

# ■ DOKUMENTASI SISTEM

## Analisis Curah Hujan

Web Application untuk Regresi dan Prediksi

Versi	1.0.0
Tanggal	02 January 2026
Platform	Next.js + ONNX Runtime
Author	Ferri Krisdiantoro

### Contact Information:

■ [WhatsApp](#): +6285351168279 | ■ [Instagram](#): @solusi.ai.praktis  
■ [Website](#): ferrikrisdiantoro.com | ■ [Fastwork](#): Ferri Krisdiantoro

# DAFTAR ISI

1. Pendahuluan	3
2. Arsitektur Sistem	4
3. Modul Analisis Regresi	6
4. Modul Prediksi Curah Hujan	10
5. Machine Learning Models	14
6. API Documentation	18
7. Komponen UI	22
8. Panduan Penggunaan	24
9. Deployment Guide	27
10. Troubleshooting	29

# 1. PENDAHULUAN

## 1.1 Latar Belakang

Sistem Analisis Curah Hujan adalah aplikasi web modern yang dirancang untuk membantu dalam analisis data hidrologi, khususnya pengolahan data curah hujan menggunakan metode regresi statistik dan prediksi berbasis Machine Learning. Aplikasi ini dikembangkan menggunakan teknologi Next.js dengan inferensi model ONNX untuk deployment yang optimal di environment serverless.

## 1.2 Tujuan Sistem

Sistem ini bertujuan untuk menyediakan:

- Analisis regresi dengan 6 metode berbeda (Linear, Polynomial, Exponential, Power, Logarithmic, Moving Average)
- Prediksi curah hujan menggunakan 3 model Machine Learning (Gradient Boosting, LSTM, BiLSTM)
- Visualisasi data interaktif dengan Chart.js
- Export hasil analisis dalam format PNG dan CSV
- Antarmuka yang responsif dan user-friendly

## 1.3 Teknologi yang Digunakan

Komponen	Teknologi	Versi
Frontend	Next.js (React)	16.1.1
Backend API	Next.js API Routes	16.1.1
ML Runtime	ONNX Runtime Node	^1.20.1
Charting	Chart.js + react-chartjs-2	4.x / 5.x
Styling	CSS Custom Properties	-
Language	TypeScript	5.x
Package Manager	npm	10.x

## 2. ARSITEKTUR SISTEM

### 2.1 Struktur Direktori

Proyek ini mengikuti konvensi standar Next.js App Router dengan struktur sebagai berikut:

```
webapp/ ├── public/ ├── models/ # ONNX model files ├── model_gbr.onnx ├── model_lstm.onnx ├── model_bilstm.onnx ├── src/ ├── app/ ├── api/ # API Routes ├── predict/ ├── regression/ ├── prediction/ # Prediction page ├── regression/ # Regression page ├── layout.tsx # Root layout ├── page.tsx # Home page ├── globals.css # Global styles ├── components/ # Reusable components ├── ChartComponent.tsx ├── CsvUploader.tsx ├── DataTable.tsx ├── Icon.tsx ├── Navbar.tsx ├── lib/ # Utility libraries ├── exportUtils.ts ├── onnxLoader.ts ├── regression.ts ├── types/ # TypeScript types ├── index.ts ├── notebook/ ├── prediksi_hujan.ipynb # Training notebook
```

### 2.2 Alur Data (Data Flow)

Sistem menggunakan arsitektur client-server dengan Next.js sebagai full-stack framework:

1. User Input: Data diinput melalui form manual atau upload CSV
2. Client Processing: Data divalidasi dan diformat di browser
3. API Request: Data dikirim ke Next.js API Routes
4. Server Processing: API melakukan kalkulasi regresi atau inferensi ML
5. Model Inference: ONNX Runtime menjalankan model prediksi
6. Response: Hasil dikembalikan dalam format JSON
7. Visualization: Chart.js merender grafik hasil analisis
8. Export: User dapat mengekspor hasil ke PNG atau CSV

### 2.3 Deployment Architecture

Aplikasi ini dirancang untuk deployment di Vercel dengan arsitektur serverless. Model ONNX disimpan di folder public dan di-load secara lazy saat pertama kali dibutuhkan. Session model di-cache menggunakan singleton pattern untuk optimasi performa.

Aspek	Konfigurasi
Hosting	Vercel (Serverless)
Build	Next.js Static + Dynamic Routes
API	Serverless Functions
Model Loading	Lazy + Singleton Cache
Static Assets	Vercel CDN

## 3. MODUL ANALISIS REGRESI

### 3.1 Deskripsi Modul

Modul Analisis Regresi menyediakan kemampuan fitting data dengan berbagai model matematika. Modul ini cocok untuk analisis hubungan antara dua variabel, seperti hubungan antara debit air (flow) dan tinggi muka air (water level).

### 3.2 Metode Regresi yang Tersedia

#### Linear Regression

Formula:  $y = a + bx$

Regresi linier sederhana untuk hubungan proporsional antara X dan Y.

Use Case: Hubungan linear seperti kecepatan vs jarak

#### Polynomial Regression

Formula:  $y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$

Regresi polynomial untuk kurva non-linear dengan degree 2-6.

Use Case: Kurva parabola, kubik, atau pola kompleks

#### Exponential Regression

Formula:  $y = a \times e^{(bx)}$

Regresi eksponensial untuk pertumbuhan/penurunan eksponensial.

Use Case: Pertumbuhan populasi, peluruhan radioaktif

#### Power Regression

Formula:  $y = ax^b$

Regresi power untuk hubungan pangkat antara variabel.

Use Case: Kurva rating (Q-H), hukum fisika

#### Logarithmic Regression

Formula:  $y = a + b \ln(x)$

Regresi logaritmik untuk kurva yang melambat.

Use Case: Kurva pembelajaran, diminishing returns

#### Moving Average

Formula:  $MA(n) = (x_1 + x_2 + \dots + x_n) / n$

Rata-rata bergerak untuk smoothing data time series.

Use Case: Trend detection, noise reduction

### 3.3 Metrik Evaluasi

Setiap hasil regresi dilengkapi dengan metrik evaluasi untuk menilai kualitas fitting:

Metrik	Rumus	Interpretasi
R <sup>2</sup> (Coefficient of Determination)	$1 - (SS_{\text{res}} / SS_{\text{tot}})$	0-1, semakin tinggi semakin baik
MAE (Mean Absolute Error)	$\sum  y - \hat{y}  / n$	Error rata-rata, satuan sama dengan Y
RMSE (Root Mean Square Error)	$\sqrt{\sum (y - \hat{y})^2 / n}$	Error kuadrat, sensitif terhadap outlier

### 3.4 Implementasi Teknis

Kalkulasi regresi diimplementasikan dalam file [src/lib/regression.ts](#) dengan algoritma yang dioptimasi untuk browser dan Node.js. Polynomial regression menggunakan Gaussian elimination dengan partial pivoting untuk stabilitas numerik.

```
// Contoh penggunaan API Regresi
const response = await fetch('/api/regression', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body:
    JSON.stringify({ data: [{ x: 100, y: 15 }, { x: 200, y: 22 }], type: 'linear' //
    atau 'polynomial', 'exponential', dll }) });
const result = await response.json();
// result.formula = "y = 11.9707 + 0.0001x" // result.r2 = 0.9624
```

## 4. MODUL PREDIKSI CURAH HUJAN

### 4.1 Deskripsi Modul

Modul Prediksi Curah Hujan menggunakan Machine Learning untuk memprediksi curah hujan 1-30 hari ke depan berdasarkan data historis. Sistem ini menggunakan tiga model yang telah dilatih dan diekspor ke format ONNX untuk inferensi yang cepat di server.

### 4.2 Model Machine Learning

#### Gradient Boosting Regressor (GBR)

Tipe	Tabular ML
Input Shape	7 features (lag, rolling stats, month)
MAE (Test)	6.42 mm
RMSE (Test)	11.28 mm

Model ensemble berbasis decision tree yang efektif untuk data tabular dengan feature engineering manual.

Features:

- lag\_1 (curah hujan kemarin)
- lag\_3 (3 hari lalu)
- lag\_7 (7 hari lalu)
- roll\_mean\_3 (rata-rata 3 hari)
- roll\_mean\_7 (rata-rata 7 hari)
- roll\_max\_7 (maksimum 7 hari)
- bulan\_idx (indeks bulan 0-11)

#### LSTM (Long Short-Term Memory)

Tipe	Deep Learning
Input Shape	Sequence [1, 7, 1]
MAE (Test)	7.15 mm
RMSE (Test)	12.03 mm

Model recurrent neural network yang mampu menangkap dependensi jangka panjang dalam data sekuensial.

Features:

- 7 nilai curah hujan berurutan sebagai input sequence

#### BiLSTM (Bidirectional LSTM)

Tipe	Deep Learning
Input Shape	Sequence [1, 7, 1]
MAE (Test)	6.89 mm

<b>RMSE (Test)</b>	11.67 mm
--------------------	----------

Varian LSTM yang memproses sequence dari dua arah untuk konteks yang lebih kaya.

*Features:*

- 7 nilai curah hujan berurutan, diproses forward dan backward

### 4.3 Recursive Forecasting

Untuk prediksi multi-step (lebih dari 1 hari), sistem menggunakan strategi recursive forecasting. Prediksi hari ke- $n$  digunakan sebagai input untuk memprediksi hari ke- $(n+1)$ . Metode ini memungkinkan prediksi hingga 30 hari ke depan dari 7 hari data historis.

**Step 1:** Input: 7 nilai historis terakhir  $[d_1, d_2, \dots, d_7]$

**Step 2:** Prediksi  $d_8$  menggunakan model

**Step 3:** Update input menjadi  $[d_2, d_3, \dots, d_7, d_8]$

**Step 4:** Prediksi  $d_9$  menggunakan model

**Step 5:** Ulangi hingga horizon tercapai (maks 30 hari)



## 5. MACHINE LEARNING MODELS

### 5.1 Training Pipeline (Notebook)

Model dilatih menggunakan Jupyter Notebook dengan file [notebook/prediksi\\_hujan.ipynb](#). Berikut adalah tahapan training yang dilakukan:

1. Data Loading: Membaca dataset curah hujan harian dari file CSV
2. Feature Engineering: Membuat lag features, rolling statistics, dan seasonal features
3. Data Splitting: Membagi data menjadi train (80%) dan test (20%)
4. Model Training: Melatih GBR, LSTM, dan BiLSTM dengan hyperparameter tuning
5. Model Evaluation: Mengevaluasi performa dengan MAE, RMSE pada test set
6. ONNX Export: Mengekspor model ke format ONNX untuk deployment web

### 5.2 Feature Engineering Details

Feature engineering adalah kunci untuk performa model GBR. Berikut adalah features yang digunakan:

Feature	Deskripsi	Formula/Metode
lag_1	Curah hujan 1 hari sebelumnya	$y[t-1]$
lag_3	Curah hujan 3 hari sebelumnya	$y[t-3]$
lag_7	Curah hujan 7 hari sebelumnya	$y[t-7]$
roll_mean_3	Rata-rata 3 hari terakhir	$\text{mean}(y[t-3:t])$
roll_mean_7	Rata-rata 7 hari terakhir	$\text{mean}(y[t-7:t])$
roll_max_7	Maksimum 7 hari terakhir	$\text{max}(y[t-7:t])$
bulan_idx	Indeks bulan (seasonality)	$\text{month}(\text{date}) - 1$

### 5.3 ONNX Runtime Integration

ONNX (Open Neural Network Exchange) digunakan sebagai format portable untuk deployment model. Keuntungan menggunakan ONNX:

- Cross-platform: Dapat dijalankan di berbagai environment (Python, Node.js, browser)
- Optimized: Runtime yang dioptimasi untuk CPU inference
- Lightweight: Ukuran model lebih kecil dibanding format asli
- Serverless-friendly: Cocok untuk Vercel dan AWS Lambda

### 5.4 Model Files

File	Size	Input Shape	Description
model_gbr.onnx	200 KB	[1, 7]	Gradient Boosting with 7 features
model_lstm.onnx	37 KB	[1, 7, 1]	LSTM sequence model
model_bilstm.onnx	96 KB	[1, 7, 1]	Bidirectional LSTM

## 6. API DOCUMENTATION

### 6.1 Regression API

**Endpoint:** `POST /api/regression`

Melakukan analisis regresi pada data X-Y.

**Request Body:**

```
{ "data": [ { "x": 33262.03, "y": 14.44 }, { "x": 48285.70, "y": 18.16 }, ... ],
"type": "linear" | "polynomial" | "exponential" | "power" | "logarithmic" |
"moving-average", "degree": 2 // Optional, for polynomial (2-6) or MA window }
```

**Response:**

```
{ "type": "linear", "formula": "y = 11.9707 + 0.0001x", "coefficients": [11.9707,
0.0001], "r2": 0.9624, "mae": 0.6594, "rmse": 0.7725, "predictions": [14.32, 17.89,
... ] }
```

### 6.2 Prediction API

**Endpoint:** `POST /api/predict`

Melakukan prediksi curah hujan menggunakan model ML.

**Request Body:**

```
{ "model": "gbr" | "lstm" | "bilstm", "horizon": 7, // 1-30 days "historicalData":
[ { "date": "2024-11-01", "value": 12.5 }, { "date": "2024-11-02", "value": 8.3 },
... // min 7 data points ] }
```

**Response:**

```
{ "success": true, "model": { "name": "Gradient Boosting Regressor", "mae": 6.42,
"rmse": 11.28 }, "predictions": [ { "date": "2024-12-01", "value": 15.32 }, {
"date": "2024-12-02", "value": 12.78 }, ... ] }
```

### 6.3 Error Handling

Status Code	Error	Penyebab
400	Invalid request body	Body request tidak valid
400	At least 2 data points required	Data kurang dari minimum
400	Invalid regression type	Tipe regresi tidak dikenal
400	At least 7 historical data points	Data historis kurang untuk ML
500	Regression calculation failed	Error internal saat kalkulasi
500	Model inference failed	Error saat inferensi ONNX

## 7. KOMPONEN UI

### 7.1 Daftar Komponen

#### Navbar

File: [src/components/Navbar.tsx](#)

Navigation bar responsif dengan hamburger menu untuk mobile. Menampilkan logo, judul aplikasi, dan link ke halaman Beranda, Regresi, dan Prediksi.

#### ChartComponent

File: [src/components/ChartComponent.tsx](#)

Wrapper untuk Chart.js yang mendukung scatter plot dan line chart. Menerima props untuk data, title, dan axis labels yang dapat dikustomisasi.

#### DataTable

File: [src/components/DataTable.tsx](#)

Tabel interaktif untuk menampilkan dan mengedit data. Mendukung checkbox toggle untuk enable/disable data point dan input edit inline.

#### CsvUploader

File: [src/components/CsvUploader.tsx](#)

Komponen upload file CSV dengan drag-and-drop. Mendukung dua mode: XY data (x,y columns) dan timeseries (date,value columns).

#### Icon

File: [src/components/Icon.tsx](#)

Library SVG icons dengan 15+ icons termasuk home, chart-line, download, copy, rocket, dll. Ukuran dan warna dapat dikustomisasi.

### 7.2 Utility Libraries

#### exportUtils.ts

Fungsi untuk export chart sebagai PNG dan data sebagai CSV.

Functions: [exportChartAsPNG\(\)](#), [exportXYDataAsCSV\(\)](#), [exportTimeSeriesAsCSV\(\)](#)

#### regression.ts

Implementasi algoritma regresi (6 metode) dengan kalkulasi metrik.

Functions: [linearRegression\(\)](#), [polynomialRegression\(\)](#), [exponentialRegression\(\)](#), [powerRegression\(\)](#), [logarithmicRegression\(\)](#), [movingAverageRegression\(\)](#)

#### onnxLoader.ts

ONNX model loader dengan singleton pattern dan feature preparation.

*Functions:* `getOnnxSession()`, `runModelInference()`, `recursiveForecast()`, `prepareGBRFeatures()`,  
`prepareLSTMFeatures()`

## 8. PANDUAN PENGGUNAAN

### 8.1 Modul Regresi

#### Langkah 1: Input Data

Upload file CSV dengan kolom X dan Y, atau input data secara manual melalui tabel. Gunakan checkbox untuk enable/disable data point tertentu.

#### Langkah 2: Pilih Metode

Pilih metode regresi dari dropdown: Linear, Polynomial (degree 2-6), Exponential, Power, Logarithmic, atau Moving Average.

#### Langkah 3: Visualisasi

Grafik akan otomatis terupdate menampilkan scatter plot data dan garis/kurva regresi.

#### Langkah 4: Hasil

Lihat formula regresi, nilai  $R^2$ , MAE, dan RMSE. Gunakan tombol Copy untuk menyalin formula.

#### Langkah 5: Export

Klik 'Export PNG' untuk download grafik atau 'Export Data CSV' untuk download data.

### 8.2 Modul Prediksi

#### Langkah 1: Input Data Historis

Upload file CSV dengan kolom date dan value, atau input manual. Minimum 7 data point diperlukan.

#### Langkah 2: Pilih Model

Pilih model ML: Gradient Boosting (tercepat, akurasi bagus), LSTM (deep learning), atau BiLSTM (bidirectional).

#### Langkah 3: Set Horizon

Tentukan berapa hari ke depan yang ingin diprediksi (1-30 hari) menggunakan slider.

#### Langkah 4: Jalankan Prediksi

Klik tombol 'Jalankan Prediksi' untuk memulai inferensi model.

#### Langkah 5: Lihat Hasil

Grafik akan menampilkan data historis dan prediksi. Tabel 'Hasil Prediksi' menampilkan nilai prediksi per hari.

#### Langkah 6: Export

Export grafik sebagai PNG atau data lengkap (historis + prediksi) sebagai CSV.

### 8.3 Format File CSV

#### Untuk Modul Regresi (XY Data):

```
x,y 33262.03,14.44 48285.70,18.16 68609.89,21.55
```

**Untuk Modul Prediksi (Time Series):**

```
date,value 2024-11-01,12.5 2024-11-02,8.3 2024-11-03,0
```

## 9. DEPLOYMENT GUIDE

### 9.1 Prerequisites

- Node.js v18.x atau lebih tinggi
- npm v10.x atau pnpm
- Git untuk version control
- Akun Vercel (opsional, untuk deployment)

### 9.2 Local Development

```
# Clone repository git clone cd webapp # Install dependencies npm install # Run development server npm run dev # Open browser at http://localhost:3000
```

### 9.3 Build Production

```
# Build for production npm run build # Start production server (local) npm start # Check build output # Route (app) # ■ ■ / (Static) # ■ f /api/predict (Dynamic - serverless) # ■ f /api/regression (Dynamic - serverless) # ■ ■ /prediction (Static) # ■ ■ /regression (Static)
```

### 9.4 Deploy to Vercel

1. Push code ke GitHub repository
2. Login ke Vercel dan import project dari GitHub
3. Vercel akan otomatis mendeteksi Next.js
4. Klik Deploy dan tunggu build selesai
5. Akses aplikasi di URL yang diberikan Vercel

### 9.5 Environment Variables

Aplikasi ini tidak memerlukan environment variables khusus. Semua konfigurasi sudah hardcoded untuk kesederhanaan. Jika perlu, tambahkan di Vercel dashboard.

## 10. TROUBLESHOOTING

### Problem: Error: Model file not found

*Penyebab:* File ONNX tidak ada di folder public/models

*Solusi:* Pastikan file model\_gbr.onnx, model\_lstm.onnx, dan model\_bilstm.onnx ada di public/models/

### Problem: Prediksi menghasilkan nilai negatif

*Penyebab:* Model output belum di-clamp

*Solusi:* Sistem sudah handle ini dengan `Math.max(0, prediction)`. Jika masih terjadi, cek preprocessing data.

### Problem: Chart tidak muncul

*Penyebab:* SSR issue dengan Chart.js

*Solusi:* ChartComponent sudah menggunakan dynamic import dengan `ssr: false`. Pastikan `react-chartjs-2` terinstall.

### Problem: CSV upload tidak terbaca

*Penyebab:* Format CSV tidak sesuai

*Solusi:* Pastikan CSV memiliki header (x,y atau date,value) dan separator koma. Encoding harus UTF-8.

### Problem: Build error: Cannot find module onnxruntime-node

*Penyebab:* Dependency belum terinstall

*Solusi:* Jalankan `npm install onnxruntime-node`. Pastikan Node.js versi 18+.

### Problem: Regresi menghasilkan NaN/Infinity

*Penyebab:* Data tidak valid untuk metode tertentu

*Solusi:* Exponential/Power regression membutuhkan  $y > 0$ . Logarithmic membutuhkan  $x > 0$ . Filter data terlebih dahulu.

## KONTAK SUPPORT

Jika mengalami masalah yang tidak tercantum di atas, silakan hubungi:

Platform	Kontak
WhatsApp	+6285351168279
Instagram	@solusi.ai.praktis
Website	ferrikrisdiantoro.com
Fastwork	Ferri Krisdiantoro



**Terima kasih telah menggunakan Sistem Analisis Curah Hujan!**

Dokumentasi ini dibuat pada 02 January 2026 11:24