

LAPORAN PEMBUATAN WEBSITE TRANSLATE BAHASA JAWA MENGGUNAKAN NEXT.JS

Disusun oleh :

22312012 Riko Maulana Al Habib

22312014 Ahmad Fadly Suryaman

22312040 Putra Hidayatul Akbar

Kelompok Mas Do



PRODI S1 INFORMATIKA

FAKULTAS TEKNIK DAN ILMU KOMPUTER

UNIVERSITAS TEKNOKRAT INDONESIA

2025

1. Latar Belakang

Bahasa Jawa merupakan salah satu bahasa daerah terbesar di Indonesia dengan jutaan penutur aktif. Namun, belum banyak platform penerjemah daring (online) yang mendukung Bahasa Jawa secara praktis dan mudah diakses. Oleh karena itu, dibuatlah sebuah website penerjemah sederhana menggunakan framework Next.js, yang mampu menerjemahkan kata/kalimat dari Bahasa Indonesia ke Bahasa Jawa atau sebaliknya, dengan antarmuka yang ringan dan mudah digunakan.

2. Tujuan

Tujuan dari proyek ini adalah:

- - Membuat website penerjemah Bahasa Jawa berbasis web.
- - Menggunakan Next.js untuk pengembangan frontend dan API handler backend secara fullstack.
- - Menyediakan fitur input teks dan hasil terjemahan secara real-time dari data lokal (JSON).

3. Teknologi yang Digunakan

- - Next.js (React Framework)
- - Tailwind CSS (untuk styling antarmuka)
- - JavaScript / TypeScript
- - JSON (sebagai penyimpanan data kosakata)
- - Vercel (untuk deployment)

4. Struktur Proyek

```
/project-root
├── /pages
│   ├── index.tsx      → Halaman utama
│   └── api/translate.ts → API untuk menerjemahkan
├── /data
│   └── dictionary.json → Data kosakata bahasa Indonesia ↔ Jawa
└── /components
    └── TranslatorBox.tsx → Komponen input dan output
```

— tailwind.config.js	→ Konfigurasi Tailwind CSS
— tsconfig.json	→ Konfigurasi TypeScript
— next.config.js	→ Konfigurasi Next.js
— package.json	→ Dependency proyek

5. Implementasi

5.1. Tampilan Antarmuka (UI)

Antarmuka pengguna terdiri dari:

- Input Box: untuk memasukkan kata atau kalimat.
- Tombol Terjemahkan
- Hasil Terjemahan: muncul secara dinamis setelah klik tombol.

5.2. Data Kosakata (dictionary.json)

```
[
  { "indonesia": "makan", "jawa": "nedha" },
  { "indonesia": "tidur", "jawa": "turu" },
  { "indonesia": "pergi", "jawa": "budhal" }
]
```

5.3. Fungsi API (pages/api/translate.ts)

```
import data from '../data/dictionary.json'

export default function handler(req, res) {
  const { text } = req.query;
  const result = data.find(entry =>
    entry.indonesia.toLowerCase() === text.toLowerCase()
  );
  res.status(200).json({ translation: result?.jawa || "Tidak ditemukan" });
}
```

5.4. Komponen Input (TranslatorBox.tsx)

```
'use client';
import { useState } from 'react';
import axios from 'axios';
```

```

export default function TranslatorBox() {
  const [text, setText] = useState("");
  const [result, setResult] = useState("");

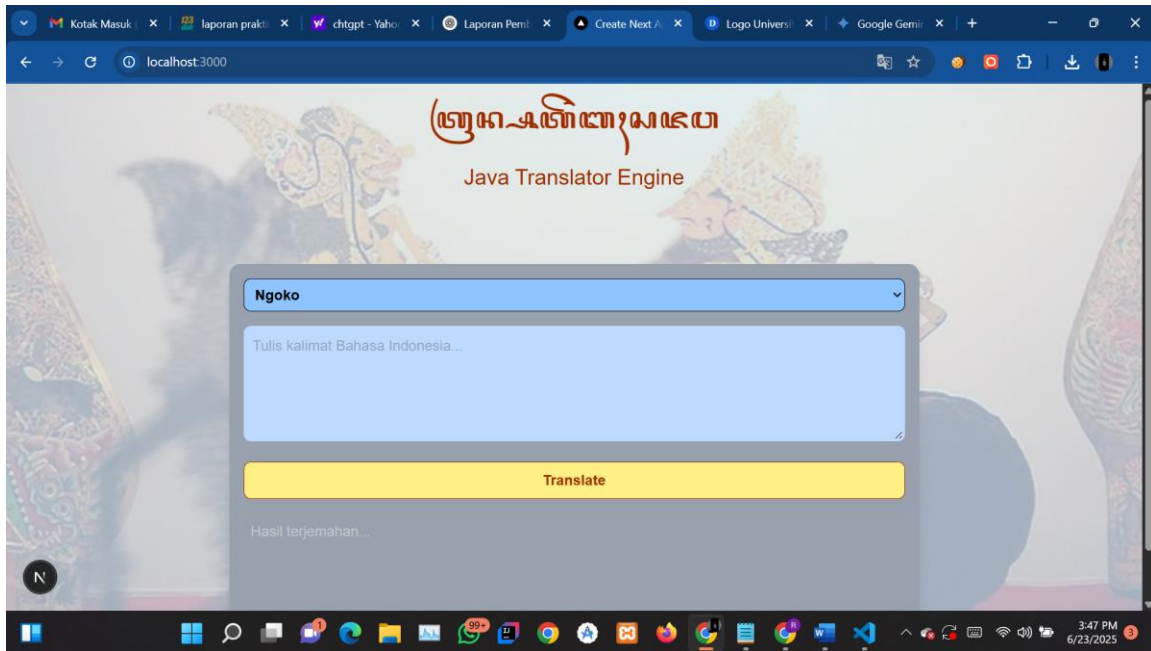
  const translate = async () => {
    const res = await axios.get(`/api/translate?text=${text}`);
    setResult(res.data.translation);
  };

  return (
    <div className="space-y-4">
      <input className="p-2 border rounded w-full" onChange={(e) =>
setText(e.target.value)} />
      <button onClick={translate} className="bg-blue-500 text-white px-4 py-2
rounded">Terjemahkan</button>
      <p className="text-xl font-bold">Hasil: {result}</p>
    </div>
  );
}

```

6. Hasil dan Uji Coba

Website berhasil menampilkan terjemahan sesuai dengan input pengguna berdasarkan data JSON. Kosakata yang tidak ditemukan akan menampilkan pesan: “Tidak ditemukan”. Aplikasi berjalan stabil saat diuji di browser desktop dan mobile.



7. Kesimpulan

Proyek ini berhasil membuktikan bahwa Next.js dapat digunakan untuk membangun aplikasi penerjemah sederhana dengan pendekatan fullstack (API + UI). Penggunaan data JSON sangat cocok untuk dataset kecil, namun untuk pengembangan lebih lanjut dapat diintegrasikan dengan database atau AI Translation API.

8. Pengembangan Selanjutnya

- - Menambahkan fitur deteksi bahasa otomatis.
- - Mengintegrasikan database MySQL atau MongoDB untuk kosakata.
- - Menambahkan fitur text-to-speech (TTS).
- - Membuat versi Progressive Web App (PWA).