

IMPLEMENTASI REST API MENGGUNAKAN BAHASA GO UNTUK OPTIMALISASI MANAJEMEN MENU DAN PEMESANAN

Bagus Ario Yudanto¹, Sabdha Putra Laudri², Ismi Felina³, Moh Rizqi Amanusa Abinovan⁴,
Muhamad Fadhil⁵, Raja Fathir Rakha Putra⁶, Dinia Fachriza Ulfa⁷, Wildan Faerus Salzi⁸,
Rahnita Nuzulah⁹

Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer
Universitas Indraprasta PGRI

Jalan Raya Tengah No 80, Kelurahan Gedong, Pasar Rebo, Jakarta Timur

bagusarioyudanto123@gmail.com¹, sabdhacipaku04@gmail.com², ismifelina12@gmail.com³,
novansch@gmail.com⁴, fadhilmuhamad272@gmail.com⁵, rajafathir3@gmail.com⁶,
diniafachrizau@gmail.com⁷, wildanfaerussalzi@gmail.com⁸, rahnita21@gmail.com⁹

Abstrak

Dalam era digital yang terus berkembang, industri restoran dituntut untuk mengadopsi teknologi canggih guna meningkatkan efisiensi operasional dan pengalaman pelanggan. Artikel ini menjelaskan implementasi REST API dengan menggunakan bahasa pemrograman Go (Golang) dalam pengembangan sistem manajemen menu dan pesanan di restoran. Bahasa Go dipilih karena kemampuannya dalam menangani performa tinggi dan pengolahan data secara paralel, sehingga cocok untuk membangun sistem API yang cepat dan andal. Penelitian ini mencakup tahapan analisis kebutuhan, perancangan sistem, dan pengembangan REST API. Implementasi REST API berbasis Go memungkinkan pengelolaan menu yang fleksibel, pemrosesan pesanan yang cepat, dan integrasi mudah dengan platform lain. Hal ini berdampak positif pada efisiensi operasional restoran dan kepuasan pelanggan. Dengan solusi ini, restoran dapat meningkatkan otomatisasi proses hingga 40% dibandingkan dengan metode manual tradisional, serta mengurangi waktu pemrosesan pesanan hingga 30%. Dengan demikian, restoran dapat meminimalkan keterlibatan manual, serta menyediakan layanan yang lebih responsif dan modern.

Kata Kunci : REST API, Golang, Manajemen Menu Dan Pemesanan

Abstract

In the ever-evolving digital era, the restaurant industry is required to adopt advanced technologies to improve operational efficiency and customer experience. This article describes the implementation of REST API using Go (Golang) programming language in the development of menu and order management system in restaurants. Go was chosen because of its ability to handle high performance and parallel data processing, making it suitable for building fast and reliable API systems. This research includes the stages of needs analysis, system design, and REST API development. The Go-based REST API implementation enables flexible menu management, fast order processing, and easy integration with other platforms. This has a positive impact on restaurant operational efficiency and customer satisfaction. With this solution, restaurants can increase process automation by 40% compared to traditional manual methods, and reduce order processing time by 30%. Thus, restaurants can minimize manual involvement, and provide more responsive and modern services.

Keyword : REST API, Golang, Menu and Order Management

PENDAHULUAN

Dalam era digital saat ini, kebutuhan akan aplikasi yang efisien dan scalable semakin meningkat, terutama pada sektor restoran yang memerlukan sistem manajemen menu dan pemesanan yang optimal. Sistem manual yang masih banyak digunakan sering kali mengakibatkan ketidakefisienan, seperti kesalahan dalam pengelolaan data menu, keterlambatan pemrosesan pesanan, hingga kesulitan dalam integrasi dengan layanan pihak ketiga. Oleh karena itu, diperlukan solusi berbasis teknologi untuk mengatasi permasalahan tersebut (Bororing & Gunawan, 2024). REST API (Representational State Transfer Application Programming Interface) adalah salah satu teknologi yang memungkinkan pengembang untuk membangun aplikasi yang modular, fleksibel, dan mudah diintegrasikan. REST API bekerja dengan arsitektur client-server, di mana client mengirimkan request kepada server, dan server merespons dengan data dalam format yang ringan seperti JSON

(JavaScript Object Notation). Teknologi ini sering digunakan dalam pengembangan aplikasi berbasis web untuk menyediakan antarmuka komunikasi yang terbuka antara client dan server, sehingga mempermudah integrasi dengan berbagai platform dan perangkat (Galindra Wardhana et al., 2020). Pada penelitian ini, REST API diimplementasikan menggunakan bahasa Go (Golang), sebuah bahasa pemrograman yang dikenal dengan performa tinggi, efisiensi, dan dukungan yang baik untuk pengembangan sistem backend. Golang dirancang untuk mendukung concurrency, sehingga sangat cocok untuk menangani banyak request secara bersamaan, sebuah kebutuhan penting dalam sistem manajemen restoran modern (Naufal Al Ghazali et al., 2024). Adapun rumusan masalah yang diangkat yaitu bagaimana implementasi REST API berbasis Go dapat meningkatkan efisiensi sistem manajemen menu dan pemesanan, serta sejauh mana performa dan keandalan API ini berkontribusi terhadap pengalaman pelanggan dan pengelolaan pesanan. Penelitian ini bertujuan untuk mengembangkan dan menganalisis implementasi REST API berbasis Golang dalam sistem manajemen menu dan pemesanan restoran. Tujuannya mencakup peningkatan efisiensi pengelolaan menu, percepatan pemrosesan pesanan, dan kemudahan integrasi dengan layanan pihak ketiga. Manfaatnya tidak hanya sebagai solusi praktis bagi restoran tetapi juga menjadi referensi pengembang aplikasi lain dalam memanfaatkan REST API berbasis Golang untuk sistem yang lebih efisien dan canggih.

PENELITIAN RELEVAN

Penelitian relevan dari Hamidi & Sakti (2023) dengan judul *Implementasi Web Service E-Menu Menggunakan REST API dan QR Code Pada Restoran Tempat Makan*. Studi ini menemukan bahwa aplikasi E-Menu dapat membantu dan memudahkan pelanggan memesan makanan dan minuman di restoran Tempat Makan; penerapan sistem E-Menu di restoran Tempat Makan mengurangi antrian dan mempercepat proses pemesanan; dan sistem E-Menu mengurangi kesalahan komunikasi antara kasir dan kitchen.

Penelitian relevan dari Adinta et al (2024) dengan judul *Implementasi RESTFUL API Pada Sistem Pemesanan dan Pengiriman Makanan*. Hasil dari penelitian ini yaitu implementasi *RESTful API* pada sistem memberikan landasan yang kuat untuk memberikan pengalaman pengguna yang efisien, terstruktur, dan andal dalam dunia pemesanan dan pengiriman makanan online.

Penelitian relevan dari (Muhammad, 2024) dengan judul *Implementasi Arsitektur REST API pada Pengembangan Sistem Transfer Dompot Digital*. Penelitian ini menghasilkan API dengan arsitektur REST sistem transfer dompot digital dengan format penulisan *JavaScript Object Notation* (JSON). *JavaScript Object Nation* (JSON) merupakan format untuk melakukan penyimpanan dan pertukaran informasi data secara terstruktur.

Penelitian relevan dari Sulistiani & Sulisty (2020) dengan judul *Implementasi Web Service Dengan Metode REST Berbasis Golang Pada Layanan Google Cloud Platform di PT Sumber Alfaria Trijaya, Tbk*. Dalam penelitian ini disimpulkan bahwa aplikasi *Web Service* dengan Metode REST berbasis Golang dapat digunakan untuk melakukan sinkronisasi Data Master secara keseluruhan.

Penelitian relevan dari Kurniawan dan Febby (2022) dengan judul *Implementasi Rest Pada Crino .Id*. Hasil dari penelitian ini yaitu penerapan REST API dapat memudahkan pertukaran data atau informasi antar sistem tidak terbatas pada perbedaan teknologi apa yang digunakan. Perbedaan teknologi antar sistem dapat dimengerti oleh sistem lain dengan memanfaatkan REST yang berjalan diatas protokol HTTP sehingga dapat berkomunikasi.

Penelitian relevan dari Nashikhuddin et al (2023) dengan judul *Implementasi Api Restful Dengan Json Web Token (Jwt) Pada Aplikasi E-Commerce Thrifty Shop Untuk Otentikasi Dan Otorisasi Pengguna*. Dalam penelitian ini, implementasi API RESTful dengan JSON Web Token (JWT) pada aplikasi E-commerce Thrifty Shop untuk otentikasi dan otorisasi pengguna memberikan solusi yang efektif dan aman dalam mengelola pengguna dan sumber daya aplikasi. Dengan menggunakan JWT, aplikasi ini akan memastikan bahwa pengguna yang menggunakan layanan tersebut telah diotentikasi dan memiliki izin yang tepat untuk mengakses sumber daya tertentu

Penelitian relevan dari Muharam & Hidayat (2024) dengan judul Pengembangan Aplikasi Back-End E-Commerce Menggunakan Rest Api Golang Untuk Optimalisasi Kinerja Server. Penelitian ini menunjukkan bahwa pengembangan aplikasi back-end e-commerce menggunakan REST API dengan bahasa pemrograman Golang memiliki potensi besar untuk meningkatkan kinerja server di PT Hacktivate Teknologi Indonesia.

METODE PENELITIAN

1. Pendekatan Pengembangan Sistem

Penelitian ini dikembangkan dengan pendekatan berbasis kebutuhan (*requirement-driven development*), dimana setiap fungsionalitas *REST API* dirancang untuk memenuhi kebutuhan pengelolaan menu dan pesanan. Arsitektur *RESTful* digunakan untuk memastikan API memiliki struktur modular, konsisten, dan mudah diintegrasikan ke berbagai platform.

2. Analisis Kebutuhan

Kebutuhan sistem dirumuskan berdasarkan simulasi studi kasus pengelolaan pesanan, di mana API dirancang untuk mendukung beberapa fitur utama seperti

- a. *CRUD (Create, Read, Update, Delete)* untuk data menu.
- b. Pembuatan dan pengelolaan pesanan
- c. Otomasi penanganan status pesanan
- d. Dokumentasi *API* untuk memudahkan integrasi

3. Perancangan Sistem

Perancangan *REST API* melibatkan penentuan struktur *endpoint*, arsitektur sistem, dan format data. *Endpoint* dirancang menggunakan standar REST dengan format data JSON untuk memastikan interoperabilitas. Struktur *endpoint* mencakup:

- a. */admins* : Untuk pengelolaan administratif,
- b. */menus* : Untuk pengelolaan data menu,
- c. */orders* : Untuk pembuatan dan pengelolaan pesanan,
- d. */transactions* : untuk pengelolaan dan riwayat pembayaran

4. Implementasi

REST API dikembangkan menggunakan *Echo Framework* sebagai *framework* utama untuk membangun aplikasi *web* yang ringan dan cepat dalam eksekusi. Basis data yang digunakan adalah *MySQL*, dipilih karena kestabilan dan skalabilitasnya dalam pengelolaan data. Untuk pengelolaan database, digunakan *GORM*, sebuah *ORM (Object Relational Mapping)* di *Go*, yang mempermudah interaksi dengan basis data. Setiap *endpoint API* dilengkapi *middleware* untuk autentikasi menggunakan *JWT* dan validasi input untuk menjamin keamanan data. Proses pembayaran terintegrasi dengan *Midtrans* sebagai *payment gateway*, memastikan transaksi online yang aman. Penyimpanan gambar menu memanfaatkan *Cloudinary* dengan fitur pengelolaan file berbasis cloud dan pengoptimalan otomatis. Struktur modular diterapkan untuk fitur utama seperti manajemen menu, pesanan, dan laporan, memudahkan pengembangan dan pemeliharaan sistem di masa depan.

5. Pengujian Sistem

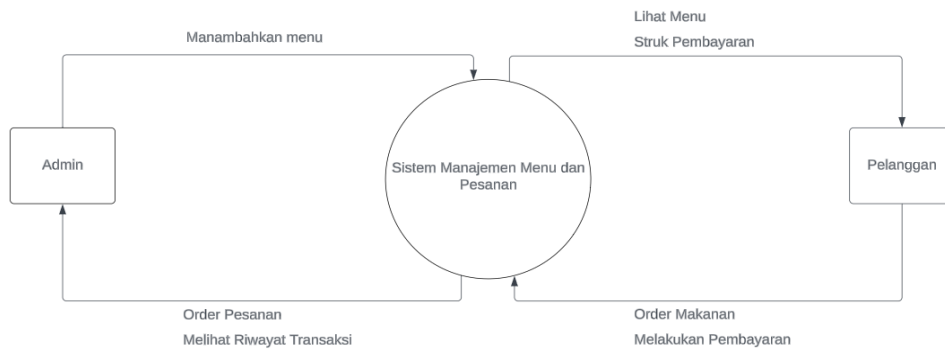
Pengujian *API* dilakukan menggunakan beberapa metode berikut:

- a. *Unit Testing*: Dilakukan dengan menggunakan *framework Testify* untuk memastikan setiap fungsi dalam *API* berjalan sesuai dengan spesifikasi. *Testify* dipilih karena kemampuannya dalam mendukung pengujian berbasis asersi, sehingga memudahkan verifikasi hasil. Contohnya, pengujian mencakup validasi input, autentikasi token, dan logika bisnis seperti kalkulasi total pembayaran.

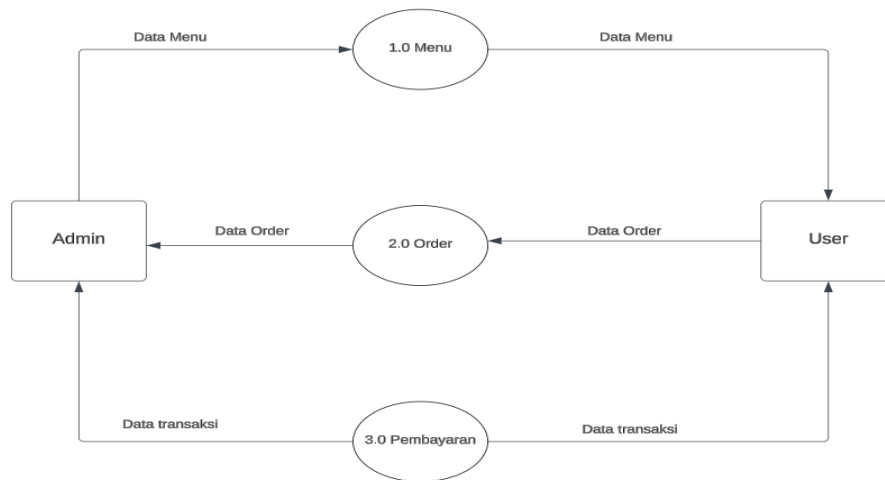
- b. Pengujian Manual: Dilakukan menggunakan *Postman* untuk memverifikasi skenario penggunaan *API* secara langsung, seperti menambahkan menu baru, memproses pesanan, mengunggah foto menu ke *Cloudinary*, dan menghasilkan laporan pesanan.

HASIL DAN PEMBAHASAN

Bagian ini mencakup analisis, hasil implementasi atau pengujian, dan pembahasan dari topik penelitian, yang bisa dibuat terlebih dahulu metodologi penelitian. *Data flow diagram* (DFD) adalah alat yang dapat digunakan untuk menggambarkan sistem yang dirancang. Penggambaran DFD diawali dengan diagram konteks (CD). CD menggunakan tiga simbol, yaitu: entitas eksternal, simbol untuk mewakili aliran data dan simbol untuk mewakili proses (Soufitri, 2019). Berikut merupakan gambaran sistem manajemen menu dan pesanan dalam bentuk diagram konteks, diagram nol, dan diagram rinci:

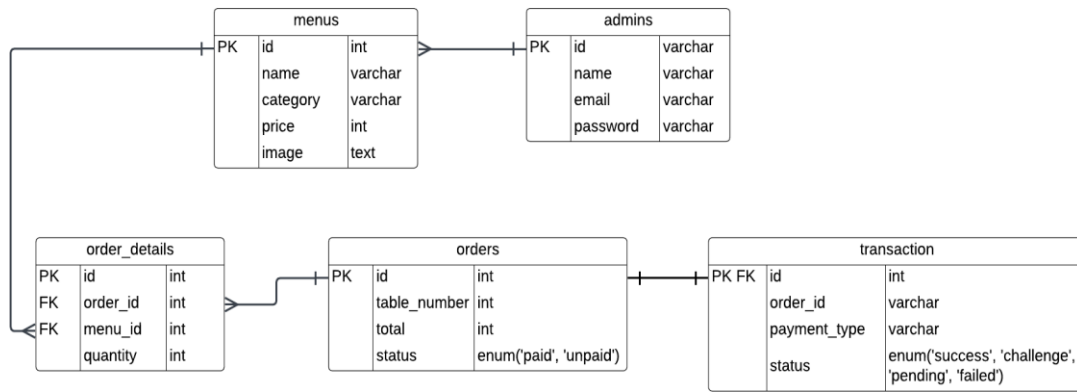


Gambar 1. Diagram Konteks



Gambar 2. Diagram Nol

Bagian ini juga memuat uraian dalam bentuk gambar, tabel, dan lainnya. Gambar 3 berikut ini menjelaskan gambaran dari *ERD* (*Entity Relationship Diagram*) pada struktur *database* yang digunakan.



Gambar 3. Entity Relationship Diagram

Implementasi *REST API* menggunakan bahasa *Go* pada sisi *server*. Selain itu, *Mysql* digunakan sebagai *database* untuk menyimpan data/sumber daya. Pada gambar tabel diatas merupakan struktur *database* yang digunakan untuk menyimpan dan mengelola data pada menu dan pemesanan. Berikut ini penjelasan terkait tiap tabel database yang digunakan :

1. Tabel *menu*

Pada tabel *menu* ini digunakan untuk menyimpan data tentang menu makanan atau minuman. Atribut yang digunakan pada tabel ini terdiri dari *id* sebagai id untuk setiap menu, *name* sebagai nama menu, *category* sebagai kategori menu, *price* sebagai harga menu, dan *image* sebagai deskripsi gambar untuk menu.

2. Tabel *admins*

Pada tabel *admin* ini digunakan untuk menyimpan data administrator. Atribut yang digunakan pada tabel ini terdiri dari *id* sebagai id untuk setiap admin, *name* sebagai nama admin, *email* sebagai alamat email admin, dan *password* sebagai kata sandi untuk melakukan login.

3. Tabel *orders*

Pada tabel *orders* ini digunakan untuk mencatat data pesanan yang dilakukan oleh pelanggan. Atribut yang digunakan pada tabel ini terdiri dari *id* sebagai id untuk setiap pesanan, *table_number* sebagai nomor meja yang memesan, *total* sebagai total untuk harga dari pesanan, dan *status* sebagai status pembayaran pesanan.

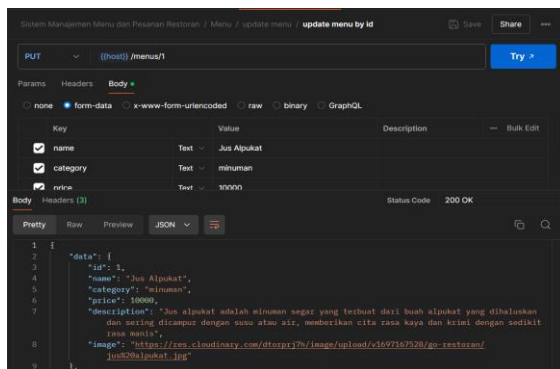
4. Tabel *order_details*

Pada tabel *order_detail* ini digunakan untuk mencatat detail dari setiap pesanan, termasuk menu yang dipesan dan jumlahnya. Atribut yang digunakan pada tabel ini terdiri dari *id* sebagai id untuk setiap detail pesanan, *order_id* yang mengacu pada id pesanan di tabel *orders*, *menu_id* yang mengacu pada id menu di tabel *menus*, dan *quantity* sebagai jumlah menu yang dipesan.

5. Tabel *transactions*

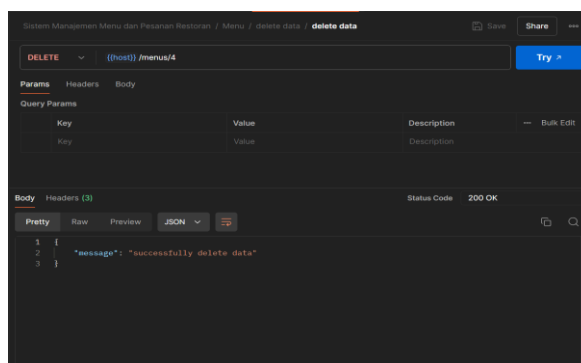
Pada tabel *transaction* ini digunakan untuk mencatat transaksi pembayaran pada setiap pesanan. Atribut yang digunakan pada tabel ini terdiri dari *id* sebagai id untuk setiap transaksi, *order_id* yang mengacu pada id pesanan di tabel *orders*, *payment_type* sebagai jenis pembayaran, dan *status* sebagai status transaksi.

Gambar 6 merupakan tampilan Endpoint menus. Saat admin mengirimkan permintaan menggunakan metode GET ke endpoint menus, server akan merespons dengan mengirimkan data-data menu yang tersedia. Endpoint ini digunakan oleh admin untuk melihat daftar menu.



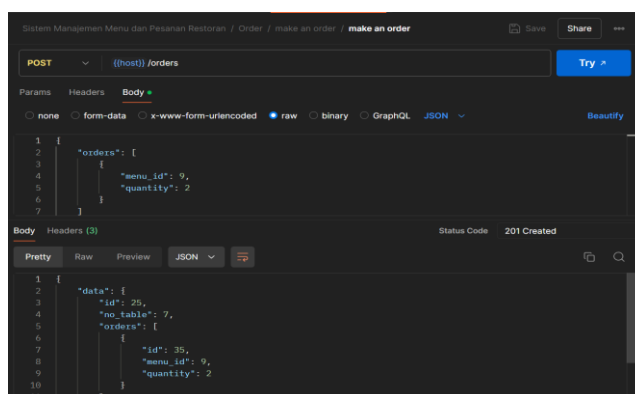
Gambar 7. Endpoint Update Menu

Gambar 7 merupakan tampilan Endpoint menus. Saat admin mengirimkan permintaan menggunakan metode PUT ke endpoint menus, permintaan tersebut berisi data berupa name, category, price, serta parameter id sebagai identifikasi menu yang akan diperbarui. Server akan memproses permintaan dan merespons dengan data menu yang telah diperbarui. Endpoint ini digunakan untuk mengelola proses pembaruan data menu



Gambar 8. Endpoint Delete Menu

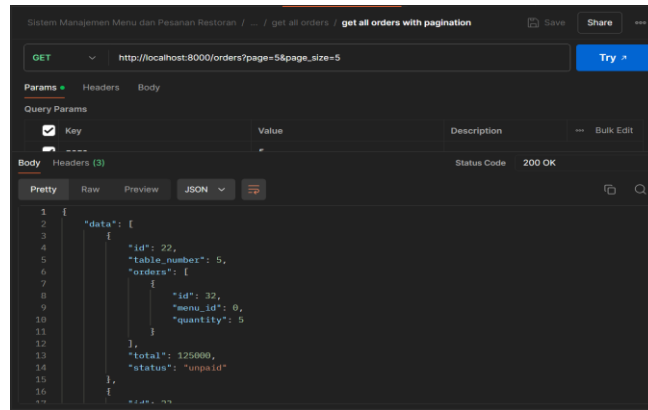
Gambar 8 merupakan tampilan Endpoint menus. Saat admin mengirimkan permintaan menggunakan metode DELETE dengan menyertakan parameter id menu yang ingin dihapus, server akan memproses permintaan tersebut. Jika penghapusan berhasil, server akan merespons dengan pesan: "message": "successfully delete data".



Gambar 9. Endpoint Order Menu

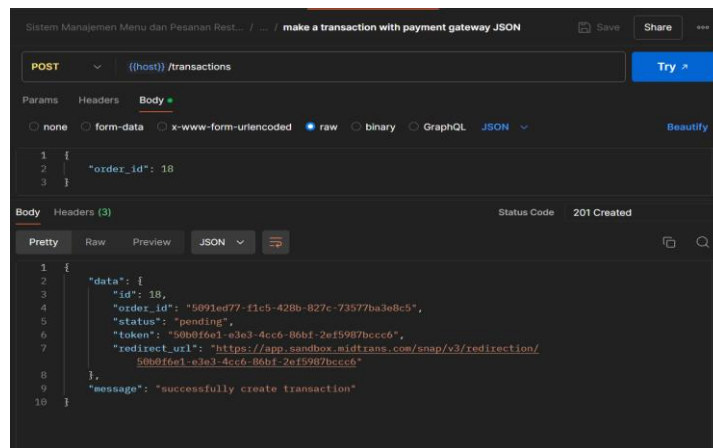
Gambar 9 merupakan tampilan Endpoint orders. Saat admin mengirimkan permintaan menggunakan metode POST ke endpoint orders, permintaan tersebut berisi data pesanan dalam format JSON pada body, yang meliputi menu_id dan quantity untuk setiap pesanan.

Server akan memproses permintaan tersebut dan merespons dengan status kode 201 Created, disertai data pesanan yang telah berhasil ditambahkan, termasuk informasi seperti id pesanan, nomor meja (no_table), dan detail pesanan. Endpoint ini digunakan untuk mencatat pesanan baru pada sistem.



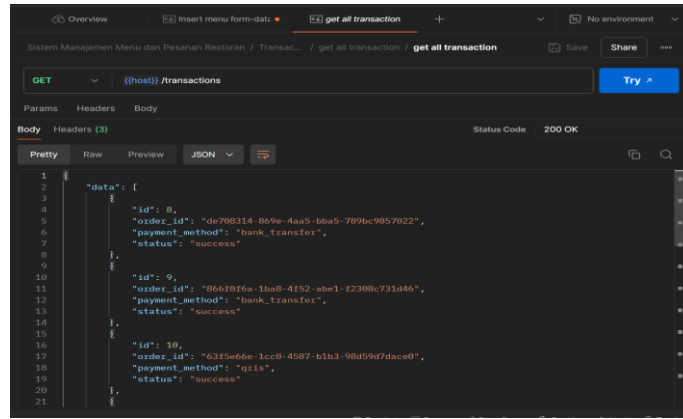
Gambar 10. Endpoint Melihat Semua order

Gambar 10 merupakan tampilan Endpoint orders. Saat admin mengirimkan permintaan menggunakan metode GET ke endpoint orders dengan query parameter page dan page_size, server akan merespons data semua pesanan yang tersedia dengan mekanisme pagination. Data yang diterima meliputi id pesanan, nomor meja (table_number), daftar orders dengan informasi seperti id menu dan jumlah pesanan (quantity), serta detail tambahan seperti total harga (total) dan status pembayaran (status).



Gambar 11. Endpoint Pembayaran

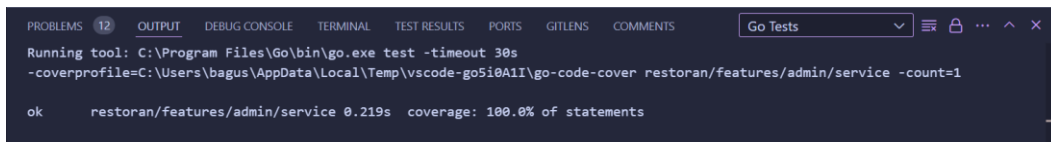
Gambar 11 merupakan tampilan Endpoint transactions. Admin mengirim permintaan menggunakan metode POST dengan order_id dalam body JSON ke endpoint ini. Server merespons dengan data transaksi, termasuk id, order_id, status pending, token, dan redirect_url untuk pembayaran melalui payment gateway, serta pesan "successfully create transaction" dengan status kode 201 Created. Endpoint ini digunakan untuk membuat transaksi baru dan memproses pembayaran.



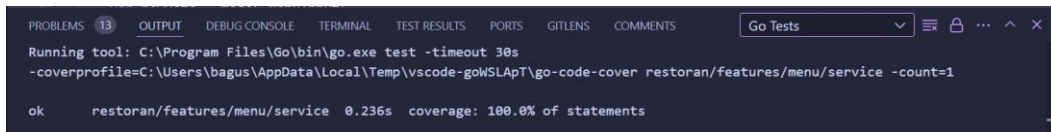
Gambar 12. Endpoint Pembayaran

Gambar 12 menampilkan Endpoint transactions. Admin mengirim permintaan dengan metode GET ke endpoint ini. Server merespons dengan data transaksi, termasuk id, order_id, payment_method, dan status. Endpoint ini digunakan untuk admin melihat semua data transaksi.

Gambar di bawah ini menunjukkan hasil unit testing untuk setiap fitur yang tersedia, yang bertujuan untuk memastikan API berfungsi sesuai dengan spesifikasi yang telah ditentukan.



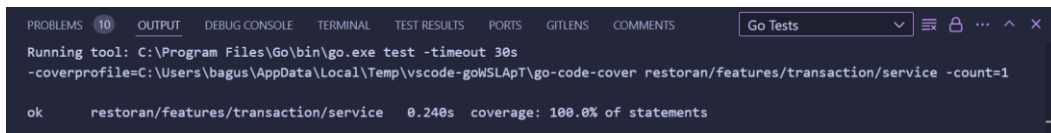
Gambar 13. Hasil Unit Testing fitur admin



Gambar 13. Hasil Unit Testing fitur menu



Gambar 14. Hasil Unit Testing fitur order



Gambar 15. Hasil Unit Testing fitur transaction

SIMPULAN

Penelitian ini menyimpulkan bahwa implementasi REST API berbasis Golang mampu meningkatkan efisiensi manajemen menu dan pemesanan di restoran. REST API memfasilitasi pembaruan data otomatis, mengurangi kesalahan pesanan, mempercepat transaksi, dan meningkatkan responsivitas sistem. Dukungan Golang dengan performa tinggi dan kemampuan menangani banyak permintaan menjadikan sistem lebih optimal dan mendukung pengalaman pelanggan yang lebih baik. Struktur modular REST API mempermudah pengelolaan jangka panjang dengan memisahkan fitur seperti menu, pemesanan, dan transaksi, sehingga

pengembangan lebih fleksibel tanpa mengganggu sistem utama. Pengujian menunjukkan API memenuhi spesifikasi, sementara autentikasi berbasis JSON Web Token (JWT) meningkatkan keamanan data dan otorisasi pengguna. Sistem ini membantu restoran meningkatkan efisiensi operasional, mengurangi antrian, dan memaksimalkan potensi digital, memberikan wawasan penting bagi sektor lain yang membutuhkan solusi teknologi serupa.

DAFTAR PUSTAKA

- Adinta, B., Winarsih, & Ningsih, S. (2024). *Implementasi Restful Api Pada Sistem Pemesanan Dan Pengiriman Makanan*. 5(1), 41–54. <https://doi.org/10.55122/junsibi.v5i1.1173>
- Bororing, G. M. G., & Gunawan, F. (2024). Sistem Pemesanan dan Pembayaran Makanan berbasis Web Terintegrasi dengan Application Programming Interface (API). *Jurnal Informatika Dan Bisnis*, 13(1), 37–48. <https://doi.org/10.46806/jib.v13i1.1147>
- Galindra Wardhana, W., Arwani, I., & Rahayudi, B. (2020). Implementasi Teknologi Restful Web Service Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 4(2), 680–689. <http://j-ptiik.ub.ac.id>
- Hamidi, & Sakti, D. V. S. Y. (2023). *Implementasi Web Service E-Menu Menggunakan Rest*. 4(1).
- Kurniawan, Febby, I. V. P. (2022). *Implementasi Rest Pada Crino*. Id.
- Muhammad, I. R. D. (2024). *Implementasi Arsitektur REST API pada Pengembangan Sistem Transfer Dompot Digital*. <https://dspace.uui.ac.id/handle/123456789/48788>
- Muharam, Y., & Hidayat, T. (2024). *Pengembangan Aplikasi Back-End E-Commerce Menggunakan Rest Api Golang Untuk Optimalisasi Kinerja Server*. 11, 7–13.
- Nashikhuddin, A. Y., Karaman, J., & Litanianda, Y. (2023). Implementasi Api Restful Dengan Json Web Token (Jwt) Pada Aplikasi E-Commerce Thrifty Shop Untuk Otentikasi Dan Otorisasi Pengguna. *METHOMIKA Jurnal Manajemen Informatika Dan Komputerisasi Akuntansi*, 7(2), 239–246. <https://doi.org/10.46880/jmika.vol7no2.pp239-246>
- Naufal Al Ghazali, M., Azizi, A., Khair, O. S., & Saifullah Kusnandar, Z. T. (2024). Pengembangan Dashboard Admin Bukupedia. *Jurnal Teknik Indonesia*, 3(1), 11–21. <https://doi.org/10.58860/jti.v3i1.320>
- Soufitri, F. (2019). Perancangan Data Flow Diagram Untuk Sistem Informasi Sekolah (Studi Kasus Pada Smp Plus Terpadu). *Ready Star*, 2(1), 240–246.
- Sulistiani, W., & Sulisty, W. (2020). Implementasi Web Service Dengan Metode REST Berbasis Golang Pada Layanan Google Cloud Platform di PT Sumber Alfaria Trijaya, Tbk. *Indonesian Journal of Economics, Entrepreneurship and Innovatio*0, 2(1), 1–6.