

Pemilihan Arsitektur Convolutional Neural Network (CNN) untuk Deteksi Tempat Wisata di Surabaya

Zakaria Nur Abidin¹, I Kadek Dwi Nuryana²

^{1,2} Sistem Informasi, Universitas Negeri Surabaya

¹zakaria.21056@mhs.unesa.ac.id

²dwinuryana@unesa.ac.id

Abstrak— Penelitian ini berfokus pada pemilihan arsitektur *Convolutional Neural Network (CNN)* untuk mendeteksi tempat wisata di Surabaya melalui klasifikasi citra digital. Latar belakang penelitian ini didasari oleh pentingnya teknologi kecerdasan buatan dalam mendukung promosi pariwisata yang lebih interaktif dan efisien. Data penelitian dikumpulkan dari berbagai sumber, yaitu Google, Google Maps, serta dokumentasi langsung, dengan total 21.500 citra yang terbagi ke dalam 43 kelas tempat wisata. Metodologi yang digunakan adalah *CRISP-DM*, mencakup tahapan pemahaman bisnis, persiapan data, pemodelan, evaluasi, hingga implementasi.

Eksperimen dilakukan dengan enam arsitektur *CNN*, yaitu *SENet*, *ResNeXt*, *Inception v4*, *ResNet*, *Inception v3*, dan *Inception v2*. Selain itu, penelitian ini juga menguji kombinasi arsitektur untuk memperoleh performa lebih optimal. Hasil pengujian menunjukkan bahwa *ResNet* merupakan arsitektur tunggal terbaik dengan akurasi 83,49%. Namun, kombinasi *ResNet* dan *SENet* dengan *optimizer RMSProp*, *learning rate* 0,0001, serta *batch size* 32, menghasilkan performa tertinggi dengan akurasi 89,02%. Model terbaik ini kemudian diimplementasikan pada aplikasi web berbasis *Flask*, yang diuji melalui *black box testing* dan terbukti berjalan sesuai kebutuhan pengguna.

Secara keseluruhan, penelitian ini berhasil menunjukkan efektivitas *CNN* dalam klasifikasi gambar tempat wisata, sekaligus memberikan kontribusi praktis berupa aplikasi web yang dapat membantu wisatawan mengenali destinasi wisata Surabaya. Dengan demikian, penelitian ini tidak hanya memperkuat pemanfaatan *deep learning* dalam sektor pariwisata, tetapi juga membuka peluang pengembangan teknologi serupa untuk promosi wisata di kota lain.

Kata Kunci— *Convolutional Neural Network*, Deteksi Citra, Tempat Wisata Surabaya, *Deep Learning*, *Flask*, *Ensemble Learning*.

I. PENDAHULUAN

Perkembangan pesat teknologi *machine learning* dan *deep learning* telah membuka peluang di berbagai bidang, termasuk pengenalan dan klasifikasi gambar. Arsitektur *Convolutional Neural Network (CNN)*, khususnya, telah menjadi standar dalam klasifikasi gambar karena kemampuannya mempelajari fitur gambar secara otomatis, menghasilkan representasi yang mendalam dan diskriminatif [1]. Penerapan *CNN* ini menawarkan solusi efisien, bahkan di sektor pariwisata.

Surabaya memiliki potensi pariwisata yang signifikan, meskipun data kunjungan wisatawan menunjukkan fluktuasi, termasuk penurunan drastis pada tahun 2020 akibat pandemi COVID-19. Meskipun terjadi pemulihan, jumlah kunjungan belum mencapai puncak tahun 2018 [2]. Oleh karena itu, diperlukan dukungan teknologi untuk meningkatkan

pengalaman wisatawan. Penggunaan *CNN* dapat membantu mengklasifikasikan dan mendeteksi tempat wisata di Surabaya melalui gambar, memudahkan wisatawan memperoleh informasi yang akurat dan interaktif. Dengan kemampuannya mengekstrak fitur visual secara otomatis, *CNN* sangat cocok untuk tugas ini. Ia dapat mengenali pola konsisten pada gambar, seperti landmark atau struktur bangunan khas, membuatnya efektif untuk klasifikasi citra yang memiliki karakteristik visual yang jelas [3].

Dalam beberapa tahun terakhir, berbagai arsitektur *CNN* telah dikembangkan untuk meningkatkan akurasi dan efisiensi, seperti *SENet*, *ResNeXt*, dan *Inception-v4*. Arsitektur-arsitektur ini terbukti unggul dengan tingkat kesalahan klasifikasi gambar kurang dari 4% [4], melampaui akurasi manusia. Penelitian ini akan mengeksplorasi performa arsitektur tersebut, termasuk *ResNet*, *Inception-v3*, dan *Inception-v2*, serta menguji kombinasi dua arsitektur untuk potensi peningkatan akurasi.

Untuk mendukung pengembangan aplikasi deteksi gambar, penelitian ini menggunakan kerangka kerja *CRISP-DM (Cross-Industry Standard Process for Data Mining)*. Kerangka ini memastikan pendekatan yang sistematis dari pemahaman bisnis, pengumpulan, persiapan data (dengan *dataset* 21.500 gambar tempat wisata di Surabaya), hingga implementasi model. Selain itu, model *CNN* yang paling optimal akan diimplementasikan ke dalam aplikasi web menggunakan framework *Flask*, sebuah micro-framework *Python* yang ringan dan fleksibel, ideal untuk pengembangan prototipe cepat [5].

Penelitian ini bertujuan mengisi celah yang ada dengan fokus pada klasifikasi gambar tempat wisata di Surabaya, menggunakan *dataset* spesifik dan komprehensif. Berbeda dengan studi sebelumnya yang sering bias atau terbatas pada *dataset* publik, penelitian ini menekankan pada pembersihan data mendetail. Selain itu, pengujian *ensemble learning* melalui kombinasi arsitektur *CNN* (*Inception-v2* + *SENet*) diharapkan dapat meningkatkan akurasi secara signifikan [6]. Kontribusi utama penelitian ini adalah pengembangan aplikasi yang memudahkan wisatawan mengenali tempat wisata hanya dengan mengambil foto, yang berpotensi mendukung promosi pariwisata yang lebih efektif dan meningkatkan ekonomi lokal Surabaya.

II. METODE PENELITIAN

Penelitian ini mengadopsi pendekatan *CRISP-DM* untuk mengklasifikasi citra tempat wisata di Surabaya menggunakan *Convolutional Neural Networks (CNN)*. Proses dimulai dari *business understanding*, dilanjutkan ke *data understanding*,

data preparation, modeling, evaluation, hingga tahap deployment ke aplikasi web berbasis *Flask*.



Gbr. 1 Alur CRISP-DM

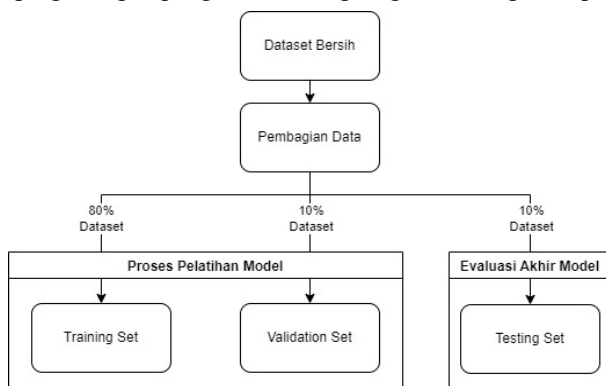
A. Business Understanding

Fokus utama penelitian ini adalah mengembangkan aplikasi berbasis *CNN* yang dapat membantu wisatawan mengidentifikasi tempat wisata melalui unggahan gambar. Tujuan ini mendukung promosi pariwisata Kota Surabaya sekaligus memberikan pengalaman interaktif kepada pengguna. Fungsionalitas utama aplikasi:

1. Identifikasi Tempat Wisata
Pengguna mengunggah gambar, lalu sistem memprediksi kategori wisata jika *confidence* $\geq 50\%$.
2. Penyediaan Informasi Tambahan
Aplikasi menampilkan deskripsi, harga tiket, jam operasional, dan rating.
3. Promosi Pariwisata
Memperluas akses informasi wisata secara digital.
4. Penanganan Gambar Kurang Jelas
Jika *confidence* $< 50\%$, sistem memberi peringatan dan saran untuk unggah ulang.

B. Data Understanding

Dataset mencakup 43 kategori tempat wisata di Surabaya dengan variasi kondisi pencahayaan, sudut pandang, dan kualitas gambar. Data dikumpulkan dari tiga sumber: web scraping Google, pengambilan langsung, dan Google Maps.



Gbr. 2 Teknik Pengumpulan Data

Selain itu, *dataset* memiliki total 27.177 gambar, dengan jumlah tiap kelas dirinci pada Tabel I.

TABEL I
JUMLAH DATA TIAP KELAS TEMPAT WISATA

No	Nama tempat wisata	Jumlah data
1	Alun Alun Surabaya	604
2	Bg Jungtion	667
3	Ciputra World Surabaya	681
4	Danau Unesa Lidah Wetan	631
5	Ekowisata Mangrove Wonorejo	617
6	Galaxy Mall	688
7	Grand City Mall	669
8	Hutan Bambu Keputih	654
9	Jembatan Merah	613
10	Jembatan Suramadu	620
11	Kebun Binatang Surabaya	606
12	Kelenteng Hong Tiek Hian	596
13	Kelenteng Sanggar Agung	627
14	Kya Kya	600
15	Masjid Al Akbar Surabaya	608
16	Masjid Muhammad Cheng Hoo Surabaya	616
17	Monumen Bambu Runcing	664
18	Monumen Jalesveva Jayamahe	625
19	Monumen Kapal Selam	618
20	Monumen Tugu Pahlawan dan Museum Sepuluh Nopember Surabaya	616
21	Museum Bank Indonesia	610
22	Museum W.R Soepratman	613
23	Pagoda Tian Ti	621
24	Pakuwon City Mall	690
25	Pakuwon Mall	666
26	Pantai Kenjeran Lama	630
27	Patung Brahma 4 Rupa	555
28	Plasa Marina	650
29	Plaza Surabaya	670
30	Royal Plaza	666
31	Surabaya Museum (Gedung Siola)	619
32	Surabaya North Quay	627
33	Taman Apsari	667
34	Taman Bungkul	601
35	Taman Flora	613
36	Taman Mayangkara	616
37	Taman Mundu	629
38	Taman Pelangi	606
39	Taman Persahabatan	627
40	Taman Prestasi	626
41	Taman Suroboyo	622
42	Tunjungan Plaza	652
43	WTC Surabaya	681
Total:		27.177

C. Data Preparation

1. Pembersihan *Dataset*
Proses pembersihan dilakukan untuk menghapus gambar buram, resolusi rendah, tidak relevan, atau berwatermark. Semua gambar yang lolos distandarisasi ke dimensi 224x224 atau 299x299 piksel.
2. Pembagian *Dataset*

Dataset dibagi menjadi: 80% *training* (Melatih model), 10% *validation* (Memantau dan mencegah *overfitting*.), 10% *testing* (Evaluasi akhir model.). Hasil dari proses ini menghasilkan distribusi data per kelas yang ditunjukkan pada Tabel II.

TABEL II
DISTRIBUSI TIAP SET DATA

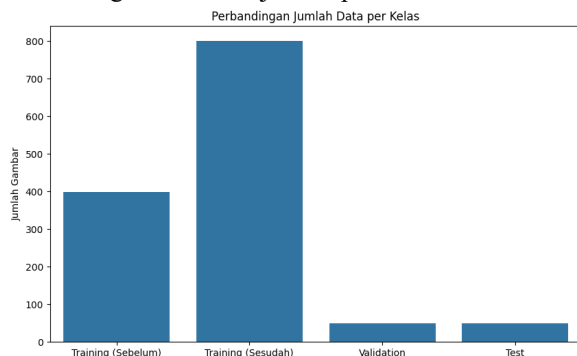
Jumlah kelas	Training	Validation	Testing	Total
1	400	50	50	500
43	17.200	2.150	2.150	21.500

3. Augmentasi Data

Untuk memperkaya variasi *dataset*, diterapkan augmentasi menggunakan *ImageDataGenerator*:

- Rotasi (hingga 30°).
- Pergeseran horizontal & vertikal.
- Transformasi shear (geser).
- Zoom in/out.
- Flip horizontal.

Hasil dari proses ini menghasilkan distribusi data per kelas sebagaimana ditunjukkan pada Gbr. 3.



Gbr. 3 Perbandingan data setelah augmentasi

D. Modeling

Tahap modeling dilakukan dengan menguji berbagai arsitektur *CNN*, yaitu *SENet*, *ResNeXt*, *Inception v4*, *ResNet*, *Inception v3*, dan *Inception v2*. Setiap arsitektur dieksplorasi secara terpisah, kemudian dipertimbangkan pula kombinasi dari dua model terbaik untuk memperoleh hasil yang lebih optimal. Langkah-langkah utama dalam proses modeling meliputi:

- Pemilihan Arsitektur
Eksplorasi model tunggal.
- Kombinasi Arsitektur
Menggabungkan dua model terbaik.
- Pelatihan & Validasi
Menggunakan *TensorFlow/Keras*.
- Strategi Anti-*Overfitting*
Menerapkan *early stopping* dan regularisasi.

Selain itu, untuk meningkatkan performa model, dilakukan pula proses *tuning hyperparameter*. Proses ini bertujuan mencari kombinasi parameter terbaik agar model dapat belajar secara lebih efektif. Rincian parameter yang diuji dapat dilihat pada Tabel III.

TABEL III
TUNING HYPERPARAMETER

Parameter	Nilai yang Diuji	Penjelasan
<i>Batch Size</i>	16, 32, 64	Jumlah sampel per update.
<i>Learning Rate</i>	0.01, 0.001, 0.0001, 0.00001	Ukuran langkah update bobot.

E. Evaluation

Evaluasi dilakukan pada *testing* set menggunakan beberapa metrik:

1 Akurasi

Akurasi merupakan metrik evaluasi yang mengukur proporsi prediksi benar terhadap seluruh prediksi yang dibuat oleh model. Rumus akurasi dirumuskan sebagai berikut:

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

Metrik ini paling tepat digunakan ketika data antar kelas seimbang, dan pada penelitian ini digunakan untuk menilai kinerja klasifikasi citra tempat wisata.

2 Presisi

Presisi mengukur proporsi prediksi positif yang benar, atau dengan kata lain seberapa akurat model dalam menetapkan data pada suatu kelas tertentu. Rumus presisi adalah:

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

Presisi sangat penting dalam konteks klasifikasi tempat wisata, karena kesalahan klasifikasi dapat menurunkan relevansi hasil sistem.

3 Recall

Recall menunjukkan proporsi data positif yang berhasil dikenali dengan benar oleh model. Metrik ini penting ketika kesalahan terlewat (*false negative*) harus diminimalkan. Rumus recall adalah:

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

Pada klasifikasi citra tempat wisata, nilai recall yang tinggi menunjukkan bahwa model mampu mengenali sebagian besar citra dalam kategori yang tepat.

4 F1-Score

F1-Score merupakan rata-rata harmonis antara presisi dan recall, digunakan untuk mengevaluasi performa model secara seimbang agar tidak bias pada salah satu metrik. Rumus *F1-Score* adalah:

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

Nilai *F1-Score* yang tinggi menunjukkan bahwa model mampu menjaga keseimbangan antara ketepatan klasifikasi (presisi) dan kemampuan mengenali data relevan (recall).

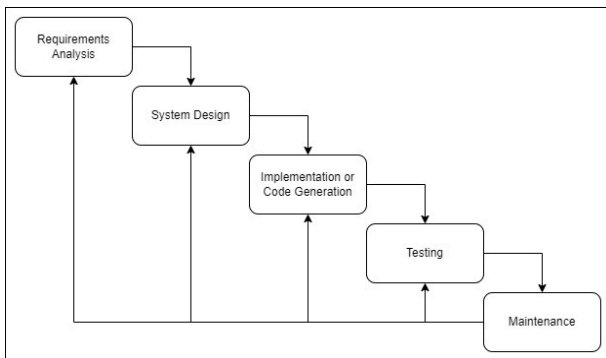
Untuk meningkatkan keandalan, digunakan juga *k-Fold Cross Validation* dengan $k=9$, dengan pembagian tiap fold (k) ditunjukkan pada Tabel IV.

TABEL IV
SKEMA PEMBAGIAN DATA UNTUK *k-FOLD CROSS-VALIDATION*

K	Folder 1	Folder 2	Folder 3	Folder 4	Folder 5	Folder 6	Folder 7	Folder 8	Folder 9	Folder 10
1	Validation	Training	Training	Training	Training	Training	Training	Training	Training	Test
2	Training	Validation	Training	Training	Training	Training	Training	Training	Training	Test
3	Training	Training	Validation	Training	Training	Training	Training	Training	Training	Test
4	Training	Training	Training	Validation	Training	Training	Training	Training	Training	Test
5	Training	Training	Training	Training	Validation	Training	Training	Training	Training	Test
6	Training	Training	Training	Training	Training	Validation	Training	Training	Training	Test
7	Training	Training	Training	Training	Training	Training	Validation	Training	Training	Test
8	Training	Training	Training	Training	Training	Training	Training	Validation	Training	Test
9	Training	Training	Training	Training	Training	Training	Training	Training	Validation	Test

F. Deployment

Tahap terakhir adalah implementasi model *CNN* terbaik ke dalam aplikasi web berbasis *Flask*. Alur deployment mengikuti metode *Waterfall*.



Gbr. 4 Alur metode *waterfall*

1. Requirement Analysis
Identifikasi kebutuhan fungsional & non-fungsional.
2. System Design
Pembuatan Use Case, Activity, Sequence Diagram dan Desain UI.
3. Implementation
Integrasi model ke aplikasi.
4. Testing
Uji coba sistem dengan menggunakan metode *black box testing*.

III. HASIL DAN PEMBAHASAN

A. Pelatihan Model *CNN*

Penelitian ini melakukan eksperimen dengan berbagai arsitektur *CNN*, baik secara tunggal maupun kombinasi, serta menguji perbedaan penggunaan *optimizer*. Tujuannya adalah untuk menemukan konfigurasi model terbaik yang mampu mengklasifikasikan objek wisata di Surabaya secara akurat.

1. Hasil Eksperimen Arsitektur *CNN*

a) *SENet*

Model klasifikasi dengan *SENet* menunjukkan kinerja yang cukup baik dengan akurasi 81,30%.

TABEL V

HASIL EVALUASI ARSITEKTUR *SENET*

Metrik	Nilai
Accuracy	81.30%
Precision (Macro)	81.64%
Recall (Macro)	81.30%
F1 Score (Macro)	81.21%

SENet berhasil mengenali beberapa objek wisata dengan akurasi tinggi, meskipun masih terdapat kesalahan pada kelas yang memiliki kemiripan visual, seperti pusat perbelanjaan.

b) *ResNet*

Arsitektur *ResNet* menghasilkan performa terbaik pada model tunggal dengan akurasi 83,49%.

TABEL VI
HASIL EVALUASI ARSITEKTUR *RESNET*

Metrik	Nilai
Accuracy	83.49%
Precision (Macro)	83.85%
Recall (Macro)	83.49%
F1 Score (Macro)	83.52%

ResNet menunjukkan keunggulan dalam mengklasifikasikan ikon wisata bersejarah seperti Masjid Muhammad Cheng Hoo dan Jembatan Merah. Namun, objek mall masih sulit dibedakan.

c) *ResNeXt*

Performa *ResNeXt* jauh di bawah arsitektur lain, dengan akurasi hanya 36,56%.

TABEL VII
HASIL EVALUASI ARSITEKTUR *RESNEXT*

Metrik	Nilai
Accuracy	36.56%
Precision (Macro)	38.74%
Recall (Macro)	36.56%
F1 Score (Macro)	35.56%

Kesalahan klasifikasi sangat tinggi, terutama pada objek wisata modern seperti mall.

d) *Inception v2*

Model ini menghasilkan kinerja moderat dengan akurasi sekitar 50,79%.

TABEL VIII
HASIL EVALUASI ARSITEKTUR *INCEPTION V2*

Metrik	Nilai
Accuracy	50.79%
Precision (Macro)	51.59%
Recall (Macro)	50.79%
F1 Score (Macro)	50.42%

e) *Inception v3*

Inception v3 menunjukkan performa konsisten dengan akurasi 79,30%.

TABEL IX
HASIL EVALUASI ARSITEKTUR *INCEPTION V3*

Metrik	Nilai
Accuracy	79.30%
Precision (Macro)	79.56%

Recall (Macro)	79.30%
F1 Score (Macro)	79.33%

Sebagian besar kelas wisata teridentifikasi dengan baik, terutama Museum dan Ekowisata.

f) *Inception v4*

Performa *Inception v4* sangat rendah dengan akurasi hanya 34,37%.

TABEL X
HASIL EVALUASI ARSITEKTUR *INCEPTION V2*

Metrik	Nilai
Accuracy	34.37%
Precision (Macro)	35.80%
Recall (Macro)	34.37%
F1 Score (Macro)	33.95%

g) Perbandingan Arsitektur Tunggal

TABEL XI
TABEL PERBANDINGAN PERFORMA ARSITEKTUR TUNGGAL

Arsitektur	Accuracy	Precision	Recall	F1 Score	Total	Rank
SENet	0.813	0.8164	0.813	0.8121	3.2545	2
ResNet	0.8349	0.8385	0.8349	0.8352	3.3435	1
ResNeXt	0.3656	0.3874	0.3656	0.3556	1.4742	5
Inception v2	0.5079	0.5159	0.5079	0.5042	2.0359	4
Inception v3	0.793	0.7956	0.793	0.7933	3.1749	3
Inception v4	0.3437	0.358	0.3437	0.3395	1.3849	6

Berdasarkan Tabel XI, *ResNet* terbukti memberikan performa terbaik, diikuti oleh *SENet* yang stabil pada peringkat kedua. Sementara itu, *Inception v3* masih menunjukkan hasil yang cukup kuat, namun tidak melampaui dua model sebelumnya. Adapun *Inception v2*, *ResNeXt*, dan *Inception v4* kurang memberikan hasil yang memuaskan sehingga tidak direkomendasikan untuk digunakan lebih lanjut.

2. Hasil Eksperimen Kombinasi Arsitektur

Untuk meningkatkan akurasi sekaligus menjaga konsistensi performa, dilakukan kombinasi arsitektur *ResNet* dan *SENet*.

TABEL XII
HASIL EVALUASI ARSITEKTUR *RESNET + SENET*

Metrik	Nilai
Accuracy	83.72%
Precision (Macro)	83.97%
Recall (Macro)	83.72%
F1 Score (Macro)	83.73%

Hasil evaluasi pada Tabel XII menunjukkan bahwa kombinasi ini berhasil memperbaiki kualitas klasifikasi, terutama dalam menyeimbangkan akurasi dan konsistensi antar kelas.

3. Perbandingan *Optimizer*

Pada model kombinasi *ResNet + SENet*, dilakukan pengujian menggunakan tiga *optimizer* berbeda, yaitu *Adam*, *RMSProp*, dan *SGD*.

TABEL XIII
PERBANDINGAN KINERJA TIAP *OPTIMIZER*

Optimizer	Accuracy	Precision	Recall	F1-Score
<i>Adam</i>	83.72%	83.97%	83.72%	83.73%

<i>RMSProp</i>	84.05%	84.53%	84.05%	84.13%
<i>SGD</i>	3.21%	4.20%	3.21%	2.04%

Hasil evaluasi pada tabel XIII menunjukkan bahwa *Adam*, yang dijadikan sebagai baseline, memperoleh akurasi sebesar 83,72%. Selanjutnya, penggunaan *RMSProp* memberikan peningkatan performa hingga 84,05%, menjadikannya sebagai opsi terbaik pada eksperimen ini. Sebaliknya, *SGD* gagal memberikan hasil yang memadai, dengan akurasi yang sangat rendah, hanya 3,21%, sehingga tidak direkomendasikan untuk digunakan pada model ini.

4. *Tuning Hyperparameter*

TABEL XIV
HASIL *TUNING HYPERPARAMETER*

Learning Rate	Batch Size	Akurasi	Presisi	Recall	F1-Score	Total	Rank
0.01	16	0.2944	0.3045	0.2944	0.2859	1.1792	12
0.01	32	0.5395	0.5476	0.5395	0.5403	2.1669	11
0.01	64	0.6158	0.6444	0.6158	0.6208	2.4968	10
0.001	16	0.7526	0.7626	0.7526	0.7548	3.0226	9
0.001	32	0.8302	0.8354	0.8302	0.831	3.3268	8
0.001	64	0.8516	0.856	0.8516	0.8522	3.4114	5
0.0001	16	0.8847	0.8867	0.8847	0.8849	3.541	2
0.0001	32	0.886	0.889	0.886	0.8867	3.5477	1
0.0001	64	0.8833	0.8859	0.8833	0.8834	3.5359	3
0.00001	16	0.8544	0.8569	0.8544	0.8545	3.4202	4
0.00001	32	0.8456	0.8482	0.8456	0.8454	3.3848	6
0.00001	64	0.8405	0.8453	0.8405	0.8413	3.3676	7

Hasil proses *tuning hyperparameter* pada tabel XIV menunjukkan bahwa konfigurasi terbaik diperoleh pada *learning rate* sebesar 0,0001 dengan *batch size* 32. Kombinasi ini terbukti mampu menghasilkan performa paling optimal, dengan akurasi tertinggi mencapai 88,6%.

B. Evaluasi

1. *Cross Validation*

Model terbaik, yaitu *ResNet + SENet* dengan *optimizer RMSProp* menggunakan konfigurasi *learning rate* 0,0001 dan *batch size* 32, kemudian diuji lebih lanjut dengan metode 9-fold *cross-validation*.

TABEL XV
HASIL *TUNING HYPERPARAMETER*

Fold	Akurasi	Presisi	Recall	F1-Score	Total	Rank
1	0.886	0.889	0.886	0.8867	3.5477	2
2	0.8842	0.8857	0.8842	0.8842	3.5383	5
3	0.8851	0.889	0.8851	0.8858	3.545	3
4	0.8823	0.8835	0.8823	0.8823	3.5304	8
5	0.8847	0.8866	0.8847	0.885	3.541	4
6	0.8819	0.8846	0.8819	0.8821	3.5305	7
7	0.8805	0.8832	0.8805	0.8804	3.5246	9
8	0.8902	0.8932	0.8902	0.8906	3.5642	1
9	0.8833	0.8852	0.8833	0.8835	3.5353	6
Rata Rata:	88.42%	88.67%	88.42%	88.45%		

Hasil pengujian pada tabel XV memperlihatkan stabilitas performa yang sangat baik, dengan rata-rata akurasi mencapai 88,42%. Temuan ini mengindikasikan bahwa model tidak hanya unggul pada data latih, tetapi

juga mampu mempertahankan konsistensi ketika diuji pada data yang berbeda.

2. Model Terbaik

Konfigurasi optimal dari model menghasilkan performa yang sangat baik dengan rincian hasil evaluasi sebagaimana ditunjukkan pada Tabel XVI.

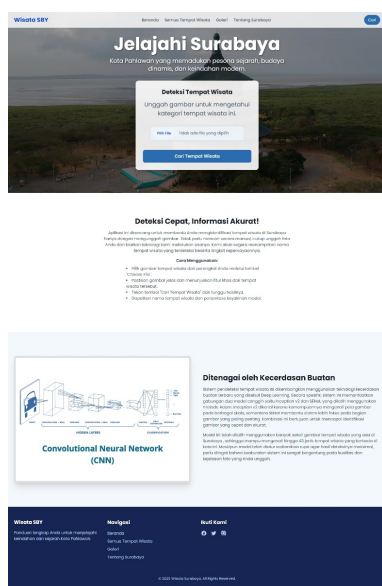
TABEL XVI
RINGKASAN METRIK KINERJA MODEL TERBAIK

Metrik	Nilai
Accuracy	89.02%
Precision (Macro)	89.32%
Recall (Macro)	89.02%
F1 Score (Macro)	89.06%

C. Implementasi Website

1. Tampilan Aplikasi Web

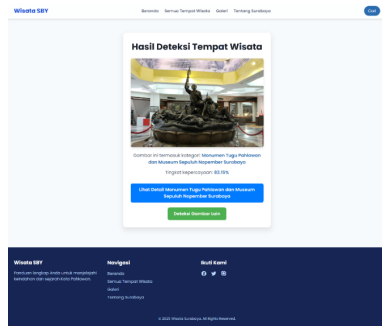
a) Halaman Deteksi



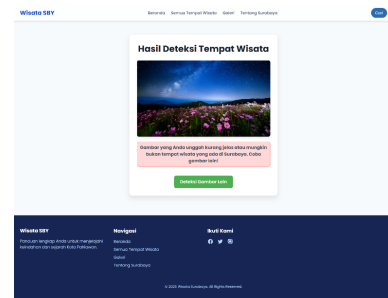
Gbr. 5 Tampilan halaman deteksi

Gbr. 5 menunjukkan tampilan Halaman Deteksi yang berfungsi sebagai antarmuka awal, di mana pengguna dapat mengunggah gambar untuk diidentifikasi. Setelah gambar diproses, sistem menampilkan hasil prediksi beserta *confidence score*.

b) Halaman Hasil Deteksi



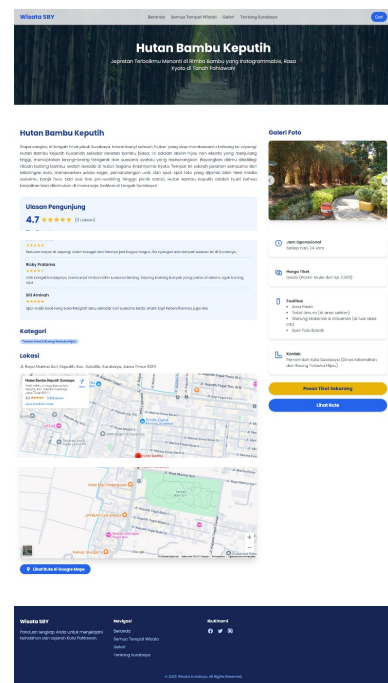
Gbr. 6 Tampilan halaman hasil deteksi (*confidence* ≥ 50%)



Gbr. 7 Tampilan halaman hasil deteksi (*confidence* < 50%)

Halaman Hasil Deteksi menyajikan informasi klasifikasi berdasarkan nilai *confidence*. Apabila nilai *confidence* ≥ 50%, sebagaimana ditunjukkan pada Gbr. 6, sistem menampilkan nama tempat wisata yang terdeteksi. Sebaliknya, jika *confidence* < 50%, sebagaimana terlihat pada Gbr. 7, pengguna akan memperoleh pesan bahwa gambar kurang jelas sehingga hasil klasifikasi tidak dapat ditentukan.

c) Detail Tempat Wisata



Gbr. 8 Tampilan halaman detail tempat wisata

Gbr. 8 menunjukkan tampilan Halaman Detail Tempat Wisata. Pada halaman ini, pengguna dapat melihat deskripsi singkat mengenai objek wisata yang sesuai dengan hasil klasifikasi.

2. Uji Coba Fungsionalitas Aplikasi

Untuk memastikan aplikasi berjalan sesuai kebutuhan pengguna, dilakukan pengujian fungsionalitas dengan metode *black box testing* pada tiga halaman utama, yaitu halaman Deteksi, Hasil Deteksi, dan Detail Tempat Wisata. Ringkasan hasil uji coba ditunjukkan pada Tabel XVII.

TABEL XVII

HASIL BLACK BOX TESTING

Halaman	Fitur yang Diuji	Hasil yang Diharapkan	Status
Deteksi	Upload gambar valid & submit	Gambar terdeteksi dan diarahkan ke halaman result	✓
	Upload gambar tidak valid	Sistem menolak file dan menampilkan pesan error	✓
	Tombol submit tanpa file	Browser menampilkan pesan error	✓
Result Deteksi	Tampil hasil klasifikasi ($\geq 50\%$ confidence)	Menampilkan nama tempat wisata, confidence, tombol detail	✓
	Gambar tidak jelas ($< 50\%$ confidence)	Pesan peringatan ditampilkan, tombol detail tidak muncul	✓
	Tombol "Deteksi Gambar Lain"	Kembali ke halaman deteksi	✓
Detail Tempat Wisata	Interaksi peta & tombol "Lihat Rute"	Peta interaktif, link ke Google Maps terbuka	✓
	Tombol "Pesan Tiket Sekarang"	Membuka halaman pemesanan tiket	✓
	Navigasi gambar (prev/next)	Gambar berganti dengan mulus	✓

Hasil pengujian menunjukkan bahwa seluruh fungsionalitas utama aplikasi telah berjalan sesuai rancangan tanpa adanya error yang mengganggu pengalaman pengguna.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang telah dilakukan mengenai klasifikasi citra tempat wisata di Surabaya menggunakan berbagai arsitektur CNN, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Penelitian ini menganalisis performa enam arsitektur CNN tunggal (*SENet*, *ResNeXt*, *Inception v4*, *ResNet*, *Inception v3*, dan *Inception v2*) dalam klasifikasi gambar tempat wisata di Surabaya. Hasil menunjukkan bahwa *ResNet* (83,49%) dan *SENet* (81,30%) memberikan akurasi terbaik, disusul *Inception v3* (~79%). *Inception v2* berada pada kategori menengah (~50%), sementara *ResNeXt* (36,56%) dan *Inception v4* (34,37%) jauh lebih rendah sehingga dinilai kurang efektif.
2. Kombinasi *ResNet* + *SENet* dengan optimizer *RMSProp* terbukti unggul. Setelah *hyperparameter tuning* menggunakan *learning rate* 0.0001 dan *batch size* 32, model ini menghasilkan akurasi 89,02%, dengan nilai

presisi 89,32%, recall 89,02%, dan F1-score 89,06%. Hasil ini menunjukkan bahwa kombinasi tersebut paling andal untuk klasifikasi gambar wisata di Surabaya.

3. Model terbaik berhasil diimplementasikan dalam aplikasi web berbasis *Flask*. Hasil pengujian *black box* menunjukkan bahwa sistem memenuhi seluruh kebutuhan fungsional, termasuk kemampuan identifikasi gambar, pemberian informasi tambahan jika *confidence* di atas 50%, penanganan gambar kurang jelas bila *confidence* di bawah 50%, navigasi antarmuka yang intuitif, serta integrasi peta interaktif dengan fitur "Lihat Rute di Google Maps".

UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Allah SWT atas rahmat dan karunia-Nya sehingga penelitian ini dapat terselesaikan dengan baik. Ucapan terima kasih juga disampaikan kepada dosen pembimbing yang telah memberikan arahan, bimbingan, dan masukan berharga sepanjang proses penelitian. Tidak lupa, apresiasi ditujukan kepada keluarga serta rekan-rekan yang selalu memberikan dukungan, motivasi, dan doa sehingga penelitian ini dapat berjalan dengan lancar.

REFERENSI

- [1] Yu, F., Zhang, Q., Xiao, J., Ma, Y., Wang, M., Luan, R., Liu, X., Ping, Y., Nie, Y., Tao, Z., & lainnya, "Progress in the application of CNN-based image classification and recognition in whole crop growth cycles," *Remote Sensing*, vol. 15, no. 12, p. 2988, Jun. 2023, doi: 10.3390/rs15122988.
- [2] Badan Pusat Statistik Provinsi Jawa Timur, Statistik Pariwisata Provinsi Jawa Timur 2023. Surabaya, Indonesia: BPS, 2023. [Online]. Available: <https://jatim.bps.go.id>.
- [3] Hsu, C.-F., Lin, C.-C., Hung, T.-Y., Lei, C.-L., & Chen, K.-T., "A detailed look at CNN-based approaches in facial landmark detection," arXiv preprint arXiv:2005.08649, 2020.
- [4] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., & Farhan, L., "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, pp. 1–74, Feb. 2021, doi: 10.1186/s40537-021-00444-8.
- [5] Ghimire, D., "Comparison of Python web frameworks: Flask vs. Django," M.S. thesis, Theseus Univ. of Applied Sciences, Finland, 2020. [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/339796/Ghimire_Devn_dra.pdf.
- [6] Bhargavi, M., Renugadevi, R., Sivabalan, S., Phani, P., Ganesh, J., & Bhanu, K., "Ensemble learning for skin lesion classification: A robust approach for improved diagnostic accuracy (ELSLC)," in Proc. 3rd Int. Conf. Innovative Mechanisms for Industry Applications (ICIMIA), Dec. 2023, pp. 390–395. IEEE. [Online]. Available: <https://www.researchgate.net/profile/Sivabalan-Settu/publication/379943350>.