Dokumentasi Teknis Musim Tanam App

Daftar Isi

1. Struktur Aplikasi

1.1 File Struktur

```
musim-tanam-app/
├── src/
│   └── app.js          # Komponen React dan logika utama
├── index.html          # Entry point dan styling
├── README.md           # Dokumentasi umum
└── docs/
    └── technical-documentation.md  # Dokumentasi teknis
```

1.2 Teknologi yang Digunakan

- React (17.0.2)
- Chart.js (4.x)
- Google Gemini AI API
- FontAwesome Icons
- Vanilla JavaScript
- CSS3 Modem

2. Komponen Utama

2.1 App Component

```
function App() {
    const [activeTab, setActiveTab] = useState('weather');
    // Komponen utama yang mengelola navigasi tab
    // dan mengatur tampilan komponen WeatherCharts atau FarmingAnalysis
}
```

2.2 WeatherCharts Component

```
function WeatherCharts({ cityData }) {
    // Komponen untuk visualisasi data cuaca menggunakan Chart.js
    // - Grafik suhu (line chart)
    // - Grafik curah hujan (bar chart)
    // - Tabel data cuaca bulanan
}
```

2.3 FarmingAnalysis Component

```
function FarmingAnalysis() {
    // Komponen untuk analisis usaha tani
    // - Form input data
    // - Perhitungan finansial
    // - Analisis AI
}
```

3. Fungsi dan Perhitungan

3.1 Perhitungan Finansial

3.1.1 Break Even Point (BEP)

```
const calculateBEP = () => {
    const totalCosts = calculateTotalCosts();
    const pricePerKg = parseInt(formData.marketPrice) || 0;

    if (pricePerKg === 0) return { units: 0, value: 0 };

    const bepUnits = totalCosts / pricePerKg; // BEP dalam kg
    const bepValue = totalCosts; // BEP dalam Rupiah

    return { units: bepUnits, value: bepValue };
};
```

Fungsi ini menghitung titik impas (BEP) dalam:

- Unit (kg): Total biaya dibagi harga per kg
- Nilai (Rp): Total biaya yang harus dicapai

3.1.2 B/C Ratio

```
const calculateBCRatio = () => {
    const totalCosts = calculateTotalCosts();
    const totalRevenue = calculateEstimatedRevenue();

    if (totalCosts === 0) return 0;
    return totalRevenue / totalCosts;
};
```

Menghitung rasio manfaat-biaya untuk menilai kelayakan usaha:

- B/C > 1: Usaha menguntungkan
- B/C = 1: Titik impas
- B/C < 1: Usaha merugi

3.1.3 ROI (Return on Investment)

```
const calculateROI = () => {
    const totalCosts = calculateTotalCosts();
    const totalRevenue = calculateEstimatedRevenue();
    const profit = totalRevenue - totalCosts;

    if (totalCosts === 0) return 0;
    return (profit / totalCosts) * 100;
};
```

Menghitung persentase pengembalian investasi.

3.2 Analisis Pertanian

3.2.1 Analisis Kesesuaian Lahan

```
const analyzeLandSuitability = () => {
    const crop = CROPS.find(c => c.name.toLowerCase() === formData.cropType.toLowerCase());
    if (!crop) return { suitable: false, message: 'Jenis tanaman tidak ditemukan dalam database' };

    const soilTypeMatch = formData.soilType.toLowerCase().includes(crop.soilType.toLowerCase());
    const phValue = parseFloat(formData.soilPh);
    const phSuitable = phValue >= 5.5 && phValue <= 7.5;

    return {
        suitable: soilTypeMatch && phSuitable,
        message: `${soilTypeMatch ? '✓' : '✗'} Jenis Tanah\n${phSuitable ? '✓' : '✗'} pH Tanah`,
        recommendations: crop.challenges
    };
};
```

Menganalisis kesesuaian lahan berdasarkan:

- Jenis tanah
- pH tanah
- Rekomendasi spesifik tanaman

3.2.2 Analisis Musim Tanam

```
const analyzePlantingSeason = (cityData) => {
    const crop = CROPS.find(c => c.name.toLowerCase() === formData.cropType.toLowerCase());
    if (!crop) return { suitable: false, message: 'Jenis tanaman tidak ditemukan' };

    const currentMonth = new Date().getMonth();
    const monthNames = ["Januari", "Februari", "Maret", "April", "Mei", "Juni",
                        "Juli", "Agustus", "September", "Oktober", "November", "Desember"];
    const currentMonthName = monthNames[currentMonth];

    const isOptimalMonth = crop.bestPlantingMonths.includes(currentMonthName);
    const weatherData = cityData?.monthlyWeather[currentMonth];

    return {
        suitable: isOptimalMonth && tempSuitable && waterSuitable,
        message: `
            ${isOptimalMonth ? '✓' : '✗'} Bulan Tanam Optimal
            ${tempSuitable ? '✓' : '✗'} Suhu Sesuai
            ${waterSuitable ? '✓' : '✗'} Curah Hujan Sesuai
        `,
        recommendations: weatherData.season === "Hujan" ?
            "Perhatikan drainase dan antisipasi serangan penyakit" :
            "Pastikan ketersediaan air irigasi dan antisipasi kekeringan"
    };
};
```

Menganalisis kesesuaian musim tanam berdasarkan:

- Bulan optimal
- Suhu
- Curah hujan
- Musim (hujan/kemarau)

4. Data dan Konstanta

4.1 Data Tanaman (CROPS)

```
const CROPS = [
  {
    name: 'Padi',
    optimalTemperature: { min: 22, max: 30 },
    waterRequirement: { min: 160, max: 200 },
    bestPlantingMonths: ['Oktober', 'November', 'Maret', 'April'],
    soilType: 'Tanah liat atau lempung berdebu',
    challenges: 'Perhatikan drainase dan pengairan yang tepat'
  },
  // Data tanaman lainnya...
];
```

4.2 Data Cuaca

```
const WEATHER_DATA = {
  "Bengkulu": {
    monthlyWeather: [
      {
        month: "Januari",
        temperature_min: 23,
        temperature_max: 31,
        precipitation: 300,
        humidity: 87,
        season: "Hujan"
      },
      // Data bulan lainnya...
    ]
  }
  // Data kota lainnya...
};
```

5. Integrasi AI

5.1 Google Gemini AI

```
const getGeminiAnalysis = async (prompt) => {
    try {
        const response = await fetch(
            `https://generativelanguage.googleapis.com/v1beta/models/gemini-pro:generateContent?key=${GEMINI_API_KEY}`,
            {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json',
                },
                body: JSON.stringify({
                    contents: [{ parts: [{ text: prompt }] }]
                })
            }
        );
        // Proses response dan format hasil analisis
    } catch (error) {
        console.error('Error calling Gemini API:', error);
        throw error;
    }
};
```

6. Styling dan UI

6.1 Tema Warna

```
:root {
    --primary-color: #059669;
    --secondary-color: #064e3b;
    --background-light: #f0fdf4;
    --border-color: #e2e8f0;
}
```

6.2 Responsive Design

```
@media (max-width: 768px) {
    .form-container {
        grid-template-columns: 1fr;
    }
    .analysis-container {
        flex-direction: column;
    }
}
```

6.3 Komponen UI

- Tab Navigation
- Form Input
- Analysis Cards
- Charts
- Loading States
- Error Messages

7. Error Handling

7.1 Validasi Input

```
const validateInput = (formData) => {
    // Validasi data numerik
    if (formData.landSize <= 0) return false;
    if (formData.soilPh < 0 || formData.soilPh > 14) return false;
    // Validasi lainnya...
    return true;
};
```

7.2 API Error Handling

```
try {
    const analysis = await getGeminiAnalysis(prompt);
    setAnalysis(analysis);
} catch (error) {
    setError('Terjadi kesalahan saat menganalisis data. Silakan coba lagi.');
} finally {
    setIsLoading(false);
}
```

8. Pengembangan Selanjutnya

8.1 Optimasi Performa

- Implementasi React.memo untuk komponen yang sering di-render
- Lazy loading untuk komponen besar
- Caching hasil analisis

8.2 Fitur Mendatang

- Integrasi data cuaca real-time
- Sistem autentikasi pengguna
- Penyimpanan riwayat analisis
- Ekspor hasil analisis ke PDF

Dokumentasi ini mencakup aspek teknis utama dari Musim Tanam App. Untuk informasi lebih lanjut, silakan merujuk ke kode sumber atau menghubungi tim pengembang.