

Rancangan Pembuatan Api Website Data Tanaman Obat Dan Langka Kabupaten Kediri

Shandy Arshad Busro Cahyono, Sucipto, Rina Firliana, M Najibuloh Muzzaki, Anita Sari Wardani, M Iqbal

Khalid, Adhi Wicak Milbar Gamas, Heru Setiawan

Fakultas Teknik, Sistem Informasi, Universitas Nusantara PGRI Kediri, Kediri, Indonesia

Email: [1shandyabc313@gmail.com](mailto:shandyabc313@gmail.com), [2sucipto@unpkediri.ac.id](mailto:sucipto@unpkediri.ac.id), [3rina@unpkediri.ac.id](mailto:rina@unpkediri.ac.id),

[4m.n.muzaki@gmail.com](mailto:m.n.muzaki@gmail.com), [5anita@unpkediri.ac.id](mailto:anita@unpkediri.ac.id), [6muhammadiqbal30933@gmail.com](mailto:muhammadiqbal30933@gmail.com),

[7adhiwicak0@gmail.com](mailto:adhiwicak0@gmail.com), [8herustiawan32@gmail.com](mailto:herustiawan32@gmail.com)

Email Penulis Korespondensi: shandyabc313@gmail.com

Abstrak– Rancangan pembuatan website informasi tanaman obat dan langka Kabupaten Kediri bertujuan memberikan kemudahan dalam mendapatkan informasi dengan mudah akses dan cepat. Perancangan website ini menggunakan metode *Software Development Life Cycle (SLDC) Waterfall* dengan tahapan identifikasi, analisis, perancangan, implementasi dan testing. Website berbasis *Application Programming Interface (API)* menggunakan Node.js sebagai *runtime environment* dari bahasa pemrograman JavaScript dengan menggunakan *framework* Express.js. *Application Programming Interface (API)* digunakan sebagai penghubung antara client dan server, pengujian *Application Programming Interface (API)* endpoint menggunakan tools Postman dengan menampilkan hasil dari pengujian berupa data dengan format JSON (*JavaScript Object Notation*) yang digunakan untuk menyimpan dan mentransfer data.

Kata Kunci: Website; Node.js; API; JSON

Abstract– The design of a website for information on medicinal and rare plants in Kediri Regency aims to provide convenience in obtaining information easily and quickly. The design of this website uses the Software Development Life Cycle (SLDC) Waterfall method with the stages of identification, analysis, design, implementation and testing. The web-based Application Programming Interface (API) uses Node.js as a runtime environment for the JavaScript programming language using the Express.js framework. The Application Programming Interface (API) is used as a liaison between the client and server, testing the endpoint Application Programming Interface (API) using Postman tools by displaying the results of the test in the form of data in JSON (*JavaScript Object Notation*) format which is used to store and transfer data.

Keywords: Website; Node.js; API; JSON

1. PENDAHULUAN

Wilayah Kabupaten Kediri secara geografis terletak pada koordinat antara 7°36'12"-8°0'32" Lintang Selatan dan 111°47'05"-112°18'20" Bujur Timur. Kabupaten Kediri memiliki luas wilayah sebesar 1.386,05 Km² atau 138.605 Ha yang terbagi menjadi 26 kecamatan antara lain yaitu Kecamatan Mojo, Semen, Ngadiluwih, Kras, Ringinrejo, Kandat, Wates, Ngancar, Plosoklaten, Gurah, Puncu, Kepung, Kandangan, Pare, Badas, Kunjang, Plemahan, Purwoasri, Papar, Pagu, Kayenkidul, Gampengrejo, Ngasem, Banyakan, Grogol, dan Tarokan[1]. Dengan kondisi geografis tersebut banyak tanaman obat yang tumbuh secara liar atau dibudidayakan di berbagai daerah Kabupaten Kediri, seperti di daerah Kecamatan Semen yang sudah dilakukan gerakan budidaya tanaman obat[2].

Data mengenai tanaman obat di Kabupaten Kediri belum tersedia secara masal masih data tanaman obat berupa laporan tentang tanaman obat yang ada di daerah-daerah yang ada pada kabupaten Kediri. Hal ini menyebabkan informasi tentang tanaman obat tersebut belum dapat diketahui oleh masyarakat secara umum[3]. Dengan adanya keberadaan teknologi yang telah menjadi kebutuhan bagi kehidupan manusia dalam mendapatkan suatu informasi yang ditandai dengan berkembang pesatnya teknologi informasi dalam berbagai bidang[4]. Keberadaan teknologi internet yang memberi kemudahan akses informasi antar perangkat dan media dalam mendapatkan informasi[5]. Teknologi banyak digunakan sebagai sarana informasi khususnya pada bidang website yang saat ini sangat berperan dalam penyampaian informasi. Website mampu memberikan informasi menjadi lebih efisien. Website lebih mudah diakses oleh masyarakat di berbagai daerah hanya dengan menggunakan internet[6].

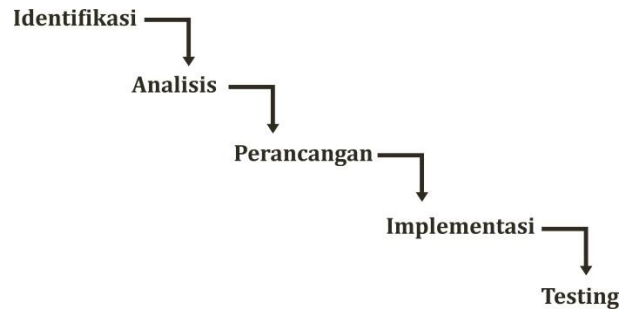
Website sebuah kumpulan halaman pada suatu domain di internet yang di buat dengan tujuan tertentu dan saling berhubungan serta dapat di akses secara luas[7]. Pengembangan layanan web menggunakan RESTful API sebagai protocol standar untuk mengkomunikasikan data yang memiliki banyak keuntungan[8]. Penggunaan Application Programming Interface (API) dapat lebih mudah bagi pengguna dalam mengakses sebuah data dengan melakukan request[9]. JSON (*JavaScript Object Notation*) sebagai media pertukaran data yang ringan saat pengguna melakukan request melalui API[10]. Penulisan yang sederhana dan berkonsekuensi pada ukuran file yang dihasilkan. Ukuran file yang dihasilkan oleh JSON sangat kecil, sehingga dapat diakses dengan waktu yang relatif cepat[11].

Dari penjelasan diatas yang mendasari membuat rancangan RESTful API untuk rancangan pembuatan website tanaman obat kabupaten Kediri menggunakan Node.js dengan framework Express.js dan hasil tampilan data tanaman obat dapat diketahui dalam format data berupa JSON (*JavaScript Object Notation*) menggunakan tools Postman sebagai pengujian API

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dilakukan menggunakan metode *Software Development Life Cycle (SLDC) Waterfall* dengan melalui beberapa tahap. Adapun tahapan penelitian dapat diuraikan sebagai berikut :



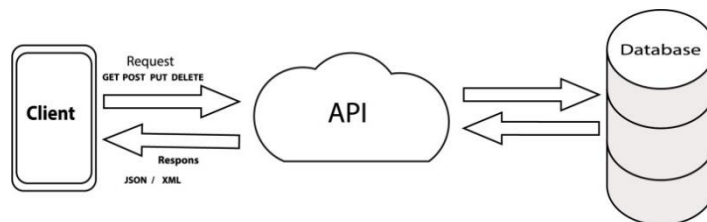
Gambar 1. Tahapan Penelitian

1. Identifikasi
Identifikasi dimana peneliti mencari segala informasi yang dibutuhkan dalam rancangan pembuatan API dan menentukan tools apa saja yang digunakan .
2. Analisis
Pada tahap ini menganalisis dari kumpulan data yang berhubungan dengan data apa saja yang nanti akan di tampilkan melalui design Figma yang sudah disediakan.
3. Perancangan
Pada tahap ini membuat rancangan arsitektur system ,dengan menggunakan bahasa pemrograman JavaScript,Node.JS dan framework Express.js disisi bagian server untuk merancang API,data yang digunakan adalah data dummy,pengujian API menggunakan tools Postman dengan tampilan format data JSON.
4. 4. Implentasi
Pada tahap ini hasil dari rancangan arsitektur akan di implementasikan menjadi kode-kode program untuk pembuatan API dengan bantuan perangkat keras dan perangkat lunak yaitu satu unit laptop dengan spesifikasi processor AMD E-350, RAM DDR3 berkapasitas 8GB, SSD dengan kapasitas 120GB, dalam pengkode an menggunakan perangkat lunak Visual Studio Code, Web Browser menggunakan Chrome, pengujian API menggunakan Postman.
5. 5. Testing / Ujicoba
Pada tahap testing akan dilakukan pengujian endpoint API menggunakan bantuan perangkat lunak Postman. Dengan dilakukannya testing ini supaya *endpoint* API sesuai atau belum dengan rancangan yang sudah dibuat, hasil dari testing akan ditampilkan dalam bentuk format data JSON

3. HASIL DAN PEMBAHASAN

3.1 Cara Kerja API

Dari hasil tahapan metode penelitian didapatkan rancangan API web serta hasil analisis dari design Figma apa saja metode API *endpoint* yang harus dibuat untuk digunakan dalam mengakses layanan sehingga dapat berinteraksi dengan berbagai data. Gambar secara umum alur kerja *Application Programming Interface (API)*. Dibawah ini gambar alur kerja *Application Programming Interface (API)*.

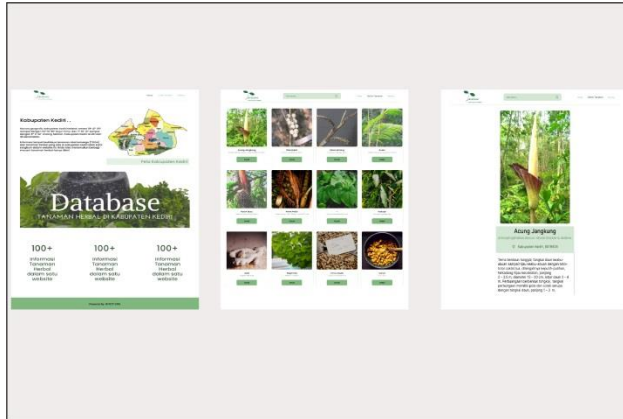


Gambar 2. Cara Kerja API

Pada gambar 2 dimana client akan berinteraksi dengan API menggunakan metode REST ,client akan melakukan permintaan menggunakan metode GET, POST, PUT atau DELETE melalui *Hypertext Transfer Protocol (HTTP)* ke server untuk mendapatkan data atau mengirim data ke database.

3.2 Analisis

Dalam rancangan pembuatan API yang akan di implementasikan ke aplikasi website data tanaman obat dan langka di kabupaten Kediri,yang dilakukan pertama adalah menganalisis API apa saja yang harus dibuat dengan bantuan melalui gambar design yang terdapat di Figma yang sudah disiapkan. Melalui gambar design tersebut mempermudah dalam menentukan API apa yang dibuat dan data apa saja yang perlu disediakan pada masing-masing API.

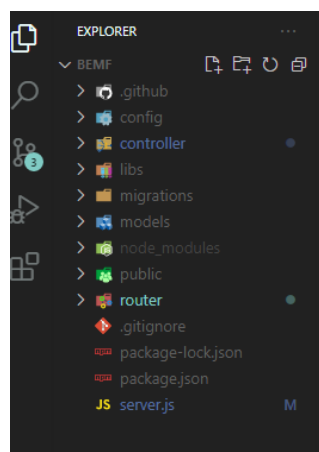


Gambar 3. Tampilan Design Website

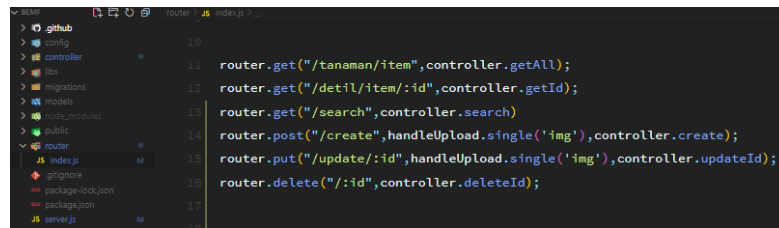
Pada gambar 3 merupakan desain tampilan website mode desktop jika diamati dari tampilan tersebut, API yang perlu dibuat yaitu endpoint API GET semua daftar tanaman untuk mendapatkan semua data tanaman, kemudian pada bagian daftar tanaman per gambar tanaman jika di klik button detil maka akan menuju ke tampilan yang memuat deskripsi detil dari tanaman tersebut maka diperlukan nya endpoint API GET byId/url untuk mendapatkan informasi detil per item tanaman dan *endpoint* API selanjutnya yang harus dibuat yaitu *endpoint* API pada form pencarian tanaman dengan menggunakan metode GET, sehingga didapatkan tiga *endpoint* API yang harus dibuat dengan menggunakan metode GET.

3.3 Implementasi

Pertama yang dilakukan menginisialisasi proyek terlebih dahulu kemudian menginstall library yang dibutuhkan,membuat struktur model,controller,router,libs,public. Dimana struktur model digunakan untuk merepresentasikan kolom table yang berhubungan dengan database,controller berisi kode program untuk mengimplementasikan logika dan mengirim respons,router digunakan untuk mendefinisikan routing program,didalam libs digunakan untuk menghandle upload gambar menggunakan multer serta folder public tempat penyimpanan gambar.



Gambar 4. Struktur Dasar Pengkodean

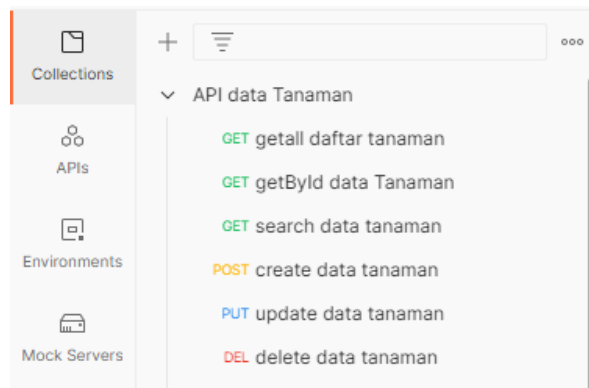


Gambar 5. Routing

Gambar 5 menampilkan *routing* yang akan diakses untuk berinteraksi dengan data melalui permintaan metode API GET, untuk menampilkan semua daftar data tanaman pada `router.get("/tanaman/item",controller.getAll)`, detail tanaman `router.get("/detil/item/:id",controller.getId)`, dan untuk form pencarian `router.get("/search",controller.search)`.

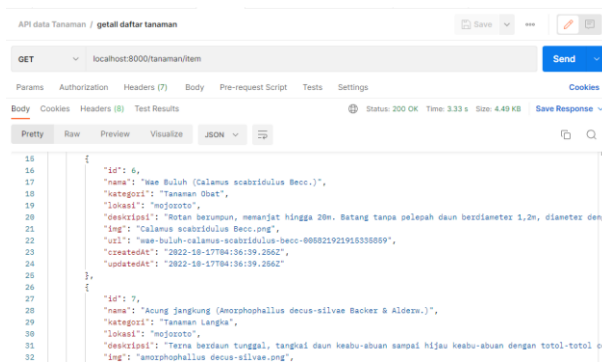
3.4 Testing

Setelah tahapan pengkodean dan membuat *endpoint* API selesai langkah selanjutnya yang harus dilakukan yaitu menguji coba masing-masing *endpoint* API yang bertujuan untuk memastikan API tersebut berjalan dengan baik dan sesuai yang diharapkan dengan menggunakan bantuan *tools* Postman. Postman memberikan fitur *collection* API untuk memberikan kemudahan kepada pengguna agar API tidak tercampur dengan API proyek lain seperti pada tampilan gambar lima, disini membuat folder API yang khusus untuk data Tanaman dengan berbagi metode API yang dibutuhkan di proyek. Dalam pengujian *endpoint* API metode GET hasil yang dikeluarkan dalam bentuk format data JSON.



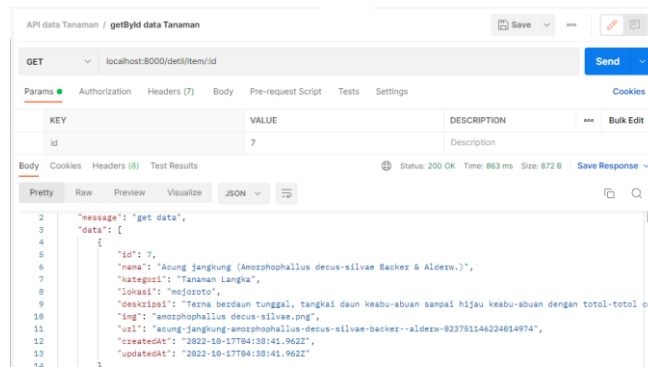
Gambar 6. Api Collection

Gambar 7 adalah tampilan hasil dari pengujian *endpoint* API untuk mendapatkan semua data tanaman yang ada pada database.



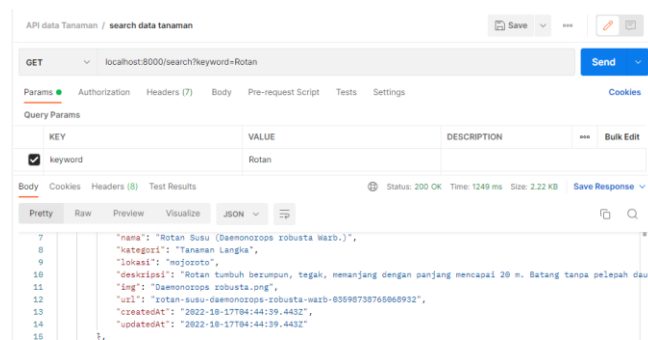
Gambar 7. GET All Data Tanaman

Gambar 8 hasil pengujian *endpoint* API GET untuk mendapatkan data satu tanaman ById dengan keluran data format JSON



Gambar 8. GET Data ById

Gambar 9 Hasil pengujian *endpoint* API search menggunakan Postman dengan keluaran data format JSON



Gambar 9. Search Data Tanaman

4. KESIMPULAN

Kesimpulan yang didapatkan bahwa API sangat membantu dalam pengembangan pembuatan website yang dijadikan sebagai penghubung antara client ke server melalui HTTP. Dimana pada penelitian ini didapatkan tiga API dengan metode Get pada rancangan pembuatan website informasi data tanaman obat dan langka di Kabupaten Kediri, API tersebut : (1) Get semua data tanaman, (2) Get tanaman ById dan (3) untuk pencarian tanaman. Tiap API di ujicoba melalui bantuan tools Postman yang mana hasil yang di tampilkan berupa data dalam format JSON.

UCAPAN TERIMA KASIH

Terimakasih disampaikan kepada seluruh pihak yang telah mendukung terlaksananya penelitian dan membantu dalam penyusunan penelitian ini mulai dari awal hingga selesai.

REFERENCES

- [1] Pradana, A. S. P. (2019). Sistem Informasi Geografis Penggunaan Lahan Dan Produksi Tanaman Pangan Kabupaten Kediri Jawa Timur. JATI (Jurnal Mahasiswa Teknik Informatika), 3(2), 9-15.
- [2] Suciyo, S., Sulistiyowati, T. I., Utami, B., & Qamaria, R. S. (2020). Gerakan Budidaya Tanaman Obat Keluarga sebagai Kepedulian Masyarakat Menuju Desa Sehat di Desa Semen Kabupaten Kediri. Cendekia: Jurnal Pengabdian Masyarakat, 2(2), 85-94.
- [3] Latief, M., Kandowanko, N., & Yusuf, R. (2017). Metode Rational Unified Process untuk Pengembangan Aplikasi Web dan Mobile (Studi Kasus Sistem Informasi Tanaman Obat Daerah Gorontalo). Jurnal Rekayasa Elektrika, 13(3), 152-160.
- [4] Arianto, M. A. (2016). Analisis dan Perancangan Representational State Transfer (REST) Web Service Sistem Informasi Akademik STT Terpadu Nurul Fikri Menggunakan Yii Framework. Jurnal Teknologi Terpadu, 2(2).
- [5] Dwi, W. (2018). Pengembangan Antarmuka Pemrograman Aplikasi Menggunakan Metode RESTful pada Sistem Informasi Akademik Politeknik Kota Malang. Smatika Jurnal, 8(02), 63-66.
- [6] Hasugian, P. S. (2018). Perancangan website sebagai media promosi dan informasi. Journal Of Informatic Pelita Nusantara, 3(1).
- [7] Pradigi, C. H., & Harlina, T. (2019). IMPLEMENTASI EXPRESS JS UNTUK MEMBANGUN REST API WEBSITE STIKOM PGRI BANYUWANGI. Jikom: Jurnal Informatika dan Komputer, 9(2), 118-122.
- [8] Manuaba, I. B. P., & Rudiastini, E. (2018). API REST web service and backend system of lecturer's assessment information system on politeknik negeri bali. In Journal of Physics: Conference Series (Vol. 953, No. 1, p. 012069). IOP Publishing.

- [9] Baharuddin, B., Wakkang, H., & Irianto, B. (2022). IMPLEMENTASI WEB SERVICE DENGAN METODE REST API UNTUK INTEGRASI DATA COVID 19 DI SULAWESI SELATAN. *Jurnal Sintaks Logika*, 2(1), 236-241.
- [10] Sahrial, R., Fauzi, D. F., & Susilawati, E. (2022). PEMANFAATAN JSON UNTUK MENAMPILKAN DATA REALTIME COVID-19 DENGAN MODEL VIEW PRESENTER. *Jurnal Teknoinfo*, 16(1), 144-149.
- [11] Sahlinal, D., Maulini, R., & Widyawati, D. K. (2018). Sistem Informasi Pemasaran Hasil Pertanian Polinela Berbasis JSON. *Jurnal SIMADA (Sistem Informasi dan Manajemen Basis Data)*, 1(1), 11-20.
- [12] E. Purba, A. Karim, and S. Trianovie, "Sistem Informasi Pendataan Usaha Micro Kecil dan Menengah Pada Dinas Umkm Labuhan Batu Berbasis web," *J. Infotek*, vol. 4, no. 3, pp. 1–8, 2019.