



Implementasi *Back-End* Sistem Otomasi Pendataan Mahasiswa pada Food Truck Undip Menggunakan *Framework Express.js*

Back-End Implementation of Student Data Collection Automation Systems at Undip Food Trucks Using the Express.js Framework

Muhammad Alifa Ramdhan^{*)}, Adnan Fauzi, Arseto Satriyo Nugroho

Program Studi Teknik Komputer, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Soedarto, SH, Kampus Undip Tembalang, Semarang, Indonesia 50275

How to cite: M. A. Ramdhan, A. Fauzi, and A. S. Nugroho, "Implementasi Back-end Sistem Otomasi Pendataan Mahasiswa pada Food Truck Undip Menggunakan Framework Express.js," *Jurnal Teknik Komputer*, vol. 2, no. 2, pp. 115-122, May 2023. doi: 10.14710/jtk.v2i2.38527 [Online].

Abstract – *Food Truck Undip is a solution offered by Undip to improve the nutrition of students by providing nutritious food assistance. However, the data collection of the students who receive the food aid is still done manually, which is inefficient and can cause errors and difficulties in managing the data. This research aims to create an automated data collection system using a Student ID Card scanner and a website. The research develops a REST API with the Express framework and MongoDB database, which will be a bridge between the website and the Student ID Card scanner at the Food Truck location. The system testing results show that the system can run well and meet the planned requirements..*

Keywords – REST API, Express, MongoDB

Abstrak – *Food Truck Undip adalah solusi yang ditawarkan Undip untuk memperbaiki gizi mahasiswa dengan memberikan bantuan makanan bergizi. Namun, pendataan mahasiswa penerima bantuan masih dilakukan secara manual yang tidak efisien dan dapat menimbulkan kesalahan serta kesulitan pengelolaan data. Penelitian ini bertujuan untuk membuat sistem otomasi pendataan dengan alat pemindai KTM dan situs web. Pada Penelitian ini REST API dikembangkan dengan framework Express dan basis data MongoDB, yang akan menjadi jembatan antara situs web dan alat pemindai KTM di lokasi Food Truck. Hasil pengujian sistem menunjukkan bahwa sistem dapat berjalan dengan baik dan sesuai dengan kebutuhan yang telah direncanakan.*

Kata kunci – REST API, Express, MongoDB

I. PENDAHULUAN

Food Truck Undip merupakan salah satu solusi yang ditawarkan oleh Undip untuk memperbaiki gizi mahasiswa Undip. Food Truck Undip memberi bantuan berupa makanan bergizi kepada mahasiswa. Agar

^{*)} Penulis Korespondensi (M. A. Ramdhan)
Email: n0psledbyte@students.undip.ac.id

penerimaan bantuan makanan oleh mahasiswa dapat merata, maka dilakukan pendataan mahasiswa yang menerima bantuan tersebut.

Pada pelaksanaannya, pendataan mahasiswa penerima bantuan makanan masih dilakukan secara manual, sehingga validasi apakah mahasiswa sudah mengambil bantuan makanan di hari tersebut menjadi sulit dan tidak kredibel. Hal tersebut dapat menimbulkan ketidakteraturan dan pembagian makanan menjadi tidak merata. Selain itu, pendataan secara manual juga memakan waktu yang tidak sedikit sehingga sering kali menyebabkan antrean yang Panjang. Masukkan data secara manual juga dapat menyebabkan terjadinya kesalahan saat memasukkan data. Pengelolaan data juga menjadi sulit karena penyimpanan data yang tidak terpusat dan data harus digabungkan secara manual.

Untuk mengatasi permasalahan tersebut, dibuatlah sistem baru yang menggunakan alat pemindai berbasis Raspberry Pi yang terintegrasi dengan server basis data dan situs web untuk membaca kode QR (*Quick Response*) pada kartu identitas mahasiswa dan mengirimkan data ke server. Sistem ini menyediakan situs web yang dapat mengelola data Food Truck. Mahasiswa juga dapat mengecek informasi tentang stok makanan dan lokasi Food Truck yang tersedia melalui situs web yang akan disediakan.

Penelitian ini bertujuan untuk mengimplementasikan sistem *backend* sebagai REST API yang menjadi pusat pengelolaan dan sumber data bagi sistem otomasi pendataan mahasiswa pada Food Truck. Sistem ini diharapkan menjadi solusi bagi masalah yang ada dan dapat memberikan manfaat bagi pelaksanaan program pembagian makanan bergizi serta para mahasiswa yang menerima bantuan makanan.

.II. TINJAUAN LITERATUR

A. Rapid Application Development

RAD (*Rapid Application Development*) adalah metodologi pengembangan perangkat lunak yang fokus pada pengembangan aplikasi dengan cepat dan efisien.

RAD mengutamakan fleksibilitas dan iterasi yang cepat dalam pengembangan perangkat lunak dengan meminimalkan dokumentasi dan proses formal yang memakan waktu. Dalam RAD, pengembang akan fokus pada pengembangan fitur-fitur yang penting terlebih dahulu dan kemudian memperbaiki dan meningkatkan fitur tersebut secara bertahap. Dengan demikian, RAD dapat membantu mempercepat pengembangan aplikasi tanpa mengorbankan kualitas dan keamanan.

Metode pengembangan ini memiliki 4 tahapan sesuai model pengembangan RAD (*Rapid Application Development*), di antaranya: Perencanaan Kebutuhan Proyek (*Requirements Planning*), Proses Perancangan (*Design Workshop*), dan Penerapan (*Implementation*) [1]. Secara garis besar pada tahap pertama, yaitu Perencanaan Kebutuhan merupakan tahap dimana data dikumpulkan dan dilakukan analisa, di tahap ini juga dilakukan pendefinisian pada kebutuhan aplikasi yang harus dipenuhi dengan cara melibatkan pengguna dan keberjalanan sistem lama yang saat ini sedang berjalan. Pada tahap perancangan, pengembang akan membuat *prototype* untuk memberikan gambaran kepada pengguna dan melakukan perbaikan jika terdapat bagian yang tidak sesuai atau kurang berdasarkan kebutuhan pengguna. Pada tahap penerapan, pengembang akan melakukan implementasi dan menguji sistem yang telah dirancang untuk memastikan bahwa sistem sudah berfungsi dengan baik dan sesuai dengan kebutuhan pengguna. Setelah selesai melakukan tahap-tahap tersebut, pengembang akan melakukan tahap terakhir yaitu pengujian sistem secara keseluruhan untuk memastikan sistem berjalan dengan baik sebelum diimplementasikan di lapangan.

B. REST API

REST (*Representational State Transfer*) API adalah sebuah arsitektur untuk membangun aplikasi web

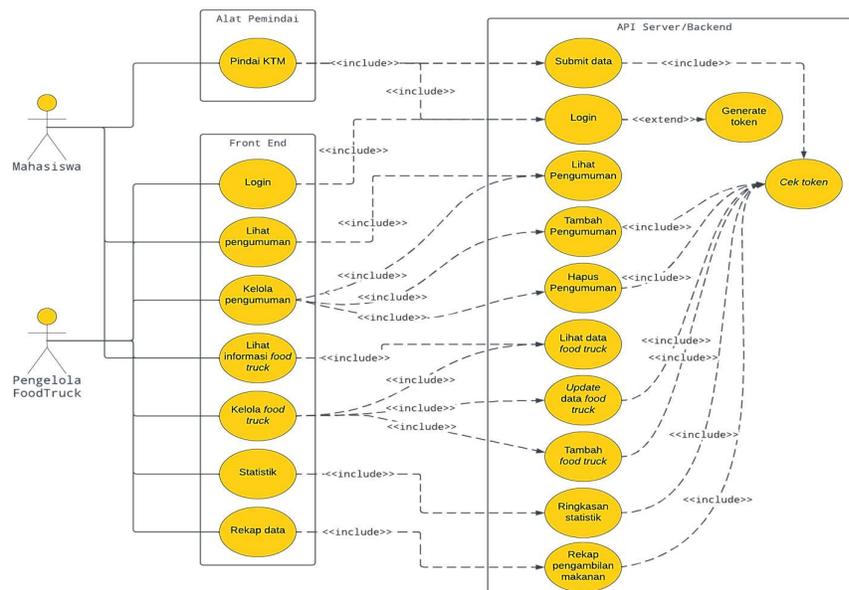
berbasis layanan web yang dapat saling berinteraksi melalui protokol HTTP [9]. Dalam REST API, terdapat dua jenis API yaitu *stateless* dan *stateful*. *Stateless* API tidak menyimpan informasi status antara *client* dan server, sehingga setiap *request* yang dikirimkan oleh *client* harus memuat informasi lengkap dan cukup untuk memenuhi permintaan tersebut. Sedangkan *stateful* API menyimpan informasi status antara *client* dan server, sehingga setiap *request* yang dikirimkan oleh *client* hanya perlu menyertakan informasi yang berbeda dengan *request* sebelumnya. Selain itu, dalam REST API, data biasanya diantarakan dalam format JSON atau XML.

C. Express

Express merupakan sebuah *web application framework* yang dibuat dengan bahasa pemrograman Javascript. Express merupakan salah satu *web framework* berbasis NodeJS dengan fitur handal untuk aplikasi web dan mobile. Express memudahkan *programmer* untuk membangun dan menjalankan *web server* dengan kode yang seminimal mungkin, sehingga mempercepat proses perancangan situs web [10].

III. PERENCANAAN DAN DESAIN

Penelitian ini berfokus pada pengembangan REST API (Application Programming Interface) untuk Sistem Otomasi Pendataan Mahasiswa pada Food Truck Undip. API akan menjadi pusat kontrol pengolahan dan penyedia data bagi alat pemindai dan situs web. Pengembangan API ini menggunakan metode perancangan sistem *Rapid Application Development* yang merupakan metode atau model pengembangan perangkat lunak yang relative singkat dan cepat.



Gambar 1. Diagram use case sistem keseluruhan

A. Perencanaan Kebutuhan

Sistem yang dibutuhkan adalah sebuah sistem yang dapat memindai kode QR pada KTM (Kartu Tanda Mahasiswa) menggunakan alat pindai berbasis Raspberry Pi yang terintegrasi dengan server basis data dan situs web. KTM Undip juga memiliki kode QR yang dapat dibaca oleh sistem secara otomatis sehingga dapat mempercepat proses pendataan mahasiswa. Selain itu, pengelola juga dapat memasukkan jumlah makanan yang dibagikan pada hari itu kemudian mengakses data mahasiswa melalui situs web. Mahasiswa juga dapat melihat data makanan yang masih tersedia melalui situs web yang disediakan

REST API yang dibuat pada server akan digunakan sebagai jembatan antara alat pemindai, situs web dan basis data, sehingga data yang tercatat dalam alat pemindai dapat secara otomatis tersimpan dalam basis data dan dapat dilihat oleh pengelola melalui situs web. Basis data akan menyimpan informasi mengenai data mahasiswa, data lokasi Food Truck, stok makanan yang tersedia, dan data lain yang diperlukan oleh sistem dan dapat diakses melalui API.

Diagram *use case* yang dibuat untuk menggambarkan proses yang terjadi pada API Server yang berinteraksi dengan sistem *front end* dan alat pemindai ditunjukkan pada Gambar 1.

B. Perancangan Basis Data

Pada sistem basis data MongoDB, data disimpan dalam format *document based* yang tergabung dalam *collections*. Dalam perancangannya, basis data ini tidak memerlukan ERD (*Entity Relationship Diagram*) karena MongoDB merupakan contoh basis data yang bersifat NoSQL, artinya tidak memerlukan *join* tabel, namun melakukan referensi kunci ke masing-masing objek, sehingga basis data yang dirancang tidak memerlukan ERD. Dalam bagian ini, dijabarkan deskripsi dari tabel atau *collection* apa saja yang digunakan.

1. Tabel PenerimaMakanan

Tabel 1 merupakan tabel yang berisi data pengambilan makanan yang dilakukan oleh mahasiswa di lokasi Food Truck. Pada tabel ini berisi identitas mahasiswa yang berisi Nama, NIM, Prodi dan Fakultas. Terdapat juga field tanggal dan *foodtruck_id* yang menjelaskan kapan dan di Food Truck mana mahasiswa tersebut melakukan pendataan.

Tabel 1. Tabel Mahasiswa Penerima Makanan

No.	Nama Kolom	Tipe Data	Keterangan
1	tanggal	Date	Tanggal pengambilan makanan
2	nama	String	Nama mahasiswa
3	nim	String	NIM mahasiswa
4	prodi	String	Prodi mahasiswa
5	fakultas	String	Fakultas mahasiswa
6	foodtruck_id	ObjectID	Referensi ke Tabel FoodTruck
7	_id	ObjectID	<i>Primary Key</i>

Tabel 2 merupakan tabel yang berisi data Food Truck yang tersedia di Undip. Data tersebut berisi nama, lokasi, stok makanan, dan status buka atau tutupnya Food Truck tersebut.

Tabel 2. Tabel Food Truck

No.	Nama Kolom	Tipe Data	Keterangan
1	nama	String	Nama Food Truck
2	lokasi	String	Lokasi Food Truck
3	stok	Number	Stok Food Truck
4	open	Boolean	Status aktif/nonaktif Food Truck
5	_id	ObjectID	<i>Primary Key</i>

Tabel 3 merupakan tabel yang digunakan untuk menyimpan akun pengelola untuk autentikasi dalam mengakses dasbor admin. Tabel ini berisi *username* dan *password* pengelola, dan juga *field superadmin* yang digunakan untuk menandakan hak akses pada akun pengelola tersebut.

Tabel 3. Tabel User

No.	Nama Kolom	Tipe Data	Keterangan
1	username	String	Nama user
2	password	String	Kata sandi
3	superadmin	Boolean	Jenis hak akses
4	_id	ObjectID	<i>Primary Key</i>

Tabel 4 merupakan tabel yang berisi data pengambilan makanan. Data pengambilan makanan berisi data keterangan yang diisi oleh pengelola ketika akan mengurangi stok makanan yang ada di lokasi Food Truck, hal ini dilakukan agar lengkapnya pencatatan untuk tiap stok yang berkurang setiap kalinya. Pada tabel, data pengambilan makanan berisi keterangan, jumlah makanan yang diambil, tanggal data itu dibuat, dan pada lokasi Food Truck mana makanan tersebut diambil.

Tabel 4. Tabel Pengambilan Makanan Oleh Pengelola

No.	Nama Kolom	Tipe Data	Keterangan
1	keterangan	String	Keterangan makanan diambil
2	jumlah	String	Jumlah makanan yang diambil
3	tanggal	Date	Tanggal stok makanan diambil
4	foodtruck_id	ObjectID	Referensi ke Tabel FoodTruck
5	_id	ObjectID	<i>Primary Key</i>

Tabel 5 merupakan tabel yang berisi data pengumuman. Data pengumuman berisi pengumuman yang akan ditampilkan ke halaman utama situs web, sehingga mahasiswa dapat mengetahui pengumuman yang diberikan oleh pengelola Food Truck. Pada tabel, data pengumuman berisi judul dan deskripsi pengumuman, serta tanggal kapan pengumuman dibuat.



Tabel 5. Tabel Pengumuman

No.	Nama Kolom	Tipe Data	Keterangan
1	judul	String	Judul pengumuman
2	deskripsi	String	Deskripsi pengumuman
3	tanggalDibuat	Date	Tanggal pengumuman dibuat
4	_id	ObjectID	<i>Primary Key</i>

C. Perancangan API

Dalam perancangan API, HTTP digunakan untuk mengatur pengiriman dan penerimaan data antara server dan aplikasi yang menggunakan API. Selain itu, perancangan API juga melibatkan pemilihan format data yang digunakan untuk pertukaran data antara aplikasi, seperti format JSON (*JavaScript Object Notation*) yang ringan dan mudah dibaca oleh mesin dan manusia[3]. Pada bagian ini akan dijelaskan deskripsi data JSON yang digunakan pada proses permintaan dan penerimaan yang ada pada tiap *endpoint* API.

1. Rancangan *endpoint* Submit Data

Endpoint submit data dipanggil melalui Alat Pemindai yang berada di lokasi Food Truck. Alat pemindai akan mengirimkan data mahasiswa yang ingin mengambil makanan. Server akan menerima data tersebut dan akan memvalidasi apakah mahasiswa dapat mengambil makanan pada saat itu atau tidak. Adapun token JWT yang disimpan di HTTP *header*, didapatkan dari *endpoint* Login. Rancangan *endpoint* Submit Data ditunjukkan pada Tabel 6.

Tabel 6. Tabel rancangan *endpoint* Submit Data

Deskripsi	Menerima data mahasiswa pengambil makanan dari alat pemindai.	
Endpoint	/submit	
HTTP method	POST	
Request Headers	Key	Deskripsi
	Authentication	Token JWT autentikasi
Request Body	Key	Deskripsi
	foodtruck	ID Food Truck
	nama	Nama Mahasiswa
	nim	NIM Mahasiswa
	prodi	Prodi Mahasiswa
	fakultas	Fakultas Mahasiswa
Response	HTTP code	Deskripsi
	200	Mahasiswa dapat mengambil makanan
	400	Mahasiswa tidak dapat mengambil makanan

2. Rancangan *endpoint* rekap data

Terdapat dua *endpoint* rekap pengambilan makanan, yaitu oleh mahasiswa dan pengelola. Pengambilan makanan oleh mahasiswa terjadi pada lokasi Food Truck ketika mahasiswa berhasil memindai KTM untuk mendapatkan makanan. Adapun token JWT yang disimpan di HTTP *header*, didapatkan dari *endpoint*

Login. Pengambilan makanan oleh pengelola dilakukan ketika pengelola mengurangi stok yang ada di sistem.

Rancangan *endpoint* rekap pengambilan makanan oleh mahasiswa ditunjukkan pada Tabel 7.

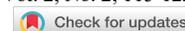
Tabel 7. Tabel rancangan *endpoint* rekap mahasiswa penerima makanan

Deskripsi	Rekap penerimaan makanan oleh mahasiswa	
Endpoint	/rekap	
HTTP method	GET	
Request Headers	Key	Deskripsi
	Authentication	Token JWT autentikasi
Request Body	Key	Deskripsi
	foodtruck	ID Food Truck
	from	Tanggal awal
	until	Tanggal akhir
Response	HTTP code	Deskripsi
	200	Permintaan sukses
	400	Permintaan gagal
Response body	Key	Deskripsi
	tanggal	Tanggal pengambilan makanan
	nama	Nama mahasiswa
	nim	NIM mahasiswa
	prodi	Prodi mahasiswa
	fakultas	Fakultas mahasiswa
	foodtruck_id	ID Food Truck
	id	ID pengambilan

Rancangan *endpoint* rekap pengambilan makanan oleh pengelola ditunjukkan pada Tabel 8.

Tabel 8. Tabel rancangan *endpoint* rekap pengambilan makanan

Deskripsi	Rekap pengambilan oleh pengelola	
Endpoint	/rekap2	
HTTP method	GET	
Request Headers	Key	Deskripsi
	Authentication	Token JWT autentikasi
Request Body	Key	Deskripsi
	foodtruck	ID Food Truck
	from	Tanggal awal
	until	Tanggal akhir
Response	HTTP code	Deskripsi
	200	Permintaan sukses
	400	Permintaan gagal diproses
Response body	Key	Deskripsi
	tanggal	Tanggal pengambilan makanan
	keterangan	Keterangan pengambilan
	Jumlah	Jumlah pengambilan
	prodi	Prodi mahasiswa
	foodtruck_id	Referensi ke Tabel Food Truck
	id	ID pengambilan



3. Rancangan *endpoint update* data Food Truck

Pembaruan data Food Truck memerlukan ID Food Truck yang ingin diperbarui. Pembaruan dapat dilakukan pada kolom nama, lokasi, stok dan open. Ketika pengelola memperbarui data stok kurang dari stok yang ada, pengelola diharuskan mengisi data keterangan kenapa stok tersebut dikurangi. Hal tersebut dilakukan sebagai upaya pencatatan, agar stok makanan tetap terjaga dan dapat ditanggung-jawabkan. Adapun token JWT yang disimpan di HTTP header, didapatkan dari *endpoint* Login. Rancangan *endpoint update* data Food Truck ditunjukkan pada Tabel 9.

Tabel 9. Tabel rancangan *endpoint update* data Food Truck

Deskripsi	Memperbarui data Food Truck	
Endpoint	/foodtruck	
HTTP method	PUT	
Request Headers	Key	Deskripsi
	Authentication	Token JWT autentikasi
Request Body	Key	Deskripsi
	foodtruck	ID Food Truck
	nama	Nama Food Truck
	lokasi	Lokasi Food Truck
	stok	Stok Food Truck
	open	Status Food Truck
Response	Key	Deskripsi
	200	Data berhasil diperbarui
	400	Data gagal diperbarui

4. Rancangan *endpoint* tambah lokasi Food Truck

Endpoint ini digunakan ketika pengelola ingin membuka lokasi Food Truck baru. Adapun token JWT yang disimpan di HTTP header, didapatkan dari *endpoint* Login. Rancangan *endpoint* tambah Food Truck ditunjukkan pada Tabel 10.

Tabel 10. Tabel rancangan *endpoint* tambah lokasi Food Truck

Deskripsi	Menambahkan Food Truck baru	
Endpoint	/foodtruck	
HTTP method	POST	
Request Headers	Key	Deskripsi
	Authentication	Token JWT autentikasi
Request Body	Key	Deskripsi
	nama	Nama Food Truck
	lokasi	Lokasi Food Truck
	stok	Stok Food Truck
	open	Status Food Truck
Response	Key	Deskripsi
	200	Food Truck berhasil ditambahkan
	400	Permintaan gagal diproses

5. Rancangan *endpoint* tambah pengumuman

Endpoint ini digunakan ketika pengelola ingin membuat pengumuman kepada mahasiswa yang nantinya dapat ditampilkan di halaman utama website. Adapun token JWT yang disimpan di HTTP header, didapatkan dari *endpoint* Login. Rancangan *endpoint* tambah pengumuman ditunjukkan pada Tabel 11.

Tabel 11. Tabel rancangan *endpoint* tambah pengumuman baru

Deskripsi	Menambahkan pengumuman baru	
Endpoint	/pengumuman	
HTTP method	POST	
Request Headers	Key	Deskripsi
	Authentication	Token JWT autentikasi
Request Body	Key	Deskripsi
	Judul	Judul pengumuman
Response	Key	Deskripsi
	deskripsi	Deskripsi pengumuman
	HTTP code	Deskripsi
	200	Pengumuman berhasil ditambahkan
	400	Permintaan gagal diproses

6. Rancangan *endpoint* hapus pengumuman

Endpoint ini digunakan ketika pengelola ingin menghapus pengumuman yang ada. Adapun token JWT yang disimpan di HTTP header, didapatkan dari *endpoint* Login. Rancangan *endpoint* hapus pengumuman ditunjukkan pada Tabel 12.

Tabel 12. Tabel rancangan *endpoint* hapus pengumuman

Deskripsi	Menghapus pengumuman	
Endpoint	/pengumuman	
HTTP method	DELETE	
Request Headers	Key	Deskripsi
	Authentication	Token JWT autentikasi
Request Body	Key	Deskripsi
	pengumuman	ID Pengumuman
Response	Key	Deskripsi
	HTTP code	Deskripsi
	200	Pengumuman berhasil dihapus
	400	Permintaan gagal diproses/Pengumuman tidak ditemukan

7. Rancangan *endpoint* login

Endpoint ini digunakan ketika pengelola hendak melakukan login ke sistem agar dapat menggunakan fitur yang ada di *dashboard* admin. Fungsi autentikasi di sistem ini menggunakan JSON Web Token (JWT), yang ditaruh pada tiap header HTTP ketika melakukan permintaan kepada *endpoint-endpoint* yang membutuhkan hak akses pengelola. *Endpoint* login ini digunakan untuk mendapatkan token tersebut. Token didapatkan jika *username* dan *password* yang



dikirimkan itu valid dan terdaftar di basis data. Rancangan *endpoint* login ditunjukkan pada Tabel 13.

Tabel 13. Tabel rancangan *endpoint* login

Deskripsi	Mendapatkan token autentikasi	
Endpoint	/user/login	
HTTP method	POST	
Request Body	Key	Deskripsi
	username	Username
	password	Kata sandi
Response	HTTP code	Deskripsi
	200	Authentikasi berhasil
	400	Permintaan gagal diproses
Response body	Key	Deskripsi
	token	Token JWT untuk autentikasi

8. Rancangan *endpoint* lihat data Food Truck

Endpoint ini digunakan untuk menampilkan data Food Truck yang ada di sistem. Data yang dapat dilihat pada *endpoint* ini seperti data stok, status aktif Food Truck, nama dan lokasi Food Truck. Rancangan *endpoint* lihat data Food Truck ditunjukkan pada Tabel 14.

Tabel 14. Tabel rancangan *endpoint* lihat data Food Truck

Deskripsi	Menampilkan data Food Truck yang tersedia.	
Endpoint	/foodtruck	
HTTP method	GET	
Response	HTTP code	Deskripsi
	200	Permintaan berhasil diproses
	400	Permintaan gagal diproses
Response body	Key	Deskripsi
	nama	Nama Food Truck
	lokasi	Lokasi Food Truck
	stok	Stok Food Truck
	open	Status aktif/nonaktif Food Truck
	id	ID Food Truck
nama	Nama Food Truck	

9. Rancangan *endpoint* lihat pengumuman

Endpoint ini digunakan untuk menampilkan data pengumuman yang ada di sistem. Data yang dapat dilihat pada *endpoint* ini seperti judul dan deskripsi pengumuman. Rancangan *endpoint* lihat data pengumuman ditunjukkan pada Tabel 15.

Tabel 15. Tabel rancangan *endpoint* lihat data pengumuman

Deskripsi	Menampilkan data pengumuman	
Endpoint	/pengumuman	
HTTP method	GET	
Response	HTTP code	Deskripsi
	200	Permintaan berhasil diproses
	400	Permintaan gagal diproses

Response body	Key	Deskripsi
	judul	Judul pengumuman
	deskripsi	Deskripsi pengumuman
	tanggalDibuat	Tanggal pengumuman dibuat
	id	ID pengumuman

10. Rancangan *endpoint* ringkasan statistik

Endpoint ini digunakan untuk menampilkan ringkasan statistik dari makanan yang telah terbagi. Data yang dapat dilihat pada *endpoint* ini seperti total terbagi hari ini, total terbagi per fakultas, total terbagi per Food Truck. Adapun token JWT yang disimpan di HTTP header, didapatkan dari *endpoint* Login. Rancangan *endpoint* ringkasan statistic ditunjukkan pada Tabel 16.

Tabel 16. Tabel rancangan *endpoint* ringkasan statistik

Deskripsi	Menampilkan ringkasan statistik	
Endpoint	/summary	
HTTP method	GET	
Response	HTTP code	Deskripsi
	200	Permintaan berhasil diproses
	400	Permintaan gagal diproses
Response body	Key	Deskripsi
		Total makanan terbagi
	totalTerbagi	Total makanan terbagi hari ini
	totalTerbagi-HariIni	Total makanan terbagi per fakultas
	totalPerFakultas	Total makanan terbagi per lokasi Food Truck

III. HASIL DAN PEMBAHASAN

A. Implementasi Basis Data

Model basis data diimplementasikan dengan pustaka Mongoose dengan kode Javascript. Untuk setiap model yang digunakan pada sistem pendataan mahasiswa pada Food Truck Undip akan dibuatkan skema terlebih dahulu dengan menggunakan *library* Mongoose. Pembuatan setiap skema mengikuti tabel-tabel yang ada pada perencanaan basis data. Dalam pembuatan skema ini, setiap *field* akan ditentukan tipe datanya berdasarkan hasil perencanaan sebelumnya. Setelah skema berhasil dibuat, selanjutnya model dapat dibuat berdasarkan skema tersebut dengan menggunakan fungsi `mongoose.model()`.

B. Pengujian

Metode pengujian *black box* merupakan metode pengujian perangkat lunak yang didasarkan pada spesifikasi fungsional suatu sistem. Dalam konteks pengujian API, metode ini digunakan untuk menguji fungsionalitas tiap *endpoint* API. Pengujian *black box* dilakukan dengan cara mengirimkan *request* dan memperhatikan *response* yang diterima dari API.



Pada bagian ini, akan diberikan tabel pengujian yang berisi hasil pengujian fungsional tiap *endpoint* API yang telah diimplementasikan pada sistem.

1. Pengujian *endpoint* /foodtruck

Tabel 17. Hasil pengujian *endpoint* /foodtruck

Bentuk Pengujian	Hasil Yang Di-harapkan	Status
Mengirimkan <i>request</i> GET untuk mendapatkan informasi Food Truck.	Menampilkan data Food Truck yang terdata di server	Berhasil
Mengirimkan <i>request</i> POST dengan nama, lokasi, dan stok Food Truck baru.	Data Food Truck baru berhasil ditampilkan	Berhasil
Mengirimkan <i>request</i> POST dengan data yang kurang maupun format yang tidak valid.	Mengembalikan pesan "Invalid value"	
Mengirimkan <i>request</i> PUT dengan nama, lokasi, stok, atau status aktif dengan ID Food Truck yang baru, lalu melihat data Food Truck melalui <i>request</i> GET.	Menampilkan data Food Truck baru yang telah dirubah	Berhasil
Mengirimkan <i>request</i> PUT dengan nama, lokasi, stok, atau status aktif dengan format yang salah dan dengan ID Food Truck yang tidak valid.	Mengembalikan pesan kesalahan "Invalid Value"	Berhasil

2. Pengujian *endpoint* /pengumuman

Tabel 18. Hasil pengujian *endpoint* /pengumuman

Bentuk Pengujian	Hasil Yang Di-harapkan	Status
Mengirimkan <i>request</i> GET untuk melihat pengumuman	Menampilkan data pengumuman yang ada di basis data	Berhasil
Mengirimkan <i>request</i> POST dengan judul dan deskripsi pengumuman baru.	Data pengumuman baru berhasil ditampilkan	Berhasil
Mengirimkan <i>request</i> POST dengan data yang kurang maupun format yang tidak valid	Mengembalikan pesan "Invalid value"	Berhasil
Mengirimkan <i>request</i> DELETE dengan ID pengumuman yang ingin dihapus, lalu melihat data Food Truck melalui <i>request</i> GET	Tidak terdapat data Food Truck yang sudah dihapus,	Berhasil
Mengirimkan <i>request</i> DELETE dengan ID pengumuman yang tidak valid	Menampilkan pesan kesalahan "Invalid Value"	Berhasil

3. Pengujian *endpoint* /submit

Tabel 19. Hasil pengujian *endpoint* /submit

Bentuk Pengujian	Hasil Yang Di-harapkan	Status
Mengirimkan <i>request</i> POST dengan data ma-	Menampilkan pesan sukses, data maha-	Berhasil

Bentuk Pengujian	Hasil Yang Di-harapkan	Status
siswa penerima bantuan makanan	siswa terdata di basis data	
Mengirimkan <i>request</i> POST dengan identitas mahasiswa di hari yang sama	Menampilkan pesan kesalahan bahwa mahasiswa telah mengambil pada hari ini	Berhasil
Mengirimkan <i>request</i> POST pada lokasi Food Truck yang memiliki stok kosong	Menampilkan pesan kesalahan bahwa stok Food Truck habis	Berhasil
Mengirimkan <i>request</i> POST pada lokasi Food Truck yang tidak aktif	Menampilkan pesan kesalahan bahwa Food Truck tidak aktif	Berhasil
Mengirimkan <i>request</i> POST dengan data yang kurang maupun format yang tidak valid	Mengembalikan pesan "Invalid value"	Berhasil

4. Pengujian *endpoint* /rekap

Tabel 20. Hasil pengujian *endpoint* /rekap

Bentuk Pengujian	Hasil Yang Di-harapkan	Status
Mengirimkan <i>request</i> GET untuk merekap penerima pada tanggal dan Food Truck tertentu	Menampilkan data mahasiswa penerima bantuan sesuai lokasi Food Truck dan tanggal yang diharapkan	Berhasil
Mengirimkan <i>request</i> POST dengan parameter yang kurang maupun format yang tidak valid	Mengembalikan pesan "Invalid value"	Berhasil

5. Pengujian *endpoint* /rekap2

Tabel 21. Hasil pengujian *endpoint* /rekap2

Bentuk Pengujian	Hasil Yang Di-harapkan	Status
Mengirimkan <i>request</i> GET untuk merekap pengambilan makanan oleh pengelola pada tanggal dan lokasi Food Truck tertentu	Menampilkan data pengambilan makanan sesuai lokasi Food Truck dan tanggal yang diharapkan	Berhasil
Mengirimkan <i>request</i> POST dengan parameter yang kurang maupun format yang tidak valid	Mengembalikan pesan "Invalid value"	Berhasil

6. Pengujian *endpoint* /login

Tabel 22. Hasil pengujian *endpoint* /login

Bentuk Pengujian	Hasil Yang Di-harapkan	Status
Mengirimkan <i>request</i> POST dengan <i>username</i> dan <i>password</i> yang terdaftar.	Mengembalikan token autentikasi	Berhasil
Mengirimkan <i>request</i> POST dengan <i>username</i> dan <i>password</i> yang tidak terdaftar.	Menampilkan pesan kesalahan bahwa pengguna tidak ditemukan	Berhasil
Mengirimkan <i>request</i>	Mengembalikan	Berhasil



Bentuk Pengujian	Hasil Yang Di-harapkan	Status
POST dengan parameter yang kurang maupun format yang tidak valid	pesan "Invalid value"	

IV. KESIMPULAN

Dalam penelitian ini, dapat disimpulkan bahwa implementasi *backend* berbasis REST API pada sistem otomasi pendataan mahasiswa pada Food Truck Undip memberikan beberapa kemudahan dan efisiensi seperti validasi data secara *real-time* dan akurat, pengelolaan data terpusat pada basis data, dan kemudahan pengelola dalam mengelola Food Truck. Hasil pengujian sistem menunjukkan bahwa sistem dapat berjalan dengan baik dan sesuai dengan kebutuhan. Pengembangan sistem ini diharapkan dapat memberikan kemudahan dan efisiensi bagi jalannya pemberian bantuan makanan pada program Food Truck Undip dan dapat menjadi acuan bagi penelitian selanjutnya dalam pengembangan sistem serupa.

DAFTAR PUSTAKA

- [1] O. Irnawati, G. Bayu, A. Listianto, M. Informatika, and A. Bsi Bekasi, "Metode Rapid Application Development (RAD) pada Perancangan Website Inventory PT. SARANA ABADI MAKMUR BERSAMA (S.A.M.B) JAKARTA 1)."
- [2] D. Flanagan, *JavaScript: the definitive guide*, vol. 1018. O'reilly, 2006.
- [3] Mithun. Satheesh, B. Joseph. D'mello, and Jason. Krol, *Web development with MongoDB and NodeJS : build an interactive and full-featured web application from scratch using Node.js and MongoDB*. Packt Publishing, 2015.
- [4] D. Kriestanto and A. B. Arnado, "Implementasi Website Pencarian Kos dengan NOSQL," *JIKO (Jurnal Informatika dan Komputer)*, vol. 2, no. 2, pp. 103–108, 2017.
- [5] A. Boicea, F. Radulescu, and L. I. Agapin, "MongoDB vs Oracle - Database comparison," in *Proceedings - 3rd International Conference on Emerging Intelligent Data and Web Technologies, EIDWT 2012*, 2012, pp. 330–335. doi: 10.1109/EIDWT.2012.32.
- [6] R. Fielding and et al, "RFC2616: Hypertext Transfer Protocol--HTTP/1.1.," 1999.
- [7] V. Surwase, "REST API modeling languages-a developer's perspective," *Int. J. Sci. Technol. Eng*, vol. 2, no. 10, pp. 634–637, 2016.
- [8] C. J. Ihrig and C. J. Ihrig, "Javascript object notation," *Pro Node. js for Developers*, pp. 263–270, 2013.
- [9] S. Karlsson, A. Čaušević, and D. Sundmark, "QuickREST: Property-based test generation of OpenAPI-described RESTful APIs," in *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*, IEEE, 2020, pp. 131–141.
- [10] E. Hahn, *Express in Action: Writing, building, and testing Node. js applications*. Simon and Schuster, 2016.
- [11] S. Holmes, *Mongoose for Application Development*. Packt Publishing Ltd, 2013.
- [12] M. Zadka and M. Zadka, "Requests," *DevOps in Python: Infrastructure as Python*, pp. 85–94, 2019.
- [13] B. Okken, *Python Testing with pytest*. Pragmatic Bookshelf, 2022.
- [14] I. Jacobson, *Object-oriented software engineering: a use case driven approach*. Pearson Education India, 1993.
- [15] K. S. Martin Fowler, "UML Distilled: Applying the Standard Object Modeling Language." Addison-Wesley Reading, MA, USA, 1997.
- [16] J. Doe, "Recommended practice for software requirements specifications (ieee)," *IEEE, New York*, 2011.
- [17] M. Glinz, "On non-functional requirements," in *15th IEEE international requirements engineering conference (RE 2007)*, IEEE, 2007, pp. 21–26.



©2023. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).