

## Perancangan *Back-end* Api pada Aplikasi *Mobile Fruityfit* Menggunakan *Framework Express JS*

Danar Hadi Bachtiar<sup>1</sup>, Paniran Paniran<sup>2</sup>, I Made Budi Suksmadana<sup>3</sup>

<sup>1,2&3</sup> Jurusan Teknik Elektro Universitas Mataram

Jl. Majapahit 62, Mataram, Lombok Nusa Tenggara Barat

Email: danarhadib@gmail.com<sup>1</sup>, paniranmt@yahoo.com<sup>2</sup>, mdbudisuk@unram.ac.id<sup>3</sup>

**Abstract.** Increasing fruit consumption is one of the challenges of public health. To address this, the authors developed a mobile application that allows users to scan and find information related to fruit. FruityFit is an application that can help users to get nutritional and processed information related to certain fruits that can increase users' interest in fruit consumption. This research focuses on constructing the back-end component of the FruityFit application. The design adheres to the REST architectural, leveraging the well-established ExpressJS framework for developmen. To realize the system, we implemented it using a suite of Google Cloud Platform (GCP) services. These services include Cloud Run, App Engine, Cloud Storage, and Cloud SQL, providing a scalable foundation for the back-end infrastructure.. The final result of this research is a REST API that has been running well and documented to help the front-end consume the API.

**Keywords:** Backend, Google Cloud Platform, Nodejs, REST API

**Abstrak.** Peningkatan konsumsi buah-buahan merupakan salah satu tantangan kesehatan masyarakat. Untuk mengatasi hal tersebut, penulis mengembangkan aplikasi mobile yang memungkinkan pengguna memindai dan mencari informasi terkait buah. FruityFit merupakan aplikasi yang dapat membantu pengguna untuk mendapatkan informasi nutrisi dan olahan terkait buah-buahan tertentu sehingga dapat meningkatkan minat pengguna dalam konsumsi buah. Penelitian ini berfokus pada konstruksi komponen back-end pada aplikasi FruityFit. Desainnya mengikuti arsitektur REST, memanfaatkan kerangka kerja ExpressJS yang sudah mapan untuk pengembangan. Untuk merealisasikan sistem tersebut, kami mengimplementasikannya menggunakan rangkaian layanan Google Cloud Platform (GCP). Layanan-layanan tersebut antara lain Cloud Run, App Engine, Cloud Storage, dan Cloud SQL, yang memberikan landasan terukur untuk infrastruktur back-end. Hasil akhir dari penelitian ini adalah REST API yang telah berjalan dengan baik dan terdokumentasi untuk membantu front-end. akhir mengkonsumsi API.

**Kata Kunci :** Backend, Google Cloud Platform, Nodejs, REST API

### PENDAHULUAN

Salah satu masalah kesehatan yang kerap dihadapi saat ini adalah obesitas. Berdasarkan *World Population Review*, Indonesia menduduki peringkat ke-18 untuk masalah obesitas. Salah satu penyebab utama obesitas adalah gaya hidup yang tidak sehat. Meskipun memiliki tanah yang subur dan hasil pertanian yang melimpah seperti buah dan sayuran, masih terdapat kurangnya pemahaman terhadap nutrisi yang terjadi di masyarakat. Berdasarkan hasil penelitian yang dilakukan oleh Mariatul Qibtiyah mengenai konsumsi buah dan sayur pada remaja. Salah satu faktor yang menghambat konsumsi buah adalah jenis buah yang disukai masih terbatas [1]. Hal ini mengarah ke kurangnya konsumsi buah-buahan dan sayuran atau hanya mengonsumsi buah-buahan tertentu saja yang bahkan bisa membuat bosan. Oleh karena

itu, pendekatan inovatif seperti aplikasi *mobile* yang dapat mendorong konsumsi buah melalui rekomendasi olahan yang menarik dapat meningkatkan asupan buah.

Melihat permasalahan ini, penulis ingin membantu masyarakat umum terutama remaja untuk meningkatkan minat dalam mengonsumsi buah-buahan melalui sebuah aplikasi *mobile*. Aplikasi ini diberi nama FruityFit dan dibangun berfokus pada *Human Healthcare and Living Wellbeings*. FruityFit merupakan aplikasi *mobile* yang dapat membantu mengenai jenis dan kesegaran buah melalui *metode Image Recognition*. Selain itu, pengguna juga mendapatkan informasi mengenai nutrisi dan rekomendasi berupa olahan dan artikel yang sesuai dengan buah yang ditunjukkan.

Dalam pembangunan sebuah *back-end* pada aplikasi, diperlukan penggunaan bahasa pemrograman yang dijalankan pada sisi server (*server-side*). Penelitian sebelumnya yang terkait dengan pembangunan *back-end* server dilakukan oleh Ahsan Mubariz dan Dahlia Nur yang membahas penggunaan REST API pada Sistem Pendaftaran Ujian Masuk Politeknik Ujung Padang [2]. Sistem tersebut dijalankan menggunakan platform Nodejs karena memiliki performa respon yang baik. Selain itu, Anugerah Christian Rompis dan Rizal Fathoni Aji melakukan penelitian yang menyatakan bahwa layanan REST API yang berjalan di atas platform Nodejs lebih sesuai untuk mengatasi respon langsung terhadap *request* yang masuk tanpa melibatkan perhitungan yang banyak [3].

*Nodejs* merupakan *runtime environment* dari javascript yang bersifat *non-blocking* sehingga *nodejs* dapat menjalankan banyak *request* secara bersamaan. Dalam pengembangan *backend* aplikasi Fruityfit, penulis menggunakan platform Nodejs dengan *framework Express* yang menyediakan fitur-fitur seperti *routing*, *request handling* dan *response* [4].

## KAJIAN TEORI

### ***Backend***

*Backend* adalah bagian internal di mana proses pada suatu sistem informasi atau aplikasi berjalan, seperti penambahan, perubahan dan penghapusan data. *Backend* menangani berbagai proses yang tidak terlihat langsung oleh pengguna, seperti pengelolaan server dan basis data [5]. Meskipun tidak berinteraksi langsung dengan pengguna, *backend* memiliki peran penting dalam pengolahan data dengan menyediakan *interface* agar *client* dapat berkomunikasi dengan *server* melalui API (*Application Programming Interface*). API berfungsi sebagai jembatan antara berbagai sistem aplikasi sehingga memungkinkan akses bersamaan terhadap fungsi sistem yang berbeda. Penggunaan API dapat meningkatkan

efisiensi pengembangan aplikasi karena memungkinkan untuk memanfaatkan fungsi dari suatu aplikasi yang telah ada tanpa perlu membangunnya dari awal. [6].

### ***Representational State Transfer (REST)***

*Representational State Transfer (REST)* merupakan salah arsitektur *backend* yang digunakan untuk sistem terdistribusi. REST menyediakan serangkaian prinsip arsitektur yang berfokus pada skalabilitas dan interaksi antar komponen sistem dan keseragaman *interface*. Dengan menerapkan prinsip independensi setiap komponen dalam implementasi REST dapat membantu untuk meminimalkan latensi. Sistem yang menerapkan prinsip-prinsip REST disebut sebagai RESTful [7].

### **NodeJS**

Node.js merupakan *runtime environment platform server-side* yang bersifat *open source* dan cross platform yang menggunakan *chrome V8 javascript engine*, sehingga nodejs dapat menjalankan javascript diluar *browser* [8].

### **ExpressJS**

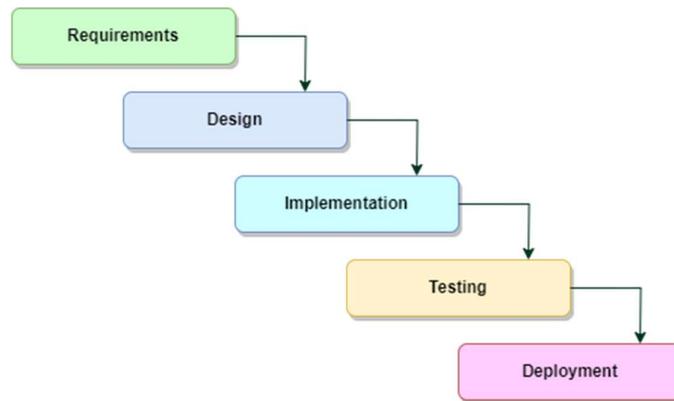
ExpressJS merupakan salah satu *framework* NodeJS yang dirancang untuk memudahkan dan mempercepat proses pembuatan aplikasi berbasis NodeJS dengan menggunakan pola desain yang fleksibel. ExpressJS juga dikenal sebagai *framework* yang ringan karena tidak memerlukan banyak dependensi tambahan sehingga ideal untuk pengembangan aplikasi web dan API [8].

### **Google Cloud Platform**

Google Cloud Platform (GCP) merupakan salah satu *service provider* milik Google yang menyediakan layanan *Cloud Computing*. Google Cloud Platform menawarkan berbagai layanan yang berfungsi sebagai solusi *cloud computing* dan pembangunan infrastruktur dengan reliabilitas dan *availability* tinggi. Setiap produk memiliki fitur dan keunggulan tersendiri yang dapat disesuaikan dengan kebutuhan pengguna, seperti komputasi, penyimpanan, jaringan, *big data* dan lain-lain [9].

## **METODE PENELITIAN**

Dalam penelitian ini, penulis menggunakan metode *waterfall* sebagai metode perancangan dan pengembangan. Metode *waterfall* adalah proses pengembangan perangkat lunak yang berjalan secara bertahap, di mana kemajuan terlihat mengalir ke bawah melalui fase-fase seperti perencanaan, pemodelan, implementasi dan pengujian [10].



Gambar 1. Metode *Waterfall* pada perancangan aplikasi FruityFit

Tahapan-tahapan yang dilakukan dalam metode *waterfall* antara lain seperti:

- 1) Analisa Kebutuhan (*Requirement Analysis*): analisis kebutuhan dilakukan dengan mengumpulkan kebutuhan-kebutuhan perangkat lunak sistem aplikasi, seperti fitur-fitur dan proses bisnis.
- 2) Desain Sistem (*System Design*): pada tahap desain sistem dilakukan perancangan arsitektur sistem yang mengarah pada analisa kebutuhan. Adapun pada tahap desain ini dihasilkan diagram ER (*Entity-Relationship*) untuk database, diagram arsitektur cloud dan *wireframe* untuk antarmuka pengguna.
- 3) Implementasi (*Implementation*): pada tahap ini dilakukan implemementasi dari arsitektur sistem ke dalam kode pemrograman untuk mengembangkan bagian *back-end* menggunakan framework ExpressJS. Impementasi arsitektur sistem dilakukan menggunakan layanan Google Cloud Platform (GCP). GCP adalah produk layanan komputasi cloud yang dimiliki Google yang menawarkan beragam layanan seperti hosting, database, dan pengelolaan aplikasi.
- 4) Pengujian (*Testing*): setelah proses implementasi selesai, langkah selanjutnya adalah menguji sistem yang telah dibuat. Pengujian dilakukan menggunakan metode *black-box testing* yang berfokus pada fungsionalitas endpoint yang telah dibuat untuk mengevaluasi *input* dan *output* dari aplikasi.
- 5) Penerapan (*Deployment*): pada tahap ini dilakukan *deployment* dan *hosting* API *back-end* yang telah selesai dibuat agar dapat diakses dan digunakan oleh bagian *front-end*.

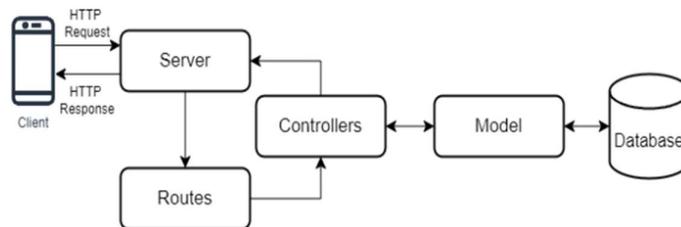
## HASIL DAN PEMBAHASAN

### Analisa Kebutuhan (*Requirement Analysis*)

Pada analisa kebutuhan, didapatkan hasil berupa fitur-fitur berdasarkan kebutuhan pengguna yang kemudian diimplementasikan ke dalam aplikasi FruityFit. Fitur-fitur yang dibutuhkan berupa *signup*, *signin* dan *signout* untuk pengguna. Kemudian, terdapat fitur bagi pengguna untuk memindai buah menggunakan kamera untuk mendapatkan informasi mengenai kandungan buah tersebut. Selain itu, terdapat fitur yang merekomendasikan olahan terkait buah-buah yang tersedia dan pengguna dapat menyimpan olahan tersebut menggunakan fitur favorit.

#### 1) Desain Sistem (*System Design*)

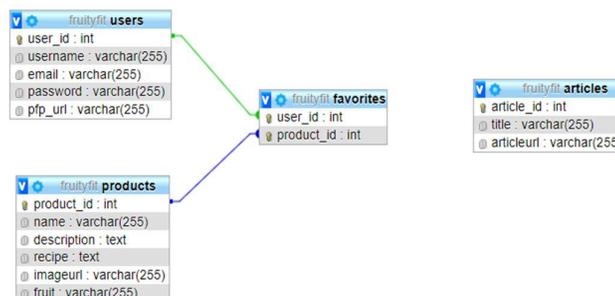
##### a) Rancangan Arsitektur Sistem



Gambar 2. Rancangan Arsitektur *Backend*

Pada rancangan *back-end* aplikasi FruityFit, terdapat beberapa komponen utama seperti *server*, *route*, *controller*, model dan *database*. Pada tahap awal, *client* mengirim request kepada server dan menentukan jalur (*route*) yang akan dieksekusi berdasarkan request yang diterima. Selanjutnya, *controllers* mengeksekusi metode yang sesuai dan diteruskan ke model untuk melakukan proses pengambilan dan manipulasi data yang dibutuhkan dari *database* [11]. Hasil yang didapatkan kemudian dikembalikan ke bagian *client* sebagai *response* sesuai *request* yang diajukan.

##### b) Struktur *Database*

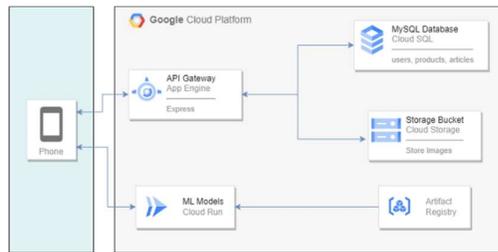


Gambar 3. Struktur *Database*

## Implementasi (*Implementation*)

Tahap awal pada proses implementasi adalah menentukan *endpoint* berdasarkan fitur yang telah ditentukan dan disesuaikan dengan kebutuhan dari bagian *front-end*. Selanjutnya adalah menentukan kebutuhan perangkat lunak, melakukan instalasi ExpressJS dan beberapa package pendukung seperti *body-parser*, *bcrypt*, *dot-env*, dan lain-lain.

Dalam pengembangan endpoint untuk keperluan autentikasi, penulis menggunakan JSON Web Token (JWT) dan *bcrypt* untuk mengenkripsi password pengguna. JWT bekerja dengan mendistribusikan token yang disimpan di *local storage* pengguna. Token ini kemudian digunakan untuk mengakses halaman tertentu dalam aplikasi dengan mengirimkan token Kembali ke server pada saat autentikasi [12]. Selain itu, terdapat *bcrypt* yang digunakan untuk mengenkripsi *password* pengguna dengan menggunakan fungsi *hashing*.

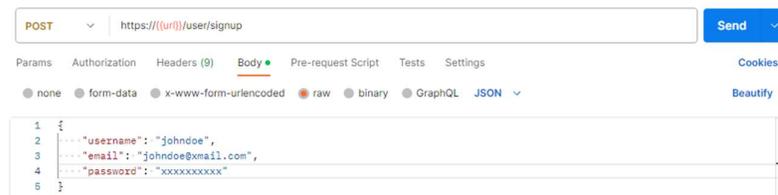


Gambar 4. Rancangan Arsitektur Google Cloud Platform

Penelitian ini memanfaatkan beberapa layanan Google Cloud Platform (GCP) untuk menjalankan aplikasinya, diantaranya App Engine dan Cloud Run untuk deployment service API dan machine learning model, Cloud SQL sebagai database relasional dan Cloud Storage sebagai penyimpanan data statis seperti gambar buah dan produk olahan.

## 2) Pengujian (*Testing*)

Pada tahapan ini dilakukan pengujian terhadap *endpoint* yang telah dibuat.



Gambar 5. Pengujian API

Pada gambar 5 ditunjukkan pengujian API menggunakan Postman untuk memverifikasi apakah fungsionalitas aplikasi sudah sesuai dengan hasil yang diinginkan. Hasil pengujian tersaji dalam tabel-tabel berikut.

Tabel 1. Pengujian untuk modul pengguna

API	Kondisi	Hasil	Status
signup	Data terisi lengkap	Data tersimpan ke <i>database</i>	Berhasil
	Terdapat <i>field</i> kosong	Muncul <i>pop up error</i>	
	Email atau username yang dimasukkan sudah ada	Pengguna diminta menggunakan email atau username lain	
signin	Email/username dan password benar	Login sukses	Berhasil
	Password salah	Muncul pop up password salah	
	Email atau username salah	Muncul pop up email atau username tidak ditemukan	
edit	Pengguna memasukkan username baru	Username diperbarui	Berhasil
getUser	Pengguna memiliki akun	Menampilkan profil pengguna	Berhasil

Tabel 2. Pengujian Modul Buah

API	Kondisi	Hasil	Status
predict	Pengguna mengunggah foto buah	Menampilkan informasi dan rekomendasi olahan buah	Berhasil
get Products	Terdapat olahan buah	Menampilkan semua olahan buah yang tersedia	Berhasil
get Product ById	Memilih olahan yang diinginkan	Menampilkan detail dan resep olahan	Berhasil
get Product ByFruit	Memilih buah yang diinginkan	Menampilkan daftar olahan terkait buah tertentu	Berhasil
get Articles	Terdapat artikel buah	Menampilkan semua artikel yang tersedia	Berhasil
get Article ById	Memilih artikel yang diinginkan	Menampilkan detail dan resep olahan	Berhasil

Tabel 3. Pengujian Modul Favorit

API	Kondisi	Hasil	Status
add Favorite	Pengguna memilih olahan yang ingin ditambahkan ke favorit	Olahan ditambahkan ke daftar favorit	Berhasil
delete Favorite	Pengguna memilih olahan yang ingin dihapus dari favorit	Olahan dihapus dari daftar favorit	Berhasil
get Favorite	Pengguna telah menambahkan olahan ke dalam favorit	Menampilkan daftar favorit	Berhasil

### Penerapan (*Deployment*)

*Deployment* API yang telah dibuat memanfaatkan layanan App Engine pada Google Cloud Platform, proses *deployment* bertujuan agar API dapat diakses dan diimplementasikan oleh bagian *front-end*. Proses *deployment* pada App Engine ditunjukkan pada gambar 6.

```

EXPLORER
  > OPEN EDITORS
  DANARHADIB
  > API
  > node_modules
  > Routes
  app.yaml
  package-lock.json
  package.json
  request.rest
  server.js
  storageCredentials.json
  README-cloudshell.txt

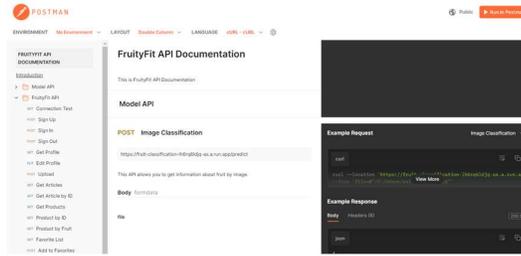
app.yaml x
API > app.yaml > ...
1 runtime: nodejs20
2 service: backendv2
3 env: standard
4 instance_class: F1
5 automatic_scaling:
6   min_idle_instances: 1
7   max_idle_instances: automatic
8   min_pending_latency: automatic
9   max_pending_latency: automatic
10

Problems dev-test-405901 x
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to dev-test-405901.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
danarhadib@cloudshell:~ (dev-test-405901)$ cd API
danarhadib@cloudshell:~/API (dev-test-405901)$ gcloud app deploy

```

Gambar 6. *Deployment* pada App Engine

Selanjutnya yaitu membuat dokumentasi API. Dokumentasi API dapat digunakan untuk membantu developer mengakses *endpoint* yang ada dengan lebih mudah. Postman merupakan *tools* yang dapat digunakan dalam pengembangan API. Selain bisa digunakan pengujian API, postman juga memiliki fitur untuk membantu dokumentasi API secara mendalam.

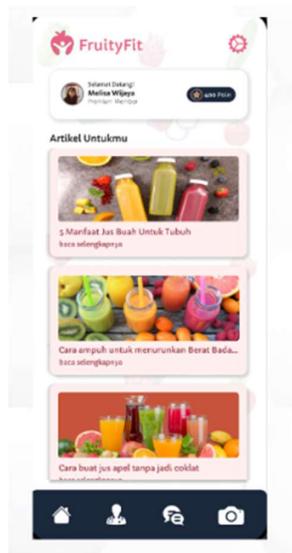


Gambar 7. Dokumentasi API

Terakhir, API yang telah dibuat diimplementasikan ke bagian front-end. Berikut merupakan hasil *consume* API yang telah dilakukan.



Gambar 8. Laman Sign In



Gambar 9. Laman Beranda



Gambar 10. Laman Hasil Klasifikasi



Gambar 11. Laman Olahan Buah

## KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa perancangan dan pengembangan *back-end* API aplikasi FruityFit berhasil diselesaikan. Proses pengembangan dilakukan menggunakan metode *waterfall* dan diimplementasikan menggunakan *framework* ExpressJS. Arsitektur sistem *cloud* memanfaatkan beberapa layanan Google Cloud Platform (GCP) seperti App Engine, Cloud Run, Cloud SQL, dan Cloud Storage. Pengujian yang telah dilakukan menunjukkan bahwa keseluruhan API berfungsi dengan baik dan didokumentasi untuk memudahkan bagian *front-end* dalam mengakses API.

## DAFTAR PUSTAKA

- A. C. Rompis, “Perbandingan Performa Kinerja Node.js, PHP, dan Python dalam Aplikasi REST Performance Comparison of Node.js, PHP, and Python Performance,” *Cogito Smart Journal*, vol. 4, no. 1, 2018.
- A. Mubariz et al., “Perancangan Back-End Server Menggunakan Arsitektur Rest dan Platform Node.JS (Studi Kasus: Sistem Pendaftaran Ujian Masuk Politeknik Negeri Ujung Pandang),” 2020.
- D. Silvi Purnia, A. Rifai, and S. Rahmatullah, “Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Aplikasi Bantuan Sosial Berbasis Android,” 2019.
- E. Nurhayati and A. Agussalim, “Rancang Bangun Back-end API pada Aplikasi Mobile AyamHub Menggunakan Framework Node JS Express,” *Jurnal Sistem dan Teknologi Informasi (JustIN)*, vol. 11, no. 3, p. 524, Jul. 2023, doi: 10.26418/justin.v11i3.66823.
- I. Ahmad Faruqi et al., “Perancangan Back-End Aplikasi Rumantara Dengan Gaya Arsitektur Rest Menggunakan Metode Iterative Incremental,” 2018.
- M. Qibtiyah, C. Rosidati, and M. H. Siregar, “Perilaku Konsumsi Buah dan Sayur Pada Remaja,” *Jurnal Gizi Kerja dan Produktivitas*, vol. 2, no. 2, p. 51, Nov. 2021, doi: 10.52742/jgkp.v2i2.12760.
- N. Ramsari and A. Ginanjar, “Implementasi Infrastruktur Server Berbasis Cloud Computing Untuk Web Service Berbasis Teknologi Google Cloud Platform,” 2022, doi: 10.28989/senatik.v7i1.472.
- R. Fajrin, “Pengembangan Sistem Informasi Geografis Berbasis Node.Js Untuk Pemetaan Mesin Dan Tracking Engineer Dengan Pemanfaatan Geolocation Pada Pt Ibm Indonesia,” 2017.
- R. Gunawan, “JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service,” *Jurnal Edukasi dan Penelitian Informatika*, vol. 5, no. 1, pp. 74–79, Apr. 2019.
- R. Pangestika and R. T. Dirgahayu, “Pengembangan Back-end Sistem Informasi Pendataan Sekolah Desa Komunitas Pendar Foundation Yogyakarta,” 2020.
- S. Nugraha, A. B. Prasetijo, and D. Eridani, “Perancangan Back-End Aplikasi Reservasi Talanoa Kopi and Space Menggunakan Framework Express.js Back End Design of the Talanoa Kopi and Space Reservation Application Using Express.js Framework,” *Jurnal Teknik Komputer*, vol. 1, no. 3, pp. 126–131, 2022, doi: 10.14710/jtk.v1i3.36901.
- S. Sauda and M. Barokah, “Penerapan Nodejs Dan Postgresql Sebagai Backend Pada Aplikasi Ecommerce Localla”, doi: 10.31949/infotech.v8i2.2944.