

Implementasi Teknologi Rest API Dengan Node Js Untuk Aplikasi Rekomendasi Destinasi Wisata

Anjas Ramadhani¹, Nandang Iriadi², Rachmat Hidayat³

Universitas Bina Sarana Informatika¹²³

anjas.ajr@bsi.ac.id¹, nandang.ndi@bsi.ac.id², rachmat.rch@bsi.ac.id³

Diterima (07-04-2025)	Direvisi (15-04-2025)	Disetujui (18-04-2025)
--------------------------	--------------------------	---------------------------

Abstrak - Kemajuan teknologi informasi terutama di era digital saat ini, pengguna menginginkan akses cepat dan mudah terkait informasi perjalanan seperti destinasi, akomodasi, dan transportasi, aplikasi rekomendasi destinasi wisata. Teknologi Rest API dengan Node JS terbukti efektif dalam menangani permintaan dan memberikan respon yang cepat, sementara Rest Api yang dikembangkan menyediakan fitur pencarian destinasi wisata berdasarkan kota, kategori, rating, dan harga. Lokis adalah aplikasi mobile inovatif yang dirancang untuk merevolusi cara wisatawan dan penduduk lokal menjelajahi objek wisata di daerah Jabodetabek yang dinamis. Lokis memanfaatkan data dari berbagai sumber Google Maps melalui API TensorFlow dalam ranah Machine Learning, diproses dan memodelkan data dengan hati-hati, membangun model untuk mengevaluasi efisiensi rute yang dioptimalkan. Untuk memastikan rute-rute ini memenuhi harapan, mengkonsolidasikan data ke dalam dataset yang komprehensif dan menerapkan sistem rekomendasi untuk destinasi yang relevan. Integrasi Cloud Computing memfasilitasi pengembangan Model Sistem Rekomendasi Pengembangan Machine Learning dan Mobile Development, yang secara khusus menangani Rest Api yang memiliki peran penting dalam menjembatani dataset yang telah diolah oleh tim Machine Learning agar dapat digunakan oleh tim Mobile Development. Keberhasilan implementasi ini menunjukkan bahwa Node.Js adalah pilihan tepat untuk pengembangan Rest Api dalam aplikasi rekomendasi destinasi wisata, membuka peluang untuk integrasi dengan sumber data lain serta penambahan fitur rekomendasi, serta peningkatan performa dan skalabilitas sistem. Metode penelitian yang digunakan dalam pengembangan aplikasi ini adalah observasi langsung terhadap objek penelitian untuk mendapatkan pemahaman mendalam tentang fenomena yang sedang dipelajari.

Kata kunci: destinasi wisata, Teknologi Rest API, Node JS, Cloud Computing

Abstract - The advancement of information technology, especially in today's digital era, users want quick and easy access to travel information such as destinations, accommodation and transportation, tourist destination recommendation applications. Rest API technology with Node JS has proven effective in handling requests and providing fast responses, while the developed Rest Api provides a search function for tourist destinations based on city, category, rating and price. Lokis is an innovative mobile application designed to revolutionize the way tourists and locals explore tourist attractions in the dynamic Greater Jakarta area. Lokis leverages data from various Google Maps sources through the TensorFlow API in the machine learning domain, carefully processing and modeling the data, and building models to evaluate the efficiency of optimized routes. To ensure that these routes meet expectations, the data is consolidated into a comprehensive dataset and a recommendation system for relevant destinations is implemented. The integration of cloud computing has facilitated the development of the machine learning and mobile development recommendation system model, specifically addressing the Rest Api, which plays an important role in bridging the datasets processed by the machine learning team for use by the mobile development team. This successful implementation demonstrates that Node.Js is the right choice for Fire Rest development in tourism destination recommendation applications, opening up opportunities for integration with other data sources and the addition of recommendation features, as well as improving system performance and scalability. This research aims to help travelers find destinations that match their preferences and provide effective and user-friendly guidance for a satisfying vacation experience in the dynamic Jabodetabek region. The research method used in the development of this application is direct observation of the research object to gain an in-depth understanding of the phenomenon under study.

Keywords: Campus tourist destination, Rest API Technology, Node JS, Cloud Computing

I. PENDAHULUAN

Kebijakan Merdeka Belajar-Kampus Merdeka (MBKM) bertujuan untuk membantu mahasiswa menguasai keahlian yang relevan dengan dunia kerja. Salah satu program studi independen yang mendukung ini adalah Bangkit Academy, sebuah inisiatif dari Google yang didukung oleh Gojek, Tokopedia, Traveloka, dan Dicoding Indonesia. Program ini menawarkan pelatihan teknologi seperti Machine Learning, Mobile Development, dan Cloud Computing. Dalam program MBKM, mahasiswa menyelesaikan Capstone Project, proyek akhir yang mengharuskan mereka bekerja dalam tim untuk mengembangkan aplikasi. Tim penulis memilih mengembangkan aplikasi "Lokis," sebuah aplikasi mobile yang merekomendasikan destinasi wisata di Jabodetabek. Lokis menggunakan metode Traveling Salesman Problem untuk mengoptimalkan rencana perjalanan berdasarkan preferensi wisatawan, seperti jarak, tema, dan biaya. Dengan memanfaatkan data dari Google Maps dan *TensorFlow*, tim mengembangkan sistem rekomendasi berbasis Machine Learning. Cloud Computing digunakan untuk mengelola Rest API yang memungkinkan data digunakan oleh tim Mobile Development. Aplikasi ini diharapkan memudahkan wisatawan dalam menemukan destinasi yang sesuai dengan preferensi mereka, sekaligus memberikan pengalaman perencanaan perjalanan yang lebih efisien dan menyenangkan. Cloud Computing atau yang bisa di sebut komputasi awan adalah gabungan pemanfaatan teknologi komputer dan pengembangan berbasis internet. Cloud Computing merupakan penyimpanan data yang menjadikan internet sebagai pengirimnya. Terdapat 3 model pengiriman dalam cloud computing yaitu :

1. SAAS (*Software as a service*), merupakan layanan cloud dalam bentuk perangkat lunak.
2. PAAS (*Platform as a Service*), layanan cloud dalam bentuk platform yang dapat dimanfaatkan penggunaannya untuk membuat aplikasi di atasnya.
3. IAAS (*Infrastructure as a Service*), merupakan layanan cloud yang di bangun sesuai kebutuhan pengguna.(Dwiyatno dkk., 2021)

REST adalah kumpulan aturan desain yang mengatur bagaimana data ditransfer melalui antarmuka standar seperti HTTP. REST API adalah istilah untuk layanan web yang menggunakan arsitektur REST sebagai antarmuka pemrograman aplikasi (API).(Maulana dkk., 2024)

Node Js adalah lingkungan runtime Java Script yang open-source dan cross-Platform, sangat populer untuk berbagai jenis proyek. Node.Js berjalan pada mesin Java Script V8, sehingga memungkinkan eksekusi kode Java Script di luar browser web. Node.Js dirancang untuk membangun aplikasi jaringan yang scalable, menjadikannya sangat ideal untuk aplikasi jaringan real-time seperti server web, server game, server chatting, dan lainnya.(Setyo Nugroho dkk., 2023)

Express Js adalah framework web yang tangguh dan efisien yang berjalan di atas Node.Js. Dengan Express Js, pengembang dapat dengan cepat membangun API yang kuat dan dapat diandalkan. Node.Js sendiri merupakan Platform Java Script berbasis event-driven yang memungkinkan penanganan banyak koneksi secara efisien.(Musyaffa dkk., 2024)

Aplikasi rekomendasi destinasi wisata (Lokis) didesain untuk mempermudah pengguna dalam menyusun rencana perjalanan mereka dengan memberikan saran tempat wisata yang sesuai dengan preferensi individu. Dengan menggunakan teknologi Rest Api yang mengatur rute, Lokis dapat menyesuaikan rekomendasi berdasarkan minat pengguna. Sebagai tambahan, Lokis menyediakan rekomendasi destinasi wisata di daerah Jabodetabek, yang merupakan kawasan metropolitan yang kaya akan berbagai pilihan atraksi wisata menarik. Dengan informasi terkini dan akurat tentang destinasi wisata di Jabodetabek, Lokis membantu pengguna untuk menemukan tempat-tempat menarik untuk dikunjungi sesuai dengan keinginan mereka.

II METODOLOGI PENELITIAN

1. Teori Pendukung

a. Activity Diagram

Activity Diagram adalah salah satu jenis diagram UML yang digunakan untuk menggambarkan aliran proses bisnis atau aktivitas yang terjadi dalam suatu sistem. Activity Diagram dapat membantu pengembang perangkat lunak untuk memodelkan proses bisnis atau alur kerja sistem dengan lebih sistematis dan terstruktur.

b. Xampp

Xampp adalah web server open source yang berjalan pada sistem operasi cross-Platform (Windows, Linux, MacOS). Semua yang diperlukan untuk mengelola website tersedia di XAMPP seperti Apache, MySQL/MariaDB, PHP, dan Perl. Meski program di dalamnya lengkap, XAMPP tetap merupakan web server yang sederhana dan ringan.(Indah Purnama Sari dkk., 22M).

- c. Visual Studio Code
Visual studio Code di kembangkan oleh Microsoft untuk berbagai sistem operasi. Visual Studio Code (atau disingkat dengan VS Code) adalah kode editor yang dapat di ekstensi dan perangkat lunak ini adalah perangkat lunak open source di bawah lisensi MIT. (Ismail Setiawan, 2022)
- d. Google Cloud Platform
Google Cloud Platform (GCP) sebagai rangkaian layanan komputasi awan yang berjalan di infrastruktur yang sama dengan yang digunakan Google untuk produk-produknya seperti Google Search dan YouTube. Google Cloud Platform menawarkan berbagai layanan termasuk komputasi, penyimpanan, dan pengembangan aplikasi yang dapat dijalankan di infrastruktur Google yang sangat scalable dan andal. Google Cloud Platform memiliki kegunaan untuk membangun dan mengelola aplikasi web modern, memanfaatkan pembelajaran mesin, dan menganalisis set data besar. (Muharram & Fawziyyah, t.t.)

2 Metode Pengumpulan Data

Metode pengumpulan data adalah cara untuk mendapatkan informasi tentang sifat, keadaan, atau kegiatan tertentu yang diperlukan dalam mencapai tujuan penelitian.

- a. Observasi
Observasi dilakukan dengan mengamati langsung objek penelitian dan mencatat secara sistematis segala gejala yang terkait dengan masalah yang sedang diteliti. Dengan demikian, observasi memungkinkan peneliti untuk mendapatkan pemahaman yang mendalam tentang fenomena yang sedang dipelajari
- b. Wawancara
Wawancara merupakan sebuah teknik untuk mendapatkan data dari objek penelitian dengan cara mengajukan pertanyaan dan mendapatkan jawaban. Data tersebut kami dapatkan dari konsultasi antara tim penulis dengan advisor dan pembimbing selama menjalani Capstone Project.
- c. Studi Pustaka
Studi Pustaka merupakan suatu percakapan ditinjau dari buku referensi yang bertujuan untuk memperkuat materi percakapan dan sebagai alasan untuk melibatkan langkah-langkah tertentu dalam menguraikan dan merencanakan suatu kegiatan.

3. Tahapan Penelitian

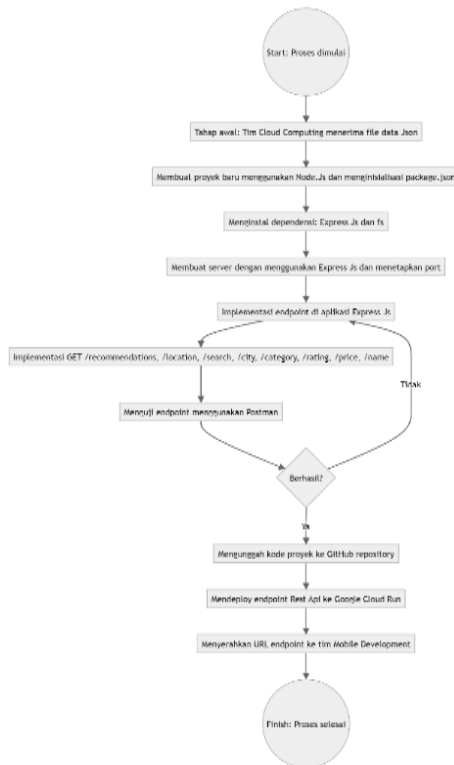
- a. Perencanaan syarat-syarat
- 1) Rekomendasi Destinasi Wisata berdasarkan Rating Kota: dapat mengakses rekomendasi destinasi wisata untuk beberapa kota (Jakarta,

Bogor, Depok, Tangerang, Bekasi) berdasarkan rating antara 4.4 dan 5 dengan melakukan permintaan GET ke endpoint

``/recommendations/cities/rating``.

- 2) Rekomendasi untuk Setiap Kota Secara Terpisah: dapat memperoleh rekomendasi destinasi wisata untuk setiap kota secara terpisah dengan melakukan permintaan GET ke endpoint seperti ``/recommendations/cities/Jakarta``, ``/recommendations/cities/Bogor``, dan seterusnya.
- 3) Data Lokasi Keseluruhan: dapat mengakses data lengkap tentang lokasi destinasi wisata dengan melakukan permintaan GET ke endpoint ``/location``.
- 4) Pencarian Data: dapat melakukan pencarian data berdasarkan nama, kota, atau kategori menggunakan endpoint ``/location/search``. Pencarian ini memanfaatkan parameter query string ``q`` untuk melakukan pencarian.
- 5) Filter Berdasarkan Kota: dapat memfilter data berdasarkan kota dengan melakukan permintaan GET ke endpoint ``/location/city=:city``.
- 6) Filter Berdasarkan Kategori: dapat memfilter data berdasarkan kategori dengan melakukan permintaan GET ke endpoint ``/location/category=:category``.
- 7) Filter Berdasarkan Rating: dapat memfilter data berdasarkan rating dengan melakukan permintaan GET ke endpoint ``/location/rating=:rating``.
- 8) Filter Berdasarkan Harga: dapat memfilter data berdasarkan harga dengan melakukan permintaan GET ke endpoint ``/location/price=:price``.
- 9) Filter Berdasarkan Nama: dapat memfilter data berdasarkan nama dengan melakukan permintaan GET ke endpoint ``/location/name=:name``.
- 10) Filter Kombinasi Kota dan Rating: dapat melakukan filter kombinasi berdasarkan kota dan rating dengan melakukan permintaan GET ke endpoint ``/location/city=:city&rating=:rating``.
- 11) Filter Kombinasi Kota dan Kategori: dapat melakukan filter kombinasi berdasarkan kota dan kategori dengan melakukan permintaan GET ke endpoint ``/location/city=:city&category=:category``.

b. Activity Diagram



Sumber : (Hasil Penelitian, 2024)
Gambar I. Activity Diagram

Berikut adalah penjelasan dari Activity Diagram di atas:

- 1) Start: Proses dimulai.
- 2) Cloud Computing menerima Data.json dari Tim machine learning: Tahap awal di mana Tim Cloud Computing menerima file data Json yang dihasilkan oleh model machine learning.
- 3) Inisialisasi Proyek: Membuat proyek Node.js dan menginisialisasi package.json: Membuat proyek baru.
- 4) menggunakan Node.js dan menginisialisasi `package.json` untuk manajemen dependensi.
- 5) Membuat Server: Membuat server menggunakan Express Js dan menetapkan port: Membuat server dengan menggunakan Express Js dan menetapkan port yang akan digunakan oleh server.
- 6) Implementasi Endpoints di Express Js: Implementasi endpoint di aplikasi Express Js untuk meng-handle berbagai permintaan HTTP.
- 7) Implementasi GET /recommendations, /location, /search, /city, /category, /rating, /price, /name: Membuat endpoint GET untuk berbagai rute seperti `/recommendations`, `/location`, `/search`, `/city`, `/category`, `/rating`, `/price`, dan `/name`.

- 8) Test endpoint di Postman: Menguji endpoint yang telah dibuat menggunakan Postman untuk memastikan semuanya bekerja dengan benar.
- 9) Berhasil?: Menentukan apakah pengujian endpoint berhasil atau tidak.
- 10) Ya: Jika pengujian berhasil, lanjutkan ke langkah berikutnya.
- 11) Tidak: Jika pengujian gagal, kembali ke tahap implementasi endpoint untuk memperbaiki kesalahan.
- 12) Push ke GitHub Repository: Mengunggah kode proyek ke GitHub repository untuk versi kontrol dan kolaborasi.
- 13) Deploy Rest Api Endpoint ke Google Cloud Run: Mendeploy endpoint Rest Api yang telah dibuat ke Google Cloud Run agar dapat diakses secara publik.
- 14) Serahkan URL Endpoint ke Mobile Development: Menyerahkan URL endpoint yang telah di deploy ke tim Mobile Development.
- 15) Finish: Proses selesai.

III. HASIL DAN PEMBAHASAN

Dalam pembuatan Rest Api ini, digunakan Visual Studio Code sebagai editor kode sumber, Node.Js sebagai lingkungan runtime untuk menjalankan Java Script di sisi server, Express Js sebagai framework untuk membangun aplikasi web dengan Node.Js, dan Postman sebagai alat yang digunakan untuk menguji dan mengirim permintaan ke Rest Api, sehingga memudahkan dalam pengujian endpoint dan memastikan bahwa Rest Api berfungsi dengan baik.

1. Mengimpor Modul dan Menginisialisasi Aplikasi

```
const Express = require('Express ');
const fs = require('fs');
const app = Express ();
const port = 8080;
app.use(Express.json());
```

Mengimpor modul *Express Js*, sebuah *framework* untuk membangun aplikasi web di *Node.Js*, dan modul file *system* bawaan *Node.Js* untuk membaca dan menulis file. Selanjutnya, membuat instance aplikasi *Express Js* dan menentukan port di mana *server* akan mendengarkan permintaan. Middleware *Express.json()* digunakan untuk mengurai *JSON* dalam *body request*.

2. Endpoint Rekomendasi Berdasarkan Rating di Kota-kota Tertentu

```
app.get('/recommendations/cities/rating', (req, res) => {
  fs.readFile('data.json', 'utf8', (err, data) => {
    if (err) {
```

```

    console.error(err);
    return res.status(500).send('Terjadi
kesalahan saat membaca file');
  }
  const jsonData = JSON.parse(data);
  const cities = ['Jakarta', 'Bogor', 'Depok',
'Tangerang', 'Bekasi'];
  const recommendationsByCityAndRating
= cities.map(city => {
    const rating4_5InCity = jsonData
    .filter(item => item.city.toLowerCase()
=== city.toLowerCase() && item.rating >= 4.4
&& item.rating <= 5)
    .map(({ id, name, rating }) => ({ id,
name, rating }));
    return {
      res.json(recommendationsByCityAndRati
ng);
    }
  });
});

```

Endpoint GET /recommendations/cities/rating membaca file data.json, mengurai data JSON, dan kemudian mencari tempat-tempat di beberapa kota (Jakarta, Bogor, Depok, Tangerang, Bekasi) dengan rating antara 4.4 dan 5. Hasilnya dikirimkan sebagai respons Json.

3. *Endpoint* Rekomendasi Berdasarkan Kota Tertentu

```

app.get('/recommendations/cities/Jakarta',
(req, res) => {
  getRecommendationsByCity(req, res,
'Jakarta');
});
app.get('/recommendations/cities/Bogor', (req,
res) => {
  getRecommendationsByCity(req, res,
'Bogor');
});

```

Endpoint di atas masing-masing untuk kota Jakarta, Bogor, Depok, Tangerang, dan Bekasi. Semuanya menggunakan fungsi getRecommendationsByCity.

4. Fungsi untuk Mendapatkan Rekomendasi Berdasarkan Kota

```

function getRecommendationsByCity(req, res,
city) {
  fs.readFile('data.json', 'utf8', (err, data) => {
    if (err) {
      console.error(err);
      return res.status(500).send('Terjadi
kesalahan saat membacafile');
    }
    const jsonData = JSON.parse(data);
    const recommendationsForCity = jsonData
    .filter(item => item.city.toLowerCase()
=== city.toLowerCase() && item.rating >= 4.4
&& item.rating <= 5)
    .map(({ id, name, rating }) => ({ id, name,
rating }));
    res.json({

```

```

    city,
    recommendations:
recommendationsForCity
  });
});
}

```

Fungsi getRecommendationsByCity membuka file data.json, mengurai data JSON, dan kemudian mencari tempat-tempat di kota tertentu dengan rating antara 4.4 dan 5. Hasilnya dikirimkan sebagai respons JSON.

5. *Endpoint* untuk Mendapatkan Semua Data

```

app.get('/location', (req, res) => {
  fs.readFile('data.json', 'utf8', (err, data) => {
    if (err) {
      console.error(err);
      return res.status(500).send('Terjadi
kesalahan saat membaca file');
    }
    const jsonData = JSON.parse(data);
    res.json(jsonData);
  });
});

```

Endpoint GET /location membaca dan mengurai file data.json dan mengirimkan semua data sebagai respons JSON.

6. *Endpoint* untuk Mencari Data Berdasarkan Query String

```

app.get('/location/search', (req, res) => {
  fs.readFile('data.json', 'utf8', (err, data) => {
    if (err) {
      console.error(err);
      return res.status(500).send('Terjadi
kesalahan saat membaca file');
    }
    const jsonData = JSON.parse(data);
    const searchTerm = req.query.q;
    if (!searchTerm) {
      return res.status(400).json({ error:
'Query parameter "q" is required for search.'
});
    }
    const searchData = jsonData.filter(item =>
item.name.toLowerCase().includes(sea
rchTerm.toLowerCase()) ||
item.city.toLowerCase().includes(searc
hTerm.toLowerCase()) ||
item.category.toLowerCase().includes(
searchTerm.toLowerCase())
);
    if (searchData.length === 0) {
      return res.status(404).json({ message:
'Data tidak ditemukan' });
    }
    res.json(searchData);
  });
});

```

Endpoint GET /location/search membaca file data.json, mengurai data JSON, dan melakukan pencarian berdasarkan query string

q untuk nama, kota, atau kategori. Hasil pencarian dikirimkan sebagai respons JSON.

7. Endpoint untuk Mendapatkan Data Berdasarkan Parameter Dinamis

```
app.get('/location/city=:city', (req, res) => {
  fs.readFile('data.json', 'utf8', (err, data) => {
    if (err) {
      console.error(err);
      return res.status(500).send('Terjadi kesalahan saat membaca file');
    }
    const jsonData = JSON.parse(data);
    const cityData = jsonData.filter(item =>
      item.city.toLowerCase() ===
      req.params.city.toLowerCase());
    if (cityData.length === 0) {
      return res.status(404).send('Data tidak ditemukan');
    }
  })
})
```

Endpoint-endpoint di atas menggunakan parameter dinamis (:city, :category, :rating, :price, :name) untuk mencari dan mengirimkan data yang sesuai dari data.json.

8. Endpoint untuk Mendapatkan Data Berdasarkan Kombinasi Parameter

```
app.get('/location/city=:city&rating=:rating',
  (req, res) => {
    fs.readFile('data.json', 'utf8', (err, data) => {
      if (err) {
        console.error(err);
        return res.status(500).send('Terjadi kesalahan saat membaca file');
      }
      const jsonData = JSON.parse(data);
      const cityDistrictData = jsonData.filter(item
        => item.city.toLowerCase() ===
        decodeURIComponent(req.params.city) &&
        item.rating === Number(req.params.rating));
      if (cityDistrictData.length === 0) {
        return res.status(404).send('Data tidak ditemukan');
      }
      res.json(cityDistrictData);
    });
  });
```

Endpoint-endpoint ini menggunakan kombinasi parameter (:city&rating dan :city&category) untuk mencari dan mengirimkan data yang sesuai dari data.json.

9. Memulai Server

```
app.listen(port, () => {
  console.log('Example app listening on port
  ${port}');
});
```

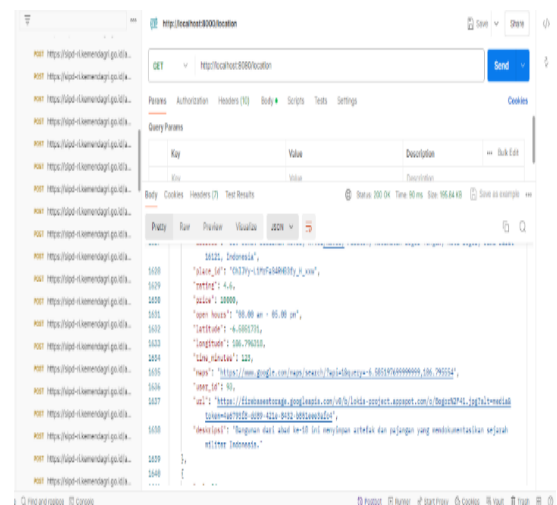
Menjalankan server *Express Js* pada port yang telah ditentukan (8080). Pesan log akan ditampilkan saat server mulai berjalan.

Kode diatas mendefinisikan sebuah server Api menggunakan *Express Js* yang menyediakan

berbagai *endpoint* untuk mendapatkan data tempat berdasarkan beberapa parameter seperti kota, kategori, rating, harga, dan nama. Data dibaca dari file data.json dan hasilnya difilter sesuai dengan parameter yang diberikan sebelum dikirimkan kembali sebagai respons dalam format Json

10. Postman Test

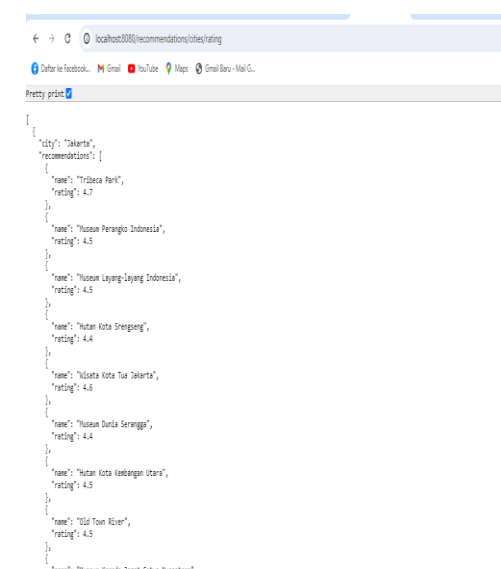
Dari gambar dibawah menunjukan Test Postman memberikan respons dengan status HTTP 200 OK. Status 200 OK menunjukkan bahwa permintaan HTTP berhasil diproses oleh server dan respons yang diharapkan dikembalikan dengan sukses. Dengan demikian, pengujian ini memverifikasi bahwa *endpoint* API berfungsi dengan baik dan dapat menangani permintaan dengan benar.



Sumber : (Hasil Penelitian, 2023)

Gambar 2. Test Postman

11. Hasil Endpoint Rest Api



Sumber : (Hasil Penelitian, 2023)

Gambar 3 Hasil Endpoint Rest Api

Penjelasan dari Endpoint Rest Api di atas: Endpoint ini mengambil rekomendasi tempat berdasarkan kota tertentu yang memiliki rating antara 4.4 dan 5. Kota yang didukung adalah Jakarta, Bogor, Depok, Tangerang, dan Bekasi. Contoh : /recommendations/cities/Jakarta.

IV. KESIMPULAN

Dalam penelitian ini, teknologi Rest Api dengan Node.Js berhasil digunakan untuk mengembangkan aplikasi rekomendasi destinasi wisata. Node.Js, sebagai Platform server-side, terbukti efektif dalam menangani permintaan dan memberikan respons yang cepat. REST API yang dikembangkan menyediakan fitur pencarian destinasi wisata berdasarkan kota, kategori, rating, dan harga. Implementasi ini mampu menangani data dalam jumlah besar dengan performa yang baik serta fleksibilitas untuk penambahan dan pembaruan fitur di masa depan.

Keberhasilan ini menunjukkan bahwa Node.Js adalah pilihan tepat untuk pengembangan REST API dalam aplikasi rekomendasi destinasi wisata. Ini membuka peluang untuk integrasi dengan sumber data lain, penambahan fitur rekomendasi berbasis machine learning, serta peningkatan performa dan skalabilitas sistem.

Selain itu, mengikuti program studi independen path learning Cloud Computing di Bangkit Academy 2023. bersama Dicoding Academy memberikan penulis wawasan dan keterampilan tambahan dalam bidang komputasi awan. Program ini memberikan pemahaman mendalam tentang teknologi Cloud Computing, termasuk konsep dasar, arsitektur, dan teknologi terkini dalam industri. Program studi independen ini mendorong kemandirian belajar, kemampuan inovasi, dan membantu peserta mengasah kemampuan inovatif mereka untuk menghadapi perubahan dan tantangan di masa depan.

Peserta juga dapat terhubung dengan komunitas yang memiliki minat dan tujuan yang sama melalui forum online, grup diskusi, atau acara terkait untuk berbagi pengetahuan dan pengalaman. Keterampilan dan pengetahuan yang diperoleh dari program ini akan meningkatkan prospek karier peserta di industri teknologi informasi, terutama di bidang Cloud Computing yang terus berkembang pesat. Peserta menjadi lebih kompetitif dalam mencari pekerjaan atau memajukan karier mereka, serta membangun kemandirian profesional yang berharga.

V. REFERENSI

Arizal (2023), "Belajar Menggunakan Fetch dan Axios untuk REST API di JavaScript: Panduan untuk Pemula yang Mudah Dipahami.

<https://www.ngulikode.com/2023/02/panduan-pemula-untuk-menggunakan-fetch.html?m=1>(

D. Westerveld (2021), *API Testing and Development with Postman*. Birmingham: Packt Publishing.

Dwiyatno, S., Rakhmat, E., & Christina, S. (2021). *PERANCANGAN PRIVATE CLOUD BERBASIS INFRASTRUCTURE AS A SERVICE*. 8(2).

H. Fajar, S. El Yamin, and M. Siregar (2019), "Desain Komunikasi Data Digital Pada Radio HF Dengan Metode Frequency Shift Keying Berbasis Thonny Python," *JTIM J. Teknol. Inf. dan Multimed.*, vol. 1, no. 3, pp. 181–187

Indah Purnama Sari, P., Syahputra, A., Zaky, N., Sibuea, R. U., & Zakhir, Z. (22M). *Perancangan Sistem Aplikasi Penjualan dan Layanan Jasa Laundry Sepatu Berbasis Website*. <https://doi.org/https://creativecommons.org/licenses/by-sa/4.0/deed.id>

Ismail Setiawan. (2022). *Komparasi Kinerja Integrated Development Environment (IDE) Dalam Mengeksekusi Perintah Python*. *SATESI: Jurnal Sains Teknologi dan Sistem Informasi*, 2(1), 52–59. <https://doi.org/10.54259/satesi.v2i1.784>

Liu, C., Yu, X., & Zhou, Z. (2014). The effectiveness of black-box testing techniques for Java applications. *Software Quality Journal*, 22(3), 593-614.

Maulana, A., Riady, N., Paniran, P., Made, I., Suksmadana, B., Majapahit, J., Nusa, L., & Barat, T. (2024). *Perancangan Backend Api Berbasis Rest-Api pada Aplikasi Rekomendasi Resep Makanan (Rest-Api Based Backend Api Design In Food Recipe Recommendation Application)*. 2(3). <https://doi.org/10.61132/mars.v2i3.137>

Muharram, A. T., & Fawziyyah, H. (t.t.). *Sistem Informasi Geografis Lokasi Sebaran Penerima Bantuan Program Keluarga Harapan di Kota Depok Berbasis Android* (Vol. 4, Nomor 2).

Musyaffa, A. I., Mulki Indana Zulfa, & Muhammad Syaiful Alim. (2024). *RANCANG BANGUN PURECOMPUTE PLATFORM E-COMMERCE UNTUK BELANJA LAPTOP BERBASIS*

WEBSITE. *Jurnal SINTA: Sistem Informasi dan Teknologi Komputasi*, 1(1), 21–29.

<https://doi.org/10.61124/sinta.v1i1.9>

Ramadhan, Muhammad & Zuhri, Zainudin. (2023). "Pengembangan Rest Api Sistem Uii Admisi Dengan Menggunakan Pendekatan Domain Driven Design". *Jurnal Ilmiah Informatika*. 11. 176-182.

S. Ahmed and Q. Mahmood (2019), "An authentication based scheme for applications using JSON web token," 2019 22nd International Multitopic Conference (INMIC), Islamabad, Pakistan, pp.1-6.

Setyo Nugroho, D., Anggara Putra, A., Luqmanulhakim, H., Farras Pratama, N., Alam Saputra, B., Alit, R., & Negeri Suarabaya, U. (2023). Inovasi Pembaruan Desain Website Penyederhana Link Unesa Menggunakan Typescript Dan Node.js. *Jurnal Ilmu Teknik*, 1(2), 35–40.

Sintiawati, N., Fajarwati, S. R., Mulyanto, A., Muttaqien, K., & Suherman, M. (2022). Partisipasi Civitas Akademik dalam Implementasi Merdeka Belajar Kampus Merdeka (MBKM). *Jurnal Basicedu*, 6(1), 902–915.

<https://doi.org/10.31004/basicedu.v6i1.2036>

T. Bratakusuma, I. U. Azmi, and S. Ayuningtiyas (2022), "Pengembangan Back End Pada Aplikasi Alat Tulis Kantor Bank Indonesia Perwakilan Purwokerto Menggunakan Nodejs Back End Development on Stationary Applications Bank Indonesia Representative Office Purwokerto Using Nodejs,"