

# INTEGRASI TEKNOLOGI PENGINDERAAN JAUH DAN *MACHINE LEARNING* PADA *WEB GIS* UNTUK PEMETAAN POTENSI BANJIR

Putu Ony Andewi<sup>1)</sup>, Ketut Agus Seputra<sup>2)</sup>, Kadek Yota Ernanda Aryanto<sup>3)</sup>, Luh Joni Erawati Dewi<sup>4)</sup>

<sup>1,2,3,4</sup> Fakultas Teknik dan Kejuruan, Universitas Pendidikan Ganesha  
Email: putuonyandewi@gmail.com, agus.seputra@undiksha.ac.id, yota.ernanda@undiksha.ac.id, joni.erawati@undiksha.ac.id

## ABSTRAK

Banjir merupakan fenomena alam yang sering kali menjadi permasalahan serius di berbagai wilayah di Indonesia dan berkaitan dengan ulah manusia yang diakibatkan oleh beberapa faktor yaitu, hujan, kondisi sungai, kondisi daerah hulu, kondisi daerah budidaya, dan pasang surut air laut. Terjadinya banjir dapat memicu berbagai masalah kesehatan, termasuk penyebaran penyakit air, kehilangan sumber daya alam, hancurnya infrastruktur, dan hilangnya mata pencaharian. Salah satu bentuk upaya yang dapat dilakukan yaitu dengan melakukan pemetaan potensi wilayah risiko banjir. Pemetaan ini dilakukan dengan menggunakan data dari citra satelit Landsat 8, level 2, collection 2, tier 1 yang diolah menggunakan platform *Google Earth Engine* (GEE) untuk menghitung indeks seperti *Digital Elevation Model* (DEM), *Topographic Position Index* (TPI), *Normalized Difference Vegetation Index* (NDVI), dan *Normalized Difference Water Index* (NDWI). Indeks ini digunakan sebagai variabel input dalam model *random forest* untuk mengklasifikasikan wilayah berdasarkan tingkat risiko banjir: tinggi, sedang, dan rendah. Model yang dikembangkan kemudian diintegrasikan ke dalam layanan *WebGIS* menggunakan *framework* Flask. Pengujian menunjukkan bahwa model *random forest* efektif dalam mengklasifikasikan data dengan akurasi sebesar 86% berdasarkan pengujian *confusion matrix* dengan mencari nilai akurasi, presisi, recall, dan *f1-score*. Rancangan integrasi model pada layanan *web services* telah terimplementasi menjadi sebuah API yang memungkinkan aplikasi pihak ketiga menampilkan data potensi risiko banjir secara *real-time*. Implementasi *front-end* dibuat menggunakan *leafletJS* yang menunjukkan visualisasi peta risiko banjir yang interaktif dan *user-friendly*. Hasil pengujian *System Integration Testing* menunjukkan bahwa seluruh fitur yang dikembangkan telah berfungsi dengan baik.

**Kata kunci:** Citra satelit, *Google Earth Engine* (GEE), *Random Forest*, *Confusion matrix*, *WebGIS*

## ABSTRACT

*Flooding is a natural phenomenon that has frequently posed significant challenges in various regions of Indonesia, driven by factors such as rainfall, river conditions, upstream landscapes, land use patterns, and sea-level rise. These events often lead to severe consequences, including the spread of waterborne diseases, destruction of infrastructure, depletion of natural resources, and economic disruption. One proactive measure to mitigate such impacts is mapping potential flood risk areas. This study utilized Landsat 8 satellite imagery Level 2, Collection 2, Tier 1 processed on the Google Earth Engine (GEE) platform to derive indices such as the Digital Elevation Model (DEM), Topographic Position Index (TPI), Normalized Difference Vegetation Index (NDVI), and Normalized Difference Water Index (NDWI). These indices served as input variables for a Random Forest model, classifying areas into high, medium, and low flood risk categories. The developed model achieved 86% accuracy when evaluated using a confusion matrix, with precision, recall, and F1-score metrics validating its performance. The integration of this model into a WebGIS service was implemented through Flask, offering an API that supports real-time flood risk data retrieval by third-party applications. The front-end interface, built using LeafletJS, provides an interactive and user-friendly map visualization of flood risk levels. The results demonstrate that the Random Forest model effectively classifies flood risk, while the WebGIS service offers a practical tool for visualizing and disseminating flood risk information. This service has the potential to support disaster management efforts and enhance community preparedness against flooding.*

**Keywords** : Citra Satellite, Google Earth Engine (GEE), Random forest, Confusion matrix, WebGIS

## 1. PENDAHULUAN

Bencana alam merupakan bencana yang terjadi akibat terganggunya keseimbangan komponen-komponen alam tanpa campur tangan manusia [1]. Banjir merupakan fenomena alam yang sering kali menjadi permasalahan serius di berbagai wilayah di Indonesia dan juga disebabkan oleh beberapa aktivitas manusia, seperti pengelolaan daerah hulu yang buruk, alih fungsi lahan budidaya, serta faktor alam seperti hujan, kondisi sungai, dan pasang surut air laut. Banjir adalah peristiwa berlimpahnya air yang meluap hingga ke daratan yang biasanya kering, akibat curah hujan yang tinggi, lelehan salju, atau masalah lain yang mengakibatkan air tak dapat diserap dengan cepat oleh tanah [2]. Bencana banjir tidak hanya merugikan secara materi, tetapi juga mengancam kehidupan manusia dan ekosistem di sekitarnya. Terjadinya banjir dapat memicu berbagai masalah kesehatan, termasuk penyebaran penyakit air, kehilangan sumber daya alam, hancurnya infrastruktur, dan hilangnya mata pencaharian. Oleh karena itu, diperlukan pemahaman mendalam terhadap penyebab banjir, sistem peringatan dini, dan upaya mitigasi menjadi sangat penting untuk mengurangi dampak negatifnya. Salah satu bentuk upaya yang dapat dilakukan yaitu dengan melakukan pemetaan potensi wilayah risiko banjir.

Pemetaan potensi wilayah risiko banjir merupakan bagian dari upaya mitigasi, perencanaan, dan respons terhadap bencana ini. Pemetaan wilayah ini dimanfaatkan untuk mengidentifikasi dan menganalisis area rawan banjir serta merencanakan tindakan pencegahan [3]. Dengan dilakukannya pemetaan potensi wilayah risiko banjir, langkah-langkah mitigasi perencanaan dapat dirancang lebih efektif dan responsif terhadap bencana dapat dilakukan secara tepat waktu. Pengurangan risiko banjir dapat dilakukan dengan memanfaatkan teknologi spasial berupa *Web Geographic Information System* (GIS) yang berfungsi sebagai sebuah sistem yang menerima input data spasial dari berbagai sumber, melakukan analisis terhadap data tersebut dan menghasilkan output berupa representasi visual dari informasi geografis yang telah diproses [4]. Dalam proses analisis, melibatkan data citra satelit yang diperoleh dan diolah pada *platform* yang disediakan oleh google yaitu *Google Earth Engine* (GEE) yang kemudian diintegrasikan menggunakan *python* API.

GEE adalah sebuah *platform* berbasis cloud yang memungkinkan pemrosesan citra satelit berskala besar untuk mendeteksi perubahan, memetakan tren, dan mengukur perbedaan di permukaan bumi [5]. Keunggulan dalam akses dan analisis data global menjadikan GEE alat yang sangat potensial untuk pemetaan potensi wilayah risiko banjir di Indonesia, khususnya di Provinsi Bali. Provinsi Bali terletak di antara 8°3'38" - 8°50'56" Lintang Selatan dan 114°25'53" - 115°42'39" Bujur Timur dengan luas wilayah sekitar 559.468 ha, membentang sepanjang 153 km dan selebar 112 km [6]. Pulau Bali seringkali dilanda hujan dengan intensitas curah hujan tinggi yang menyebabkan banjir di sejumlah daerah pesisir, terutama di daerah perkotaan yang padat penduduk. Berdasarkan pernyataan Pusdalops (2024), curah hujan yang tinggi dengan cuaca ekstrem (hujan disertai angin kencang) terjadi di seluruh kabupaten di Bali. Sehingga mengakibatkan sungai-sungai meluap dan air hujan tidak dapat dengan cepat disalurkan ke laut karena drainase yang kurang optimal. Banjir tertinggi terjadi pada periode 1 Januari hingga 25 Februari 2023 dengan cuaca ekstrem (20 kejadian di Kabupaten Klungkung, Tabanan, Gianyar) dan banjir 6 kejadian di Kabupaten Jembrana [7]. Pada periode akhir 1 s.d 31 Desember 2023, kejadian bencana hidrometeorologi basah juga meningkat, cuaca ekstrem (hujan disertai angin kencang) sebesar 60.16%, terjadi banjir di beberapa kabupaten yaitu Kab. Tabanan, Kab. Badung, dan Kota Denpasar [8]. Banjir bisa menjadi gangguan serius bagi masyarakat, terutama dalam hal mobilitas dan keamanan. Karena banyak jalan yang tergenang air, transportasi menjadi terhambat, dan rumah serta bisnis terendam yang mengakibatkan kerugian materiil yang signifikan.

Hal tersebut menunjukkan bahwa penting untuk mengambil tindakan yang tepat dalam menjaga keberlanjutan lingkungan dan mengelola potensi wilayah risiko banjir di Provinsi Bali. Dengan upaya bersama dari pemerintah, masyarakat, dan pihak terkait lainnya untuk menciptakan lingkungan yang aman, lestari, dan meminimalkan risiko bencana banjir. Kesadaran akan tantangan yang dihadapi, peneliti mengangkat topik pengembangan layanan *web services* untuk pemetaan potensi wilayah risiko banjir secara lebih rinci. Layanan ini dapat digunakan oleh pihak ketiga untuk diintegrasikan ke dalam *WebGIS* yang dapat memberikan kejelasan dan relevansi informasi berupa visualisasi peta spasial yang informatif dan mudah dipahami serta membantu pengguna dalam pengambilan keputusan terkait mitigasi bencana.

## 2. METODE

Sistem dimulai dari mendefinisikan *Area Of Interest* (AOI) yang akan dipetakan untuk wilayah risiko banjir, untuk AOI disini penulis memilih Provinsi Bali. Kemudian dilakukan pemrosesan data di GEE dengan citra satelit dari Landsat 8 Level 2, Collection 2, Tier 1 untuk memetakan potensi wilayah risiko banjir berdasarkan perhitungan indeks seperti *distance score* untuk mencari nilai jarak bahaya (*distance hazard score*) dari sumber air permanen, *elevasi score* untuk mencari nilai *Digital Elevation Model* (DEM); *topo score* untuk mencari nilai *Topographic Position Index* (TPI); *veg score* untuk mencari nilai *Normalized Difference Vegetation Index* (NDVI) (1); dan *wet score* untuk mencari nilai *Normalized Difference Water Index* (NDWI) (2) [9].

$$NDVI = \frac{(NIR-RED)}{(NIR+RED)} \quad (1) \quad [10]$$

$$NDWI = \frac{(GREEN-NIR)}{(GREEN+NIR)} \quad (2) \quad [11]$$

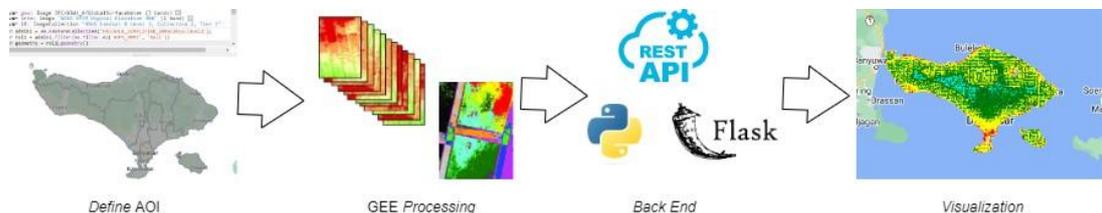
Keterangan:

NIR: *Near-Infrared* (band 5)

RED: Merah (band 4)

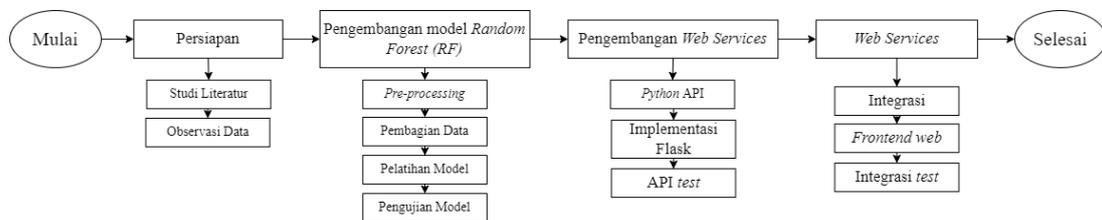
GREEN: Hijau (band 3)

Setelah itu, pemrosesan pada GEE tersebut diintegrasikan ke dalam *python* untuk melihat pengelolaan data dan analisis data lebih lanjut. Hasil analisis ini kemudian diintegrasikan dengan *framework* Flask, yang digunakan untuk membangun layanan *web*. Flask berperan dalam pengintegrasian antara model menjadi layanan REST API yang dapat diakses secara eksternal. REST API ini kemudian diintegrasikan ke dalam aplikasi web untuk memvisualisasikan informasi mengenai potensi wilayah risiko banjir. Dengan demikian, pengguna dapat dengan mudah mengakses dan mengeksplorasi informasi melalui interface web yang ramah pengguna, dapat dilihat pada Gambar 1.



Gambar 1. Gambaran umum sistem

Mekanisme atau tahapan pelaksanaan dalam penelitian ini dimulai dari persiapan, pengembangan model *random forest*, pengembangan *web services*, dan *web services*. Tahapan penelitian dapat dilihat pada Gambar 2.



Gambar 2. Tahapan penelitian

### A. Persiapan

#### 1. Studi Literatur

Studi literatur dalam penelitian ini mencakup berbagai sumber dari internet, termasuk jurnal ilmiah, dokumentasi resmi, dan tutorial. Literatur yang dikaji meliputi pemahaman tentang proses hidrologi, faktor risiko banjir, serta teknik pemetaan menggunakan citra satelit. Penelitian terkait penggunaan *Google Earth Engine* (GEE) dan *Python API* untuk analisis citra satelit juga menjadi bagian dari studi ini, dengan informasi yang diperoleh dari jurnal, *YouTube*, dan dokumentasi resmi GEE.

#### 2. Observasi Data

Observasi data dilakukan secara tidak langsung melalui analisis dan eksplorasi citra satelit seperti MODIS, Sentinel, dan Landsat. Proses ini bertujuan untuk meningkatkan pemahaman dan keterampilan dalam interpretasi data satelit, termasuk eksplorasi fitur-fitur seperti distribusi vegetasi dan bentangan sungai.

## B. Pengembangan Model *Random Forest*

### 1. *Pre-processing*

Tahap *pre-processing* data melibatkan pengumpulan citra satelit Landsat 8 dari GEE untuk wilayah Bali, diikuti dengan pembersihan data dari *noise*, penanganan *missing values*, normalisasi, serta pemilihan fitur yang relevan. Proses ini mencakup *filtering*, penyesuaian radiometrik, dan koreksi atmosfer untuk meningkatkan kualitas data. Data kemudian diklasifikasikan ke dalam tiga kategori risiko banjir (tinggi, sedang, rendah), diikuti dengan langkah-langkah *data cleaning*, penanganan *outlier*, dan standarisasi menggunakan *Python*.

### 2. Pembagian Data

Pembagian data dilakukan dengan proses pemisahan data menjadi dua subset yang berbeda, yaitu data latih (*training data*) dan data uji (*testing data*).

### 3. Pelatihan Model

Dalam pelatihan model, data latih (*training data*) digunakan untuk melatih model yang melibatkan pembuatan serangkaian pohon keputusan berdasarkan subset data latih. Di mana setiap pohon keputusan dalam ensemble belajar dari subset acak dari data latih dan fitur-fiturnya [12].

### 4. Pengujian Model

Pengujian model dilakukan menggunakan data uji untuk mengevaluasi kinerjanya dalam menggeneralisasi data baru. Validasi model menggunakan *confusion matrix* [13], yang membandingkan hasil prediksi dengan label sebenarnya. Matriks ini menghasilkan metrik akurasi (3), recall (4), presisi (5), dan *f1-score* (6) untuk mengukur performa klasifikasi model pada tiga kelas yang berbeda [14].

Nilai *accuracy* (akurasi), mengukur seberapa akurat model dalam melakukan klasifikasi [15].

$$Akurasi = \frac{(True\ Positive + True\ Negative)}{(True\ Positive + False\ Positive + False\ Negative + True\ Negative)} \quad (3)$$

Nilai recall, mengukur keberhasilan model dalam menemukan kembali informasi tertentu [15].

$$Recall = \frac{(True\ Positive)}{(True\ Positive + False\ Negative)} \quad (4)$$

Nilai presisi, mengukur akurasi antara data yang diminta dengan hasil prediksi yang dihasilkan oleh model [15].

$$Presisi = \frac{(True\ Positive)}{(True\ Positive + False\ Positive)} \quad (5)$$

Nilai *f1-score*, mengukur kebaikan suatu model dengan memperhitungkan *precision* (presisi) dan recall [15].

$$f1 - score = 2 \cdot \frac{precision \times recall}{precision + recall} \quad (6)$$

## C. Pengembangan *Web Services*

### 1. *Python API*

*Python API* berfungsi sebagai perantara yang mengintegrasikan model dengan layanan web, memungkinkan aplikasi web untuk berkomunikasi dengan model. Dengan *Python API*, pengguna dapat mengirim data input ke model, menerima hasil prediksi, dan menjalankan berbagai fungsi lainnya dalam sistem yang dikembangkan,

### 2. Implementasi Flask

Implementasi menggunakan *framework* Flask membangun *back-end* layanan web dengan menangani routing HTTP, pemrosesan data, dan menyediakan *interface API*. Flask mengintegrasikan *python API* untuk menjalankan pemrosesan data dan memanggil model *random forest*, kemudian mengirimkan respons hasil prediksi kembali ke pengguna.

### 3. *API Test*

*API test* ini dilakukan menggunakan *platform* Postman untuk menguji inputan parameter sesuai dengan yang diinginkan. Pengujian dapat dilakukan dengan mengirimkan permintaan POST ke API dan memverifikasi bahwa data yang diterima dan diproses oleh server sesuai dengan spesifikasi yang telah ditentukan.

#### D. Implementasi *Web Services*

##### 1. Integrasi

Proses integrasi *web services* dimulai dengan mengidentifikasi layanan yang akan digunakan, kemudian dikembangkan menggunakan API untuk menghubungkan layanan tersebut ke dalam aplikasi yang dibangun.

##### 2. *Front-end Web* pada Interaksi Peta

Pengembangan *front-end web* difokuskan pada pembuatan antarmuka yang interaktif dan responsif untuk kenyamanan pengguna. HTML digunakan sebagai struktur dasar, CSS untuk tata letak dan desain visual, sementara *JavaScript* menambahkan interaktivitas dan fungsionalitas dinamis pada halaman *web*.

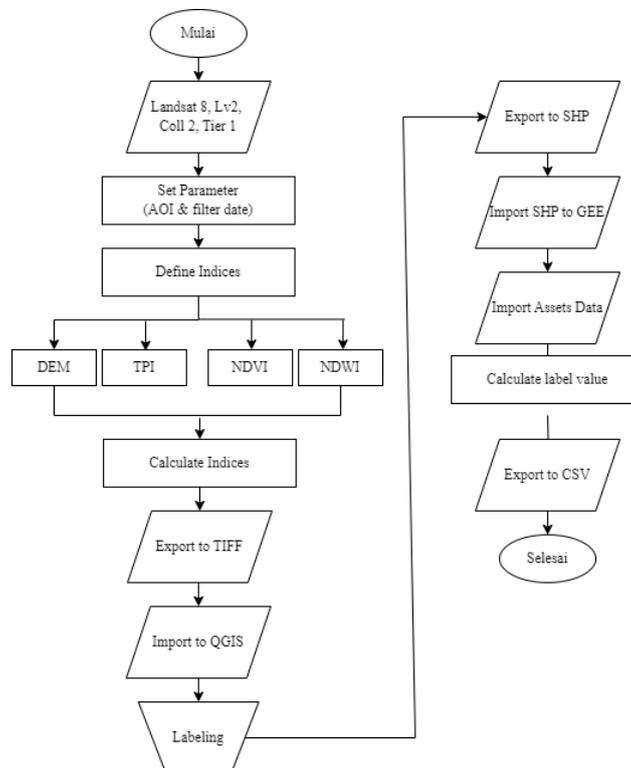
##### 3. *Integration Test*

Pengujian integrasi dilakukan untuk memastikan API berfungsi dengan baik dalam berkomunikasi dengan *front-end web*. Berbagai skenario penggunaan diuji untuk menjamin keandalan dan akurasi informasi yang diberikan, sehingga layanan *web services* dapat beroperasi optimal dalam menyajikan informasi risiko banjir.

### 3. HASIL DAN PEMBAHASAN

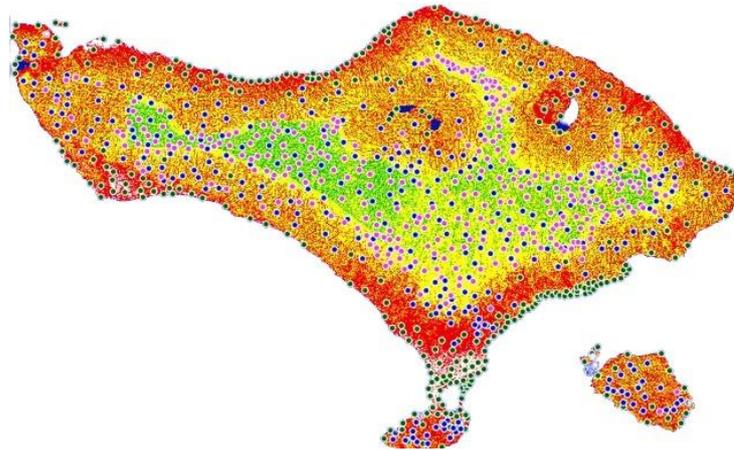
#### 1. Pengembangan Model

Pengumpulan data dilakukan melalui *platform Google Earth Engine (GEE)* yang kemudian di ekstrak untuk dilakukan pelabelan dengan 3 kelas, yaitu 1 berarti tinggi, 2 berarti sedang, dan 3 berarti rendah. Adapun detail dalam melakukan pengumpulan data ini dapat dilihat pada Gambar 3.



Gambar 3. *Flowchart* Akuisisi Data dengan *Google Earth Engine (GEE)*

Dalam melakukan pelabelan data, data yang dilabelkan menghasilkan sebanyak 900 data dengan 300 data setiap kelas yang akan menjalani tahap *pre-processing*. Detail sampel data dapat dilihat pada Gambar 4.



Gambar 4. Ilustrasi titik sampel pelabelan

Titik label berwarna hijau (tinggi), biru (sedang), dan ungu (rendah) merepresentasikan koordinat yang diekspor untuk dijadikan file dengan format CSV dengan mengekstrak nilai-nilai dari setiap indeks tersebut yaitu, *distance score* untuk mencari nilai jarak bahaya dari sumber air permanen, *elevasi score* untuk mencari nilai *Digital Elevation Model* (DEM); *topo score* untuk mencari nilai *Topographic Position Index* (TPI); *veg score* untuk mencari nilai *Normalized Difference Vegetation Index* (NDVI); dan *wet score* untuk mencari nilai *Normalized Difference Water Index* (NDWI) [16]. Sehingga menghasilkan nilai-nilai seperti pada Tabel 1.

Tabel 1. Data hasil ekspor *Google Earth Engine*

Label	DEM	TPI	NDVI	NDWI
1	0	-18.125	-0.3050561521	0.3758395605
1	5	-13.58730159	0.1063946655	-0.222584479
1	90	-11.03389831	0.4686777015	-0.4726990322
1	5	-3.323076923	0.3773372433	-0.4105986629
1	5	-3.407407407	0.2469831883	-0.4575007171
...	...	...	...	...
3	647	-6.407407407	0.8273615515	-0.7560276275
3	668	10.35802469	0.8610289673	-0.7782565255
3	430	-6.358024691	0.8848301559	-0.8085755879
3	444	-6.185185185	0.9085679704	-0.8363665588
3	477	-7.209876543	0.8728494852	-0.7767338241

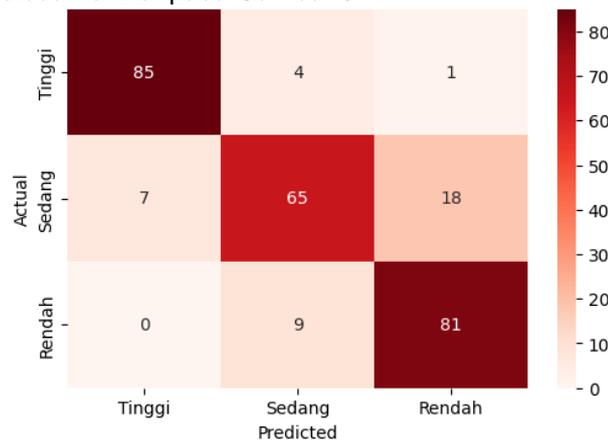
Dalam pembuatan model *random forest*, langkah awal melibatkan pengimporan pustaka yang diperlukan, khususnya *scikit-learn*. Setelah itu, tahap penting adalah data *pre-processing*, yang mencakup penanganan nilai yang hilang dan standarisasi data. Penanganan nilai yang hilang memastikan tidak ada informasi penting yang terabaikan, sementara standarisasi menjamin bahwa semua fitur memiliki skala yang sama, sehingga model dapat berfungsi dengan baik. Kemudian data dibagi menjadi data latih (*training data*) dan data uji (*testing data*) dengan perbandingan 70 : 30. Artinya 70% untuk *training data* dan 30% untuk *testing data*. Dalam hal ini, dataset dibagi menjadi  $x_{train}$ ,  $x_{test}$ ,  $y_{train}$ ,  $y_{test}$  dengan melakukan *stratify* untuk memastikan bahwa pembagian data mempertahankan distribusi kelas yang sama seperti dataset asli [17]. Pemodelan dilakukan dengan menerapkan parameter dalam *random forest* seperti pada Tabel 2.

Tabel 2. Parameter *random forest*

Parameter	Nilai
<i>n_estimator</i>	700
<i>criterion</i>	<i>entropy</i>
<i>max_depth</i>	60
<i>max_features</i>	<i>sqrt</i>
<i>min_samples_split</i>	5
<i>min_samples_leaf</i>	2

Berdasarkan eksperimen, jumlah pohon optimal ( $n\_estimators$ ) dalam model *random forest* adalah 700, yang menunjukkan performa terbaik pada jumlah pohon tersebut. Kriteria *entropy* dipilih sebagai yang terbaik dibandingkan *log\_loss* dan *gini*, karena nilai *entropy* yang lebih rendah mengindikasikan pemisahan *node* yang lebih baik. Kedalaman pohon maksimum (*max\_depth*) yang optimal adalah 60, sedangkan nilai maksimum fitur (*max\_features*) yang memberikan hasil terbaik adalah *sqrt*, menunjukkan pemanfaatan jumlah fitur yang setara dengan akar kuadrat dari total fitur pada setiap split. Parameter *min\_samples\_split* yang optimal adalah 5, dan *min\_samples\_leaf* yang terbaik adalah 2, menandakan pembagian *node* hanya dilakukan jika memiliki lebih dari 5 sampel dan setiap *leaf node* harus memiliki setidaknya 2 sampel. Dengan akurasi terbaik sebesar 86%, model menunjukkan kinerja yang baik dalam klasifikasi data, mengindikasikan tidak adanya *overfitting* signifikan dan kemampuan model untuk menangkap pola serta menggeneralisasi data baru secara efektif [18].

Berdasarkan model *random forest* yang sudah dibuat, diperoleh hasil *confusion matrix* yang menggambarkan hasil dari prediksi kelas yang telah dilakukan oleh model. Berikut merupakan hasil dari *confusion matrix* yang telah dibuat terlihat pada Gambar 5.



Gambar 5. *Confusion matrix*

Pada Gambar 5, terlihat bahwa kelas “Tinggi” memiliki 85 prediksi data yang benar, 4 prediksi yang salah ke kelas “Sedang”, dan 1 prediksi yang salah ke kelas “Rendah”. Kelas “Sedang” memiliki 65 prediksi yang benar, tetapi terdapat 7 prediksi yang salah ke kelas “Tinggi” dan 18 prediksi salah ke kelas “Rendah”. Kelas “Rendah” memiliki 81 prediksi yang benar, namun terdapat 9 prediksi yang salah ke kelas “Sedang” tanpa adanya prediksi yang salah ke kelas “Tinggi”.

*Classification report* dari *confusion matrix* berdasarkan model *random forest*, adapun informasi lebih rinci terkait perhitungan presisi, recall, dan *f1-score* yaitu terlihat pada Tabel 3.

Tabel 3. *Classification report*

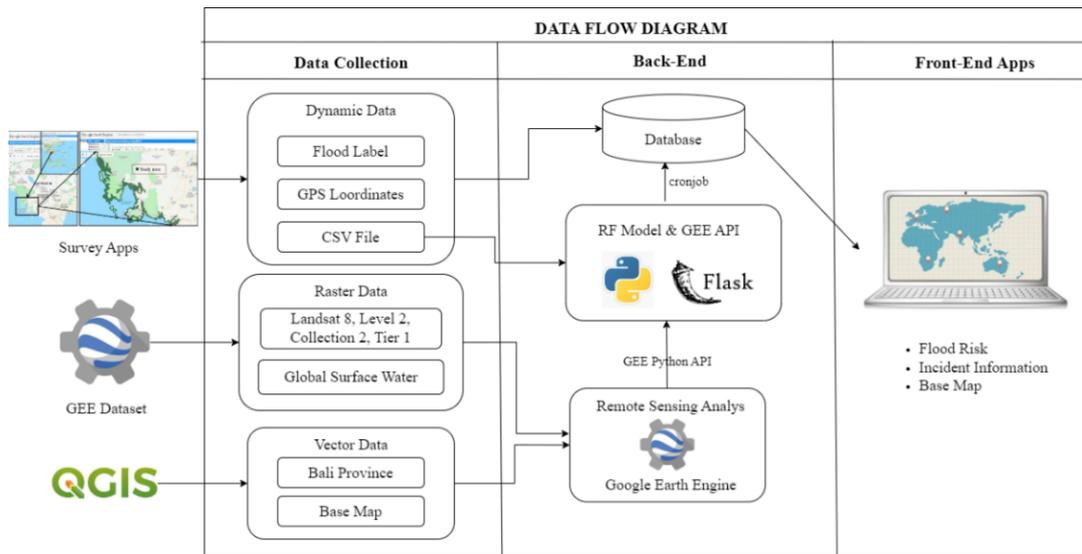
Kelas	Presisi	Recall	F1-score
<b>Tinggi</b>	0.92	0.94	0.93
<b>Sedang</b>	0.83	0.82	0.77
<b>Rendah</b>	0.81	0.90	0.85

Kelas “Tinggi” memiliki presisi dan recall yang tinggi, masing-masing sebesar 0.92 dan 0.94. *F1-score* juga memiliki tingkat yang baik, yaitu 0.93 yang menunjukkan keseimbangan yang baik antara presisi dan recall. Pada kelas “Sedang” memiliki presisi sebesar 0.83, recall sebesar 0.82, dan *f1-score* sebesar 0.77 yang menunjukkan keseimbangan yang baik antara presisi dan recall. Kemudian pada kelas “Rendah” memiliki presisi sebesar 0.81, recall sebesar 0.90, dan *f1-score* sebesar 0.85 yang menunjukkan keseimbangan relatif baik antara presisi dan recall. Secara keseluruhan, model menunjukkan performa yang baik untuk semua kelas.

## 2. Pengembangan *Web Services*

Dalam integrasi model, data dinamis seperti label banjir yang mengklasifikasikan tingkat risiko banjir berdasarkan data historis dan prediksi model, dicatat dengan *GPS coordinates* untuk akurasi lokasi dan disimpan dalam file CSV untuk kemudahan akses dan transfer. Data raster menggunakan Landsat 8, Level 2, Collection 2, Tier 1 untuk menghitung indeks seperti DEM, TPI, NDVI, dan NDWI, serta memantau perubahan dan distribusi air melalui *global surface water*. Sementara itu, data vektor

digunakan untuk meningkatkan fungsionalitas *WebGIS* dengan peta dasar interaktif di Provinsi Bali. Rincian integrasi model ini dapat dilihat pada Gambar 6.



Gambar 6. Integrasi model

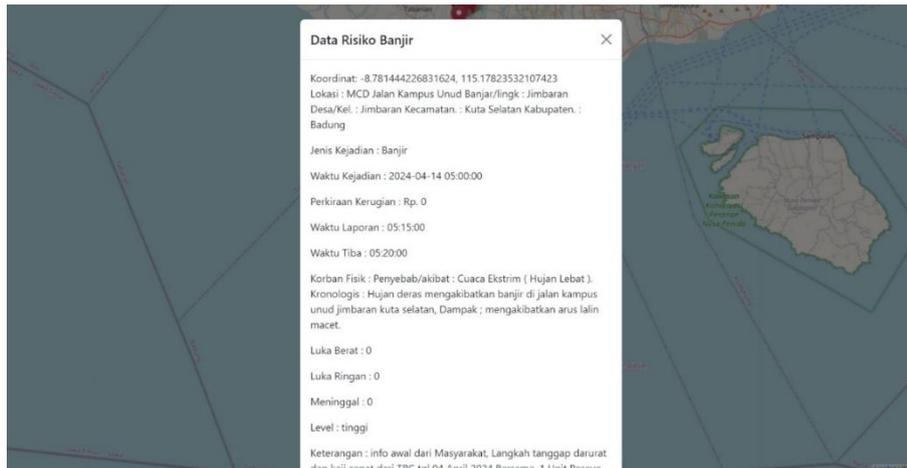
Pada bagian *back-end*, *database MySQL* digunakan untuk menyimpan informasi terkait kejadian banjir, dengan data dari Badan Penanggulangan Bencana Daerah (BPBD) Badung yang mencakup 38 catatan kejadian banjir, termasuk informasi hari, waktu, lokasi, korban jiwa, dan kerugian. Flask digunakan untuk membangun API yang berkomunikasi dengan *Google Earth Engine (GEE) API*, memungkinkan pengambilan data dari proses penginderaan jauh sebagai layanan mikro *remote sensing* [19]. Model *Random Forest* diterapkan untuk analisis dan prediksi risiko banjir berdasarkan data koordinat, dengan pemrosesan dijadwalkan secara otomatis menggunakan *cronjob* yang terhubung ke *database* untuk pembaruan level risiko banjir. Respons dari *endpoint* API disajikan dalam format JSON, yang rinciannya dapat dilihat pada Tabel 4.

Tabel 4. Metode integrasi *endpoint*

No	Method	Endpoint	Keterangan
1.	GET	<u>{base_url}/history</u>	Mengambil atau menampilkan semua data histori
2.	GET	<u>{base_url}/history/2</u>	Mengambil atau menampilkan histori sesuai id yang diinginkan
3.	POST	<u>{base_url}/history</u>	Menambahkan data histori baru
4.	PATCH	<u>{base_url}/history?id=3</u>	Mengubah nilai pada data sesuai id yang diinginkan
5.	DEL	<u>{base_url}/history?id=2</u>	Menghapus data sesuai dengan id
6.	GET	<u>{base_url}/dokumentasi/3</u>	Mengambil atau menampilkan gambar sesuai dengan id
7.	POST	<u>{base_url}/dokumentasi</u>	Menambahkan gambar pada <i>history</i>
8.	DEL	<u>{base_url}/dokumentasi?id=2</u>	Menghapus gambar berdasarkan id

### 3. Hasil Implementasi *Web Services*

Pada bagian *front-end*, aplikasi dibangun menggunakan *JavaScript* dan *LeafletJS* untuk menyajikan peta interaktif. *Front-end* ini menampilkan informasi dari *back-end*, seperti risiko banjir, informasi kejadian, dan peta dasar. Data risiko banjir divisualisasikan pada peta dengan marker yang ditambahkan melalui permintaan *AJAX* ke *endpoint* */history*. Peta menampilkan wilayah Provinsi Bali dengan titik koordinat yang telah ditentukan, di mana marker dengan warna berbeda menunjukkan tingkat risiko banjir: merah untuk risiko tinggi, oranye untuk risiko sedang, dan hijau untuk risiko rendah. Pengguna dapat melihat detail kejadian banjir dengan mengklik marker pada peta, yang memberikan informasi lengkap tentang lokasi tersebut. Tampilan peta interaktif ini dapat dilihat pada Gambar 7.



Gambar 7. Visualisasi peta pada web

Integrasi testing dalam pengembangan *WebGIS* sederhana dilakukan untuk memastikan bahwa komponen *front-end*, *back-end*, dan model berfungsi dengan baik. Pengujian ini melibatkan verifikasi bahwa semua dependensi terpasang dan server *back-end* berjalan dengan baik, serta pengujian *endpoint* menggunakan *Postman*. Model *random forest* yang telah disimpan dalam format *pickle* dipanggil dan dijalankan melalui layanan web yang dibangun dengan *Flask*. Layanan ini memungkinkan pengguna mengirim permintaan *POST* dengan koordinat lokasi melalui *Postman*. Sistem kemudian menghitung nilai indeks (*DEM*, *TPI*, *NDVI*, *NDWI*) dan mengembalikan hasil analisis dalam format *JSON*, termasuk potensi risiko banjir. Rincian integrasi antara *back-end* dan model dapat dilihat pada Tabel 5.

Tabel 5. *User scenario* integrasi antara *back-end* dengan model

Judul	Integrasi antara <i>back-end</i> dengan model ( <i>flask</i> dan model)
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. Model telah dilatih dengan data yang sesuai dan disimpan dalam format ".<i>pk</i>"</li> <li>2. <i>Flask</i> terkonfigurasi dengan benar dalam server</li> <li>3. Mengkonfigurasi <i>database</i></li> <li>4. Konsistensi dalam input data</li> </ol>
Objek uji	<i>Endpoint API, data input, respons API</i>
Tujuan	<ol style="list-style-type: none"> <li>1. Fungsional: API dapat memberikan respon berupa level banjir berdasarkan data inputan yang diterima.</li> <li>2. Non-fungsional: API memberikan respon dengan waktu respons yang cepat.</li> </ol>
Hasil	Seluruh <i>Test Case</i> telah berhasil dilaksanakan dan menunjukkan semua fitur berfungsi

*Cronjob* digunakan untuk meng-*update* level potensi risiko banjir secara otomatis, menggambarkan langkah-langkah otomatisasi yang diperlukan untuk pembaruan data secara berkala [20]. Proses ini memastikan bahwa sistem secara rutin memperbarui informasi risiko banjir dan menjalankan pembaruan sesuai dengan jadwal yang ditetapkan. Rincian lebih lanjut mengenai proses pembaruan ini dapat dilihat pada Tabel 6.

Tabel 6. *User scenario cronjob* untuk *update level*

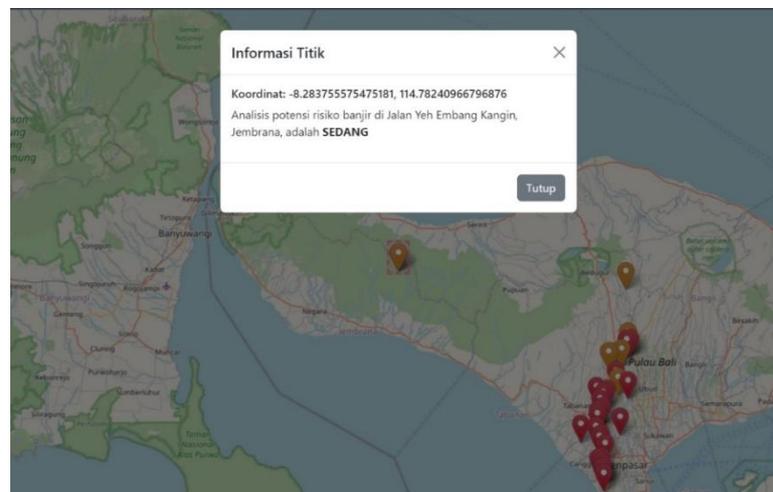
Judul	<i>Cronjob</i> untuk <i>update level</i>
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. Sistem yang mengelola data banjir dan level terpasang dan berfungsi dengan baik, termasuk <i>database</i></li> <li>2. <i>Cronjob</i> sudah terkonfigurasi dengan server</li> <li>3. Koneksi ke <i>database</i> agar dapat diakses</li> </ol>
Objek uji	Validasi <i>cronjob</i> terjadwal dan aktif, eksekusi, <i>update level banjir</i>
Tujuan	<ol style="list-style-type: none"> <li>1. Fungsional: memperbarui level banjir pada <i>database</i></li> <li>2. Non-fungsional: meminimalkan penggunaan sumber daya dan waktu proses</li> </ol>
Hasil	Sistem penjadwalan pada sisi backend berfungsi

Kemudian pengujian integrasi sebelumnya pada *endpoint back-end* menggunakan *tools postman* dan penggunaan cronjob juga dilakukan secara manual menggunakan browser dengan memeriksa peta, mengklik marker, dan memverifikasi informasi yang tampil, rincian dapat dilihat pada Tabel 7.

Tabel 7. *User scenario front-end dan back-end*

Judul	<b>Front-end dan back-end</b>
<i>Preconditions</i>	1. API sudah dikonfigurasi dengan benar 2. Pengguna dapat mengakses <i>interface web</i> yang telah terhubung dengan <i>back-end</i>
Objek uji	Komponen tampilan detail aplikasi
Tujuan	1. Fungsional: pengguna dapat mengakses dan menampilkan detail dari suatu objek 2. Non-fungsional: tampilan <i>user-friendly</i> dan mudah dinavigasi dengan informasi yang diatur dengan jelas
Hasil	Integrasi REST API pada frontend web berhasil dilakukan, dan seluruh <i>Test Case</i> telah berhasil dilaksanakan

Hasil pengembangan *endpoint* pemetaan potensi banjir diterapkan pada frontend melalui menu pin marker analisis potensi banjir dapat dilihat pada Gambar 8. Selain berfungsi untuk memetakan area rawan banjir, fitur ini juga memiliki manfaat lain, seperti mendukung analisis tata ruang wilayah dan membantu perencanaan pembangunan fasilitas publik yang bebas banjir.



Gambar 8. Fitur Analisis Potensi Banjir

#### 4. SIMPULAN DAN SARAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa implementasi model *Random Forest* berhasil dengan akurasi 86% dalam mengklasifikasikan risiko banjir, memanfaatkan fitur-fitur terkait seperti DEM, TPI, NDVI, dan NDWI. Integrasi model dalam layanan web services, yang mencakup sinkronisasi data dinamis mengenai bencana banjir dan prediksi risiko banjir, telah sukses diterapkan melalui API yang memungkinkan aplikasi pihak ketiga menampilkan data secara *real-time*. Implementasi *WebGIS* menunjukkan peta risiko banjir yang interaktif dan *user-friendly*, memungkinkan pengguna mengakses informasi potensi risiko di Provinsi Bali dengan mudah. Uji integrasi membuktikan bahwa sistem secara efektif menghubungkan model, *back-end*, dan *front-end* untuk memberikan informasi yang akurat dan bermanfaat.

Rekomendasi tindak lanjut untuk pemanfaatan Web Services pemetaan banjir di Kabupaten Badung dapat dimulai dengan pengembangan sistem berbasis spesifik wilayah. Model *Random Forest* perlu disesuaikan menggunakan data spasial dan lingkungan yang lebih rinci untuk Kabupaten Badung, seperti data elevasi (DEM), vegetasi (NDVI), dan indeks kelembapan (NDWI) yang diperbarui secara lokal. Selain itu, pelatihan ulang model menggunakan data historis banjir di wilayah ini dapat dilakukan untuk meningkatkan akurasi prediksi. Kolaborasi dengan pemangku kepentingan lokal juga menjadi langkah penting, dengan melibatkan Badan Penanggulangan Bencana Daerah (BPBD) Kabupaten Badung untuk validasi data dan integrasi sistem dengan rencana mitigasi lokal.

Selanjutnya, integrasi dengan sistem yang sudah ada di pemerintah daerah dapat dilakukan dengan menghubungkan layanan Web Services ke platform pelaporan bencana atau sistem tanggap darurat. Akses API juga dapat disediakan untuk aplikasi pihak ketiga agar masyarakat Kabupaten Badung dapat menerima data prediksi banjir secara real-time. Pengayaan fitur pada WebGIS menjadi langkah penting lainnya, seperti menambahkan simulasi aliran banjir berdasarkan curah hujan aktual, jalur evakuasi, dan lokasi penampungan terdekat. Selain itu, penyediaan layer tambahan pada peta yang menampilkan infrastruktur kritis, seperti jembatan dan jalan utama yang rentan terhadap banjir, akan meningkatkan nilai informasi peta dan membantu dalam perencanaan mitigasi yang lebih efektif.

#### DAFTAR PUSTAKA

- [1] Dedi Hermon, "Geografi Bencana Alam." Accessed: Feb. 24, 2024. [Online]. Available: [https://books.google.co.id/books?hl=en&lr=&id=GgDeEAAQBAJ&oi=fnd&pg=PA1&dq=buku+dedi+hermon+Geografi+Bencana+Alam&ots=8KozWt7w5U&sig=w6x-iveSZrEiZ-7SZ4d4UstuCfo&redir\\_esc=y#v=onepage&q&f=false](https://books.google.co.id/books?hl=en&lr=&id=GgDeEAAQBAJ&oi=fnd&pg=PA1&dq=buku+dedi+hermon+Geografi+Bencana+Alam&ots=8KozWt7w5U&sig=w6x-iveSZrEiZ-7SZ4d4UstuCfo&redir_esc=y#v=onepage&q&f=false)
- [2] Dino, "Banjir: Pengertian, Penyebab, dan Dampaknya," web.bppd.jatimprov.go.id. Accessed: Feb. 24, 2024. [Online]. Available: <https://web.bppd.jatimprov.go.id/2023/10/19/banjir-pengertian-penyebab-dan-dampaknya/>
- [3] B. E. Cahyono, E. Ikke, S. Putri, and A. T. Nugroho, "Pemetaan Daerah Genangan Banjir dan Keterkaitan dengan Penggunaan Lahan , Jenis Tanah dan Curah Hujan di Kabupaten Konawe Utara Flooded Area Mapping and Its Relationship to the Land Use , Soil Type , and Rainfall in North Konawe Regency," vol. 23, no. 2, pp. 93–100, 2022.
- [4] A. Sukmono, A. L. Nugraha, M. Awaluddin, F. Janu, and S. D. Kirana, "Pendampingan Penyusunan Peta Kelurahan Berbasis WebGIS Untuk Penunjang Pembangunan Bagi Pemerintah Kelurahan Rowosari , Kota Semarang," vol. 4, no. 1, pp. 21–28, 2021.
- [5] G. E. Engine, "Meet Earth Engine." Accessed: Feb. 26, 2024. [Online]. Available: <https://earthengine.google.com/>
- [6] Tarubali, "Sekilas Bali," tarubali.baliprov.go.id/. Accessed: Feb. 26, 2024. [Online]. Available: <https://tarubali.baliprov.go.id/sekilas-bali/>
- [7] Pusdalops, "Informasi Bencana Bulanan Provinsi Bali Periode 1 Januari - 25 Februari 2023," bppd.baliprov.go.id. Accessed: Apr. 20, 2024. [Online]. Available: <https://bppd.baliprov.go.id/article/2933/informasi-bencana-bulanan-provinsi-bali-periode-1-januari-25-februari-2023>
- [8] Pusdalops, "Informasi Bencana Bulanan Provinsi Bali," bppd.baliprov.go.id. Accessed: Apr. 20, 2024. [Online]. Available: <https://bppd.baliprov.go.id/article/3182/informasi-bencana-bulanan-provinsi-bali>
- [9] N. Sugandhi and H. Rakuasa, "Utilization of Google Earth Engine for Flood Hazard Analysis in DKI Jakarta Province," vol. 1, no. 02, pp. 41–49, 2023.
- [10] B. Laurensz, F. Lawalata, and S. Y. J. Prasetyo, "Potensi Resiko Banjir dengan Menggunakan Citra Satelit (Studi Kasus : Kota Manado, Provinsi Sulawesi Utara)," 2019.
- [11] A. Y. Isnaen and S. Y. J. Prasetyo, "Klasifikasi Wilayah Potensi Risiko Kerusakan Lahan Akibat Bencana Tsunami Menggunakan Machine Learning," vol. 8, no. April, pp. 33–42, 2022.
- [12] M. V. A. Wadud, E. Hermawan, and N. Kamilah, "Analisis Pola Distribusi Spasial Perubahan Penggunaan Lahan Dan Urban Heat Island Menggunakan Google Earth Engine," vol. 9, no. 1, pp. 259–269, 2023.
- [13] N. E. Kin, "Statistics of Confusion Matrix." Accessed: Feb. 28, 2024. [Online]. Available: <https://medium.com/machine-learning-türkiye/statistics-of-confusion-matrix-d762eaf53f74>
- [14] A. N. Fauziah, "ANALISIS SENTIMEN MENGGUNAKAN NAÏVE BAYES CLASSIFIER DAN INSET LEXICON PADA TWITTER ( Studi Kasus : Mie Gacoan ) PADA TWITTER ( Studi Kasus : Mie Gacoan )," 2023.
- [15] M. S. Anggreany, "Confusion Matrix," socs.binus.ac.id/. Accessed: Feb. 27, 2024. [Online]. Available: <https://socs.binus.ac.id/2020/11/01/confusion-matrix/>
- [16] Ramadhan, Indonesia. *Flood Hazard Mapping in Earth Engine*, (2023). [Online]. Available: <https://www.youtube.com/watch?v=35-YGQzxSzE&list=LL&index=13&t=26s>
- [17] T. Mahesti, K. D. Hartomo, S. Yulianto, and J. Prasetyo, "Penerapan Algoritma Random Forest dalam Menganalisa Perubahan Suhu Permukaan Wilayah Kota Salatiga," vol. 6, pp. 2074–2085, 2022, doi: 10.30865/mib.v6i4.4603.
- [18] J. Hu and S. Szymczak, "A Review on Longitudinal Data Analysis with Random Forest," vol. 24, no. January, pp. 1–11, 2023.

- [19] M. Habibi, "Definisi, Komponen dan Manfaat Remote Sensing," [www.technogis.co.id](http://www.technogis.co.id). Accessed: Apr. 20, 2024. [Online]. Available: <https://www.technogis.co.id/definisi-komponen-dan-manfaat-remote-sensing/>
- [20] A. Juantoro and N. Ratama, "Sistem Notifikasi Monitoring Server Pada BOT Telegram Menggunakan Cronjob Berbasis Web ( Studi Kasus : PT . Ekanuri Group )," vol. 1, no. 12, pp. 2346–2351, 2022.