



REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android

Irfan Kurniawan[#], Humaira[#], Fazrol Rozi [#]

[#]Jurusan Teknologi Informasi, Politeknik Negeri Padang, Limau Manis, Padang, 25164, Indonesia
E-mail: humaira@pnp.ac.id, fazrol@pnp.ac.id

ABSTRACTS

Previously, sales and service offerings were only carried out directly, which of course made consumers feel less interested, because they had to exhaust themselves to go to places of people who opened services. From this problem, a system regarding service transactions was developed based on the Application programming Interface (API) as the backend and implemented on the mobile android as the frontend. In this final project, produce an API-based system with REST architecture in terms of backend to facilitate the service transaction process and apply it to an android application as a user interface.

ABSTRAK

Sebelumnya penjualan dan penawaran jasa hanya dilakukan secara langsung, yang tentunya ini membuat konsumen merasa kurang berminat, karena harus menguras tenaga pergi ketempat orang yang buka jasa. Dari permasalahan ini maka dikembangkan sebuah sistem mengenai transaksi jasa dengan basis Application Programming Interface (API) sebagai backend dan diimplementasikan ke mobile android sebagai frontend. Dalam tugas akhir ini menghasilkan sistem berbasis API dengan arsitektur REST dari segi backend untuk memudahkan dalam proses transaksi jasa dan diterapkan pada aplikasi android sebagai antarmuka pengguna.

KATA KUNCI

*API,
REST,
REST API,
Transaksi Online,
Penawaran Jasa Online,
Client Server*

1. PENDAHULUAN

Pada umumnya penjualan dan penawaran jasa hanya dilakukan secara langsung atau melalui perantara, yang tentunya ini membuat konsumen merasa kurang berminat, karena harus menguras tenaga ketempat pembelian barang atau pergi ke tempat orang yang buka jasa, walaupun bisa melalui perantara tetapi gambaran tentang apa yang dibutuhkan bisa tidak sesuai dengan selera. Dengan berkembangnya teknologi informasi sekarang ini, membuat banyak bermunculannya gagasan baru tentang transaksi barang dan penawaran jasa yang bisa dilakukan secara online. Konsumen dapat melihat langsung jenis jasa apa saja yang ditawarkan dari suatu barang. Kemudian konsumen dapat melakukan proses transaksi dari mana saja. Untuk mengatasi masalah tersebut, menganggap perlu dibuatnya sistem mengenai transaksi jasa dengan basis Application Programming Interface (API) sebagai backend dan diimplementasikan ke mobile android sebagai frontend.

Backend sendiri adalah yang bertanggung jawab mengenai sisi server dan database. Dari itu peneliti mengangkat Representational State Transfer (REST) Application Programming Interface (API) untuk pengembangan dari segi backend dan aplikasi mobile android sebagai fontend sebagai transaksi jasa. REST sendiri merupakan sebuah standar arsitektur komunikasi yang sering diterapkan dalam pengembangan layanan berbasis web dan aplikasi, sedangkan API adalah suatu penghubung yang memungkinkan suatu aplikasi untuk berinteraksi dengan aplikasi lainnya dan berbagi data.

Application Programming Interface (API) merupakan antarmuka yang dibangun oleh pengembang sistem supaya sebagian atau keseluruhan fungsi sistem dapat diakses secara programmatic. Penelitiannya menunjukkan

bawa pengembangan API berhasil dilakukan dan REST yang diterapkan mempermudah pengembangan struktur API. Dengan adanya bantuan dari API, konsumen dapat melakukan registrasi atau login ke aplikasi hanya dengan email ke aplikasi tersebut dan memudahkan dalam proses transaksi jasa. API juga dapat berperan memudahkan konsumen untuk melakukan berbagai metode pembayaran sehingga proses pembayaran akan lebih cepat dan mudah.

2. METODOLOGI PENELITIAN

1. Pemrograman Frontend

Frontend developer menghubungkan sebuah situs atau aplikasi dengan pengguna. Secara umum, ia membuat teks, gambar, tombol, dan menu serta interaksi antara situs atau aplikasi dengan pengguna. Oleh karena itu, peran ini juga biasa disebut dengan client-side. Frontend tidak benar-benar merancang desain dari sebuah situs atau aplikasi. Pasalnya, hal tersebut adalah tugas UI designer. Frontend bertugas untuk memindahkan desain dari UI designer ke dalam bentuk yang interaktif dan membuat desain tersebut menjadi hidup.

Alat Bantu Pemrograman Frontend yang digunakan:

1. Java

Java adalah suatu bahasa pemrograman yang dapat membuat seluruh bentuk aplikasi, desktop, web, mobile dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. Bahasa Pemrograman Java ini berorientasi objek Object Oriented Programming (OOP), dan dapat dijalankan pada berbagai platform sistem operasi,[3]

Bahasa pemrograman java adalah perkembangan dari bahasa pemrograman C++ yang dimana bahasa pemrograman java jauh lebih dapat diandalkan sehingga java bisa membuat seluruh bentuk pemrograman dari aplikasi desktop hingga ke aplikasi mobile, sehingga java sangat cocok digunakan sebagai bahasa pemrograman untuk kalangan programmer yang menggunakanannya.

2. Android Studio

Android Studio adalah lingkungan pengembangan terpadu, Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor code IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan lebih banyak fitur. Pertama kali Android Studio diumumkan di Google I/O Conference pada tahun 2013 dan dirilis ke publik pada tahun 2014. Sebelum lahirnya Android Studio, aplikasi pada Android dikembangkan dengan Eclipse IDE yaitu IDE Java. Setelah adanya android studio yang open source dapat memudahkan bagi Anda yang ingin membuat aplikasi dengan Android Studio.

2. Pemrograman Backend

Backend adalah bagian belakang layar dari sebuah website atau aplikasi. Bahasa pemrograman untuk backend development diantaranya adalah PHP, Ruby, Python, dan banyak lainnya. Mereka adalah mesin yang bekerja di balik layar, semua yang tidak dilihat oleh pengguna akhir atau berinteraksi langsung, tetapi itu memberi kekuatan pada apa yang terjadi. Backend fokus pada database, scripting, dan arsitektur dari sebuah website atau aplikasi. Kode yang ditulis oleh mereka akan membantu mengomunikasi database informasi kepada browser

Alat Bantu Pemrograman Backend yang digunakan :

1. Application Programming Interface (API)

API adalah antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari sebuah program. API memungkinkan pengembang untuk memakai fungsi yang sudah ada dari aplikasi lain sehingga tidak perlu membuat ulang dari awal. Pada konteks web, API merupakan pemanggilan fungsi lewat Hyper Text Transfer Protocol (HTTP) dan mendapatkan respon berupa Extensible Markup Language (XML) atau JavaScript Object Notation (JSON).[5] Tujuan penggunaan dari API adalah untuk saling berbagi data antar aplikasi yang berbeda tersebut, Tujuan penggunaan API lainnya yaitu untuk mempercepat proses pengembangan aplikasi dengan cara menyediakan sebuah function yang terpisah sehingga para developer tidak perlu lagi membuat fitur yang serupa. API yang bekerja pada tingkat sistem operasi membantu aplikasi berkomunikasi dengan layer dasar dan satu sama lain mengikuti serangkaian protokol dan spesifikasi. Contoh yang dapat menggambarkan spesifikasi tersebut adalah POSIX (Portable Operating System Interface). Dengan menggunakan standar POSIX, aplikasi yang di-compile untuk bekerja pada sistem operasi tertentu juga dapat bekerja pada sistem lain yang memiliki kriteria yang sama. Software library juga memiliki peran penting dalam menciptakan compatibility antar sistem yang berbeda.

2. Representational State Transfer (REST)

REST merupakan seperangkat prinsip arsitektur yang melakukan transmisi data melalui antarmuka yang terstandarisasi seperti HTTP. REST API bekerja layaknya seperti aplikasi web biasa. Client dapat mengirimkan permintaan kepada server melalui protokol HTTP dan kemudian server memberikan respons balik kepada klien. REST dikembangkan oleh Roy Fielding yang merupakan co-founder dari Apache HTTP Server Project.[6]

Pada arsitektur REST, REST server menyediakan resources (sumber daya/data) dan REST client mengakses dan menampilkan resource tersebut untuk penggunaan selanjutnya. Setiap resource diidentifikasi oleh URIs (Universal Resource Identifiers) atau global ID. Resource tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML.

3. JavaScript Object Notation (JSON)

JSON adalah sebuah format untuk berbagi data. Seperti dapat kita lihat dari namanya, JSON diturunkan dari bahasa pemrograman javaScript, akan tetapi format ini tersedia bagi banyak bahasa lain termasuk Python, Ruby, PHP, dan Java. JSON biasanya dilafalkan seperti nama "Jason." JSON menggunakan ekstensi .json saat ia berdiri sendiri. Saat didefinisikan di dalam format file lain (seperti di dalam .html), ia dapat tampil didalam tanda petik sebagai JSON string, atau ia dapat dimasukkan kedalam sebuah variabel. Format ini sangat mudah untuk ditransfer antar server web dengan klien atau browser.

4. NodeJS

Node.js adalah perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis dalam sintaks bahasa pemrograman JavaScript. Bila selama ini kita mengenal JavaScript sebagai bahasa pemrograman yang berjalan di sisi client / browser saja, maka Node.js ada untuk melengkapi peran JavaScript sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi server, seperti halnya PHP, Ruby, Perl, dan sebagainya. Node.js dapat berjalan di sistem operasi Windows, Mac OS X dan Linux tanpa perlu ada perubahan kode program. Node.js memiliki pustaka server HTTP sendiri sehingga memungkinkan untuk menjalankan server web tanpa menggunakan program server web seperti Apache atau Nginx.

Perilisan Node.js dibagi menjadi 3 fase, yaitu:

1. Current (saat ini),
2. Long Term Support / LTS (dukungan aktif jangka panjang)
3. Maintenance (pemeliharaan).

5. Sequelize

Sequelize adalah ORM Node.js yang berbasis promise. Ini bisa digunakan dengan PostgreSQL, MySQL, MariaDB, SQLite, dan MSSQL. Fitur ini mendukung transaksi yang solid, hubungan, lazy loading dan eager, membaca replikasi dan banyak lagi.

6. JSON Web Token (JWT)

JWT ini adalah sebuah token berbentuk string panjang yang sangat random yang gunanya sendiri untuk melakukan sistem Autentikasi dan Pertukaran Informasi. Umumnya untuk melakukan login tidak seperti pada aplikasi website biasa dimana kita menggunakan session untuk mengingat siapa yang sedang login. Tapi didalam API sendiri kita menggunakan konsep JWT atau dibacanya sebagai "jot". Website resminya ada di jwt.io

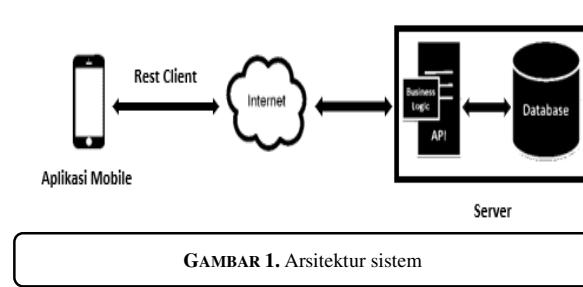
JWT terdiri dari tiga struktur yang dipisahkan oleh tanda titik (.), yaitu:

1. Header untuk memuat jenis encoding yang digunakan.
2. Payload untuk memuat nilai-nilai informasi yang ditransaksikan.
3. Signature untuk memuat nilai hash untuk memverifikasi payload

7. Dotenv

File.env (dotenv) berfungsi untuk menyimpan variabel env, file ini berisi deklarasi atau pembuatan variabel env yang nantinya dapat kita load dari Nodejs. Agar dapat menggunakan file .env, kita membutuhkan modul dotenv. Modul ini nanti akan membantu kita untuk me-load semua variabel yang ada di file .env.

3. Arsitektur Sistem

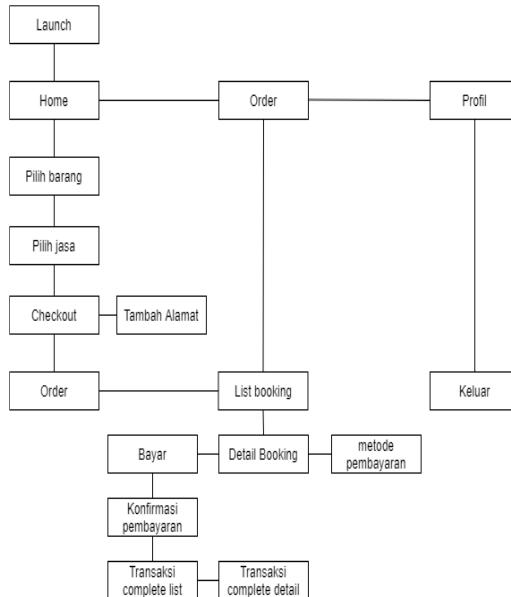


Aplikasi client (android) yang di bangun pada penelitian ini terintegrasi oleh Application Programming Interface yang menggunakan metode REST. Dapat dilihat pada gambar 1 merupakan skema pengambilan data menggunakan REST API. Data jasa yang sudah dipesan oleh pengguna nantinya akan digabungkan dan dikirim dalam bentuk json. Kemudian data tersebut akan dikirim ke server API. Server API akan melakukan proses permintaan ke database.

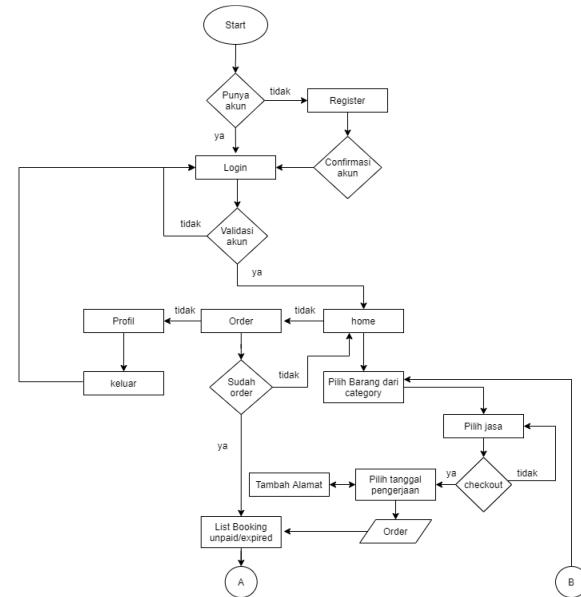
Yang kemudian hasil dari proses permintaan akan dikembalikan lagi ke pengguna dalam bentuk json dan diuraikan oleh aplikasi sehingga data hasil proses bisa digunakan dan ditampilkan. Perancangan dimulai dari perancangan Application Programming Interface (API) dengan REST sebagai arsitektur yang dipakai untuk menghubungkan antara frontend dalam hal ini adalah aplikasi mobile berbasis android dengan database yang memuat segala data yang digunakan. API yang sudah dibuat akan dilakukan uji coba terlebih dahulu dan dites

menggunakan aplikasi Postman yang merupakan sebagai salah satu tools penguji API, lalu dibangun content management system (CMS) sebagai website admin yang bertujuan untuk mengelola content yang akan dipakai untuk aplikasi nantinya, kemudian desain antarmuka untuk aplikasi sesuai dengan konsep API yang telah dibuat sebelumnya.

4. Perancangan Navigasi dan Flowchart



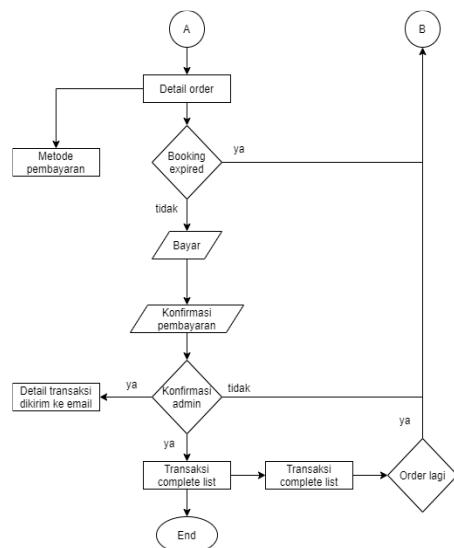
GAMBAR 2. Perancangan navigasi untuk aplikasi



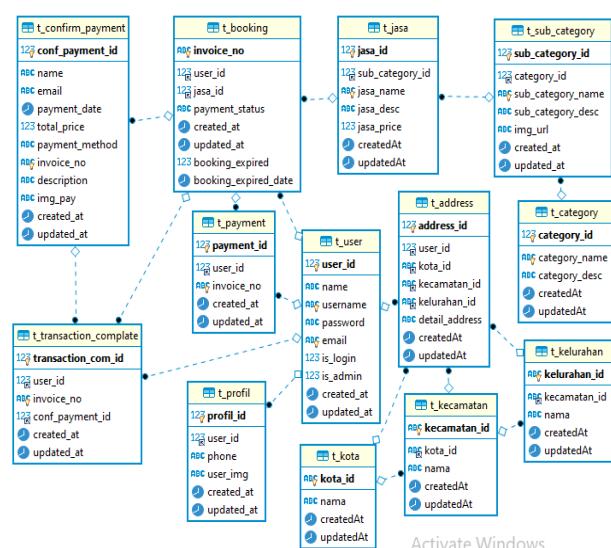
GAMBAR 3. Flowchart aplikasi bagian 1

Dapat dilihat pada gambar 2 struktur perancangan navigasi bahwa perpindahan antar fitur dapat dilakukan dengan mudah melalui menu dan pada android studio terdapat fitur Bottom Navigation sehingga keseluruhan menu dapat dilihat pada bagian bawah aplikasi,

Flowchart aplikasi merupakan alur keseluruhan dari sebuah aplikasi yang dibuat dan dikembangkan, dimulai dari halaman login dan register, kemudian masuk ke halaman menu utama yang terdapat 3 menu navigasi, yakni menu home, menu order, dan menu profil, menu home diisi dengan barang yang ditawarkan lalu pemilihan jasa kemudian melakukan proses order yang ditujukan ke menu order sebagai list booking. Menu order melakukan proses transaksi dengan akhir mendapatkan list transaksi complete, keseluruhan proses dari aplikasi dapat dilihat melalui alur gambar 3



GAMBAR 4. Flowchart aplikasi bagian 2



GAMBAR 5. Perancangan database untuk menampung data

Dari database yang dibuat dapat dihubungkan dengan frontend menggunakan API yang nanti akan diterapkan pada Android Studio, API disini berperan sebagai interface penghubung antara database dengan forntend dan merupakan otak dari pembuatan aplikasi.

5. Perancangan API

TABEL 1. Perancangan PI

No	Method	Kandidat API URI	Entitas	Proses
1	POST	/auth	User	User melakukan login dengan authentikasi
2	GET	/user/<user_id>	User	Menampilkan data user berdasarkan user id
3	GET	/user	User	Menampilkan daftar semua user
4	POST	/user	User	Mendaftarkan user ke database
5	PUT	/user/<user_id>	User	Mengedit user berdasarkan user id
6	DELETE	/user/<user_id>	User	Menghapus user berdasarkan user id
7	GET	/user/active-account/<user_id>	User	Melakukan aktifasi pada akun user yang terdaftar
8	POST	/super-user	User	Menjadikan akun user menjadi admin
9	GET	/category	Category	Menampilkan daftar semua kategori
10	GET	/category/<category_id>	Category	Menampilkan data kategori berdasarkan category id
11	POST	/category	Category	Menambahkan kategori baru
12	PUT	/category/<category_id>	Category	Mengedit kategori berdasarkan category id
13	DELETE	/category/<category_id>	Category	Menghapus kategori berdasarkan category id
14	GET	/sub-category	Sub_category	Menampilkan daftar semua sub kategori
15	GET	/sub_category/<sub_category_id>	Sub_category	Menampilkan data kategori berdasarkan sub
16	POST	/sub_category	Sub_category	Menambahkan kategori baru
17	PUT	/sub_category/<sub_category_id>	Sub_category	Mengedit kategori berdasarkan sub category id
18	DELETE	/sub_category/<sub_category_id>	Sub_category	Menghapus kategori berdasarkan sub category id
19	GET	/jasa	Jasa	Menampilkan daftar semua jasa
20	GET	/jasa/<jasa_id>	Jasa	Menampilkan data jasa berdasarkan jasa id
21	POST	/jasa	Jasa	Menambahkan jasa baru
22	PUT	/jasa/<jasa_id>	Jasa	Mengedit kategori berdasarkan jasa id
23	DELETE	/jasa/<jasa_id>	Jasa	Menghapus jasa berdasarkan jasa id
24	GET	/profil/<user_id>	Profil	Menampilkan data user berdasarkan user id
25	PUT	/update-profil	Profil	Mengedit profil user
26	POST	/delete-profil-picture	Profil	Menghapus foto profil user
27	POST	/kota	Kota	Membuat kota
28	GET	/kota	Kota	Menampilkan daftar semua kota
29	GET	/kota-detail/<kota_id>	Kota	Menampilkan kota berdasarkan kota id
30	POST	/kecamatan	Kecamatan	Membuat kecamatan
31	GET	/kecamatan	Kecamatan	Menampilkan daftar semua kecamatan
32	GET	/kecamatan/<kota_id>	Kecamatan	Menampilkan daftar semua kecamatan berdasarkan
33	GET	/kecamatan-detail/<kecamatan_id>	Kecamatan	Menampilkan kecamatan berdasarkan kecamatan id
34	POST	/kelurahan	Kelurahan	Membuat kelurahan
35	GET	/kelurahan	Kelurahan	Menampilkan daftar semua kelurahan
36	GET	/kelurahan/<kecamatan_id>	Kelurahan	Menampilkan daftar semua kelurahan berdasarkan
37	GET	/kelurahan-detail/<kelurahan_id>	Kelurahan	Menampilkan kelurahan berdasarkan kelurahan id
38	POST	/add-address	Address	Menambahkan alamat
39	GET	/address-list/<user_id>	Address	Menampilkan alamat user
40	POST	/address-detail	Address	Menampilkan detail alamat user
41	PUT	/address-update	Address	Mengedit alamat
42	DELETE	/address-delete	Address	Menghapus alamat
43	POST	/booking	Booking	Membuat order
44	GET	/booking-list	Booking	Menampilkan semua booking
45	GET	/booking-detail/<invoice_no>	Booking	Menampilkan booking berdasarkan invoice no
46	GET	/booking-list-user/<user_id>	Booking	Menampilkan daftar booking berdasarkan user id
47	POST	/booking-update-status	Booking	Mengubah status booking
48	POST	/payment	Payment	Mengirim payment method ke email user
49	POST	/confirm-payment	Confirm_payment	Membuat konfirmasi pembayaran
50	GET	/confirm-payment-list	Confirm_payment	Menampilkan daftar konfirmasi pembayaran
51	GET	/confirm-payment-detail/<invoice_no>	Confirm_payment	Menampilkan konfirmasi pembayaran Berdasarkan invoice no
52	GET	/transaction-complete-list	Transaction_complete	Menampilkan semua daftar selesai transaksi
53	GET	/transaction-complete-list/<user_id>	Transaction_complete	Menampilkan semua daftar selesai transaksi Berdasarkan user id
54	GET	/transaction-complete-detail/<invoice_no>	Transaction_complete	Menampilkan detail selesai transaksi Berdasarkan invoice no

3. HASIL DAN PEMBAHASAN

Implementasi dan pengujian pemesanan jasa Online, mulai dari pembuatan backend serta menerapkan ke frontend untuk aplikasi mobile. Implementasi adalah penerapan atau pelaksanaan. Implementasi juga bisa diartikan sebagai tindakan untuk menjalankan rencana yang telah dibuat. Implementasi merupakan tahap lanjutan dari tahapan analisis dan perancangan. Pada tahap ini akan dilakukan implementasi terhadap rancangan yang telah dibuat pada pembahasan sebelumnya. Mulai dari implementasi terhadap pengkodean sistem dalam segi backend dan aplikasi dari implementasi terhadap design interface yang telah dibuat. Lalu tahap selanjutnya yaitu tahap pengujian terhadap sistem yang dibangun guna untuk memastikan sistem serta aplikasi yang dibangun telah sesuai dengan apa yang diharapkan.

REST API

API ini dibuat untuk diimplementasikan pada aplikasi Android, dengan konsep agar aplikasi mendapatkan data dari API yang dibuat dalam android studio. Pada API dibuat models menggunakan sequelize untuk database dan controllers dari models tersebut. API yang dibuat menggunakan NodeJS yang merupakan server-side JavaScript dan terkoneksi pada database.

4. KESIMPULAN

Membangun sistem berbasis API dengan arsitektur REST dari segi backend dan aplikasi android sebagai frontend untuk transaksi jasa secara online. Dalam pembangunan REST API untuk tugas akhir menggunakan nodejs dan melakukan installasi dependencies yang dibutuhkan. User dapat memesan jasa secara online melalui aplikasi dengan memilih jenis jasa berdasarkan barang yang dipilih user serta menambahkan alamat lengkap dan waktu pengerjaan jasa yang dipesan. Sistem yang dibuat untuk pembayaran jasa menggunakan konfirmasi pembayaran melalui aplikasi yang didalamnya terdapat form yang harus diisi terkait pemesanan user dan upload foto struk sebagai bukti pemesanan. Konfirmasi pembayaran yang dilakukan user akan dilakukan check oleh admin melalui CMS, kalau data dan bukti pembayaran asli admin akan menukar booking status user dari unpaid menjadi paid, selanjutnya detail pembayaran user akan dikirim melalui email..

REFERENSI

- [1] Interface (Api) Berbasis Web Menggunakan Gaya Arsitektur Representational State Transfer (Rest) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit,” J. Komputasi, vol. 6, no. 1, pp. 33–42, 2018, doi: 10.23960/komputasi.v6i1.1554.
- [2] P. J. Saputro and N. Nuryantini, “Pembuatan Sistem Layanan Penjualan Berbasis,” vol. 1, no. November, pp. 18–27, 2016.
- [3] A. Kadir, “Buku Pertama Belajar Pemrograman Java Untuk Pemula,” no. January 2014, p. 444, 2014, doi: 10.13140/2.1.3686.2084.
- [4] A. Kadir, “From Zero To A Pro Pemrograman Aplikasi Android,” no. August, p. 408, 2014, doi: 10.13140/2.1.1589.0563.
- [5] S. Surahman and E. B. Setiawan, “Aplikasi Mobile Driver Online Berbasis Android Untuk Perusahaan Rental Kendaraan,” J. Ultim. InfoSys, vol. 8, no. 1, pp. 35–42, 2017, doi: 10.31937/si.v8i1.554.
- [6] M. Akbar, “Pengembangan Restful Api Untuk Application Specific High Level Location Service,” p. 9, 2018.
- [7] R. FAJRIN, “Pengembangan Sistem Informasi Geografis Berbasis NodeJS untuk Pemetaan Mesin dan Tracking Engineer dengan Pemanfaatan Geolocation pada PT IBM Indonesia,” J. Inform., vol. 11, no. 2, pp. 40–47, 2017, doi: 10.26555/jifo.v11i2.a6090.
- [8] Hidra Amnur, “OPEN DATA (OD) EXPLORING THE POTENTIAL OF FOOD INDUSTRY OF PADANG , INDONESIA,” ASCNI-Tech, pp. 191–198, 2016.
- [9] N. Suri, “Bab II Landasan Teori,” J. Chem. Inf. Model., vol. 53, no. 9, pp. 1689–1699, 2019, doi: 10.1017/CBO9781107415324.004..