



KERJA PRAKTIK - IF184801

Pemanfaatan XAI (Xplainable Artificial Intelligence) dalam Pemetaan Kerentanan Bencana Tanah Longsor di Kecamatan Pacet Kab. Mojokerto

Departemen Perencanaan Wilayah dan Kota FTSPK ITS
Jl. Despro, Gedung A Lantai 2 Departemen Perencanaan Wilayah dan Kota Kampus ITS, Sukolilo Surabaya 60111
Periode: 1 April 2025 - 30 Juni 2025

Oleh:

Rano Noumi Sulistyو

5025211185

Pembimbing Departemen

Ary Mazharuddin S.Kom., M.Comp.Sc.

Pembimbing Lapangan

Prananda Navitas, S.T., M.Sc., Ph.D.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2025

DAFTAR ISI

DAFTAR ISI	iv
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
LEMBAR PENGESAHAN	xii
KATA PENGANTAR	xvi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan	2
1.3. Manfaat	2
1.4. Rumusan Masalah	3
1.5. Lokasi dan Waktu Kerja Praktik	3
1.6. Metodologi Kerja Praktik	3
1.6.1. Perumusan Masalah	3
1.6.2. Studi Literatur	4
1.6.3. Perancangan dan Implementasi Program	4
1.6.4. Hasil dan Evaluasi	4
1.6.5. Kesimpulan dan Saran	4
1.7. Sistematika Laporan	4
1.7.1. Bab I Pendahuluan	4
1.7.2. Bab II Profil Perusahaan	5
1.7.3. Bab III Tinjauan Pustaka	5

1.7.4.	Bab IV Perancangan dan Implementasi Program 5	
1.7.5.	Bab VI Hasil dan Evaluasi	5
1.7.6.	Bab VII Kesimpulan dan Saran	5
	BAB II PROFIL PERUSAHAAN	7
2.1.	Profil Departemen Perencanaan Wilayah dan Kota (PWK) ITS	7
2.2.	Lokasi	7
	BAB III TINJAUAN PUSTAKA	10
3.1.	Machine Learning	10
3.2.	Python	10
3.3.	OSGEO dan GDAL	11
3.4.	CNN	12
3.5.	Squeeze-and-Excitation (SE) dan Spatial Pyramid Pooling (SPP)	12
3.6.	SHAP	13
	BAB IV PERANCANGAN DAN IMPLEMENTASI PROGRAM	15
4.1.	Perancangan Program	15
4.1.1.	Desain Arsitektur Program	15
4.2.	Implementasi Program	17
4.2.1.	Preprocessing data	17
4.2.2.	CNN	22
4.2.3.	Training	26

4.2.4. XAI	29
BAB V HASIL DAN EVALUASI	33
5.1. Training CNN	33
5.2. Hasil SHAP	34
5.3. Perbandingan SHAP dengan PFI	40
BAB VI KESIMPULAN DAN SARAN	43
6.1. Kesimpulan	43
6.2. Saran	44
DAFTAR PUSTAKA	46
LAMPIRAN	48
BIODATA PENULIS	49

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 1 Gambar Desain Program.....	17
Gambar 2 Skrip data check	19
Gambar 3 Skrip ambil data tif dan transformasi menjadi data untuk.....	20
Gambar 4 Kode preparasi data menjadi CNN.....	22
Gambar 5 Kode Model CNN	25
Gambar 6 Struktur Model CNN.....	26
Gambar 7 Kode Training CNN	28
Gambar 8 Kode XAI Program.....	31
Gambar 9 Data hasil training CNN.....	34
Gambar 10 Global Feature Important Plot.....	37
Gambar 11 Summary Plot terhadap Longsor.....	37
Gambar 12 Force Plot SHAP individu untuk a. Tanah Longsor dan b. Non-.....	38
Gambar 13 Dependence Plot Elevasi dengan Slope	38
Gambar 14 Matriks Uji Korelasi antar Fitur	39
Gambar 15 Diagram Perbandingan SHAP dengan PF	40

DAFTAR TABEL

Tabel 1 Label Data.....	21
-------------------------	----

**LEMBAR PENGESAHAN
KERJA PRAKTIK**

**Pemanfaatan XAI (Xplainable Artificial Intelligence)
dalam Pemetaan Kerentanan Bencana Tanah Longsor di
Kecamatan Pacet Kab. Mojokerto**

Oleh:

Rano Noumi Sulistyo

5025211185

Disetujui oleh Pembimbing Kerja Praktik:

1. Ir. Ary Mazharuddin
Shiddiqi, S.Kom.,
M.Comp.Sc., Ph.D
NIP. 198106202005011003



(Pembimbing Departemen)

2. Prananda Navitas, S.T.,
M.Sc., Ph.D
NIP. 198205302009121005



(Pembimbing Lapangan)

Perancangan dan Implementasi Infrastruktur Sistem Penerimaan Peserta Didik Baru (PPDB) SMA Negeri Jawa Timur

Nama Mahasiswa : Rano Noumi Sulistyo
NRP : 5025211185
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Ary Mazharuddin S.Kom.,
M.Comp.Sc.
Pembimbing Lapangan : Prananda Navitas, S.T., M.Sc., Ph.D

ABSTRAK

Praktik ini mengembangkan sistem klasifikasi kerentanan tanah longsor berbasis Convolutional Neural Network (CNN) yang menerima masukan berupa data raster (.tif) multi-kanal. Proses pemrosesan data dilakukan dengan menggunakan pustaka GDAL untuk membaca dan mengelola data citra spasial. Dataset hasil ekstraksi kemudian dilatih dalam model CNN yang dilengkapi dengan Squeeze-and-Excitation (SE) dan Spatial Pyramid Pooling (SPP). Hasil pelatihan menunjukkan akurasi tinggi, dengan nilai Area Under Curve (AUC) sebesar 97,94% pada data pelatihan dan 97,31% pada data validasi, serta kestabilan loss selama 500 epoch.

Untuk meningkatkan interpretabilitas model, digunakan pendekatan Explainable AI (XAI) dengan metode SHAP (SHapley Additive exPlanations). Analisis SHAP menunjukkan fitur-fitur dominan seperti curvature profile, kemiringan lereng, penggunaan lahan, dan jarak dari jalan sebagai kontributor utama dalam prediksi. Visualisasi summary plot dan force plot berhasil mengungkap kontribusi positif dan negatif setiap fitur pada setiap prediksi, serta menunjukkan bahwa model membentuk keputusan

berdasarkan kombinasi fitur yang kompleks. Hasil ini membuktikan bahwa integrasi CNN dan SHAP efektif dalam mendukung sistem pemetaan kerentanan bencana berbasis data geospasial.

Kata Kunci : CNN, SHAP, GDAL, Longsor, Geospasial

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Pemanfaatan XAI (Xplainable Artificial Intelligence) dalam Pemetaan Kerentanan Bencana Tanah Longsor di Kecamatan Pacet Kab. Mojokerto

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Bapak Ary Mazharuddin S.Kom., M.Comp.Sc. selaku dosen pembimbing kerja praktik sekaligus koordinator kerja praktik.
3. Bapak Prananda Navitas, S.T., M.Sc., Ph.Dselaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.


Surabaya, 30 Juni 2025
Rano Noumi Sulisty

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pemetaan kerentanan tanah longsor menjadi sangat penting dalam upaya mitigasi bencana, terlebih di negara seperti Indonesia yang memiliki topografi kompleks dan risiko tinggi terhadap bencana geologis. Seiring berkembangnya teknologi, pendekatan tradisional dalam pemetaan mulai digantikan oleh kecerdasan buatan (AI), khususnya machine learning (ML), yang mampu mengolah berbagai variabel geografis dan lingkungan untuk menghasilkan prediksi yang lebih akurat dan efisien (Ado et al., 2022a; Melati et al., 2024). Dalam konteks ini, algoritma seperti Convolutional Neural Network (CNN) digunakan untuk menangkap pola-pola spasial dari data geospasial secara mendalam, yang relevan dalam mengidentifikasi daerah rawan longsor (Pradhan et al., 2023a).

Namun, kendala utama dari penggunaan model AI adalah kurangnya transparansi dalam pengambilan keputusan. Oleh karena itu, integrasi Explainable AI (XAI), khususnya metode SHAP (Shapley Additive exPlanations), menjadi sangat krusial. SHAP memberikan interpretasi kuantitatif terhadap kontribusi masing-masing fitur input dalam menghasilkan prediksi, sehingga memungkinkan para peneliti dan pengambil kebijakan untuk memahami faktor-faktor utama yang memengaruhi tingkat kerentanan tanah longsor (Pradhan et al., 2023a; Ado et al., 2022a). Dengan memahami peran variabel seperti curah hujan, kemiringan lereng, jenis tanah, dan penggunaan lahan secara lebih jelas, strategi mitigasi dapat disusun dengan lebih tepat sasaran (Irawan et al., 2021).

Penelitian terkini menunjukkan bahwa integrasi CNN dengan SHAP tidak hanya meningkatkan akurasi prediksi, tetapi juga memperkuat pemahaman terhadap data dan meningkatkan kepercayaan dalam penggunaannya untuk pengambilan keputusan (Pradhan et al., 2023a). Di Indonesia, pendekatan ini masih jarang digunakan, khususnya dalam konteks lokal seperti di Kecamatan Pacet, Kabupaten Mojokerto. Penelitian ini menjadi relevan karena menawarkan pendekatan yang lebih informatif dan transparan dalam memetakan risiko bencana. Dengan demikian, penerapan XAI seperti SHAP dalam konteks pemetaan kerentanan tanah longsor tidak hanya memberikan nilai teknis, tetapi juga mendukung pengambilan kebijakan yang lebih berbasis data dan bertanggung jawab (Irawan et al., 2022).

1.2. Tujuan

Tujuan kerja praktik ini adalah menyelesaikan kewajiban nilai kerja praktik sebesar 2 sks dan membantu Mas Mohammad Alvi Adi Bani dalam mengerjakan Tesis S2 dalam sisi machine learning dan XAI.

1.3. Manfaat

Manfaat yang diperoleh dengan penyelesaian Tesis Mas Alvi Adi Bani dengan judul “Pemetaan Kerentanan Tanah Longsor Berbasis Convolutional Neural Network (CNN) dan Analisis Interpretasi SHAP di Kecamatan Pacet, Kabupaten Mojokerto”

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana bisa mengaplikasikan CNN untuk data gambar longsor di Kab. Mojokerto?
2. Bagaimana menjelaskan bekerjanya CNN menggunakan XAI dan SHAP?

1.5. Lokasi dan Waktu Kerja Praktik

Praktik ini dilakukan dengan leluasa sesuai dengan kebutuhan Mas Alvi, sehingga banyak pekerjaan dilakukan secara *online* dengan beberapa pertemuan *offline* yang dilaksanakan di ITS.

Adapun kerja praktik dimulai pada tanggal 1 April 2025 hingga 30 Juni 2025.

1.6. Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik meliputi :

1.6.1. Perumusan Masalah

Untuk mengetahui kebutuhan dan spesifikasi dari kerja praktik dilakukan *Zoom Meeting* dengan kedua dosen pembimbing dan Mas Alvi. Disana dijelaskan bahwa perlu mengubah data peta Kab. Mojokerto yang berupa gambar sehingga bisa diterima oleh CNN dan dapat melakukan *Machine Learning* sehingga bisa mengetahui daerah di Mojokerto yang rawan longsor. Setelah itu model CNN perlu dijelaskan menggunakan *Explainable AI (XAI)* berupa *Shaply Additive Explanations (SHAP)*.

1.6.2. Studi Literatur

Setelah melakukan analisis tentang masalah dan keperluan dari Mas Alvi Adi Bani, kita akan menggunakan Python untuk Machine Learning dan ditentukan memerlukan 3 teknik utama, yaitu Melakukan translasi dari data peta menjadi data yang diterima CNN, Model CNN yang digunakan, dan implementasi XAI. Disini kami menentukan untuk menggunakan GDAL yang mampu membaca data peta, Model CNN dengan *Squeeze-and-Excitation* (SE) dan *Spatial Pyramid Pooling* (SPP), serta SHAP untuk menjelaskan hasil *training*.

1.6.3. Perancangan dan Implementasi Program

Setelah komponen utama sudah diidentifikasi, program perlu dirancang strukturnya dan diimplementasikan dengan menggunakan data peta Kab. Mojokerto sebagai bahan *training* model CNN, dilanjutkan dengan penjelasan model menggunakan SHAP.

1.6.4. Hasil dan Evaluasi

Setelah program yang telah direncanakan sudah terimplementasi dan berjalan, perlu dilakukan evaluasi hasil dari hasil *training* dan SHAP. Kita perlu memastikan tidak ada kesalahan dalam program sehingga hasil sesuai harapan.

1.6.5. Kesimpulan dan Saran

Program yang dibuat ini telah memenuhi syarat yang diinginkan, dan berjalan dengan baik dan lancar.

1.7. Sistematika Laporan

1.7.1. Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2. Bab II Profil Perusahaan

Bab ini berisi gambaran umum Dinas Pendidikan Provinsi Jawa Timur mulai dari profil, lokasi perusahaan.

1.7.3. Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4. Bab IV Perancangan dan Implementasi Program

Bab ini berisi mengenai tahap perancangan dan implementasi program dalam menyelesaikan proyek kerja praktik.

1.7.5. Bab VI Hasil dan Evaluasi

Bab ini berisi hasil dan evaluasi dari program yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.6. Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL PERUSAHAAN

2.1. Profil Departemen Perencanaan Wilayah dan Kota (PWK) ITS

Departemen Perencanaan Wilayah dan Kota (PWK) ITS berawal dari Laboratorium Perencanaan dan Perancangan Kota di Jurusan Arsitektur ITS, dan resmi berdiri pada tahun 2001 dengan penerimaan mahasiswa pertama di 2002. Departemen ini telah terakreditasi A oleh BAN-PT pada 2015 dan tengah berproses menuju sertifikasi Asean University Network (AUN). Saat ini PWK ITS memiliki 23 dosen dengan berbagai latar belakang pendidikan, serta kurikulum yang fokus pada perencanaan wilayah, kota, dan kawasan pesisir. Pembelajaran didukung dengan kelas berbahasa Inggris, kegiatan studio, dan riset.

Mahasiswa PWK ITS mempelajari berbagai aspek seperti kependudukan, lingkungan, dan kebijakan, serta dapat memilih spesialisasi seperti ekonomi wilayah, perancangan kota, perubahan iklim, GIS, transportasi, dan manajemen bencana. Hingga kini, PWK ITS telah meluluskan 636 sarjana yang bekerja di sektor publik dan swasta sebagai perencana, konsultan, peneliti, maupun pengelola proyek pembangunan.

2.2. Lokasi

Jl. Despro, Gedung A Lantai 2 Departemen Perencanaan Wilayah dan Kota Kampus ITS, Sukolilo Surabaya 60111

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

3.1. Machine Learning

Machine Learning (ML) adalah cabang dari kecerdasan buatan (*Artificial Intelligence/AI*) yang memungkinkan sistem belajar dari data dan membuat prediksi tanpa pemrograman eksplisit. Dalam konteks pemetaan kerentanan tanah longsor, ML digunakan untuk menganalisis hubungan kompleks antara faktor geospasial seperti curah hujan, kemiringan lereng, dan penggunaan lahan dengan kejadian longsor. Algoritma ML seperti *Random Forest*, *Support Vector Machine* (SVM), dan *Convolutional Neural Network* (CNN) mampu mengenali pola dan tren dari data historis serta data penginderaan jauh dengan akurasi tinggi (Ado et al., 2022a).

Penerapan ML dalam penelitian ini mempercepat proses analisis dan meningkatkan ketepatan pemetaan wilayah berisiko tinggi. ML juga menjadi dasar integrasi dengan metode *Explainable AI* (XAI) seperti SHAP, yang memperkuat aspek interpretabilitas hasil model.

3.2. Python

Python adalah bahasa pemrograman tingkat tinggi yang populer dalam bidang *data science* dan AI karena sintaksnya sederhana dan memiliki banyak pustaka pendukung. Dalam penelitian ini, *Python* digunakan untuk pemrosesan data, pelatihan model ML, dan integrasi dengan pustaka visualisasi serta XAI seperti SHAP. Ekosistem *Python* mencakup pustaka seperti *NumPy*, *Pandas*, *Scikit-learn*, *TensorFlow*, dan *PyTorch*.

Keunggulan *Python* terletak pada kemudahan integrasi dengan berbagai sumber data dan alat geospasial. Dalam pemetaan kerentanan tanah longsor, *Python* memungkinkan otomatisasi analisis dari ekstraksi fitur hingga evaluasi model. Kombinasi *Python* dan SHAP mendukung interpretasi hasil model ML secara transparan dan akuntabel

3.3. OSGEO dan GDAL

OSGeo (*Open Source Geospatial Foundation*) adalah organisasi yang mendukung pengembangan perangkat lunak geospasial sumber terbuka. Salah satu pustaka utama yang dikembangkan dan dikelola oleh *OSGeo* adalah *GDAL* (*Geospatial Data Abstraction Library*), sebuah pustaka yang dapat membaca, menulis, dan mengonversi berbagai format data geospasial raster dan vektor (Christl, 2014).

Dalam penelitian ini, *GDAL* berperan penting dalam membaca data spasial berbasis citra, seperti peta kerentanan atau *Digital Elevation Model* (DEM), sehingga data tersebut dapat digunakan langsung sebagai input dalam model *machine learning*. Dengan *GDAL*, peneliti dapat melakukan transformasi data geospasial secara efisien, seperti reproyeksi, pemotongan wilayah, dan ekstraksi nilai piksel, yang menjadi tahapan krusial dalam menyiapkan data pelatihan yang sesuai untuk CNN.

3.4. CNN

Convolutional Neural Network (CNN) adalah jenis jaringan saraf tiruan yang dirancang untuk memproses data berbentuk grid seperti citra atau peta spasial. CNN secara otomatis mengenali pola visual dan spasial melalui lapisan konvolusi yang mengekstraksi fitur penting dari data. Dalam pemetaan tanah longsor, CNN efektif dalam mendeteksi fitur spasial dari data geospasial dan menghasilkan prediksi kerentanan wilayah (Pradhan et al., 2023).

CNN unggul dalam menangani kompleksitas data spasial. CNN dapat memproses peta multilapis yang mencakup informasi seperti ketinggian, kemiringan, curah hujan, dan tekstur tanah. Integrasi CNN dan SHAP memungkinkan interpretasi mendalam terhadap fitur yang paling memengaruhi kerentanan tanah longsor.

3.5. Squeeze-and-Excitation (SE) dan Spatial Pyramid Pooling (SPP)

Squeeze-and-Excitation (SE) adalah mekanisme perhatian dalam CNN yang memberikan bobot lebih besar pada fitur penting dan mengabaikan fitur tidak relevan. SE bekerja dengan merangkul informasi spasial menjadi deskriptor *channel* dan kemudian menguatkan *channel* yang relevan melalui mekanisme perhatian (Hu et al., 2019).

Spatial Pyramid Pooling (SPP) adalah teknik *pooling* yang memungkinkan ekstraksi fitur dari input berdimensi variabel tanpa kehilangan informasi spasial penting (He et al., 2014). SPP mendukung pemrosesan data spasial dari berbagai skala secara bersamaan, berguna dalam menangani peta geospasial dengan resolusi berbeda. Gabungan SE dan SPP memperkuat representasi fitur spasial dan meningkatkan akurasi prediksi kerentanan tanah longsor.

3.6. SHAP

SHAP (*SHapley Additive exPlanations*) adalah metode XAI berbasis teori permainan *Shapley value* yang menjelaskan kontribusi setiap fitur input terhadap output model prediktif. SHAP memberikan nilai kuantitatif yang menunjukkan seberapa besar pengaruh suatu fitur terhadap hasil prediksi (Lundberg & Lee, 2017). Dalam penelitian ini, SHAP menjelaskan prediksi model CNN mengenai wilayah yang rentan terhadap tanah longsor.

SHAP unggul karena menghasilkan interpretasi konsisten, adil, dan dapat diukur untuk setiap prediksi. Dalam konteks pemetaan kerentanan, SHAP membantu peneliti dan pengambil kebijakan memahami alasan suatu wilayah dikategorikan sebagai rawan longsor. Dengan demikian, SHAP meningkatkan transparansi model dan mendukung pengambilan keputusan berbasis data. Untuk menonjolkan poin ini dalam tesis, akan dilakukan komparasi dengan XAI lain, yaitu *Permutation Feature Importance*

BAB IV

PERANCANGAN DAN IMPLEMENTASI PROGRAM

4.1. Perancangan Program

Pada bab ini akan dijelaskan mengenai tahapan dalam membangun arsitektur system. Hal tersebut dijelaskan ke dalam dua bagian Desain secara meluruh dan Algoritma program.

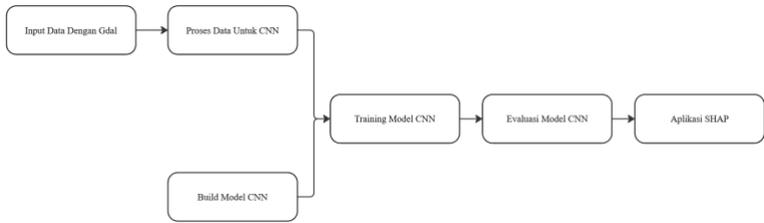
4.1.1. Desain Arsitektur Program

Desain program pemodelan kerentanan tanah longsor menggunakan *Convolutional Neural Network* (CNN) dan *SHAP* terdiri dari beberapa tahapan terstruktur. Proses dimulai dengan memuat data raster geospasial menggunakan pustaka *GDAL*. Data raster ini terdiri dari beberapa *band* yang masing-masing dibaca dan diratakan menjadi array satu dimensi. Selanjutnya, semua *band* tersebut ditumpuk untuk membentuk matriks fitur (*feature matrix*) X . Label longsor yang disediakan dalam bentuk raster juga diratakan menjadi vektor label Y . Kedua elemen ini (X dan Y) akan digunakan dalam proses pembelajaran mesin.

Setelah data berhasil dimuat, tahap berikutnya adalah praproses data agar sesuai dengan format masukan CNN. Matriks fitur dinormalisasi dengan skala antara 0 dan 1, sementara label Y dikodekan dalam format biner. Data kemudian diubah bentuknya menjadi format empat dimensi (n, h, w, c), yaitu jumlah sampel, tinggi, lebar, dan jumlah saluran, sesuai kebutuhan arsitektur CNN. Selanjutnya, data dibagi menjadi dua bagian yaitu data latih dan data uji, masing-masing terdiri dari pasangan (X_{train}, Y_{train}) dan (X_{test}, Y_{test}).

Model CNN kemudian dibangun dengan beberapa lapisan utama, termasuk lapisan masukan, konvolusi dengan fungsi aktivasi *ReLU*, *MaxPooling*, dan lapisan *fully connected*. Lapisan keluaran menggunakan fungsi aktivasi *sigmoid* karena tugas klasifikasi bersifat biner. Model dikompilasi dengan fungsi kerugian *binary crossentropy* dan *optimizer* Adam. Setelah itu, model dilatih menggunakan data latih, dengan jumlah *epoch* dan ukuran *batch* yang telah ditentukan sebelumnya. Model yang telah dilatih kemudian dievaluasi menggunakan data uji untuk menghitung akurasi dan *loss*.

Tahap akhir dari proses ini adalah menjelaskan hasil prediksi model menggunakan pendekatan *Explainable AI (XAI)*, khususnya dengan metode *SHAP GradientExplainer*. Penjelasan dilakukan dengan menghitung nilai SHAP untuk setiap fitur input menggunakan data uji. Nilai ini menunjukkan seberapa besar kontribusi masing-masing fitur terhadap hasil prediksi. Visualisasi seperti *summary plot* SHAP kemudian dihasilkan untuk membantu peneliti memahami pengaruh relatif dari setiap variabel. Dengan demikian, integrasi CNN dan SHAP tidak hanya menghasilkan model prediktif yang akurat, tetapi juga model yang dapat dijelaskan secara transparan dan informatif.



Gambar 1 Gambar Desain Program

4.2. Implementasi Program

4.2.1. Preprocessing data

Sebelum data digunakan dalam pelatihan model *Convolutional Neural Network* (CNN) untuk *Landslide Susceptibility Mapping* (LSM), penting untuk memastikan kesesuaian dimensi spasial dan sistem referensi geografis antara seluruh fitur dan label. Untuk itu, dilakukan tahap validasi data awal menggunakan skrip *Python* berbasis pustaka *GDAL*. Skrip ini membaca properti dari seluruh berkas raster (*GeoTIFF*) dan melakukan pemeriksaan terhadap tiga aspek utama, yaitu kesesuaian ukuran raster, transformasi spasial, dan kesiapan data untuk digunakan dalam pipeline CNN. Pemeriksaan ini bertujuan memastikan bahwa setiap *layer* memiliki lebar dan tinggi yang konsisten, serta berada dalam sistem proyeksi dan resolusi spasial yang seragam. Ketidaksesuaian dalam aspek-aspek tersebut dapat menyebabkan kesalahan pembacaan data atau gangguan dalam proses normalisasi.

Skrip ini juga bertindak sebagai filter awal untuk mendeteksi kesalahan potensial seperti tumpang tindih yang tidak tepat antar *layer*, perbedaan sistem koordinat, atau format data yang tidak sesuai. Dengan memverifikasi kesamaan transformasi geospasial dan dimensi raster sejak awal, sistem dapat mencegah kegagalan proses saat pelatihan model CNN. Validasi awal ini sangat penting untuk menjaga integritas data serta memastikan input yang bersih dan konsisten, sehingga proses pelatihan dapat berjalan optimal dan hasil model memiliki tingkat akurasi yang tinggi.

```

1 import os
2 from osgeo import gdal
3 import argparse
4 import sys
5 parser = argparse.ArgumentParser()
6 parser.add_argument("--feature_path", default='data_kotak/feature/', type=str)
7 parser.add_argument("--label_path", default='data_kotak/label/label1.tif', type=str)
8 args = parser.parse_args()
9
10 def get_tiff_attributes(tiff_path):
11     tiff = gdal.Open(tiff_path)
12     width, height = tiff.RasterXSize, tiff.RasterYSize
13     transform = tiff.GetGeoTransform()
14     return width, height, transform
15
16 consistency = {"w": [], "h": [], "transform": []}
17 #label
18 w, h, transform = get_tiff_attributes(args.label_path)
19 consistency["w"].append(w)
20 consistency["h"].append(h)
21 consistency["transform"].append([int(value) for value in transform])
22 #feature
23 for tif_data in os.listdir(args.feature_path):
24     w, h, transform = get_tiff_attributes(os.path.join(args.feature_path, tif_data))
25     print('Reading '+tif_data+' data.')
26     consistency["w"].append(w)
27     consistency["h"].append(h)
28     consistency["transform"].append([int(value) for value in transform])
29 consistency["w"] = set(consistency["w"])
30 consistency["h"] = set(consistency["h"])
31 if(len(consistency["w"])-1):
32     print("The width of the data is different")
33     print(consistency["w"])
34     sys.exit()
35 if(len(consistency["h"])-1):
36     print("The height of the data is different")
37     print(consistency["h"])
38     sys.exit()
39 flag = consistency["transform"][0] # Get the first tuple in the list
40 if(not(all(i == flag for i in consistency["transform"]))):
41     print("The transform of the data is different")
42     print(consistency["transform"])
43     sys.exit()
44
45 print("datamu aman lue....")

```

Gambar 2 Skrip data check

Sebelum memulai pelatihan model *Convolutional Neural Network* (CNN), dilakukan tahap persiapan data untuk memastikan validitas dan konsistensi antara citra raster dan label longsor. Data fitur dibaca dari berkas *GeoTIFF* *multi-layer* yang merepresentasikan berbagai variabel seperti kemiringan lereng, curah hujan, litologi, dan lain-lain.

Pembacaan ini dilakukan menggunakan pustaka *GDAL*, lalu seluruh nilai piksel pada setiap *layer* dinormalisasi ke dalam rentang [0, 1] menggunakan metode *min-max normalization*. Tujuannya adalah untuk menyamakan skala antar fitur agar proses pembelajaran CNN tidak terpengaruh oleh perbedaan skala numerik antar variabel. Selain itu, dilakukan *padding* pada raster untuk memastikan bahwa semua piksel, termasuk yang berada di tepi wilayah studi, tetap dapat diproses secara utuh dalam jendela konvolusi.

```
def read_data_from_tif(tif_path):
    """
    Read impact Factor data and convert to nparray
    """
    tif = gdal.Open(tif_path)
    w, h = tif.RasterXSize, tif.RasterYSize
    img = np.array(tif.ReadAsArray(0, 0, w, h).astype(np.float32))
    return img

def normalize_min_max(data):
    min_val = np.min(data)
    max_val = np.max(data)
    normed = (data - min_val) / (max_val - min_val)
    return normed, min_val, max_val

def get_feature_data(tif_paths, window_size, norm_param_path=None):
    """
    Read features and min-max normalization, optionally save normalization params
    """
    n = int(window_size/2)
    data = []
    norm_params = {}
    for tif_data in os.listdir(tif_paths):
        img = read_data_from_tif(os.path.join(tif_paths, tif_data))
        normed, min_val, max_val = normalize_min_max(img)
        norm_params[tif_data] = {'min': float(min_val), 'max': float(max_val)}
        w, h = img.shape[0], img.shape[1]
        print(str(tif_data)+' amann enjoy...')
        normed = np.pad(normed, ((n, n), (n, n)), 'constant', constant_values=(0, 0)) # Filling edges
        data.append(normed)
    if norm_param_path:
        with open(norm_param_path, 'w') as f:
            json.dump(norm_params, f)
    return w, h, len(data), np.array(data)

def get_label_data(tif_path, window_size):
    """
    Read label data
    """
    n = int(window_size/2)
    img = read_data_from_tif(tif_path)
    img = np.pad(img, ((n, n), (n, n)), 'constant', constant_values = (0.1, 0.1)) #Filling 0.1
    return img
```

Gambar 3 Skrip ambil data tif dan transformasi menjadi data untuk

Langkah berikutnya adalah pembuatan *dataset* untuk pelatihan CNN. Label disiapkan dalam bentuk raster yang menyimpan informasi spasial mengenai lokasi longsor dan area non-longsor, yang kemudian dikodekan ke dalam empat nilai berbeda. Tabel berikut menjelaskan skema pengkodean label:

Tabel 1 Label Data

Nilai	Keterangan
0	Titik Longsor Training
1	Titik Longsor Test
2	Titik Tidak Longsor Training
3	Titik Tidak Longsor Test

Setelah pengkodean, dilakukan pengacakan citra dan label menggunakan fungsi `shuffle_image_label` untuk menjamin distribusi data yang acak. Data kemudian diubah menjadi format array *NumPy*, dan label disusun ulang dalam bentuk vektor kolom. Sistem mencetak bentuk (*shape*) dari dataset sebagai verifikasi akhir, kemudian mengembalikan citra pelatihan, label pelatihan, citra validasi, dan label validasi sebagai keluaran siap pakai untuk proses pelatihan model.

```

53 def shuffle_image_label(images, labels):
54     data_shuffle
55     """
56     randomnum = random.randint(0, len(images))
57     random.seed(randomnum)
58     random.shuffle(images)
59     random.seed(randomnum)
60     random.shuffle(labels)
61     return images, labels
62
63
64 def get_CNN_data(data, label, n):
65     """
66     Creating a CNN dataset
67     """
68     n = int(n/2)
69     #Create train sets, labels
70     train_data = []
71     mask_0 = label == 0
72     i_indices_0, j_indices_0 = np.where(mask_0)
73     mask_1 = label == 2
74     i_indices_1, j_indices_1 = np.where(mask_1)
75     for i, j in zip(i_indices_0, j_indices_0):
76         train_data.append((data[:,i-n:i+n+1,j-n:j+n+1],0))
77     for i, j in zip(i_indices_1, j_indices_1):
78         train_data.append((data[:,i-n:i+n+1,j-n:j+n+1],1))
79     #Create validation sets, labels
80     val_data = []
81     mask_2 = label == 1
82     i_indices_2, j_indices_2 = np.where(mask_2)
83     mask_3 = label == 3
84     i_indices_3, j_indices_3 = np.where(mask_3)
85     for i, j in zip(i_indices_2, j_indices_2):
86         val_data.append((data[:,i-n:i+n+1,j-n:j+n+1],0))
87     for i, j in zip(i_indices_3, j_indices_3):
88         val_data.append((data[:,i-n:i+n+1,j-n:j+n+1],1))
89
90     train_imgs = [item[0] for item in train_data]
91     train_labels = [item[1] for item in train_data]
92     val_imgs = [item[0] for item in val_data]
93     val_labels = [item[1] for item in val_data]
94
95     train_imgs, train_labels = shuffle_image_label(train_imgs, train_labels)
96     val_imgs, val_labels = shuffle_image_label(val_imgs, val_labels)
97
98     train_imgs, val_imgs = np.array(train_imgs), np.array(val_imgs)
99     train_labels, val_labels = np.array(train_labels).reshape((-1,1)), np.array(val_labels).reshape((-1,1))
100     print(train_imgs.shape, val_imgs.shape, train_labels.shape, val_labels.shape)
101     return train_imgs, train_labels, val_imgs, val_labels

```

Gambar 4 Kode preparasi data menjadi CNN

4.2.2. CNN

Model Convolutional Neural Network (CNN) yang digunakan dalam penelitian ini dikembangkan secara khusus untuk memetakan kerentanan longsor berdasarkan data *raster* multivariat. Model ini dinamakan *LSM_cnn*, yang terdiri dari beberapa blok konvolusi dalam, disertai

dengan mekanisme *self-attention* ringan berbasis *Squeeze-and-Excitation* (SE) dan diakhiri dengan agregasi spasial melalui lapisan *Spatial Pyramid Pooling* (SPP).

Model LSM_cnn dirancang dengan struktur hierarkis yang terdiri atas tiga blok konvolusi utama. Blok pertama (Conv Block 1) terdiri dari dua lapisan konvolusi berukuran 3×3 dengan 128 filter, diikuti oleh fungsi aktivasi ReLU, batch normalization, max pooling, dan dropout dengan rate 0,3. Blok ini juga dilengkapi dengan *Squeeze-and-Excitation Layer* (SELayer) untuk menyesuaikan bobot antar channel berdasarkan informasi spasial global. Blok kedua (Conv Block 2) mengulangi pola yang sama namun dengan peningkatan jumlah filter menjadi 256, serta penggunaan SELayer kedua untuk memperkuat pemilihan fitur penting antar kanal. Pada Conv Block 3, digunakan dua lapisan konvolusi dengan 512 filter dan dropout, namun tanpa pooling untuk mempertahankan resolusi spasial sebelum ekstraksi fitur akhir.

Setelah proses konvolusi, fitur spasial diekstraksi menggunakan *Modified Spatial Pyramid Pooling* (SPP). Teknik ini memungkinkan model menerima masukan dalam ukuran patch yang bervariasi serta menghasilkan representasi fitur dengan ukuran tetap. Pendekatan ini sangat bermanfaat dalam pengolahan data raster yang memiliki dimensi bervariasi, terutama untuk kasus pemetaan kerentanan tanah longsor. Integrasi SPP memperkuat fleksibilitas model dalam menangani input spasial tanpa mengorbankan informasi kontekstual dari berbagai skala spasial.

Selanjutnya, fitur yang telah diekstraksi dari SPP diteruskan ke lapisan fully connected untuk proses

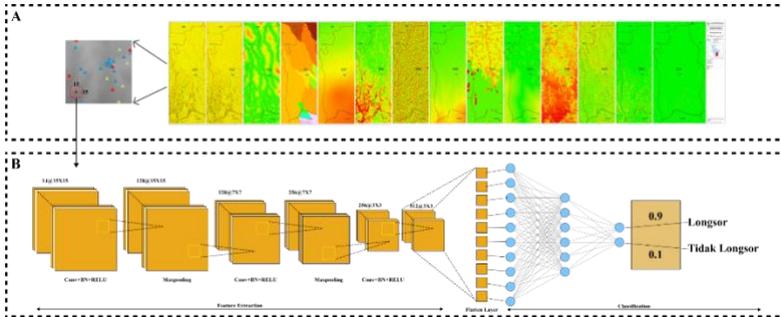
klasifikasi. Struktur lapisan ini terdiri dari dua bagian utama, yaitu fc1, yang merupakan lapisan linier dengan 512 neuron dan dropout sebesar 0,5 untuk mencegah overfitting. Kemudian, dilanjutkan dengan fc2, yakni lapisan keluaran dengan 2 neuron yang merepresentasikan kelas biner: longsor dan non-longsor. Keluaran dari lapisan ini akan digunakan dalam proses pelatihan menggunakan fungsi loss CrossEntropyLoss, yang secara efektif menghitung kesalahan klasifikasi dan membimbing proses pembaruan bobot model secara iteratif.

```

1 #attention
2 class SELayer(nn.Module):
3     def __init__(self,channel,reduction=16):
4         super(SELayer,self).__init__()
5         self.avg_pool = nn.AdaptiveAvgPool2d(1)
6         self.fc = nn.Sequential(
7             nn.Linear(channel,channel//reduction,bias=False),
8             nn.ReLU(inplace=True),
9             nn.Linear(channel//reduction,channel,bias=False),
10            nn.Sigmoid()
11        )
12    def forward(self,x):
13        b,c,_ = x.size()
14        y = self.avg_pool(x).view(b,c)
15        y = self.fc(y).view(b,c,1,1)
16        return x*y.expand_as(x)
17
18 class LPM_cnn(nn.Module):
19     def __init__(self, in_chanel):
20         super(LPM_cnn, self).__init__()
21         self.conv1 = nn.Sequential(
22             nn.Conv2d(in_chanel, 128, 3, 1, 1),
23             nn.ReLU(),
24             nn.BatchNorm2d(128),
25             nn.Conv2d(128, 128, 3, 1, 1),
26             nn.ReLU(),
27             nn.BatchNorm2d(128),
28             nn.MaxPool2d(2),
29             nn.Dropout(0.3)
30         )
31         self.se1 = SELayer(channel=128, reduction=16)
32
33         self.conv2 = nn.Sequential(
34             nn.Conv2d(128, 256, 3, 1, 1),
35             nn.ReLU(),
36             nn.BatchNorm2d(256),
37             nn.Conv2d(256, 256, 3, 1, 1),
38             nn.ReLU(),
39             nn.BatchNorm2d(256),
40             nn.MaxPool2d(2),
41             nn.Dropout(0.3)
42         )
43         self.se2 = SELayer(channel=256, reduction=16)
44
45         self.conv3 = nn.Sequential(
46             nn.Conv2d(256, 512, 3, 1, 1),
47             nn.ReLU(),
48             nn.BatchNorm2d(512),
49             nn.Conv2d(512, 512, 3, 1, 1),
50             nn.ReLU(),
51             nn.BatchNorm2d(512),
52             nn.Dropout(0.3)
53         )
54
55         self.fc1 = nn.Linear(2560, 512)
56         self.dropout = nn.Dropout(0.5)
57         self.fc2 = nn.Linear(512, 2)
58
59     def forward(self, x):
60         x = self.conv1(x)
61         x = self.se1(x)
62         x = self.conv2(x)
63         x = self.se2(x)
64         x = self.conv3(x)
65         spp = Modified_SPPayer(2).forward(x)
66         x = self.fc1(spp)
67         x = self.dropout(x)
68         out = self.fc2(x)
69         return out

```

Gambar 5 Kode Model CNN



Gambar 6 Struktur Model CNN

4.2.3. Training

Kode pelatihan model *LSM_cnn* dimulai dengan membaca argumen input yang ditentukan pengguna, seperti jalur data fitur dan label, ukuran jendela (*window size*), laju pembelajaran (*learning rate*), ukuran *batch*, dan jumlah *epoch*. Program kemudian memuat data fitur raster multi-kanal dan label menggunakan fungsi yang tersedia di kodel `data_prepare`. Data ini diubah ke dalam format yang sesuai untuk CNN melalui proses ekstraksi patch berdasarkan ukuran jendela yang ditentukan, lalu dibagi menjadi *training set* dan *validation set*, masing masing 70/30. Setiap subset data ini dimasukkan ke dalam *DataLoader* untuk proses pelatihan dan validasi bertahap, dengan ukuran *batch* sebesar 128.

Model CNN dibangun dari kelas `LSM_cnn` dengan jumlah fitur yang disesuaikan, dan dijalankan di perangkat *GPU*

(*cuda*). Untuk optimisasi, digunakan *Stochastic Gradient Descent* (SGD) dengan laju pembelajaran awal 0,0001 serta penurunan laju bertahap menggunakan *StepLR* setiap 50 *epoch* dengan faktor pengurang 0,1. Fungsi kerugian yang digunakan adalah *CrossEntropyLoss*, sesuai dengan kebutuhan klasifikasi biner. Proses pelatihan berlangsung selama 500 *epoch*, di mana pada setiap iterasi, model melakukan proses *forward pass*, menghitung *loss*, melakukan *backpropagation*, dan memperbarui bobot model. Data akurasi dan *loss* untuk setiap *epoch* dicatat dan disimpan untuk keperluan evaluasi.

Pada tahap evaluasi, model memasuki mode inferensi dan dilakukan penghitungan akurasi serta *loss* terhadap data validasi. Jika akurasi validasi terbaik tercapai, model akan menyimpan bobot terbaik ke dalam berkas `best.pth`, serta menghasilkan visualisasi kurva AUC untuk data latih dan validasi. Kurva akurasi dan *loss* juga digambarkan menggunakan fungsi `draw_acc` dan `draw_loss`. Di akhir pelatihan, seluruh rekaman nilai akurasi dan *loss* tersimpan dalam berkas `record.json`, dan model versi terakhir disimpan sebagai `latest.pth`. Dengan struktur pelatihan ini, model *LSM_cnn* dibangun untuk menghasilkan klasifikasi kerentanan longsor secara akurat, efisien, dan dapat ditinjau ulang melalui catatan pelatihan dan visualisasi performa.

```

def parse_args():
    parser = argparse.ArgumentParser(description="Train CNN Processes on data")
    parser.add_argument("--feature_path", default="data/kotak/feature/", type=str)
    parser.add_argument("--label_path", default="data/kotak/label/label.tif", type=str)
    parser.add_argument("--window_size", default=15, type=int)
    parser.add_argument("--lr", default=0.0001, type=float)
    parser.add_argument("--batch_size", default=128, type=int)
    parser.add_argument("--epochs", default=500, type=int)
    args = parser.parse_args()
    return args

def main():
    args = parse_args()

    n_feature, data = dp.get_feature_data(args.feature_path, args.window_size)
    label = dp.get_label_data(args.label_path, args.window_size)
    alldata_train, alltarget_train, alldata_val, alltarget_val = dp.get_CNN_data(data, label, args.window_size)

    train_dataset = TensorDataset(torch.from_numpy(alldata_train).float(), torch.from_numpy(alltarget_train).float())
    train_loader = torch.utils.data.DataLoader(dataset=train_dataset, batch_size=args.batch_size, shuffle=True)
    val_dataset = TensorDataset(torch.from_numpy(alldata_val).float(), torch.from_numpy(alltarget_val).float())
    val_loader = torch.utils.data.DataLoader(dataset=val_dataset, batch_size=args.batch_size, shuffle=True)

    model = LR_CNN(n_feature).to('cuda')
    # loss function and optimizers
    criterion = nn.CrossEntropyLoss().to('cuda')
    optimizer = optim.Adam(model.parameters(), lr=args.lr)
    scheduler = optim.lr_scheduler.StepLR(optimizer, step_size=50, gamma=0.1)
    max_acc=0

    record = {"train": {"acc": [], "loss": []}, "val": {"acc": [], "loss": []}}
    for epoch in range(args.epochs):
        train_acc = 0.0
        train_loss = 0.0
        val_acc = 0.0
        val_loss = 0.0
        train_outputs_list = []
        train_labels_list = []
        val_outputs_list = []
        val_labels_list = []
        model.train()
        for images, target in train_loader:
            #backward
            images, target = images.to('cuda'), target.to('cuda')
            optimizer.zero_grad()
            outputs = model(images)
            _, preds = torch.max(outputs, data=1)
            loss = criterion(outputs, target.squeeze()).log()
            loss.backward()
            optimizer.step()

            train_outputs_list.append(outputs.detach().cpu().numpy())
            train_array=np.array(train_outputs_list)
            train_labels_list.extend(target.cpu().numpy())

            train_acc += (preds[... ,None]==target.squeeze()).sum().cpu().numpy()
            train_loss += loss.item()

        record["train"]["loss"].append(train_loss/len(train_dataset))
        record["train"]["acc"].append(train_acc / len(train_dataset))

        model.eval()
        with torch.no_grad():
            for images, target in val_loader:
                images, target = images.to('cuda'), target.to('cuda')
                outputs = model(images)
                loss = criterion(outputs, target.squeeze()).log()
                val_loss += loss.item()

                val_outputs_list.extend(outputs.detach().cpu().numpy())
                val_labels_list.extend(target.cpu().numpy())
                score_array=np.array(val_outputs_list)
                _, preds = torch.max(outputs, data, 3)
                val_acc += (preds[... ,None]==target.squeeze()).sum().cpu().numpy()

        print("%30s/%30s Train Acc: %3.0f loss: %3.0f | Val Acc: %3.0f loss: %3.0f" % \
              (epoch + 1, args.epochs, \
               train_acc / len(train_dataset), train_loss / len(train_dataset), val_acc / len(val_dataset), \
               val_loss / len(val_dataset)))
        if (val_acc / len(val_dataset)) > max_acc:
            max_acc = val_acc / len(val_dataset)

            drawACC_TwoClass(val_labels_list, score_array[:,1], "val_ACC.png")
            drawACC_TwoClass(train_labels_list, train_array[:,1], "train_ACC.png")
            torch.save(model.state_dict(), "model1/best.pth")

```

Gambar 7 Kode Training CNN

4.2.4. XAI

Kode SHAP ini bertujuan menjelaskan kontribusi setiap fitur terhadap prediksi model CNN dalam pemetaan kerentanan longsor. Pertama, model *LSM_cnn* dimuat dari berkas parameter terlatih dan disiapkan dalam mode inferensi. Data fitur dan label diproses dari citra raster menggunakan `data_prepare`, lalu dibagi menjadi *background data* (100 sampel pelatihan) dan *test data* (200 sampel validasi). Model menerima data dalam format tensor dan dijalankan di perangkat *GPU* jika tersedia. Proses ini menggunakan *SHAP GradientExplainer*, yang menghitung nilai kontribusi (nilai SHAP) untuk masing-masing fitur dalam konteks kelas longsor (1) dan non-longsor (0). Hasil nilai SHAP dirata-ratakan secara spasial untuk menyederhanakan interpretasi.

Nilai SHAP yang diperoleh dari kelas longsor dan non-longsor kemudian divisualisasikan menggunakan *summary plot* dalam bentuk *violin plot*. Plot ini menyajikan sebaran nilai kontribusi masing-masing fitur (seperti kemiringan, curah hujan, NDVI, dan TWI) terhadap prediksi model, sehingga memudahkan interpretasi fitur mana yang paling berpengaruh. Visualisasi disimpan dalam direktori `Final_SHAP_Results` dalam dua berkas gambar: satu untuk kelas longsor dan satu untuk non-longsor. Proses ini memberikan pemahaman kuantitatif dan visual yang kuat mengenai faktor geospasial penyebab longsor, yang penting untuk pengambilan keputusan berbasis data.

Selain SHAP, kode ini juga menyertakan metode pembandingan menggunakan *Permutation Feature Importance* (PFI). PFI mengukur penurunan akurasi model saat nilai-nilai dari satu fitur diacak secara acak. Proses ini dilakukan untuk semua fitur dan perbedaan akurasi (sebelum dan sesudah permutasi) disimpan sebagai skor pentingnya fitur. Hasilnya, skor PFI menunjukkan sejauh mana ketergantungan model terhadap fitur tertentu.

```

17 # KODINGRASI -----
18 FEATURE_NAMES = [
19     "aspect", "elevasi", "hujan", "jalan", "landuse", "litologi", "mvt",
20     "plan", "prof", "slope", "spt", "sti", "sungai", "tmt"
21 ]
22 FEATURE_FILES = [{"name".tif" for name in FEATURE_NAMES]
23 MODEL_PATH = "hasil_kotak_02_05/hasil_terbaik.pth"
24 FEATURE_PATH = "data_kotak/feature/"
25 LABEL_PATH = "data_kotak/label/label1.tif"
26 WINDOW_SIZE = 15
27 DEVICE = torch.device("cuda" if torch.cuda.is_available() else "cpu")
28 OUTPUT_DIR = "final_SHAP_Results"
29 os.makedirs(OUTPUT_DIR, exist_ok=True)
30
31 # ----- BACA PARAMETER NORMALISASI -----
32 with open("norm_params.json", "r") as f:
33     norm_params = json.load(f)
34
35 def inverso_normalise(value, min_val, max_val):
36     return value * (max_val - min_val) + min_val
37
38 # ----- LOADING DATA & MODEL -----
39 n_feature, data = dp.get_feature_data(FEATURE_PATH, WINDOW_SIZE)
40 label = dp.get_label_data(LABEL_PATH, WINDOW_SIZE)
41 train_x, train_y, val_x, val_y = dp.get_CMR_data(data, label, WINDOW_SIZE)
42
43 X_background = train_x[:100]
44 X_test = val_x
45 y_test = val_y[:100].squeeze()
46
47 X_bg_tensor = torch.tensor(X_background, dtype=torch.float32).to(DEVICE)
48 X_test_tensor = torch.tensor(X_test, dtype=torch.float32).to(DEVICE)
49
50 model = LSP_cnn(n_feature).to(DEVICE)
51 model.load_state_dict(torch.load(MODEL_PATH, map_location=DEVICE))
52 model.eval()
53
54 # ----- SHAP EXPLANATION -----
55 explainer = shap.GradientExplainer(model, X_bg_tensor)
56 shap_values = explainer.shap_values(X_test_tensor)
57
58 # SHAP untuk kelas 1 (longsor)
59 shap_vals_1 = shap_values[1][:, 1, :, :].mean(axis=(2, 3))
60 # SHAP untuk kelas 0 (tidak longsor)
61 shap_vals_0 = shap_values[0][:, 1, :, :].mean(axis=(2, 3))
62
63 X_flat = X_test[:, shap_vals_1.shape[0]].mean(axis=(2, 3))
64
65 # Menormalisasi kolom jika perlu
66 if shap_vals_1.shape[1] > X_flat.shape[1]:
67     shap_vals_1 = shap_vals_1[:, :X_flat.shape[1]]
68 if shap_vals_0.shape[1] > X_flat.shape[1]:
69     shap_vals_0 = shap_vals_0[:, :X_flat.shape[1]]
70
71 # ----- PERMUTATION FEATURE IMPORTANCE -----
72 pfi_scores = []
73 acc_orig = accuracy_score(y_test, model(X_test_tensor).argmax(dim=1).cpu().numpy())
74
75 for i in range(n_feature):
76     X_perm = X_test.copy()
77     np.random.shuffle(X_perm[:, i, :, :])
78     X_perm_tensor = torch.tensor(X_perm, dtype=torch.float32).to(DEVICE)
79     y_pred_perm = model(X_perm_tensor).argmax(dim=1).cpu().numpy()
80     acc_shuffled = accuracy_score(y_test, y_pred_perm)
81     pfi_scores.append(acc_orig - acc_shuffled)
82
83 # ----- SHAP vs PFI COMPARISON -----
84 mean_shap_vals = np.abs(shap_vals_1).mean(axis=0)
85 x = np.arange(n_feature)
86 bar_width = 0.15
87
88 plt.figure(figsize=(12, 6))
89 plt.bar(x - bar_width/2, mean_shap_vals, width=bar_width, label="SHAP (Class 1)")
90 plt.bar(x + bar_width/2, pfi_scores, width=bar_width, label="PFI")
91 plt.xticks(x, FEATURE_NAMES, rotation=45)
92 plt.ylabel("Feature Importance")
93 plt.title("SHAP (Longsor) vs Permutation Feature Importance")
94 plt.legend()
95 plt.tight_layout()
96 plt.savefig(os.path.join(OUTPUT_DIR, "shap_vs_pfi.png"))
97 plt.close()

```

Gambar 8 Kode XAI Program

BAB V HASIL DAN EVALUASI

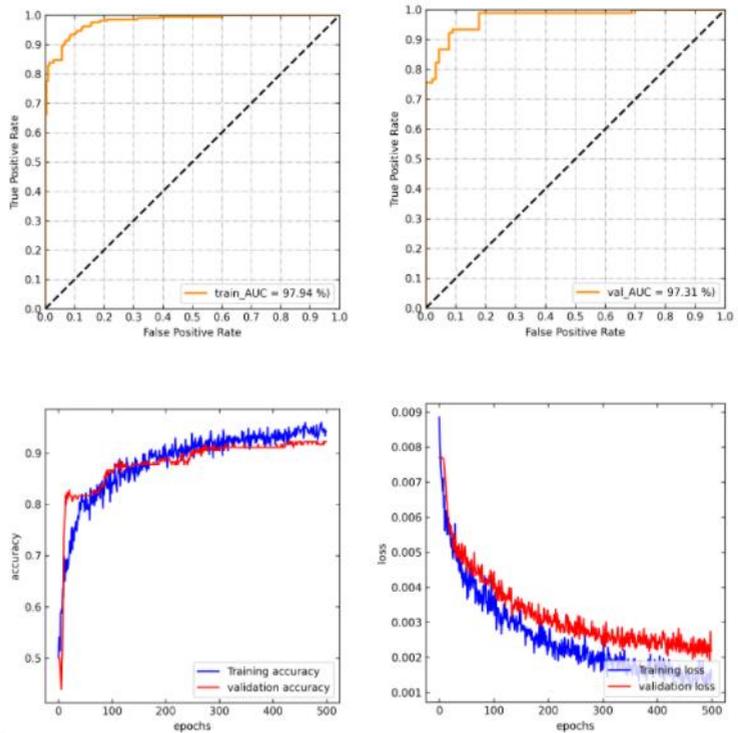
5.1. Training CNN

Hasil pelatihan model *LSM_cnn* menunjukkan performa klasifikasi yang sangat baik berdasarkan grafik *Receiver Operating Characteristic (ROC)*. Nilai Area Under Curve (AUC) untuk data pelatihan tercatat sebesar 97,94%, sedangkan AUC untuk data validasi mencapai 97,31%. Nilai AUC yang tinggi dan konsisten antara data pelatihan dan validasi ini mengindikasikan bahwa model memiliki kemampuan generalisasi yang sangat baik dan tidak mengalami *overfitting* secara signifikan.

Grafik akurasi memperlihatkan tren peningkatan yang stabil sepanjang 500 *epoch*. Akurasi pelatihan meningkat dari sekitar 65% di awal menjadi lebih dari 93% pada akhir pelatihan. Sementara itu, akurasi validasi juga meningkat secara bertahap dan sejajar dengan pelatihan, yang menunjukkan bahwa model berhasil belajar secara efektif dari data tanpa kehilangan kemampuan generalisasi. Kestabilan antara kedua kurva ini menegaskan bahwa konfigurasi pelatihan, termasuk pemilihan *learning rate*, *batch size*, dan arsitektur model, telah berjalan optimal.

Dari sisi fungsi kerugian (*loss*), grafik menunjukkan penurunan tajam pada awal pelatihan, dari nilai di atas 0,008 hingga stabil di bawah 0,002. Penurunan *loss* yang konsisten pada data pelatihan dan validasi mengindikasikan proses pembelajaran yang sehat dan konvergen. Selisih kecil antara *training loss* dan *validation loss* pada akhir pelatihan menguatkan kesimpulan bahwa model tidak mengalami *overfitting* yang berarti. Dengan

demikian, hasil visualisasi ini menunjukkan bahwa model *LSM_cnn* telah dilatih secara efektif dan memiliki kinerja yang sangat baik dalam mengklasifikasikan kerentanan longsor.



Gambar 9 Data hasil training CNN

5.2. Hasil SHAP

Analisis menggunakan *SHAP* berhasil mengungkap logika internal model *Convolutional Neural Network* (CNN) yang telah dilatih untuk memprediksi

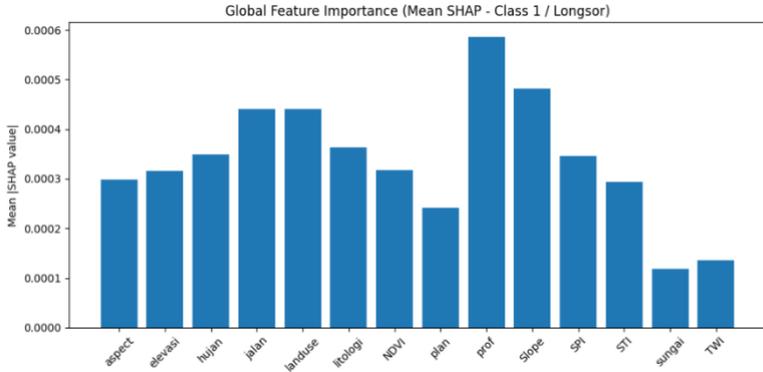
kejadian longsor. Melalui visualisasi seperti *Global Feature Importance Plot* dan *Summary Plot*, diperoleh pemahaman mendalam tentang kontribusi setiap fitur terhadap prediksi model. Secara global, fitur *curvature profile*, jarak ke jalan, *NDVI*, curah hujan, dan *slope* (kemiringan lereng) menjadi yang paling dominan mendorong klasifikasi ke arah kelas longsor. *Summary Plot* juga menunjukkan bahwa sebaran nilai SHAP untuk fitur-fitur tersebut cukup luas, menandakan peran yang signifikan dan bervariasi, tergantung nilai aktualnya. Temuan ini menegaskan bahwa karakteristik geomorfologi kompleks di area studi berpengaruh besar terhadap logika model.

Analisis nilai SHAP menunjukkan bahwa fitur dengan nilai tinggi seperti *curvature profile* dan *slope* dapat memiliki pengaruh positif maupun negatif terhadap prediksi. Misalnya, nilai *slope* rendah (warna biru) cenderung berkontribusi negatif, menunjukkan bahwa lereng landai lebih stabil. Sebaliknya, nilai tinggi dari fitur tersebut meningkatkan probabilitas klasifikasi sebagai area rawan longsor. Fitur lain seperti *landuse* dan jarak dari jalan juga menunjukkan korelasi positif terhadap kelas longsor, khususnya pada wilayah terbuka dan dekat infrastruktur. Fitur *TWI* dan jarak dari sungai, meskipun tidak terlalu dominan, tetap relevan dalam kondisi tertentu, terutama karena adanya interaksi spasial dan pengaruh lokal terhadap kerentanan.

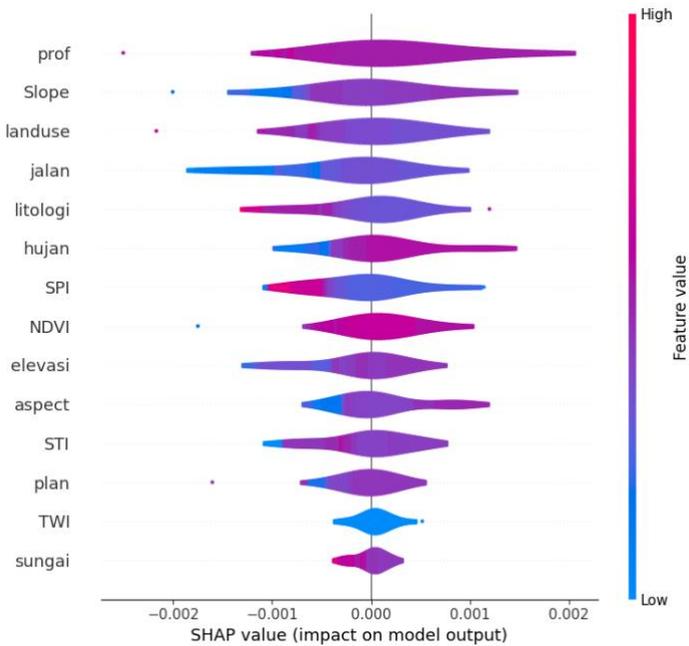
Nilai SHAP yang secara umum rendah dan terkonsentrasi di sekitar nol menandakan adanya distribusi kontribusi yang merata di antara banyak fitur. Hal ini disebabkan oleh tingginya korelasi antar fitur—misalnya antara *SPI* dan *STI* ($r = 0,90$), *SPI* dan *slope* ($r = 0,85$), serta *NDVI* dan *STI* ($r = 0,90$). Dalam prinsip SHAP,

ketika dua fitur memiliki korelasi tinggi, kontribusi mereka akan dibagi secara adil. Selain itu, karakteristik model CNN yang mengandalkan kombinasi banyak fitur secara simultan turut menyebabkan nilai SHAP menjadi tersebar. Hal ini menegaskan bahwa model tidak tergantung pada satu fitur tunggal, tetapi membentuk keputusan dari interaksi kompleks antar variabel geospasial.

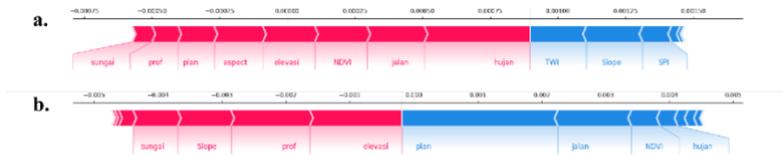
Pada tingkat individu, visualisasi *force plot* SHAP menunjukkan bagaimana fitur berkontribusi terhadap prediksi per titik data. Contohnya, pada titik yang diprediksi sebagai rawan longsor, fitur seperti sungai, jalan, *curvature profile*, *aspect*, dan elevasi memberikan kontribusi positif, sedangkan *slope*, SPI, dan TWI berkontribusi negatif. Sebaliknya, pada titik yang tidak rawan longsor, *curvature profile*, elevasi, dan jarak dari sungai memberikan kontribusi negatif dominan. Visualisasi ini memperkuat pemahaman bahwa model CNN mempertimbangkan kombinasi spasial yang kompleks dan menghasilkan prediksi yang kontekstual. Dengan demikian, SHAP tidak hanya menjelaskan hasil pelatihan CNN secara global, tetapi juga memperlihatkan dinamika lokal yang memperkuat kepercayaan terhadap validitas prediksi model dalam konteks pemetaan kerentanan bencana longsor.



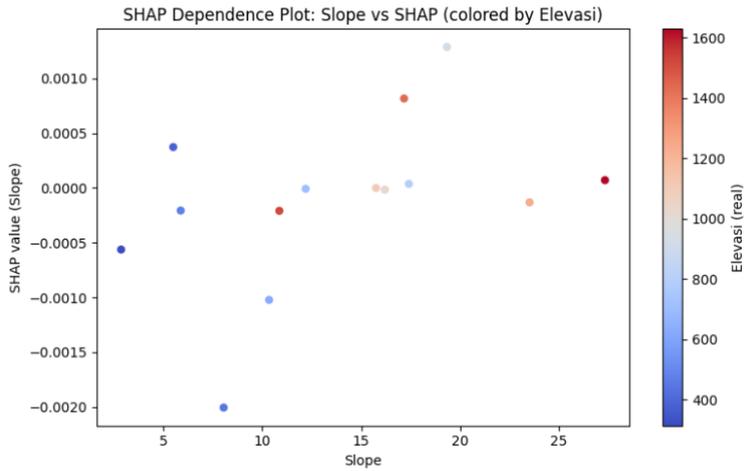
Gambar 10 Global Feature Important Plot



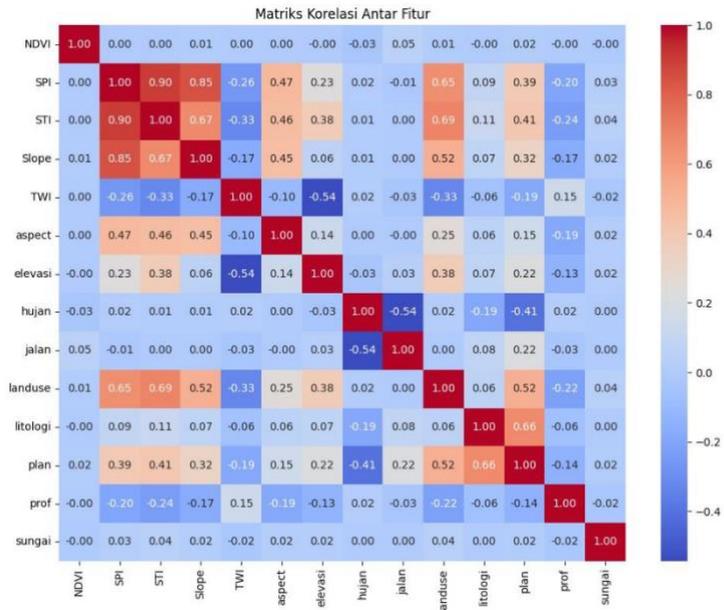
Gambar 11 Summary Plot terhadap Longsor



Gambar 12 Force Plot SHAP individu untuk a. Tanah Longsor dan b. Non-Longsor

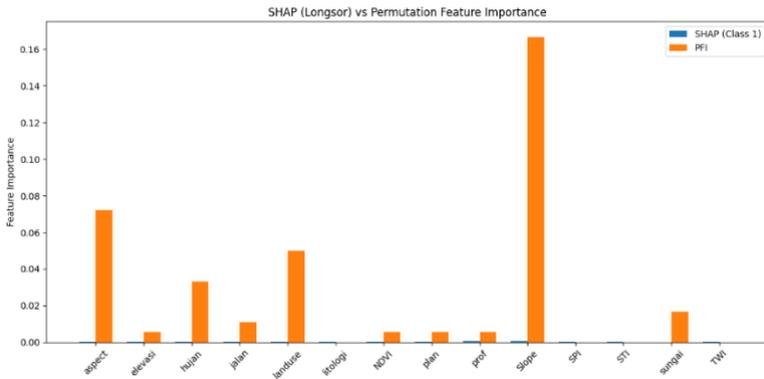


Gambar 13 Dependence Plot Elevasi dengan Slope



Gambar 14 Matriks Uji Korelasi antar Fitur

5.3. Perbandingan SHAP dengan PFI



Gambar 15 Diagram Perbandingan SHAP dengan PF

Interpretasi fitur dalam model *machine learning* menjadi aspek penting untuk menjamin transparansi, validasi ilmiah, dan pengambilan keputusan yang akurat. Dua metode populer dalam menjelaskan pentingnya fitur adalah *SHAP* dan *Permutation Feature Importance (PFI)*. PFI menilai pengaruh fitur dengan menghitung penurunan akurasi saat fitur tersebut diacak, memberikan gambaran pentingnya fitur secara global. Dalam hasil PFI, fitur *slope* menunjukkan nilai paling tinggi, seolah menjadi satu-satunya variabel dominan dalam klasifikasi. Namun, metode ini tidak menjelaskan arah pengaruh fitur terhadap prediksi (positif atau negatif), dan mengabaikan variasi spasial yang krusial dalam konteks geospasial seperti pemetaan longsor. Satu fitur yang sama, seperti *slope*, dapat berkontribusi berbeda tergantung pada kondisi lokalnya.

Sebaliknya, SHAP menghitung kontribusi setiap fitur terhadap prediksi individual dan menyajikan arah

pengaruh, baik positif maupun negatif. Ini menjadikan SHAP lebih unggul dalam menjelaskan mengapa suatu lokasi diklasifikasikan sebagai rawan atau tidak rawan longsor. Dengan mempertimbangkan interaksi antar fitur dan variasi lokal, SHAP memberikan pemahaman yang jauh lebih rinci dan relevan dibandingkan PFI. Studi oleh Flora et al. (2022) mendukung hal ini, dengan menyatakan bahwa SHAP secara konsisten lebih akurat dalam menggambarkan kontribusi aktual setiap fitur, terutama pada data dengan korelasi tinggi antar variabel. Oleh karena itu, dalam aplikasi geospasial yang kompleks, SHAP menjadi pendekatan yang lebih tepat dan informatif.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Praktik ini berhasil menunjukkan bahwa data geospasial dalam format raster (.tif) dapat diolah secara efektif menggunakan pustaka *GDAL*, sehingga data tersebut dapat dimasukkan ke dalam pipeline *machine learning* berbasis *Convolutional Neural Network* (CNN). Proses validasi dan transformasi data yang mencakup normalisasi, padding, serta pengkodean label berjalan lancar dan menghasilkan dataset yang kompatibel dengan CNN. Hal ini membuktikan bahwa pendekatan pemrosesan citra geospasial untuk keperluan klasifikasi spasial berskala besar dapat diimplementasikan secara efisien dengan Python.

Model *LSM_cnn* yang dirancang menggunakan arsitektur CNN, SE block, dan Spatial Pyramid Pooling (SPP) menunjukkan performa pelatihan yang sangat baik. Nilai *Area Under Curve* (AUC) mencapai 97,94% untuk data pelatihan dan 97,31% untuk validasi. Evaluasi ini menegaskan bahwa arsitektur model dan konfigurasi pelatihan yang digunakan telah berhasil dalam mengklasifikasikan wilayah rawan longsor secara efektif. Penggunaan SHAP juga telah mencerminkan sifat model yang menggunakan semua fitur untuk memprediksi dengan fokus khusus ke hujan dan elevasi.

6.2. Saran

Ke depan, integrasi pustaka geospasial seperti GDAL dengan model CNN sebaiknya dikembangkan lebih lanjut untuk mendukung pemrosesan berbagai tipe data spasial dalam skala nasional. Disarankan pula agar proses validasi dan *data pipeline* dibuat lebih modular dan terdokumentasi, sehingga dapat digunakan ulang dalam studi sejenis dengan data geospasial berbeda. Pendekatan ini berpotensi besar untuk diaplikasikan dalam domain mitigasi bencana lainnya seperti banjir dan kekeringan.

Selain itu, pemanfaatan *Explainable AI* (XAI) khususnya SHAP terbukti memberikan nilai tambah signifikan dalam menjelaskan hasil pelatihan model. Visualisasi SHAP membantu mengungkap kontribusi masing-masing fitur geospasial terhadap prediksi model secara transparan dan detail. Disarankan agar ke depan, analisis SHAP digunakan secara rutin dalam proyek berbasis *machine learning* spasial untuk memperkuat kepercayaan publik dan pengambil kebijakan terhadap sistem prediktif berbasis AI.

DAFTAR PUSTAKA

Irawan, L. Y., Panoto, D., Hartono, R., Pradana, I. H., & Darmansyah, A. C. (2022).

Landslide Susceptibility Assessment in Mojokerto Regency Using Logistic Regression. *KnE Social Sciences*, 7(16 SE-Articles).

<https://doi.org/10.18502/kss.v7i16.12170>

Irawan, L. Y., Sumarmi, Bachri, S., Panoto, D., Nabila, Pradana, I. H., Faizal, R.,

Devy, M. M. R., & Prasetyo, W. E. (2021). The Use of Machine Learning for

Assessing Landslide Susceptibility Class: Study Case of Kecamatan Pacet,

Kabupaten Mojokerto. *IOP Conference Series: Earth and Environmental*

Science, 884(1), 12006. <https://doi.org/10.1088/1755-1315/884/1/012006>

Ado, M., Amitab, K., Maji, A. K., Jasińska, E., Gono, R., Leonowicz, Z., & Jasiński,

M. (2022). Landslide Susceptibility Mapping Using Machine Learning: A

Literature Survey. *Remote Sensing*, 14(13), 3029.

<https://doi.org/10.3390/rs14133029>

Melati, D., Umbara, R., Astiasari, A., Wisyanto, W., Trisnafiah, S., Trinugroho,

T., Prawiradisatra, F., Arifianti, Y., Ramdhani, T. I., Arifin, S., & Anggreainy,

M. (2024). A comparative evaluation of landslide susceptibility mapping using

85

machine learning-based methods in Bogor area of Indonesia.

Environmental

Earth Sciences, 83. <https://doi.org/10.1007/s12665-023-11402-3>

Pradhan, B., Dikshit, A., Lee, S., & Kim, H. (2023a). An explainable AI (XAI) model for landslide susceptibility modeling. *Applied Soft Computing*, 142, 110324.
<https://doi.org/https://doi.org/10.1016/j.asoc.2023.110324>

Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E. (2019). Squeeze-and-Excitation Networks. arXiv preprint.
<https://www.robots.ox.ac.uk/~vgg/publications/2019/Hu19a/hu19a.pdf>

He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. arXiv:1406.4729. <https://arxiv.org/abs/1406.4729>

Christl, A. (2014). OSGeo: Professionally Leveraging Open Source. FIG Congress Proceedings.
http://www.fig.net/resources/proceedings/fig_proceedings/fig2014/papers/TS08H/TS08H_christl_7371.pdf

Lundberg, S. M., & Lee, S.-I. (2017). SHAP (SHapley Additive exPlanations): A game theoretic approach to explain the output of any machine learning model. GitHub repository.
<https://github.com/slundberg/shap>

Flora, M., Potvin, C., McGovern, A., & Handler, S. (2022). Comparing Explanation Methods for Traditional Machine Learning Models Part 2: Quantifying Model Explainability Faithfulness and Improvements with Dimensionality Reduction.
<http://arxiv.org/abs/2211.10378>

LAMPIRAN

Seluruh dataset dan script dapat diakses pada link berikut:

https://its.id/m/Dataset_LSM_CNN_SHAP

BIODATA PENULIS

Nama : Rano Noumi Sulistyو
Tempat, Tanggal Lahir : Surabaya, 10 November 2004
Jenis Kelamin : Laki-Laki
Telepon : +6285931148833
Email : noumirano@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2021
Semester : 8 (Delapan)