Technical Documentation for Web-Based Flood Vulnerability Index (FVI) Analysis Application

Application Name: Bojonegoro Flood Vulnerability Index (FVI) Analyzer
Version: 1.0
Date: October 26, 2023
Developer: [Your Name/Team Name]
Platform: Client-side Web Application (HTML, CSS, JavaScript)

1.0 Introduction and Overview

This document provides a comprehensive technical description of the Bojonegoro Flood Vulnerability Index (FVI) Analyzer, a single-page web application designed to process, analyze, and visualize flood vulnerability across 28 sub-districts (Kecamatan) within Bojonegoro Regency, Indonesia.

The primary objective of this application is to automate the complex calculations of FVI, which integrates physical, social, and flood occurrence indicators. By transforming raw tabular data into an interactive geospatial visualization, the tool serves as a critical asset for researchers, policymakers, and disaster management agencies in identifying high-risk areas and formulating evidence-based mitigation strategies. The application operates entirely on the client-side, requiring no server infrastructure, thereby ensuring ease of deployment and use.

2.0 System Architecture

The application is built upon a purely client-side architecture, leveraging standard web technologies to perform data processing and visualization directly within the user's web browser.

2.1 Core Technologies

- **HTML5:** Defines the structure and semantic content of the application interface.
- **CSS3:** Provides styling for a responsive and user-friendly interface, including the control panel, map container, and data table.
- **JavaScript (ES6+):** The core engine of the application, responsible for data ingestion, processing, calculation, and rendering.

2.2 External Libraries and Dependencies

The application utilizes three key open-source JavaScript libraries to extend its functionality:

- **Leaflet.js (v1.9.4):** An open-source JavaScript library for interactive maps. It is used for rendering the base map (OpenStreetMap), displaying FVI markers, and managing user interactions (zoom, pan, popups).
- **SheetJS (xlsx.js v0.18.5):** A powerful library for reading and writing spreadsheet files. It enables the application to parse user-uploaded Excel (`.xlsx`, `.xls`) files directly in the browser without server-side processing.

- **shpjs:** A JavaScript library for handling geospatial vector data in ESRI Shapefile format. It allows users to upload a ZIP file containing `.shp`, `.shx`, `.dbf`, and `.prj` components to display administrative boundaries as an overlay on the main map.

2.3 Data Flow and Processing Workflow

The application follows a linear, event-driven workflow:

1. **Input:** The user initiates the process by selecting an Excel file via the HTML file input element.
2. **Ingestion:** The FileReader API reads the file into an ArrayBuffer, which is then parsed by SheetJS into a JSON object.
3. **Processing & Calculation:** The JavaScript engine executes the FVI calculation algorithm (detailed in Section 3.0) on the JSON data.
4. **Rendering:** The processed data is rendered in two primary views:
    - **Map View:** Leaflet.js generates an interactive map with colored circle markers representing FVI scores.
    - **Table View:** A dynamic HTML table displays the full processed dataset for detailed inspection.
5. **Output:** The user can download the processed results, including calculated indices and coordinates, as a new Excel file.

2.4 Application Components

The user interface is modularly structured into three main components:

- **Control Panel:** The top section for file inputs (Excel, SHP), action buttons (Process, Download), and status messages.
- **Map View:** The central interactive map displaying FVI markers and optional SHP overlays.
- **Data Table:** The bottom section presenting a scrollable table of all processed data points.

3.0 Data Requirements and Processing Methodology

3.1 Input Data Specification

The application requires a primary input file in Microsoft Excel format (`.xlsx` or `.xls`) with the following mandatory columns:

Column Name	Data Type	Description
Nama Kecamatan	Text	Name of sub-district.
Slope (%)	Numeric	Average slope of terrain in percentage.
Jarak Sungai (km)	Numeric	Average distance to the nearest river in kilometers.
Densitas Jalan (km/km^2)	Text or Numeric	Road density, categorized as "Sangat Tinggi", "Tinggi", "Sedang", "Rendah". The application converts this to a numerical scale (4, 3, 2, 1).

Column Name	Data Type	Description
Kepadatan Penduduk (jiwa/km²)	Numeric	Population density in persons per square kilometer.
Kemiskinan (%)	Numeric	Percentage of the population living below the poverty line.
Pendidikan Rendah (%)	Numeric	Percentage of the population with low educational attainment.
Kedalaman (m)	Numeric	Average flood depth in meters.
Luas Genangan (ha)	Numeric	Average inundated area in hectares.
Frekuensi (10 thn)	Numeric	Number of flood events in the last 10 years.
Latitude	Numeric	Geographic latitude of the sub-district centroid.
Longitude	Numeric	Geographic longitude of the sub-district centroid.

An optional input is a ZIP file containing a Shapefile (`.shp`, `.shx`, `.dbf`, `.prj`) for displaying administrative boundaries.

3.2 Data Preprocessing and Validation

Before calculation, the data undergoes several preprocessing steps to ensure quality and consistency:

1. **Header Trimming:** All column names from the Excel file are trimmed of leading/trailing whitespace to prevent matching errors.
2. **Data Cleaning:** Rows with duplicate sub-district names are processed, retaining only the first occurrence. Rows with missing or non-numeric values in critical columns are skipped, with warnings logged to the browser's developer console.
3. **Data Transformation:** The `Densitas Jalan` column is transformed from categorical text data into a numerical scale to enable quantitative analysis.

3.3 FVI Calculation Methodology

The FVI is calculated using a multi-step aggregation method, where each indicator is first normalized to a common scale, then aggregated into sub-indices, and finally averaged to produce the final FVI score.

Step 1: Indicator Normalization (Min-Max Scaling)

All numeric indicators are normalized to a [0, 1] range using Min-Max scaling to eliminate the influence of different units and scales. The formula is: `Normalized Value = (Value - Min) / (Max - Min)` For indicators where a lower value indicates higher vulnerability (e.g., `Jarak Sungai`), the formula is inverted: `Normalized Value = (Max - Value + Min) / (Max - Min)`

Step 2: Sub-Index Calculation

Three sub-indices are calculated by averaging the normalized values of their constituent indicators:

- **Physical Vulnerability Index (PVI):** `PVI = (Norm(Slope) + Norm(Jarak Sungai) + Norm(Densitas Jalan)) / 3`
- **Social Vulnerability Index:** `Social Vulnerability = (Norm(Kepadatan Penduduk) + Norm(Kemiskinan) + Norm(Pendidikan Rendah)) / 3`
- **Flood Occurrence Index:** `Flood Occurrence = (Norm(Kedalaman) + Norm(Luas Genangan) + Norm(Frekuensi)) / 3`

Step 3: Final FVI Calculation

The final FVI is the arithmetic mean of the three normalized sub-indices: `FVI = (Norm(PVI) + Norm(Social Vulnerability) + Norm(Flood Occurrence)) / 3`

Step 4: Vulnerability Categorization

The final FVI score is categorized into five levels to simplify interpretation:

- 0.00 - 0.20: Sangat Rendah (Very Low)
- 0.21 - 0.40: Rendah (Low)
- 0.41 - 0.60: Sedang (Medium)
- 0.61 - 0.80: Tinggi (High)
- 0.81 - 1.00: Sangat Tinggi (Very High)

4.0 User Interface and Functionality

The application provides a rich, interactive user experience:

- **Interactive Map:** Users can pan, zoom, and click on markers to view detailed FVI information for each sub-district in a popup.
- **Data Table:** A sortable, scrollable table provides a complete overview of all processed data.
- **SHP Overlay:** Users can load a Shapefile to visualize administrative boundaries. Clicking on a polygon displays its `NAMOBJ` attribute.
- **Data Export:** A function allows users to download the complete processed dataset, including all calculated indices and coordinates, as a new Excel file.

5.0 Deployment and Accessibility

5.1 Deployment

This application is designed for zero-configuration deployment. As a single, self-contained HTML file, it requires no web server, database, or installation process. It can be hosted on any static web server or run locally by simply opening the file in a modern web browser.

5.2 Browser Compatibility

The application is compatible with all modern web browsers that support ES6+ JavaScript, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. An active internet connection is required for loading base map tiles from OpenStreetMap.

6.0 Error Handling and Validation

The application includes robust error handling and user feedback mechanisms:

- **File Type Validation:** The file input restricts selections to `.xlsx` and `.xls` formats.
- **Data Integrity Checks:** The application validates the presence of all required columns and provides specific error messages if any are missing.
- **Data Quality Alerts:** Users are notified of skipped rows due to duplicate names or invalid/empty data, with detailed logs available in the developer console (accessed by pressing F12).

Note: This documentation format can be easily adapted to meet the specific templates of Q1/Q2 journals.