preprocessing Dataset</i>	41
4.3	Pembuatan Model.....	49
4.4	Pembuatan Web	54
4.5	Pengujian	55
4.6	Pembahasan	63
V.	SIMPULAN DAN SARAN.....	89
5.1	Simpulan.....	89
5.2	Saran	90
	DAFTAR PUSTAKA.....	91
	LAMPIRAN	95

DAFTAR TABEL

Tabel 1. Penelitian Terdahulu.....	6
Tabel 2. Kategori prediksi model pada confusion matrix	21
Tabel 3. Hyperparameter training model.....	37
Tabel 4. Pengujian Black Box Testing	39
Tabel 5. Total dataset gabungan	43
Tabel 6. Total Dataset Setelah Split Dataset.....	45
Tabel 7. Hyperparameter training model Swin Transformer.....	52
Tabel 8. Confusion Matrix Untuk Kelas Jahe	58
Tabel 9. Confusion Matrix Untuk Kelas Kencur.....	58
Tabel 10. Confusion Matrix Untuk Kelas Kunyit	59
Tabel 11. Confusion Matrix Untuk Kelas Lengkuas	59
Tabel 12. Hasil perhitungan confusion matrix dengan metrik macro-average.....	60
Tabel 13. Hasil pengujian web	62
Tabel 14. Perbandingan konfigurasi parameter patch dan windows size pada Arsitektur Swin Transformer.....	64
Tabel 15. Perbandingan konfigurasi parameter drop rate dan drop path rate pada Arsitektur Swin Transformer.....	69
Tabel 16. Perbandingan model dengan dan tanpa menggunakan augmentasi pada dataset pada Arsitektur Swin Transformer	80
Tabel 17. Perbandingan Arsitektur Transformer dengan CNN	83

DAFTAR GAMBAR

Gambar 1. Rimpang	8
Gambar 2. Jahe.....	9
Gambar 3. Kunyit.....	10
Gambar 4. Kencur	11
Gambar 5. Lengkuas	12
Gambar 6. Arsitektur Vision Transformer (ViT).....	15
Gambar 7. Hierarcichal Patches	16
Gambar 8. Arsitektur Swin Transformer.....	17
Gambar 9. Window-based Multi-Head Self-Attention	18
Gambar 10. Shifted Window based Multi-Head Self-Attention	19
Gambar 11. Gambar Kurva ROC dengan Closest to (0,1) Criteria.....	25
Gambar 12. Tahap Penelitian	28
Gambar 13. Website roboflow.....	29
Gambar 14. Ilustrasi Augmentasi Dataset.....	32
Gambar 15. Ilustrasi Resize Dataset	33
Gambar 16. Sumber Model Swin Transformer	34
Gambar 17. Sumber Pre-trained Model	34
Gambar 18. Ilustrasi modifikasi output layer.....	36
Gambar 19. Alur kerja web	38
Gambar 20. Contoh gambar dataset dari website Roboflow.....	40
Gambar 21. Contoh gambar dataset yang diambil secara langsung.....	41
Gambar 22. Code Join Dataset.....	43
Gambar 23. Code Split Dataset.....	45
Gambar 24. Hasil dataset setelah augmentasi	47
Gambar 25. Code augmentasi dataset	47

Gambar 26. Code reszie dataset.....	48
Gambar 27. Contoh hasil resize dataset	48
Gambar 28. Code transfer learning	49
Gambar 29. Code freeze parameter.....	51
Gambar 30. Code modifikasi output layer	51
Gambar 31. Grafik akurasi model pada tahap training	53
Gambar 32. Grafik loss model pada tahap training.....	53
Gambar 33. Code untuk menyimpan model.....	54
Gambar 34. Hasil tampilan antarmuka web.....	55
Gambar 35. Hasil pengujian dengan confusion matrix.....	57
Gambar 36. Confusion Matrix Untuk Konfigurasi Patch 2 dan Windows 7.....	65
Gambar 37. Confusion Matrix Untuk Konfigurasi Patch 4 dan Windows 7.....	66
Gambar 38. Confusion Matrix Untuk Konfigurasi Patch 4 dan Windows 14.....	67
Gambar 39. Confusion Matrix Untuk Konfigurasi drop rate 0 dan drop path rate 0,1.....	70
Gambar 40. Kurva ROC untuk konfigurasi drop rate 0 dan drop path rate 0,1	71
Gambar 41. Confusion Matrix Untuk Konfigurasi drop rate 0 dan drop path rate 0,2.....	72
Gambar 42. Kurva ROC untuk konfigurasi drop rate 0 dan drop path rate 0,2.....	73
Gambar 43. Confusion Matrix Untuk Konfigurasi drop rate 0 dan drop path rate 0,3.....	74
Gambar 44. Kurva ROC untuk konfigurasi drop rate 0 dan drop path rate 0,3	75
Gambar 45. Confusion Matrix Untuk Konfigurasi drop rate 0,1 dan drop path rate 0,1.....	76
Gambar 46. Kurva ROC untuk konfigurasi drop rate 0,1 dan drop path rate 0,1 ..	77
Gambar 47. Confusion Matrix Untuk Konfigurasi drop rate 0,1 dan drop path rate 0,3.....	78
Gambar 48. Kurva ROC untuk konfigurasi drop rate 0,1 dan drop path rate 0,3 ..	79
Gambar 49. Confusion Matrix Untuk Dataset Tanpa Augmentasi.....	81
Gambar 50. Confusion Matrix Untuk Dataset Dengan Augmentasi.....	82
Gambar 51. Confusion Matrix Untuk Arsitektur Transformer.....	83
Gambar 52. Confusion Matrix Untuk Arsitektur CNN	84

Gambar 53. Kurva ROC Untuk Nilai Threshold Optimum	86
Gambar 54. data OOD yang berhasil tidak teridentifikasi	86
Gambar 55. Data ID yang tidak berhasil teridentifikasi.....	87

I. PENDAHULUAN

1.1 Latar Belakang dan Masalah

Indonesia merupakan negara dengan sumber daya alam yang melimpah, salah satunya adalah rempah-rempah. Rempah-rempah dapat berupa bunga, daun, biji, batang, *rhizoma* (rimpang), dan akar (Tanuwijaya dan Roseanne, 2021). Adapun beberapa jenis rempah yang tumbuh di Indonesia adalah jahe, kunyit, kencur, lengkuas, dan masih banyak lagi (Riska dan Farokhah, 2021). Sejak zaman dahulu, rempah-rempah dikenal sebagai komoditas unggulan bernilai tinggi yang menjadi incaran bangsa-bangsa asing, terutama bangsa Eropa, karena rempah-rempah tidak hanya digunakan sebagai penyedap masakan tetapi juga dapat berfungsi sebagai obat tradisional yang bermanfaat bagi kesehatan, khususnya untuk menjaga daya tahan tubuh (Nisa dan Candra, 2023). Sebagai contoh, jahe sering dimanfaatkan untuk meredakan masuk angin, batuk, dan mual, kunyit digunakan sebagai penurun demam, kencur dipakai untuk mengobati radang lambung, *influenza*, masuk angin, sakit kepala, dan batuk, dan lengkuas sering digunakan untuk mengatasi diare.

Penelitian (Tanuwijaya dan Roseanne, 2021) menyatakan dengan banyaknya jenis rempah-rempah yang ada di Indonesia membuat masyarakat, kesulitan dalam membedakan setiap nama rempah-rempah. Di antara rempah yang sulit dibedakan tersebut adalah rimpang, seperti jahe, kunyit, kencur, dan lengkuas. Rimpang tersebut memiliki karakteristik, bentuk, dan warna kulit yang hampir mirip sehingga sulit untuk membedakan antara rimpang satu dengan rimpang yang lainnya. Hal tersebut didukung dengan survei yang telah dilakukan oleh penelitian (Putra *et al.*, 2023), dimana 100 orang diminta untuk menebak 5 jenis rempah asli Indonesia, dan hanya 31% orang yang berhasil menebak lebih dari 3 jenis rempah

dengan benar. Oleh karena itu dapat disimpulkan bahwa pengetahuan mengenai jenis rempah masih rendah.

Machine Learning merupakan teknik kecerdasan buatan yang dapat mengolah data secara otomatis untuk menemukan pola atau karakteristik khusus dalam data. Metode ini bekerja dengan cara memprediksi hasil berdasarkan data yang telah dipelajari selama proses pelatihan (Roihan *et al.*, 2020). *Deep Learning* merupakan cabang ilmu dari *Machine Learning* yang merupakan pengembangan dari *Multi Layer Perceptron* (MLP) yang dirancang untuk mengolah data dalam bentuk dua dimensi lebih cepat (Ilahiyah dan Nilogiri, 2018). Metode *Machine Learning* dan *Deep Learning* sudah diterapkan pada beberapa penelitian sebelumnya untuk mengidentifikasi rempah dan herbal, termasuk yang berbentuk rimpang, dan hasilnya menunjukkan efektivitas yang cukup tinggi dalam proses pengidentifikasian tersebut.

Beberapa penelitian telah dilakukan untuk mengidentifikasi jenis rempah menggunakan berbagai metode dan pendekatan. Misalnya, penelitian oleh (Riska dan Farokhah, 2021) mengklasifikasikan rempah, seperti jahe, kunyit, kencur, dan lengkuas menggunakan metode *K-Nearest Neighbors* (K-NN) dengan tingkat akurasi sebesar 73%. Selanjutnya, penelitian oleh (Carnegie *et al.*, 2023) dan (Nisa dan Candra, 2023) menggunakan metode *Convolutional Neural Network* (CNN) untuk identifikasi rempah dan herbal, mencapai akurasi 93% dan 90%. Penelitian (Khrisne dan Suyadnya, 2018) dan (Tanuwijaya dan Roseanne, 2021) juga mengimplementasikan arsitektur CNN lainnya, yaitu VGGNet dan VGG16, yang masing-masing mencapai akurasi sebesar 70% dan 85%.

Vision Transformer (ViT) merupakan salah satu arsitektur *deep learning* yang mengadaptasi mekanisme transformer, yang awalnya dirancang untuk *Natural Language Processing* (NLP), ke dalam tugas *computer vision*. ViT bekerja dengan cara membagi citra menjadi *patch-patch* kecil yang kemudian diproses sebagai *sequence tokens* (Dosovitskiy *et al.*, 2020). Namun, pendekatan ini memiliki keterbatasan dalam hal komputasi yang berat serta kurangnya kemampuan untuk menangkap detail pada suatu citra. Untuk mengatasi kelemahan tersebut,

dikembangkan metode *Swin Transformer* yang menggunakan pendekatan *hierarchical feature mapping* dan teknik *shifted windows*. Metode *Swin transformer* pertama kali diperkenalkan oleh (Liu *et al.*, 2021) dalam jurnal yang berjudul *Swin Transformer : Hierarchical Vision Transformer using Shifted Windows*. Dalam penelitiannya, *Swin Transformer* sangat efisien dalam mengerjakan berbagai tugas computer vision melalui pendekatan berbasis *shifted windows* yang menghasilkan akurasi sebesar 83,7% dalam tugas mengklasifikasi citra pada *dataset* ImageNet-1K.

Penerapan metode *Swin transformer* sampai saat ini belum diterapkan dalam pengidentifikasian jenis rempah yang berbentuk rimpang. Padahal, *Swin Transformer* dapat bersaing dengan CNN walaupun arsitektur *transformer* masih baru dalam bidang *computer vision* (Moutik *et al.*, 2023). *Swin Transformer* juga mendapatkan skor AUC yang lebih tinggi dibandingkan CNN dalam mendeteksi *Diabetic Retinopathy*, yaitu sebesar 95,7%. Oleh karena itu, pada penelitian ini akan digunakan metode *Swin Transformer* untuk mengidentifikasi citra rempah yang berbentuk rimpang karena *Swin Transformer* memiliki potensi untuk menghasilkan akurasi yang lebih baik dari CNN.

1.2 Rumusan Masalah

Berdasarkan paparan pada latar belakang maka rumusan masalah dalam penelitian ini adalah bagaimana mengimplementasikan arsitektur *Swin Transformer* untuk membantu dalam mengidentifikasi jenis-jenis rempah yang berbentuk rimpang melalui citra gambar.

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini yaitu sebagai berikut:

1. Penelitian ini berfokus pada pembuatan model dengan metode *Swin Transformer* dan pembuatan web dengan menggunakan *framework* Streamlit untuk menerima dan memprediksi foto yang dikirimkan.
2. *Dataset* yang digunakan merupakan kumpulan gambar rempah berbentuk rimpang dengan format warna RGB yang memiliki format ekstensi *file* berupa (.jpg dan .png) yang terdiri dari beberapa foto jahe, kunyit, kencur, dan lengkuas.
3. Model yang dibuat mengimplementasikan arsitektur *Swin Transformer* yang telah dilatih sebelumnya (*pre-trained*).

1.4 Tujuan Penelitian

Penelitian ini memiliki beberapa tujuan, yaitu:

1. Mengetahui akurasi model yang menggunakan arsitektur *Swin Transformer*.
2. Mengembangkan model dengan arsitektur *Swin Transformer* yang dapat mengidentifikasi jenis rempah yang berbentuk rimpang, seperti jahe, kunyit, kencur, dan lengkuas.
3. Mengembangkan web dengan menggunakan *framework* Streamlit yang dapat menerima foto rempah yang berbentuk rimpang dan mengembalikan respon berupa hasil prediksi nama jenis rimpang yang ada pada foto tersebut.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Penelitian ini akan memudahkan pengguna dalam melakukan identifikasi jenis rempah yang berbentuk rimpang.
2. Penelitian ini memberikan kontribusi dalam pengembangan sistem *computer vision* khususnya untuk identifikasi jenis rempah yang berbentuk rimpang .

II. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu bertujuan untuk membandingkan penelitian yang sudah ada dengan penelitian yang akan dilakukan. Selain itu, penelitian terdahulu akan dijadikan sebagai acuan dalam upaya tinjauan Pustaka terkait dengan penelitian yang dilakukan. Penelitian tersebut antara lain:

Tabel 1. Penelitian Terdahulu

Penulis	Judul	Metode	Hasil
(Nisa dan Candra, 2023)	Klasifikasi Jenis Rempah-Rempah Menggunakan Algoritma Convolutional Neural Network	<i>Convolutional Neural Network</i> (CNN)	Penelitian tersebut berhasil membuat model CNN untuk mengklasifikasikan jenis rempah-rempah, seperti jahe, kunyit, kencur, dan lengkuas. Model yang dibuat mendapatkan akurasi sebesar 95% pada tahap pelatihan, 98% pada tahap validasi, dan 90% pada tahap pengujian dengan total <i>dataset</i> sebanyak 1000 gambar.

Penulis	Judul	Metode	Hasil
(Bi et al., 2022)	<i>Development of Deep Learning Methodology for Maize Seed Variety Recognition Based on Improved Swin Transformer</i>	<i>Swin Transformer</i>	Pada penelitian tersebut berhasil membuat model dengan arsitektur <i>Swin Transformer</i> yang telah dimodifikasi untuk mengidentifikasi jenis jagung. Model yang dibuat mendapatkan rata-rata akurasi sebesar 96,4% dan memiliki tingkat akurasi lebih besar dibandingkan dengan arsitektur CNN, seperti <i>ResNet50</i> , <i>AlexNet</i> , dan VGG16.
(Subburaj et al., 2023)	<i>Transfer Learning based Image Classification of Diseased Tomato Leaves with Optimal Fine-Tuning combined with Heat Map Visualization</i>	<i>Convolutional Neural Network (CNN)</i>	Pada penelitian tersebut berhasil membuat model dengan arsitektur <i>InceptionV3</i> untuk mengklasifikasi penyakit pada daun tomat dan mengimplementasikannya pada web menggunakan <i>framework</i> Streamlit.

2.2 Rempah

Rempah adalah bumbu yang berasal dari tumbuhan, baik segar maupun kering, yang dicampurkan pada masakan sebagai penyedap. Sebagai sumber daya hayati, rempah-rempah telah lama memainkan peran penting dalam kehidupan manusia. Bagian tumbuhan yang bersifat aromatik ini digunakan sebagai bumbu, penguat cita rasa, pengharum, dan pengawet makanan secara terbatas. Dalam seni kuliner,

rempah-rempah dimanfaatkan dari berbagai bagian tumbuhan, seperti batang, daun, kulit kayu, umbi, rimpang, akar, biji, bunga, atau bagian lain yang memberikan aroma dan rasa khas (Robi *et al.*, 2019).

2.3 Rimpang

Rimpang atau yang dalam bahasa ilmiah disebut *rhizome* merupakan modifikasi batang tumbuhan yang tumbuh secara horizontal di bawah permukaan tanah (Trimanto. *et al.*, 2018). Sebagai modifikasi dari batang, rimpang memiliki struktur anatomi yang menyerupai anatomi batang dan memiliki peran penting dalam perkembangbiakan vegetatif tumbuhan serta penyimpanan cadangan makanan. Secara tradisional, rimpang telah lama dimanfaatkan oleh berbagai kelompok masyarakat sebagai bahan pangan, obat-obatan tradisional, dan rempah-rempah, dengan beberapa contoh tumbuhan rimpang seperti kunyit (*Curcuma longa*), jahe (*Zingiber officinale*), lengkuas (*Alpinia galanga*), dan kencur (*Kaempferia galanga*). Ilustrasi rimpang dapat dilihat pada Gambar 1.



Gambar 1. Rimpang

2.3.1 Jahe

Jahe (*Zingiber officinale*) merupakan rimpang yang berbentuk beruas-ruas. Kulit luarnya berwarna coklat muda kekuningan dan permukaannya memiliki mata tunas di setiap ruas. Tanaman ini telah dimanfaatkan di kawasan Asia sebagai bahan

bumbu masakan dan obat tradisional sejak ribuan tahun yang lalu. Dalam masakan, jahe tidak hanya melengkapi zat gizi pada menu utama, tetapi juga membantu melancarkan proses pencernaan (Aryanta, 2019). Gambar jahe dapat dilihat pada Gambar 2.



Gambar 2. Jahe

2.3.2 Kunyit

Kunyit (*Curcuma longa*) merupakan rimpang yang berbentuk silindris dengan ruas-ruas, diselubungi kulit berwarna coklat keoranyean dengan permukaan yang berkerut halus. Teksturnya padat dan tidak berserat, serta memiliki karakteristik unik yaitu akan meninggalkan warna kuning pada permukaan yang disentuhnya. Selain sebagai tanaman herbal yang telah lama dikenal, terutama di wilayah Asia Tenggara, kunyit sering digunakan dalam berbagai pengobatan tradisional karena memiliki banyak khasiat, seperti antibakteri, antiinflamasi, dan sebagai penyembuh luka (Azis, 2019). Kunyit juga dikenal sebagai obat demam, penghilang bau badan, serta mampu mengurangi motilitas usus. Manfaat-manfaat ini menjadikan kunyit sebagai salah satu alternatif yang potensial dalam terapi penyakit ringan, terutama yang berkaitan dengan peradangan atau infeksi. Kandungan aktif utama dalam kunyit, yaitu kurkumin, bertanggung jawab atas sebagian besar manfaat kesehatannya (Prabowo *et al.*, 2019). Gambar kunyit dapat dilihat pada Gambar 3.



Gambar 3. Kunyit

2.3.3 Kencur

Kencur (*Kaempferia galanga*) merupakan rimpang yang relatif kecil, dengan bentuk bulat pendek atau menyerupai jemari pendek yang ringkas. Kulitnya berwarna coklat muda dengan permukaan yang halus, sedangkan dagingnya berwarna putih bersih dengan tekstur yang renyah dan padat. Meskipun ukurannya kecil dibandingkan dengan rimpang lainnya, kencur tetap padat berisi. Tanaman ini dikenal sebagai salah satu obat potensial yang banyak digunakan dalam pengobatan tradisional di Asia, dengan manfaat untuk kesehatan seperti mengobati gangguan pencernaan, mengatasi batuk, dan meredakan peradangan. Kencur juga mengandung senyawa bioaktif, termasuk minyak atsiri dan flavonoid, yang berperan penting dalam efek farmakologisnya. Dengan manfaat yang beragam dan potensi ekonomi yang menjanjikan, kencur menjadi salah satu komoditas penting dalam sektor pertanian tanaman obat di Indonesia (Subaryanti *et al.*, 2020). Gambar kencur dapat dilihat pada Gambar 4.



Gambar 4. Kencur

2.3.4 Lengkuas

Lengkuas (*Alpinia galanga*) merupakan rimpang besar yang memanjang dengan struktur yang kokoh. Kulit luarnya memiliki gradasi warna coklat keputihan. Teksturnya keras dan memiliki serat yang lebih kasar dibandingkan dengan jahe, dengan permukaan yang beruas besar dan bertekstur kasar. Sebagai tanaman umbi-umbian yang termasuk dalam famili *Zingiberaceae*, lengkuas sering dimanfaatkan sebagai rempah dalam masakan, bahan minuman, serta obat tradisional. Lengkuas dikenal karena aroma dan rasa pedasnya yang khas, mirip dengan jahe, namun memiliki perbedaan dalam rasa dan tekstur. Tanaman ini banyak dibudidayakan di berbagai negara Asia, seperti Indonesia, India, Cina, dan Sri Lanka, karena permintaan yang tinggi dan manfaatnya yang beragam (JANNAH *et al.*, 2022). Gambar lengkuas dapat dilihat pada Gambar 5.



Gambar 5. Lengkuas

2.4 Machine Learning

Machine learning adalah suatu cabang dari kecerdasan buatan (*Artificial Intelligence*) yang memungkinkan komputer untuk belajar secara mandiri dari data yang diberikan tanpa perlu diprogram secara eksplisit (Roihan *et al.*, 2020). Sistem ini menggunakan algoritma dan model matematis untuk mengenali pola dari data, kemudian membuat prediksi atau keputusan berdasarkan pola tersebut.

Keberhasilan sistem *machine learning* dalam menghasilkan prediksi dan keputusan yang tepat sangat dipengaruhi oleh mutu data yang digunakan untuk pembelajaran (Sarker, 2021). Hal ini disebabkan sistem komputer akan memproses dan belajar dari *dataset* yang disediakan sebagai bahan pembelajaran. Semakin tinggi kualitas data yang digunakan, semakin akurat pula hasil prediksi yang dihasilkan. Begitu pula sebaliknya, *dataset* yang berkualitas rendah cenderung menghasilkan prediksi yang kurang akurat. *Machine learning* sendiri dapat diklasifikasikan ke dalam tiga kategori utama, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*.

2.4.1 *Supervised Learning*

Supervised Learning adalah metode dalam *machine learning* di mana model dilatih menggunakan data yang sudah dilabeli, yaitu data yang disertai dengan *output* yang benar (Roihan *et al.*, 2020). Proses ini dimulai dengan mengumpulkan data yang telah diberi label, kemudian model dilatih untuk mempelajari hubungan antara *input* dan *output* yang benar. Selama pelatihan, algoritma menyesuaikan parameter model untuk mengurangi kesalahan dalam prediksi. Setelah model dilatih, kinerjanya dievaluasi menggunakan data yang terpisah dari data pelatihan untuk mengukur akurasi prediksi pada data baru. Contoh penerapan *supervised learning* termasuk regresi linier, yang digunakan untuk memprediksi nilai kontinu seperti harga rumah berdasarkan fitur-fitur tertentu, dan klasifikasi, yang digunakan untuk mengkategorikan data, seperti mengidentifikasi email sebagai spam atau bukan spam.

2.4.2 *Unsupervised Learning*

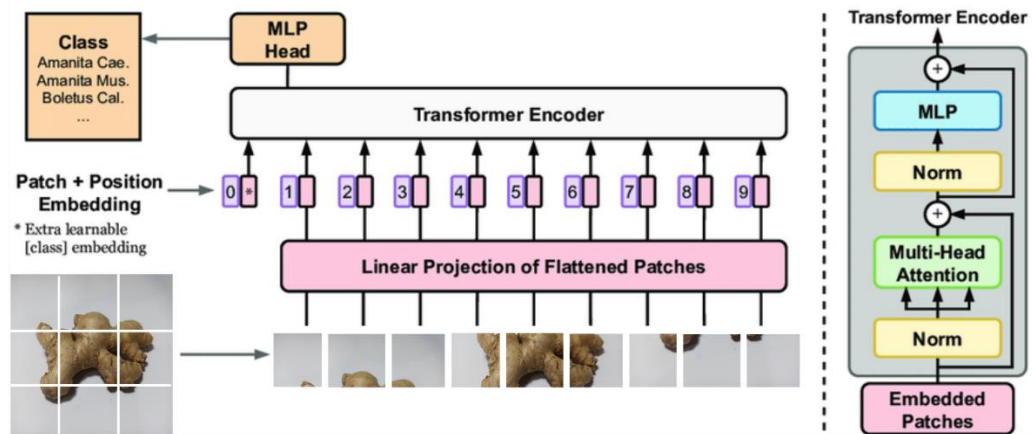
Unsupervised Learning adalah metode dalam *machine learning* yang menggunakan data yang tidak dilabeli, yaitu data yang tidak disertai dengan *output* atau hasil yang benar (Roihan *et al.*, 2020). Tujuan dari metode ini adalah untuk menemukan pola atau struktur tersembunyi dalam data tanpa adanya panduan dari label yang sudah ditentukan. Prosesnya dimulai dengan menganalisis data yang belum dilabeli untuk mengidentifikasi pola atau hubungan yang ada. Contoh penerapan *unsupervised learning* termasuk *clustering*, yang mengelompokkan data ke dalam grup berdasarkan kesamaan fitur, seperti dalam segmentasi pelanggan, dengan algoritma seperti *K-Means* dan *Hierarchical Clustering*. Selain itu, asosiasi menemukan aturan atau hubungan yang sering muncul bersamaan dalam data, seperti dalam sistem rekomendasi produk, dengan algoritma seperti *Apriori* dan *FP-Growth*.

2.4.3 Reinforcement Learning

Reinforcement Learning adalah metode dalam *machine learning* di mana agen belajar melalui proses coba-coba dan mendapatkan umpan balik dari lingkungan. Dalam metode ini, agen membuat keputusan secara berurutan dan menerima umpan balik berupa hadiah atau hukuman berdasarkan hasil tindakannya, dengan tujuan untuk memaksimalkan total hadiah yang diperoleh seiring waktu (Roihan *et al.*, 2020). Proses ini melibatkan penggunaan algoritma seperti *Q-Learning* dan *Deep Q-Network* (DQN), yang membantu agen mengoptimalkan strateginya berdasarkan pengalaman. Contoh penerapan *reinforcement learning* termasuk permainan video, di mana agen belajar bermain dengan mengeksplorasi berbagai strategi, dan robotika, di mana agen belajar menyelesaikan tugas fisik seperti berjalan atau mengambil objek dengan menyesuaikan tindakan terhadap kondisi lingkungan.

2.5 Vision Transformer (ViT)

Vision Transformer merupakan sebuah arsitektur pemrosesan gambar yang berlandaskan pada arsitektur *transformer*. *Vision Transformer* bekerja dengan cara memperlakukan gambar sebagai kumpulan "*token*" yang kemudian diproses secara berurutan (Dosovitskiy *et al.*, 2020). Arsitektur *Vision Transformer* mirip dengan BERT, tetapi masukan pada ViT berupa gambar, bukan kata. Gambaran umum arsitektur *Vision Transformer* dapat dilihat pada Gambar 6.



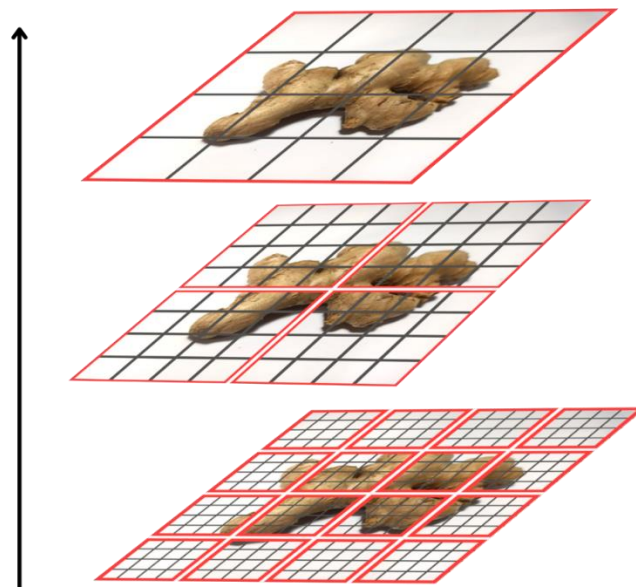
Gambar 6. Arsitektur *Vision Transformer* (ViT)

Alur kerja ViT diawali dengan memecah gambar menjadi beberapa bagian kecil atau *patch* yang berfungsi sebagai *token*, seperti kata dalam *Natural Language Processing* (NLP). Setiap *patch* ini kemudian diubah menjadi representasi vektor melalui modul *patch embedding*. Setelah itu, informasi posisi masing-masing *patch* ditambahkan menggunakan *position embedding* agar model dapat mengenali urutan atau letak setiap bagian dalam gambar. *Token* yang telah diperkaya dengan informasi posisi ini kemudian diproses oleh *encoder transformer*, yang terdiri dari blok *multihead self-attention* (MSA) dan blok *multilayer perceptron* (MLP). Di akhir proses, sebuah MLP tambahan digunakan untuk menghasilkan prediksi akhir.

Dalam penelitian Dosovitskiy et al., 2020), ViT telah menunjukkan kinerja yang sangat baik, dengan akurasi mencapai 99,74% pada *dataset* Oxford 102 Flowers dan 97,56% pada Oxford-IIIT-Pets. Akan tetapi, model ViT yang digunakan merupakan *pre-trained* model yang sebelumnya dilatih pada *dataset* besar seperti ImageNet-21k. Hal ini dilakukan karena ViT membutuhkan data dalam jumlah besar untuk menghasilkan akurasi yang tinggi.

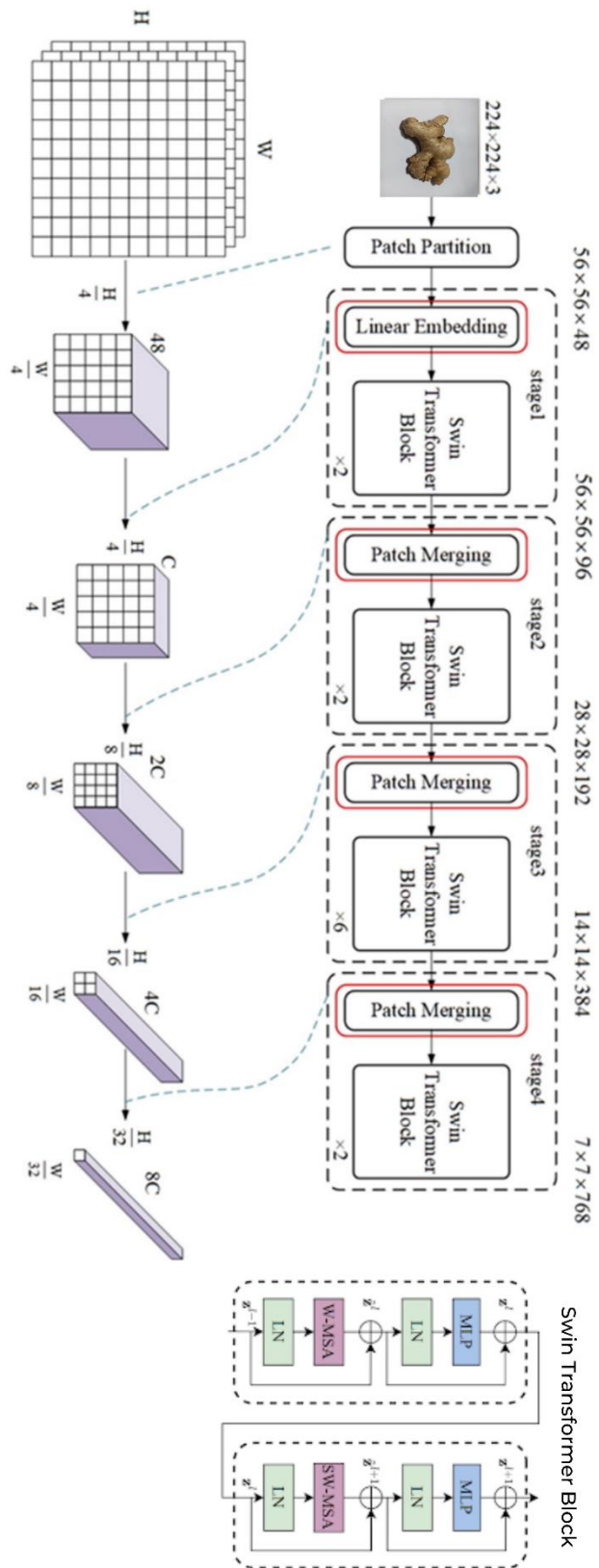
2.5.1 Swin Transformer

Swin Transformer merupakan varian dari *Vision Transformer* (ViT) dengan keunggulan mampu menangani gambar beresolusi tinggi secara efisien. Arsitektur ini memperkenalkan mekanisme *shifted windows* untuk mengurangi kompleksitas komputasi. Selain itu, *Swin Transformer* menggunakan *hierarchical patches* yang memungkinkan ekstraksi informasi dari detail kecil hingga struktur global gambar, sehingga dapat menjadi lebih efektif dalam memahami suatu gambar (Liu *et al.*, 2021). Mekanisme *hierarcichal patches* dapat dilihat pada Gambar 6.



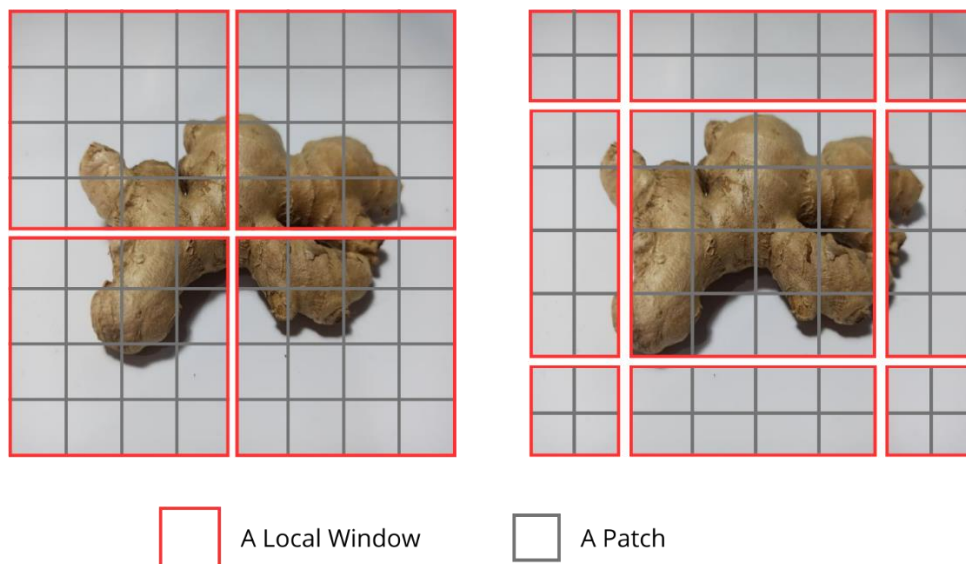
Gambar 7. Hierarcichal Patches

Selanjutnya, arsitektur *Swin Transformer* terdiri dari beberapa komponen penting yang dapat dilihat pada Gambar 8, seperti tahap *patch partitioning*, *linear embedding*, dan beberapa *Swin Transformer* Blok. Setiap blok terdiri dari dua *Multihead Self-Attention*, yaitu *Windows-MSA* (W-MSA) dan *Shfited Windows-MSA* (SW-MSA) dan diikuti dengan lapisan normalisasi (*LayerNorm*) dan *residual connection* serta *Multi-Layer Perceptron* (MLP).



Gambar 8. Arsitektur Swin Transformer

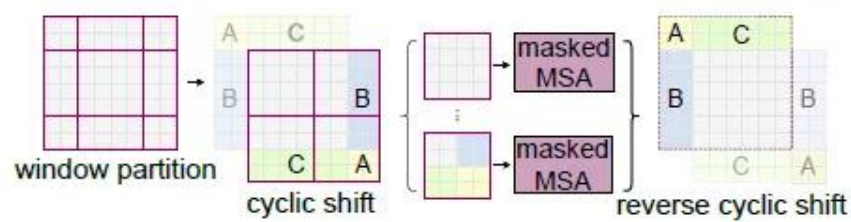
Pada tahap *patch partition*, gambar dibagi menjadi beberapa *patch* dengan menggabungkan setiap 4×4 *pixel* menjadi satu *patch* (*patch size* = 4). Nilai empat dipilih karena merupakan ukuran yang optimal antara kompleksitas komputasi dan kemampuan untuk mengekstraksi detail pada gambar. Dengan demikian, gambar berukuran $H \times W \times 3$ diubah menjadi *patch* dengan ukuran $(H/4) \times (W/4) \times 48$ dan *patch* diperlakukan seperti *token* dalam konsep NLP (Liu *et al.*, 2021). Pada *stage* 1, *patch* diproses melalui modul *Linear Embedding* untuk mengubah dimensinya menjadi C dimensi, di mana C adalah *hyperparameter* yang menentukan kapasitas model *transformer*. Di dalam blok *Swin Transformer*, setiap *patch* melewati modul W-MSA (*Window-based Multi-Head Self-Attention*), yang menerapkan MSA pada *patch* di dalam satu *window* untuk memahami hubungan antar *patch*. Ilustrasi W-MSA dapat dilihat pada Gambar 9. Setelah itu, *patch* diproses melalui modul MLP (*Multi-Layer Perceptron*), dan *Layer Normalization* dan *residual connection* diterapkan setelah setiap modul W-MSA dan MLP untuk menjaga stabilitas dan efisiensi pada saat *training*.



Gambar 9. *Window-based Multi-Head Self-Attention*

Pada blok kedua, proses serupa dilakukan, tetapi modul W-MSA digantikan oleh modul SW-MSA (*Shifted Window Multi-Head Self-Attention*). Pada modul ini,

dilakukan pergeseran *window* sebesar $M/2$, di mana M adalah ukuran *window* dengan dimensi $M \times M$. Setelah itu, *cyclic shift* diterapkan untuk mengisi bagian *window* yang kosong akibat pergeseran tersebut. Selanjutnya, dilakukan *Mask-MSA* untuk membatasi interaksi hanya pada *patch* yang berdekatan dari posisi sebelumnya, karena terdapat beberapa *patch* yang sebelumnya tidak berdekatan. Terakhir, *reverse cyclic shift* diterapkan untuk mengembalikan posisi *patch* seperti semula. Ilustrasi SW-MSA dapat dilihat pada Gambar 10.



Gambar 10. *Shifted Window based Multi-Head Self-Attention*

Setelah melalui blok *Swin Transformer* pada *stage* 1, *patch* akan digabung (*merging*) untuk membentuk *hierarchical patch*. Proses *patch merging* dilakukan dengan cara menyatukan empat *patch* berukuran 2×2 menjadi satu *patch*, sehingga menghasilkan ukuran *patch* yang baru sebesar $(H/8) \times (W/8) \times 4C$. Namun, setelah melewati blok *Swin Transformer* pada *stage* 2, ukuran *patch* berubah karena dinormalisasi menjadi $(H/8) \times (W/8) \times 2C$. Proses serupa juga dilakukan pada *stage* 3 dan 4. Perbedaannya, pada *stage* 3 terdapat 6 blok *Swin Transformer*, sedangkan pada *stage* 1, 2, dan 4 hanya terdapat 2 blok *Swin Transformer*.

Pada penelitian yang dilakukan (Liu *et al.*, 2021), *Swin Transformer* berhasil mendapatkan akurasi yang cukup tinggi, yaitu sebesar 83,7% dalam tugas mengklasifikasi citra pada *dataset* ImageNet-1K. Meskipun *Swin Transformer* menunjukkan performa yang menjanjikan, arsitektur ini memiliki keterbatasan, yaitu diperlukannya jumlah *dataset* yang sangat besar untuk mencapai tingkat akurasi yang optimal. Namun, kendala ini dapat diatasi dengan menerapkan teknik *transfer learning*, yaitu dengan menggunakan model yang telah dilatih sebelumnya (*pre-trained model*) pada *dataset* yang besar seperti ImageNet. Setelah

menggunakan *pre-trained* model, kemudian dilakukan *fine-tuning* menggunakan *dataset* yang lebih kecil dan spesifik sesuai dengan tugas yang diinginkan.

2.6 Transfer Learning

Transfer learning adalah teknik yang memanfaatkan model yang sebelumnya telah dilatih menggunakan *dataset* yang lebih besar, untuk mengekstraksi fitur pada *dataset* yang lebih kecil atau *dataset* dalam kasus yang berbeda (Morales *et al.*, 2018). Proses ini dilakukan dengan mengambil model yang telah dilatih sebelumnya dengan *dataset* yang besar (*pre-trained* model) dan menambahkan beberapa lapisan *fully-connected layer* baru untuk melakukan klasifikasi pada *dataset*, baik dengan kasus yang sama dengan *dataset* yang baru, maupun dengan kasus yang baru.

2.7 Preprocessing

Tahap *preprocessing* merupakan langkah penting dalam mempersiapkan data agar siap digunakan untuk analisis atau pembuatan model. Dalam konteks *machine learning*, tahap ini berperan vital dalam mentransformasi data mentah ke dalam bentuk yang dapat diolah oleh model. Berbagai teknik dapat diterapkan dalam *preprocessing*, termasuk pembersihan data, normalisasi, *scaling*, ekstraksi fitur, dan reduksi dimensi. Tujuan utama dari *preprocessing* adalah untuk meningkatkan mutu data serta memudahkan model dalam mengenali pola dan menghasilkan prediksi yang lebih tepat. Selain itu, *preprocessing* juga bermanfaat dalam mengurangi kerumitan setiap iterasi dan mencegah pembaruan berulang pada struktur data yang identik (Song *et al.*, 2021).

2.8 Confusion Matrix

Confusion matrix adalah alat yang sangat berguna untuk mengevaluasi model karena memberikan gambaran komprehensif mengenai kinerja suatu model. Matriks ini menampilkan tabel yang membandingkan prediksi model dengan nilai

sebenarnya (*ground truth*). Dengan menganalisis berbagai kategori dalam matriks, seperti *true positive*, *false positive*, *true negative*, dan *false negative*, dapat diperoleh wawasan tentang metrik utama seperti akurasi, presisi, *recall*, dan lainnya. Ini memungkinkan penilaian yang lebih mendalam terhadap efektivitas model dalam mengaitkan fitur dengan output model, serta dapat mengidentifikasi area yang perlu diperbaiki (Arias-Duart *et al.*, 2023). Kategori prediksi model pada *confusion matrix* dalam bentuk tabel dapat dilihat pada Tabel 1.

Tabel 2. Kategori prediksi model pada *confusion matrix*

Prediksi \ Sampel	Positif	Negatif
Positif	<i>True Positive</i>	<i>False Negative</i>
Negatif	<i>False Positive</i>	<i>True Negative</i>

Keterangan:

True Positive (TP) = Data positif yang terdeteksi positif

False Positive (FP) = Data positif yang terdeteksi negatif

True Negative (TN) = Data negatif yang terdeteksi negatif

False Negative (FN) = Data negatif yang terdeteksi positif

2.8.1 Akurasi

Akurasi adalah nilai pengukuran paling umum untuk mengevaluasi efektivitas dari sebuah algoritma dengan mengestimasi kemungkinan nilai benar dari sebuah kelas (Bekkar *et al.*, 2013). Akurasi adalah metrik evaluasi yang mengukur proporsi prediksi yang benar dari seluruh prediksi yang dilakukan oleh model klasifikasi. Metrik ini dihitung dengan menjumlahkan kasus positif benar dan negatif benar, kemudian membaginya dengan jumlah total kasus. Persamaan akurasi dapat dilihat pada Persamaan 1.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (1)$$

2.8.2 Presisi

Presisi (*precision*) adalah metrik yang digunakan untuk mengukur ketepatan prediksi model pada kelas positif. Presisi menunjukkan proporsi prediksi positif yang benar dibandingkan dengan keseluruhan prediksi positif yang dihasilkan oleh model. Dengan kata lain, semakin tinggi presisi, semakin sedikit kesalahan prediksi positif yang dilakukan oleh model (Arias-Duart *et al.*, 2023). Persamaan presisi dapat dilihat pada Persamaan 2.

$$Presisi = \frac{TP}{TP + FP} \times 100\% \quad (2)$$

2.8.3 Recall

Recall adalah metrik yang digunakan untuk mengukur sensitivitas atau kemampuan model dalam mendeteksi seluruh kasus positif yang sebenarnya. *Recall* menunjukkan proporsi prediksi positif yang benar dibandingkan dengan total kasus positif yang ada dalam data. Semakin tinggi nilai *recall*, semakin baik model dalam mengidentifikasi semua data positif (Arias-Duart *et al.*, 2023). Persamaan *recall* dapat dilihat pada Persamaan 3.

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (3)$$

2.8.4 F1-Score

F1 Score merupakan metode pengujian yang menghitung gabungan antara nilai presisi dan *recall*. *F1 Score* dihitung dengan cara membagi hasil perkalian nilai

presisi dan *recall* dengan hasil penjumlahan nilai presisi dan *recall*. Persamaan *F1 score* dapat dilihat pada Persamaan 4.

$$F1\ Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (4)$$

2.9 Streamlit

Streamlit adalah *framework open-source* yang dirancang untuk memudahkan pengembangan aplikasi web interaktif dengan mudah, khususnya untuk proyek-proyek *data science* dan *machine learning*. Streamlit memungkinkan pengguna untuk membuat aplikasi hanya dengan menggunakan kode Python, tanpa memerlukan keahlian dalam pengembangan antarmuka web tradisional. Pada penelitian (Kharisma *et al.*, 2023) yang menggunakan algoritma *Decision Tree* untuk ulasan aplikasi, Streamlit berperan dalam memfasilitasi proses *deployment* model ke dalam bentuk aplikasi berbasis web, yang memudahkan pengguna untuk melakukan analisis secara langsung.

2.10 Black Box Testing

Menurut (Febiharsa *et al.*, 2018), *black box testing* adalah metode pengujian yang berfokus pada fungsionalitas, yaitu menguji bagaimana perangkat lunak merespons input yang diberikan oleh pengguna untuk menghasilkan *output* yang diinginkan, tanpa memperhatikan proses internal atau kode program yang dijalankan. Dengan demikian, dapat disimpulkan bahwa *black box testing* adalah teknik pengujian perangkat lunak di mana proses internalnya tidak diketahui, sehingga pengujian diperlukan. Teknik ini bertujuan untuk memastikan bahwa aplikasi bekerja sesuai dengan kebutuhan pengguna dan tidak memiliki kesalahan dalam implementasi fungsionalnya. Blackbox testing sering digunakan dalam berbagai tahap

pengembangan perangkat lunak, seperti pengujian unit, integrasi, dan penerimaan, karena memberikan perspektif pengguna akhir terhadap kualitas perangkat lunak.

2.11 Kurva ROC

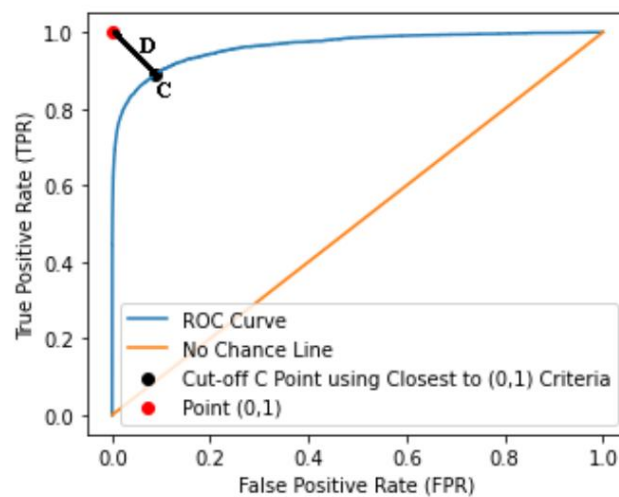
Receiver Operating Characteristic (ROC) curve adalah alat evaluasi yang umum digunakan untuk menilai kinerja model klasifikasi biner. ROC curve menggambarkan hubungan antara True Positive Rate (TPR) dan False Positive Rate (FPR) pada berbagai nilai ambang keputusan, memberikan wawasan tentang kemampuan model dalam membedakan antara dua kelas. Area Under the Curve (AUC) dari ROC curve, yang dikenal sebagai AUC-ROC sehingga memberikan ukuran kuantitatif dari kemampuan diskriminatif model secara keseluruhan. Nilai AUC berkisar antara 0,5 (tidak ada kemampuan diskriminatif) hingga 1,0 (kemampuan diskriminatif sempurna), dengan nilai yang lebih tinggi menunjukkan kinerja klasifikasi yang lebih baik (Carrington et al., 2023).

Dalam penelitian ini, ROC curve dan AUC-ROC digunakan untuk mengevaluasi sejauh mana model dapat membedakan antara data in-distribution (ID) dan data out-of-distribution (OOD). Model yang overfit cenderung memberikan prediksi dengan kepercayaan tinggi pada data OOD, yang seharusnya tidak dikenalnya, sehingga menghasilkan nilai AUC yang lebih rendah. Sebaliknya, model yang mampu mengenali ketidakpastian pada data OOD akan memberikan prediksi dengan kepercayaan lebih rendah, menghasilkan nilai AUC yang lebih tinggi, dan menunjukkan generalisasi yang lebih baik. Oleh karena itu, ROC curve dan AUC-ROC merupakan alat penting dalam menilai kemampuan generalisasi model dan mendeteksi overfitting.

Selain sebagai alat evaluasi, ROC curve juga dapat digunakan untuk menentukan threshold klasifikasi optimal menggunakan pendekatan *Closest to (0,1) Criteria*. Metode ini memilih titik pada ROC curve yang paling dekat dengan titik ideal (TPR=1, FPR=0), yaitu titik di mana model mencapai sensitivitas maksimal dengan kesalahan minimal. Secara matematis, jarak ke titik (0,1) dihitung menggunakan

rumus Euclidean distance seperti yang dapat dilihat pada persamaan 5, dan threshold yang menghasilkan jarak terpendek dianggap optimal. Dengan menggunakan kriteria ini, model dapat diatur untuk mencapai keseimbangan terbaik antara deteksi benar (true positives) dan kesalahan positif palsu (false positives), yang sangat penting dalam aplikasi OOD detection (Kanika et al., 2021). Gambar kurva ROC dapat dilihat pada Gambar 11.

$$D = \sqrt{(1 - TPR)^2 + (FPR)^2} \quad (5)$$



Gambar 11. Gambar Kurva ROC dengan Closest to (0,1) Criteria

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan di Laboratorium Rekayasa Perangkat Lunak (RPL) di Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung. Tempat pelaksanaan berada di Jalan Prof. Dr. Soemantri Brojonegoro No. 1, Gedung Meneng, Bandar Lampung yang berlangsung pada semester ganjil tahun akademik 2024/2025 dimulai dari bulan Oktober 2024 hingga penyelesaian pada bulan Februari 2025.

3.2 Perangkat Penelitian

Penelitian ini akan menggunakan dua jenis alat, yaitu perangkat lunak dan perangkat keras. Berikut merupakan spesifikasi alat yang akan digunakan selama penelitian.

3.2.1 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian ini adalah sebuah laptop dengan detail sebagai berikut.

Merk : Dell

System Model : Dell G15 5515

CPU : AMD Ryzen 5 5600H

GPU : NVIDIA GeForce RTX 3050 laptop GPU

RAM : 8GB DDR4

Penyimpanan : SSD M.2 NVME Generasi 3 512GB Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam proses penelitian ini yaitu:

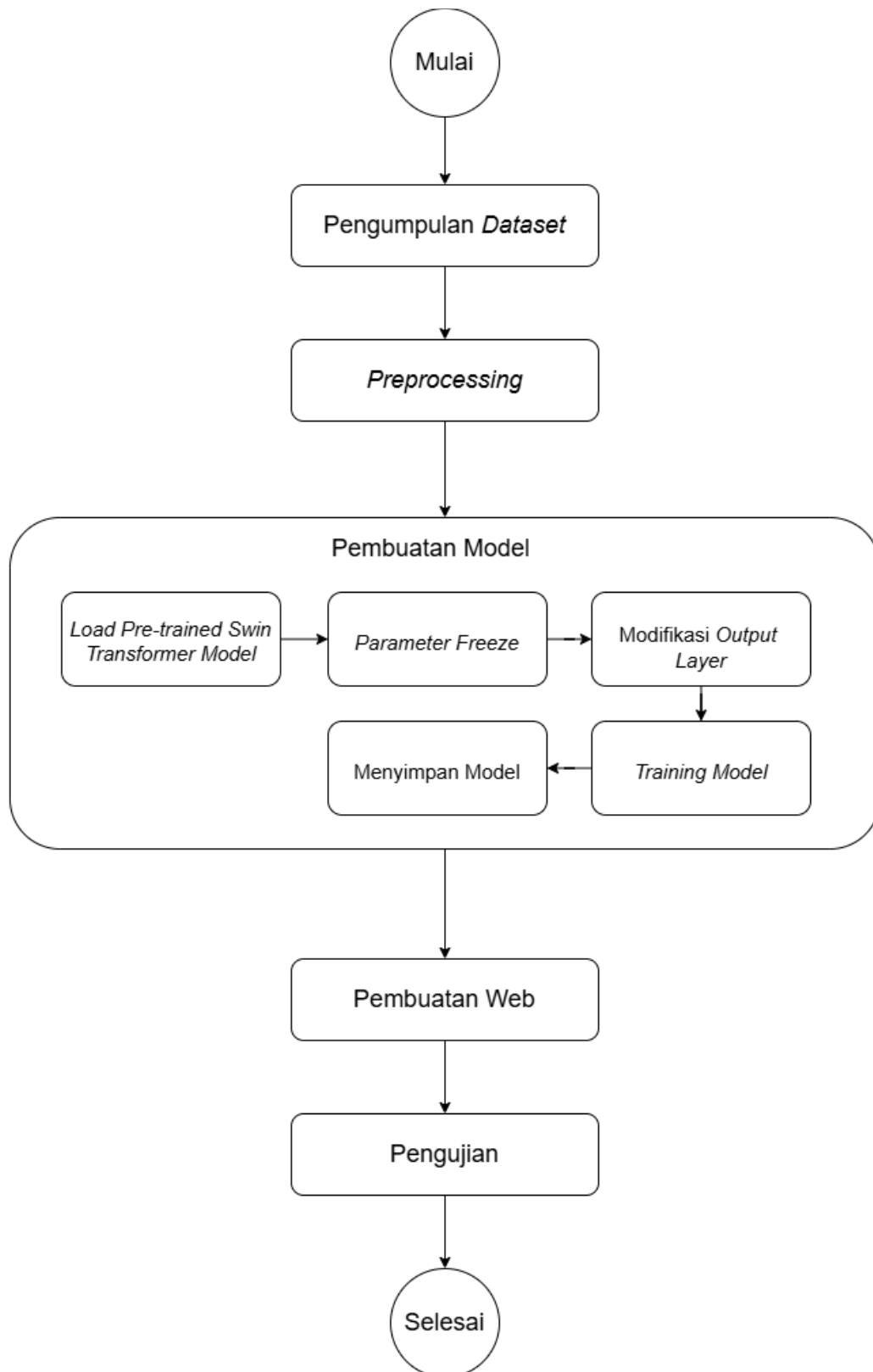
1. Sistem Operasi Windows 11 Home 64-bit
2. Google Chrome versi 129
3. Visual Studio Code versi 1.94
4. Python versi 3.11.9 dengan *library* PyTorch dan Streamlit

3.3 Dataset

Dataset yang digunakan dalam penelitian ini berupa kumpulan foto rempah berbentuk rimpang, seperti jahe, kunyit, kencur, dan lengkuas. Total *dataset* yang digunakan dalam penelitian ini adalah 1400 foto dimana terdapat 350 foto untuk masing-masing kelas. *Dataset* tersebut dipisahkan menjadi data latih, data validasi, dan data uji.

3.4 Tahap Penelitian

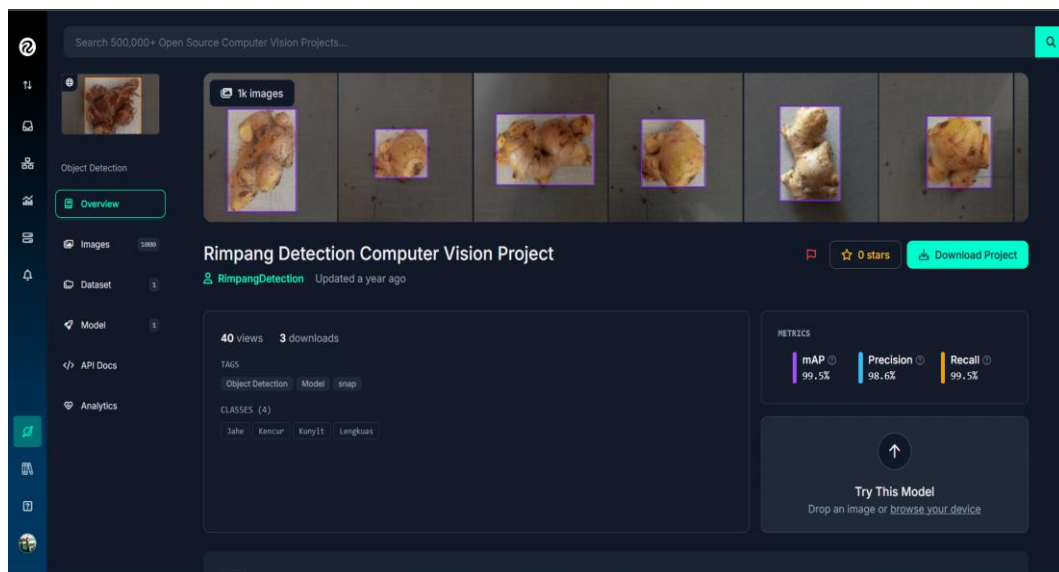
Tahapan yang akan dilakukan pada penelitian ini antara lain dimulai dari pengumpulan *dataset*, *preprocessing dataset*, pembuatan model, pembuatan web, dan pengujian. Alur dari tahapan yang akan dilakukan pada penelitian ini diilustrasikan pada Gambar 12.



Gambar 12. Tahap Penelitian

3.4.1 Pengumpulan *Dataset*

Dataset pada penelitian ini diperoleh dari internet dan pengambilan data secara langsung. Data yang di ambil dari internet bersumber dari *website* roboflow, roboflow adalah *platform* serupa dengan Kaggle yang menyediakan layanan pengelolaan *dataset* dan pelatihan model untuk aplikasi *computer vision* seperti yang dapat dilihat pada Gambar 13. Sedangkan, *dataset* yang diambil secara langsung dilakukan dengan cara memfoto rempah yang berbentuk rimpang dengan menggunakan kamera *handphone*. *Dataset* terpilih sebanyak 1400 citra gambar rempah berbentuk rimpang dengan 4 kelas yang terdapat 350 gambar pada tiap kelasnya, kelas tersebut yaitu jahe, kunyit, kencur, dan lengkuas.



Gambar 13. *Website* roboflow

3.4.2 *Preprocessing Dataset*

Tahapan ini bertujuan untuk meningkatkan kualitas *dataset* agar meningkatkan efisiensi dan efektifitas pada saat training model dilakukan. Langkah-langkah preprocessing pada penelitian ini adalah sebagai berikut:

a) *Merge Dataset*

Tahapan awal dalam proses *preprocessing* data pada penelitian ini adalah penggabungan (*merge*) dataset dari dua sumber yang berbeda. *Dataset* pertama diperoleh dari *platform* Roboflow yang terdiri dari 1.000 gambar dengan empat kelas rimpang, yakni jahe, kunyit, kencur, dan lengkuas, dengan total gambar masing-masing kelas sebanyak 250 gambar. Sedangkan *dataset* kedua merupakan hasil pengambilan gambar secara langsung dengan menggunakan kamera yang menghasilkan 400 gambar dengan total masing-masing kelas sebanyak 100 gambar. Untuk memudahkan proses pengolahan data lebih lanjut, kedua dataset tersebut digabungkan ke dalam satu direktori utama yang bernama *merge_dataset*, dan setiap kelas memiliki subdirektornya masing-masing. Pada tahap ini, dilakukan penyesuaian format gambar agar seragam dalam format JPEG, serta penomoran ulang file untuk memastikan tidak terjadi duplikasi nama file saat proses penggabungan. Proses ini dilakukan secara otomatis menggunakan skrip Python yang memanfaatkan *library os*, *shutil*, dan *PIL (Python Imaging Library)*, dengan mempertimbangkan pengelompokan berdasarkan kelas. Penggabungan *dataset* ini menjadi langkah penting untuk memastikan data yang digunakan dalam pelatihan model Swin Transformer berada dalam struktur yang terorganisir dan siap untuk tahapan *preprocessing* lanjutan.

b) *Split Dataset*

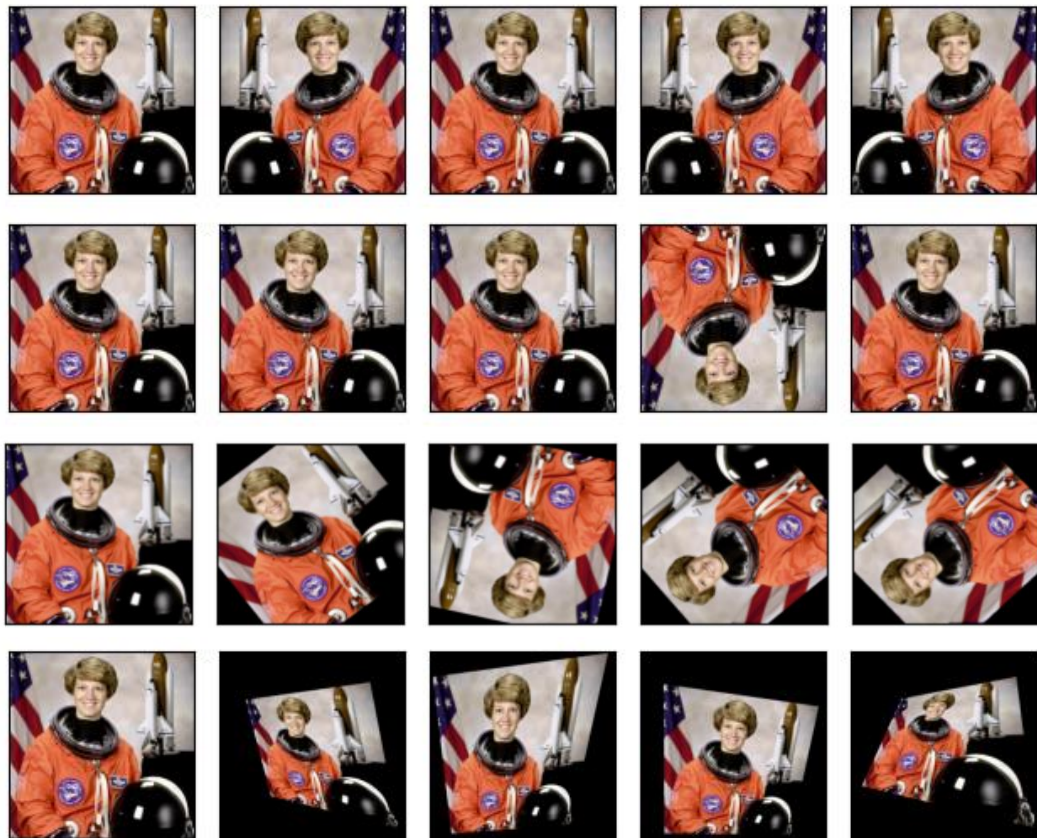
Setelah *dataset* berhasil digabungkan, tahap berikutnya adalah membagi *dataset* ke dalam tiga subset, yaitu *train*, *validation*, dan *test*. Pembagian dilakukan dengan rasio 60% untuk *train*, 20% untuk *validation*, dan 20% untuk *test*. Dari total 1.400 gambar yang tersedia, diperoleh sebanyak 840 gambar untuk *train*, 280 gambar untuk *validation*, dan 280 gambar untuk *test*. Pembagian dengan rasio ini dipilih karena rasio 60:20:20 sering digunakan dalam berbagai penelitian sebagai pendekatan yang optimal untuk menghindari overfitting dan memastikan model dapat diuji pada data yang representatif (Ye et al., 2019). Setiap subset disimpan ke dalam folder terpisah dengan struktur berdasarkan kelas masing-masing. Tahapan

ini bertujuan untuk mempersiapkan data yang akan digunakan dalam proses pelatihan, validasi, dan pengujian model Swin Transformer secara sistematis.

c) Augmentasi

Setelah pembagian *dataset* selesai, tahap berikutnya adalah melakukan augmentasi data pada *training set* untuk meningkatkan keragaman dan variasi dalam dataset pelatihan. Augmentasi ini diterapkan hanya pada data *train* untuk membantu model belajar mengenali pola yang lebih beragam. Teknik augmentasi yang digunakan meliputi *random horizontal flip*, *random vertical flip*, *random rotation*, dan *random perspective* seperti yang dapat dilihat pada Gambar 14. *Random Horizontal Flip* merupakan Teknik membalik gambar secara horizontal dengan acak. *Random Vertical Flip* mirip dengan teknik pertama, tetapi membalik gambar secara vertikal. *Random Rotation* merupakan Teknik memutar gambar secara acak dalam rentang 0 hingga 180 derajat, sehingga model bisa mengenali objek yang mungkin muncul dalam orientasi yang berbeda. Terakhir, *Random Perspective*, yaitu teknik mengubah sudut pandang gambar, mensimulasikan kondisi pengambilan gambar yang berbeda untuk objek yang sama, yang membantu model menangani distorsi perspektif.

Meskipun jumlah data pelatihan tetap 840 gambar, teknik-teknik ini memungkinkan model untuk belajar dari gambar-gambar yang telah diubah-ubah, memperkaya dataset pelatihan dengan variasi gambar yang lebih banyak. Proses augmentasi ini menggunakan pustaka *PyTorch* dengan berbagai fungsi bawaan, dan bertujuan untuk meningkatkan kemampuan model dalam menghadapi berbagai variasi gambar pada data uji yang belum pernah dilihat sebelumnya.



Gambar 14. Ilustrasi Augmentasi Dataset

d) *Resize*

Tahap *preprocessing* yang kedua adalah proses *resize* gambar, pada proses ini setiap gambar dalam *dataset* akan diubah ukurannya agar memiliki dimensi yang seragam. Pada proses ini gambar yang ada pada *dataset* diubah ukurannya menjadi 224x224 pixel. Hal ini bertujuan agar ukuran *dataset* sama dengan ukuran *input image* pada arsitektur *Swin Transformer*, yaitu 224x224 pixel. Proses *resize* ini penting dilakukan karena gambar-gambar dalam *dataset* mentah umumnya memiliki dimensi yang beragam, sementara model *deep learning* membutuhkan input dengan ukuran yang konsisten untuk dapat melakukan pembelajaran dengan baik. Penyeragaman ukuran ini juga membantu dalam optimalisasi komputasi dan memastikan bahwa setiap fitur visual dapat diproses secara konsisten. Ilustrasi *resize* dapat dilihat pada Gambar 15.

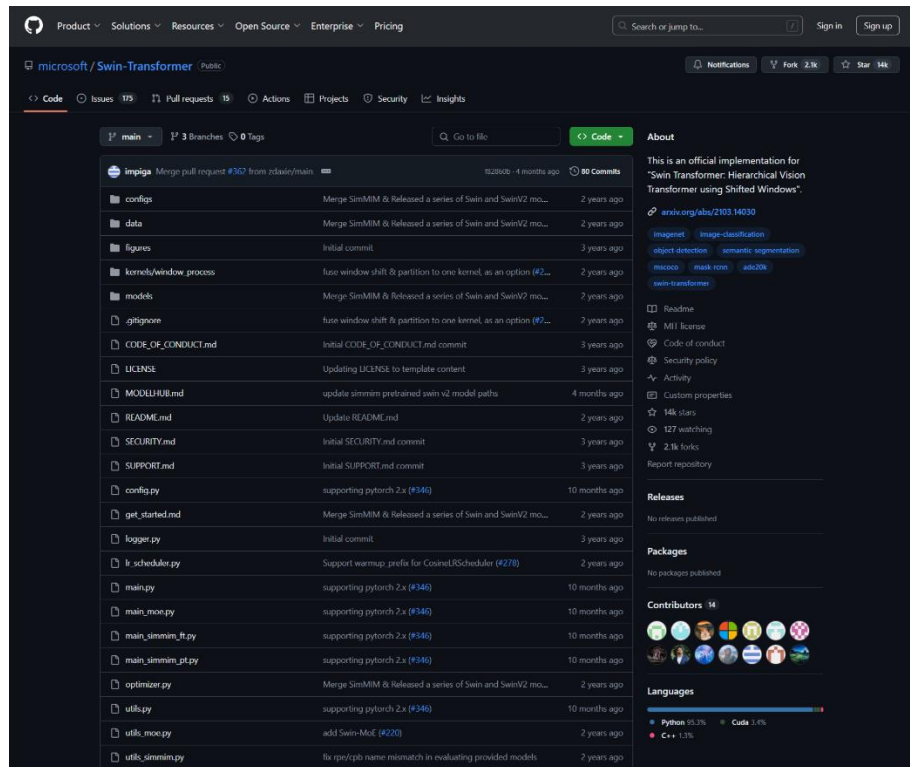


Gambar 15. Ilustrasi *Resize Dataset*

3.4.3 Pembuatan Model

a) *Load pre-trained model*

Tahap pertama dalam proses pengembangan model adalah *load pre-trained model*. Pada tahap ini, dilakukan pembuatan model *Swin Transformer* dengan inisialisasi parameter awal. Setelah model berhasil dibuat, dilakukan pengunduhan parameter yang berasal dari model Swin Transformer yang telah dilatih sebelumnya (*pre-trained model*) menggunakan *dataset* ImageNet1K. Parameter tersebut disimpan dalam format file dengan ekstensi (.pth). Selanjutnya, parameter dari pre-trained model tersebut dimasukkan ke dalam model Swin Transformer yang telah diinisialisasi sebelumnya untuk meningkatkan performa dan mempercepat proses pelatihan. Kode untuk model Swin Transformer dan parameter *pre-trained* yang digunakan dalam penelitian ini berasal dari penelitian (Liu et al., 2021) yang dapat dilihat pada Gambar 16 dan Gambar 17



Gambar 16. Sumber Model Swin Transformer

Main Results on ImageNet with Pretrained Models									
ImageNet-1K and ImageNet-22K Pretrained Swin-V1 Models									
name	pretrain	resolution	acc@1	acc@5	#params	FLOPs	FPS	22K model	
Swin-T	ImageNet-1K	224x224	81.2	95.5	28M	4.5G	755	-	g
Swin-S	ImageNet-1K	224x224	83.2	96.2	50M	8.7G	437	-	g
Swin-B	ImageNet-1K	224x224	83.5	96.5	88M	15.4G	278	-	g
Swin-B	ImageNet-1K	384x384	84.5	97.0	88M	47.1G	85	-	
Swin-T	ImageNet-22K	224x224	80.9	96.0	28M	4.5G	755	github/baidu/config	
Swin-S	ImageNet-22K	224x224	83.2	97.0	50M	8.7G	437	github/baidu/config	
Swin-B	ImageNet-22K	224x224	85.2	97.5	88M	15.4G	278	github/baidu/config	
Swin-B	ImageNet-22K	384x384	86.4	98.0	88M	47.1G	85	github/baidu	
Swin-L	ImageNet-22K	224x224	86.3	97.9	197M	34.5G	141	github/baidu/config	
Swin-L	ImageNet-22K	384x384	87.3	98.2	197M	103.9G	42	github/baidu	

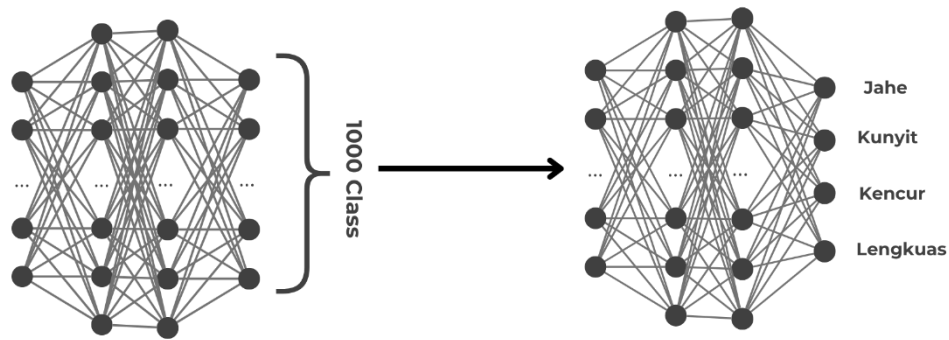
Gambar 17. Sumber Pre-trained Model

b) *Parameter freeze*

Tahap kedua dalam pengembangan model adalah *parameter freeze*. Pada tahap ini, parameter yang sebelumnya telah dimasukkan ke dalam model dari hasil *pre-trained* akan dibekukan (*freeze*), sehingga nilai-nilai pada parameter tersebut tidak akan mengalami perubahan selama proses pelatihan. Pembekuan parameter ini bertujuan untuk mempertahankan "pengetahuan" atau nilai parameter optimal yang telah diperoleh dari pelatihan awal menggunakan *dataset* skala besar, seperti ImageNet1K. Dengan menjaga stabilitas parameter tersebut, proses pelatihan pada *dataset* yang lebih kecil, seperti *dataset* rempah berbentuk rimpang yang digunakan dalam penelitian ini, dapat lebih terfokus pada penyesuaian lapisan-lapisan tertentu tanpa mengganggu performa optimal yang telah diperoleh sebelumnya.

c) *Modifikasi Output Layer*

Tahap ketiga dalam pengembangan model adalah modifikasi *output layer*. Pada tahap ini, lapisan keluaran (*output layer*) dari model Swin Transformer yang sebelumnya telah dilatih menggunakan *dataset* ImageNet1K dimodifikasi agar sesuai dengan jumlah kelas pada *dataset* rempah berbentuk rimpang yang digunakan dalam penelitian ini. Secara bawaan, *output layer* dari model *pre-trained* Swin Transformer memiliki 1000 neuron, yang mewakili 1000 kelas pada *dataset* ImageNet1K. Untuk menyesuaikannya dengan kebutuhan penelitian, *output layer* tersebut dimodifikasi sehingga hanya memiliki empat neuron, yang sesuai dengan jumlah kelas pada *dataset* rimpang. Modifikasi dilakukan dengan mengganti *MLP head* pada arsitektur Swin Transformer, yaitu dengan menetapkan jumlah input pada MLP head sama dengan jumlah neuron di lapisan sebelumnya, dan jumlah outputnya disesuaikan dengan jumlah kelas dalam *dataset* penelitian, yaitu empat. Ilustrasi dari modifikasi *ouput layer* dapat dilihat pada Gambar 18.



Gambar 18. Ilustrasi modifikasi *output layer*

d) *Training Model*

Tahap keempat dalam pengembangan model adalah *training model*. Pada tahap ini, model Swin Transformer yang telah dimodifikasi sebelumnya dilatih dan divalidasi menggunakan *dataset* rempah berbentuk rimpang pada folder train dan val yang telah dipersiapkan pada tahap *preprocessing*. Proses pelatihan dilakukan agar model dapat mengenali dan mengidentifikasi jenis-jenis rempah berbentuk rimpang, seperti jahe, kunyit, kencur, dan lengkuas. Selama pelatihan, model akan belajar dari data latih untuk memetakan pola-pola pada citra ke dalam kelas yang sesuai, sedangkan data validasi digunakan untuk mengevaluasi kinerja model dan mencegah terjadinya *overfitting*. *Hyperparameter* yang digunakan dalam proses pelatihan ini, seperti *learning rate*, jumlah *epochs*, ukuran *batch*, dan lainnya mengacu pada penelitian (Liu *et al*, 2021) dan dapat dilihat secara rinci pada Tabel 2. Tahap ini merupakan bagian penting dalam memastikan bahwa model dapat memberikan hasil yang akurat dalam mengklasifikasikan jenis rempah berbentuk rimpang.

Tabel 3. *Hyperparameter training model*

Nama Parameter	Nilai
Epoch	30
Learning Rate	0.0001
Batch-size	32
Optimizer	AdamW
Weight Decay	0.0001

e) Menyimpan Model

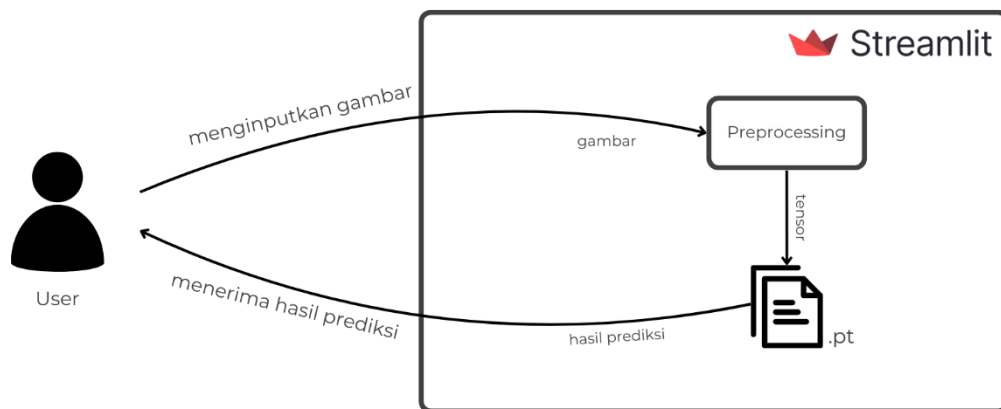
Tahap terakhir dalam pembuatan model adalah menyimpan model. Pada tahap ini, model Swin Transformer yang telah selesai dilatih dan divalidasi sebelumnya disimpan ke dalam file dengan format PyTorch Tensors (.pt). Penyimpanan model ini bertujuan agar model dapat digunakan kembali di masa mendatang tanpa perlu melakukan pelatihan ulang, sehingga menghemat waktu dan sumber daya. Model yang disimpan ini dapat dimanfaatkan untuk berbagai keperluan, seperti pengujian lebih lanjut, implementasi dalam aplikasi nyata, atau integrasi dengan *platform* berbasis PyTorch, termasuk aplikasi berbasis web. Langkah ini merupakan bagian penting untuk memastikan hasil penelitian dapat diterapkan secara praktis dan mendukung pengembangan lebih lanjut.

3.4.4 Pembuatan Web

Tahap Tahap pembuatan web dalam penelitian ini menggunakan *framework* Streamlit. *Framework* ini dipilih karena kemudahannya dalam membangun antarmuka berbasis web secara cepat serta kemampuan integrasinya yang baik dengan model *deep learning*. Web yang dirancang pada penelitian ini memiliki fitur utama untuk menerima input berupa gambar yang diunggah oleh pengguna. Setelah gambar diunggah, sistem akan melakukan proses *preprocessing*, yakni mengubah ukuran gambar menjadi 224x224 *pixel* agar sesuai dengan dimensi input yang dibutuhkan oleh model Swin Transformer. Selain itu, gambar yang telah diubah

ukurannya akan dikonversi menjadi bentuk tensor, format yang diperlukan untuk diproses oleh model *deep learning*.

Setelah tahap *preprocessing* selesai, gambar akan diteruskan ke model Swin Transformer yang sebelumnya telah dilatih dan disimpan dalam format PyTorch Tensors (.pt). Model akan melakukan prediksi untuk mengidentifikasi jenis rempah berbentuk rimpang dari gambar tersebut, seperti jahe, kunyit, kencur, atau lengkuas. Hasil prediksi kemudian ditampilkan dalam antarmuka web dalam bentuk teks, sehingga pengguna dapat dengan mudah memahami hasil identifikasi. Proses kerja dan alur web secara keseluruhan dapat dilihat pada Gambar 19.



Gambar 19. Alur kerja web

3.4.5 Pengujian

Pengujian dilakukan untuk melihat apakah model dan web yang dibuat memenuhi kriteria yang diinginkan. Pengujian yang dilakukan terdiri dari dua jenis pengujian, yaitu pengujian model dan pengujian web.

1) Pengujian model

Pengujian model dilakukan menggunakan *confusion matrix*. Pengujian dilakukan menggunakan *dataset* pengujian yang sebelumnya telah dipisahkan dari *dataset* pelatihan dan validasi. Hasil dari keberhasilan model dapat dianalisa menggunakan

confusion matrix dengan menentukan besar *accuracy*, *f1-score*, *recall*, dan *precision*.

2) Pengujian Web

Pengujian web dilakukan untuk memastikan bahwa web yang dibuat telah memenuhi kebutuhan yang diperlukan. Pengujian yang dilakukan pada web akan memfokuskan kepada fungsional keluaran *input* dan *output* dari fungsi web menggunakan *black box testing*. Pengujian akan dilakukan dengan beberapa skenario yang telah ditentukan. Skenario pengujian dan hasil yang diharapkan tertera pada Tabel 3.

Tabel 4. Pengujian *Black Box Testing*

ID Tes	Skenario Pengujian	Input	Ekspektasi Output
1	Unggah file dengan format yang valid	Gambar rimpang (jpg/png/jpeg)	Gambar ditampilkan dan prediksi muncul di layar
2	Unggah file dengan format yang tidak valid	File pdf atau txt	Muncul pesan error
3	Tidak mengunggah file dan klik tombol prediksi	Tidak ada file yang diunggah	Muncul pesan: "Tidak ada gambar yang dipilih!"
4	Mengunggah gambar bukan rimpang	Gambar bukan rimpang	Muncul pesan: "Gambar bukan bagian dari class"

V. SIMPULAN DAN SARAN

5.1 Simpulan

Simpulan dari penelitian yang telah dilakukan sebagai berikut:

1. Model Swin Transformer yang dikembangkan mendapatkan akurasi 99,11% pada data uji, tetapi mengalami degradasi performa menjadi 93,75% saat diuji melalui *platform* web dengan data internet yang mengindikasikan keterbatasan kapabilitas generalisasi model terhadap variasi data dunia nyata.
2. Penelitian ini berhasil membuktikan potensi bahwa model Vision Transformer yang diwakilkan oleh arsitektur Swin Transformer dapat mengalahkan model CNN yang diwakilkan oleh arsitektur Xception dalam mengidentifikasi jenis rempah yang berbentuk rimpang.
3. Penelitian ini juga telah berhasil membuat web yang dapat digunakan sebagai *platform* untuk model dapat melakukan prediksi untuk mengidentifikasi jenis rempah yang berbentuk rimpang.

5.2 Saran

Saran yang dapat diberikan untuk melanjutkan penelitian ini adalah sebagai berikut:

1. Memperbanyak variasi data latih untuk setiap kelas rimpang (jahe, kencur, kunyit, lengkuas), termasuk variasi bentuk, ukuran, sudut pengambilan, dan kondisi pencahayaan, agar model dapat mengenali dan mengeneralisasi kelas dengan lebih baik.
2. Mengembangkan model Swin Transformer yang mampu mendeteksi dua objek berbeda dalam satu frame untuk meningkatkan fleksibilitas dalam melakukan pengidentifikasian

DAFTAR PUSTAKA

- Arias-Duart, A., Mariotti, E., Garcia-Gasulla, D., & Alonso-Moral, J. M. 2023. A Confusion Matrix for Evaluating Feature Attribution Methods. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops 2023-June(Cv)*:3709–3714.
- Aryanta, I. W. R. 2019. Manfaat Jahe Untuk Kesehatan. *Journal Widya Kesehatan* 1(2):39–43.
- Azis, A. 2019. Kunyit (*Curcuma domestica* Val) Sebagai Obat Antipiretik. *Jurnal Ilmu Kedokteran Dan Kesehatan* 6(2):116–120.
- Bekkar, M., Djemaa, H. K., & Alitouche, T. A. 2013. Evaluation Measures for Models Assessment over Imbalanced Data Sets. *Journal of Information Engineering and Applications* 3(10):27–38.
- Bi, C., Hu, N., Zou, Y., Zhang, S., Xu, S., & Yu, H. 2022. Development of Deep Learning Methodology for Maize Seed Variety Recognition Based on Improved Swin Transformer. *Agronomy* 12(8):1843.
- Carnegie, J. O., Prabowo, A. R., Istanto, I., Budiana, E. P., Singgih, I. K., Yaningsih, I., & Mikšik, F. 2023. Technical review of supervised machine learning studies and potential implementation to identify herbal plant dataset. *Open Engineering* 13(1).
- Carrington, A. M., Manuel, D. G., Fieguth, P. W., Ramsay, T., Osmani, V., Wernly, B., Bennett, C., Hawken, S., Magwood, O., Sheikh, Y., Mcinnes, M., & Holzinger, A. 2023. Deep ROC Analysis and AUC as Balanced Average Accuracy, for Improved Classifier Selection, Audit and Explanation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45(1):329–341.

- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR 2021 - 9th International Conference on Learning Representations*.
- Febiharsa, D., Sudana, I. M., & Hudallah, N. 2018. Uji Fungsionalitas (Blackbox Testing) Sistem Informasi Lembaga Sertifikasi Profesi (SILSP) Batik dengan AppPerfect Web Test dan Uji Pengguna. *Joined Journal (Journal of Informatics Education)* 1(2):117.
- Ilahiyah, S., & Nilogiri, A. 2018. Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *JUSTINDO(Jurnal Sistem & Teknologi Informasi Indonesia)* 3(2):49–56.
- Jannah, A. B. S. N., Ramadanti, K., & Uyun, K. 2022. Identifikasi Ciri Morfologi pada Lengkuas (*Alpinia galanga*) dan Bangle (*Zingiber purpureum*) di Desa Mesjid Priyayi, Kecamatan Kasemen, Kota Serang, Banten. *Tropical Bioscience: Journal of Biological Science* 2(1):27–34.
- Kanika, Singla, J., & Nikita. 2021. Comparing ROC curve based thresholding methods in online transactions fraud detection system using deep learning. *Proceedings - IEEE 2021 International Conference on Computing, Communication, and Intelligent Systems ICCIS 2021*:119–124.
- Kharisma, I. L., Septiani, D. A., Fergina, A., & Kamdan. 2023. Penerapan Algoritma Decision Tree untuk Ulasan Aplikasi Vidio di Google Play. *Jurnal Nasional Teknologi Dan Sistem Informasi* 9(2):218–226.
- Khrisne, D. C., & Suyadnya, I. M. A. 2018. Indonesian Herbs and Spices Recognition using Smaller VGGNet-like Network. *2018 International Conference on Smart Green Technology in Electrical and Information Systems (ICSGTEIS)* 7:221–224.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *2021*

- IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC. Canada, pp. 9992–10002.
- Morales, G., Huam, S. G., & Telles, J. 2018. Artificial Neural Networks and Machine Learning – ICANN 2018. In V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, & I. Maglogiannis (Eds.), *27th International Conference on Artificial Neural Networks* 11141(November): 270-279.
- Moutik, O., Sekkat, H., Tigani, S., Chehri, A., Saadane, R., Tchakoucht, T. A., & Paul, A. 2023. Convolutional Neural Networks or Vision Transformers: Who Will Win the Race for Action Recognitions in Visual Data? *Sensors* 23(2): 734.
- Nisa, C., & Candra, F. 2023. Klasifikasi Jenis Rempah-Rempah Menggunakan Algoritma Convolutional Neural Network. *MALCOM: Indonesian Journal of Machine Learning and Computer Science* 4(1):78–84.
- Prabowo, H., Chaya, I. A. P. D., Arisanti, C. I. S., & Samirana, P. O. 2019. Standardisasi Spesifik dan Non-spesifik Simplisia dan Ekstrak Etanol Rimpang Kunyit (*Curcuma domestica* Val.). *Jurnal Farmasi Udayana* 8(1): 29.
- Putra, A. E., Naufal, M. F., & Prasetyo, V. R. 2023. Klasifikasi Jenis Rempah Menggunakan Convolutional Neural Network dan Transfer Learning. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)* 9(1):12.
- Riska, S. Y., & Farokhah, L. 2021. Klasifikasi Bumbu Dapur Indonesia Menggunakan Metode K-Nearest Neighbors (K-NN). *SMATIKA JURNAL* 11(1):37–42.
- Robi, Y., Kartikawati, S. M., & Muflihati, . 2019. Etnobotani Rempah Tradisional Di Desa Empoto Kabupaten Sanggau Kalimantan Barat. *Jurnal Hutan Lestari* 7(1): 130–142.
- Roihan, A., Sunarya, P. A., & Rafika, A. S. 2020. Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. *IJCIT (Indonesian Journal on Computer and Information Technology)* 5(1):75–82.
- Sarker, I. H. 2021. Machine Learning: Algorithms, Real-World Applications and

- Research Directions. *SN Computer Science* 2(3):160.
- Song, Z., Yang, S., & Zhang, R. 2021. Does Preprocessing Help Training Over-parameterized Neural Networks? *Advances in Neural Information Processing Systems* 27:22890–22904.
- Subaryanti, S., Sulistyaningsih, Y. C., Iswantini, D., & Triadiati, T. 2020. The Growth and Production of Galanga (*Kaempferia galanga* L.) in Different Altitudes. *Jurnal Ilmu Pertanian Indonesia* 25(2):167–177.
- Subburaj, S. D. R., Rengarajan, V. K. V., & Palaniswamy, S. 2023. Transfer Learning based Image Classification of Diseased Tomato Leaves with Optimal Fine-Tuning combined with Heat Map Visualization. *Tarim Bilimleri Dergisi* 29(4):1003–1017.
- Tanuwijaya, E., & Roseanne, A. 2021. Modifikasi Arsitektur VGG16 untuk Klasifikasi Citra Digital Rempah-Rempah Indonesia. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer* 21(1):189–196.
- Trimanto., Dwiyaniti, D., & Indriyani, S. 2018. Morfologi, Anatomi dan Uji Histokimia Rimpang *Curcuma aeruginosa* Valetton dan Zipj. *Jurnal Ilmu-Ilmu Hayati : LIPI* 17(2):123–133.
- Ye, Z., Yang, Y., Li, X., Cao, D., & Ouyang, D. 2019. An Integrated Transfer Learning and Multitask Learning Approach for Pharmacokinetic Parameter Prediction. *Molecular Pharmaceutics* 16(2):533–541.