



# The Reading Ibay

"UAS II3160 - Teknologi Sistem Terintegrasi"

"Penyusun:" M Ikhbar A (18223050)

## Ringkasan Proyek

**The Reading Ibay** adalah layanan integrasi cerdas yang menghubungkan **API Katalog Buku** (Service A - Rekan) dengan **API Katalog Musik** (Service B - Internal).

Layanan ini memanfaatkan **Gemini AI** untuk menganalisis deskripsi dan tema buku guna menentukan profil musik yang paling sesuai (genre, suasana, dan energi). Hasil akhirnya adalah sebuah "Soundtrack" kurasi AI yang memberikan pengalaman membaca yang lebih imersif.

### [!NOTE]

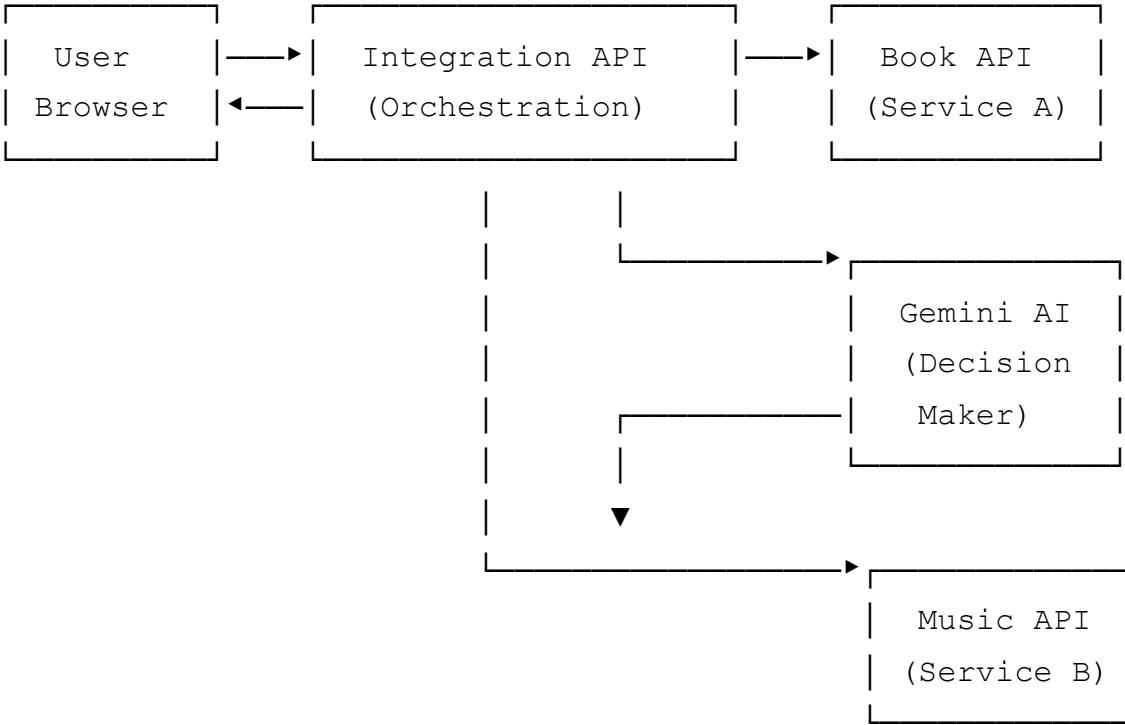
Proyek ini merupakan pemenuhan **Tugas 3 (Integrasi Layanan)** untuk proyek akhir mata kuliah Teknologi Sistem Terintegrasi (TST).

## Fitur Utama

- **✓ Integrasi End-to-End:** Alur otomatis mulai dari pengambilan data Buku → Analisis Gemini AI → Pencarian Musik.
- **✓ AI-Powered Recommendation:** Rekomendasi musik yang akurat berdasarkan analisis sentimen dan tema buku.
- **✓ Antarmuka Modern:** Interface web yang responsif dengan animasi yang halus.
- **✓ Profil Musik Real-time:** Menghasilkan metadata musik (genre, mood, energi) secara instan.
- **✓ Strategi Fallback:** Penanganan error yang kuat jika salah satu service mengalami kendala.
- **✓ RESTful API:** Mendukung integrasi pihak ketiga dengan format JSON standar.

## Arsitektur Sistem

Layanan ini menggunakan pola **Orchestration**, di mana *Integration API* bertindak sebagai konduktor utama yang mengatur aliran data antar layanan.



## 🚫 Strategi Deployment

Layanan ini **TIDAK dideploy pada STB (Set-Top Box)** dengan pertimbangan:

- Latency:** Panggilan API ke Gemini AI membutuhkan waktu pemrosesan yang dapat membebani resource STB.
- Resource Intensif:** Pengolahan data dan koordinasi antar service memakan memori dan CPU yang lebih besar.
- Separation of Concerns:** Menjaga agar STB tetap fokus pada layanan inti (Music API).

**Rekomendasi:** Gunakan server terpisah seperti VPS, Cloud (Google Cloud/AWS), atau mesin lokal untuk performa optimal.

## 🔧 Tech Stack

### Backend

- Runtime:** Node.js 18+
- Framework:** Express.js 4.x
- HTTP Client:** Axios
- AI Engine:** @google/generative-ai (Gemini SDK)
- Security:** Helmet, CORS

# Frontend

- HTML5 & Vanilla CSS (Custom UI/UX)
  - Vanilla JavaScript (Modern ES6+)
- 

## Panduan Instalasi

### Prasyarat

- Node.js versi 18 ke atas
- npm (Node Package Manager)
- API Keys:
  - Gemini AI API Key ([Dapatkan di Google AI Studio](#))
  - Kredensial API Buku (dari Rekan)
  - Kredensial API Musik

### Langkah-langkah Setup

#### 1. Clone dan Instal Dependensi:

```
npm install
```

#### 2. Konfigurasi Environment:

```
cp .env.example .env
```

#### 3. Isi variabel di file .env :

```
PORt=3000  
GEMINI_API_KEY=isi_api_key_gemini_anda  
BOOK_API_BASE_URL=http://url-api-buku.com/api  
BOOK_API_KEY=api_key_buku  
MUSIC_API_BASE_URL=http://url-api-musik.com/api  
MUSIC_API_KEY=api_key_musik
```

---

## Penggunaan

# Mode Pengembangan (Auto-reload)

```
npm run dev
```

## Mode Produksi

```
npm start
```

Aplikasi dapat diakses melalui: <http://localhost:3000>

---

## Dokumentasi API

### 1. Mendapatkan Rekomendasi Soundtrack

**Endpoint:** GET /api/soundtrack/:bookId

**Contoh Response Sukses:**

```
{
  "success": true,
  "data": {
    "book": {
      "id": "123",
      "title": "The Underground Detective",
      "genre": "Mystery",
      "description": "Detektif yang mencari jejak di kota gelap..."
    },
    "musicProfile": {
      "primaryGenre": "jazz",
      "mood": "dark",
      "energy": 4,
      "reasoning": "Karena buku memiliki tema noir dan misteri urban."
    },
    "recommendations": [
      {
        "id": "m456",
        "title": "Midnight Blues",
        "artist": "Miles Davis"
      }
    ]
  }
}
```

```
]
}
}
```

## 2. Cek Kesehatan Sistem

**Endpoint:** GET /api/health

---

## Struktur Folder

```
integrated_service/
├── src/
│   ├── controllers/      # Logika orkestrasi utama
│   ├── services/         # Integrasi API (Book, Music, Gemini)
│   ├── prompts/          # Template prompt untuk Gemini AI
│   ├── routes/           # Definisi endpoint API
│   └── app.js             # Inisialisasi Express
├── public/               # File statis (Frontend)
│   ├── index.html
│   ├── css/
│   └── js/
└── .env                  # Konfigurasi rahasia
```

---