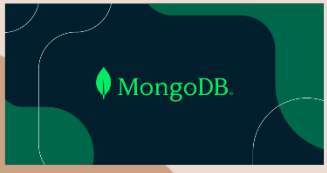
PEMANFAATAN

TEKNOLOGI PEMROGRAMAN DALAM MEMBANGUN SISTEM INFORMASI WISATA WEGOTOUR

Daffa Audya Pramana
Prisya Haura Febrianti
Noviana Riza
Rolly Maulana Awangga



Pemanfaatan Teknologi Pemrograman Dalam Membangun Sistem Informasi Wisata Wegotour

Memahami cara menggunakan beberapa Bahasa Pemrograman beserta API dan Github

Daffa Audya Pramana

Prisya Haura Febrianti

bukupedia

Pemanfaatan Teknologi Pemrograman Dalam Membangun Sistem Informasi Wisata Wegotour

Memahami cara menggunakan beberapa Bahasa Pemrograman beserta API dan Github

Penulis :
Daffa Audya Pramana
Prisya Haura Febrianti

ISBN : -

Editor :
Rolly Awangga

Penyunting :
Rolly Awangga

Desain sampul dan Tata letak :
Daffa Audya Pramana

Penerbit :
Penerbit Buku Pedia

Redaksi :
Athena Residence Blok. E No. 1, Desa Ciwaruga,
Kec. Parongpong, Kab. Bandung Barat 40559
Tel. 628-775-2000-300
Email : penerbit@bukupedia.co.id

Distributor :
Informatics Research Center
Jl. Sariasih No. 54
Bandung 40151
Email : irc@poltekpos.ac.id

Cetakan Pertama, 2023

KATA PENGANTAR

Puji syukur kita panjatkan ke hadirat Allah SWT, karena atas berkat dan rahmat-Nya, buku ini dapat hadir sebagai panduan yang membahas Tutorial Implementasi Frontend dan Backend dan Github API menggunakan Golang. Dengan judul "Pemanfaatan Teknologi Pemrograman Dalam Membangun Sistem Informasi Wisata Wegotour," buku ini membawa kita dalam sebuah perjalanan mendalam untuk memahami integrasi antara Frontend Backend Golang dengan Github API dalam pengembangan aplikasi web.

Dalam era perkembangan teknologi yang cepat, penggunaan Github API menjadi fundamental dalam kolaborasi pengembangan perangkat lunak. Buku ini akan membimbing pembaca melalui langkah-langkah praktis dalam memanfaatkan Github API, terutama dalam konteks penerapan Golang. Pembaca akan mendapatkan pemahaman yang mendalam tentang konsep dan teknik yang diperlukan untuk secara efisien dan efektif mengunggah file ke repository Github menggunakan bahasa pemrograman Golang.

Kami, penulis, merasa sangat bahagia dapat berbagi pengetahuan dan pengalaman kami dalam dunia pengembangan aplikasi web, terutama dalam konteks integrasi antara Frontend Backend Golang dengan Github API. Buku ini bukan hanya sekadar panduan praktis, tetapi juga merupakan upaya kami untuk memberikan pemahaman menyeluruh dan mendalam tentang topik ini.

Kami berharap bahwa buku ini akan menjadi sumber ilmu yang bermanfaat bagi pembaca, terutama bagi para pengembang web yang ingin menguasai implementasi Github API dengan menggunakan Golang.

Akhir kata, terima kasih kepada semua pihak yang telah mendukung dan mendorong terwujudnya buku ini. Semoga bermanfaat dan selamat membaca!

Bandung, Januari 2024

Penulis

PRAKATA

Buku ini membahas tentang pengembangan aplikasi web untuk pemesanan tiket tempat wisata secara online. Dalam buku ini, pembaca akan mempelajari bagaimana membuat aplikasi web yang dapat digunakan untuk memesan tiket tempat wisata dengan mudah dan cepat. Buku ini ditulis dengan menggunakan bahasa pemrograman JavaScript ES6+ dan Golang. Dalam buku ini, pembaca akan mempelajari bagaimana mengembangkan aplikasi web dengan menggunakan kedua bahasa pemrograman tersebut.

Pentingnya kerja sama antara Frontend dan Backend dalam pengembangan layanan web yang handal dan efisien. Namun, literatur yang membahas implementasi spesifik ini masih terbilang terbatas. Oleh karena itu, melalui buku ini, kami berupaya untuk mengisi celah tersebut dan menyajikan panduan praktis untuk para pengembang yang ingin memahami cara menggabungkan Frontend dengan Backend juga dengan bantuan dari Github API.

Buku ini terdiri dari beberapa bab yang membahas tentang konsep dasar pengembangan aplikasi web, penggunaan JavaScript ES6+ dan Golang dalam pengembangan aplikasi web, serta cara membuat aplikasi web untuk pemesanan tiket tempat wisata secara online. Buku ini ditujukan untuk para pengembang aplikasi web yang ingin mempelajari bagaimana membuat aplikasi web untuk pemesanan tiket tempat wisata secara online. Buku ini juga dapat digunakan sebagai referensi bagi para mahasiswa yang sedang mempelajari pengembangan aplikasi web.

Kami berharap buku ini dapat menjadi sumber rujukan yang bermanfaat bagi pembaca, khususnya bagi para pengembang yang tengah menjelajahi kemungkinan integrasi Frontend dan Backend. Dan juga kami mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dan inspirasi dalam proses penulisan buku ini.

DAFTAR ISI

DAFTAR GAMBAR

BAB I

BAB II

BAB III

BAB IV

Bab V

PENDAHULUAN

Dalam era digital yang terus berkembang pesat, teknologi informasi memainkan peran kunci, terutama dalam pengembangan aplikasi web. JavaScript, sebagai bahasa pemrograman yang umum digunakan dalam pengembangan web, telah mengalami evolusi signifikan dengan munculnya berbagai framework dan library. Perkembangan ini mendukung pengembangan aplikasi yang lebih dinamis dan interaktif, menciptakan pengalaman pengguna yang lebih unggul.

Salah satu sektor yang sangat dipengaruhi oleh kemajuan teknologi ini adalah sektor pariwisata. Pariwisata memiliki peran penting dalam perekonomian suatu daerah, dan untuk meningkatkan daya saing serta pengalaman pengguna dalam menjelajahi destinasi wisata, diperlukan sistem informasi wisata yang informatif, interaktif, dan mudah diakses.

Dalam upaya mencapai tujuan tersebut, konsep serverless telah menjadi pilihan utama dalam pengembangan aplikasi modern. Dengan menerapkan arsitektur serverless, pengembang dapat fokus sepenuhnya pada pengembangan fitur tanpa perlu khawatir tentang manajemen infrastruktur. Pendekatan ini tidak hanya meningkatkan skalabilitas aplikasi tetapi juga meningkatkan efisiensi pengembangan, sambil memungkinkan pemanfaatan sumber daya secara optimal.

Selain itu, penggunaan konsep micro frontend menjadi strategi yang efektif dalam pengembangan aplikasi. Dengan membagi aplikasi menjadi modul-modul kecil, pengembang dapat mengembangkan dan memelihara bagian-bagian tertentu secara terpisah. Pendekatan ini tidak hanya memudahkan pengelolaan kode, tetapi juga memfasilitasi pemeliharaan aplikasi serta memungkinkan tim pengembang untuk bekerja secara independen pada komponen-komponen tertentu. Dengan menggabungkan konsep serverless dan micro frontend, pengembang dapat menciptakan sistem informasi pariwisata yang lebih adaptif, responsif, dan efisien.

BAB I

PENDAHULUAN

1.1 PERKEMBANGAN TEKNOLOGI

Pentingnya pemahaman tentang teknologi semakin menonjol, mengingat perannya yang semakin mendalam dalam menghubungkan kita dengan berbagai aspek kehidupan modern. Aspek-aspek tersebut melibatkan tidak hanya komunikasi, tetapi juga meluas ke ranah seperti belanja, hiburan, pendidikan, dan banyak lagi. Oleh karena itu, eksplorasi peran teknologi tidak hanya terbatas pada tingkat global, namun juga akan dipelajari secara lebih spesifik dalam konteks Indonesia.

Perkembangan teknologi informasi telah membawa dampak signifikan dalam berbagai aspek kehidupan. Salah satu area yang terpengaruh adalah sistem pelaporan pengaduan masyarakat. Seiring dengan kemajuan teknologi, pelaporan pengaduan menjadi lebih efisien dan terintegrasi, memungkinkan pihak berwenang dan instansi terkait untuk lebih responsif terhadap masalah yang dihadapi oleh masyarakat.

Gambar 1. 1 Logo WeGoTour

2.2 PENGARUH TEKNOLOGI DALAM PEMESANAN TIKET SECARA ONLINE

Teknologi memainkan peran krusial dalam pemesanan tiket secara online, menyediakan berbagai kemudahan dan efisiensi bagi pengguna serta penyedia layanan. Berikut adalah beberapa peran utama teknologi dalam pemesanan tiket online:

1. Agar memudahkan dan mempersingkat waktu dalam registrasi akun secara online agar pengguna tidak perlu mengirim surat formulir registrasi kepada admin secara langsung.
2. Memudahkan admin dalam mengelola berkas data pengguna online.
3. Teknologi memungkinkan pengguna untuk melakukan pemesanan tiket kapan saja, tanpa terbatas oleh jam operasional kantor atau loket fisik. Platform online memberikan aksesibilitas 24/7, meningkatkan kenyamanan dan fleksibilitas bagi konsumen.
4. Teknologi juga memiliki peran penting dalam menjaga keamanan transaksi dan perlindungan data pengguna. Penggunaan protokol keamanan seperti SSL untuk enkripsi data dan sistem otentikasi dua faktor membantu melindungi informasi sensitif pengguna.

BAB II

PENGENALAN

1.1 Tentang JavaScript ES6+

Kita akan masuk ke bagian selanjutnya dari buku ini yaitu, mempersiapkan kebutuhan software dan cara instalasi software dan Bahasa yang dibutuhkan. Kebutuhan dasar yang dibutuhkan adalah JavaScript

2.1.1 JavaScript ES6+

Abdulloh (2018,193) menyatakan, "JavasScript merupakan bahasa pemrograman web yang pemrosesanya dilakukan di sisi client. Karena berjalan di sisi client, JavaScript dapat dijalankan hanya dengan menggunakan browser". Sari, dkk (2019:73) menyatakan "Javascript merupakan suatu Bahasa script yang banyak di gunakan dalam dunia teknologi internet yang dapat bekerja di sebagian besar web browser". JavaScript ES6+ merupakan versi terbaru dari bahasa pemrograman JavaScript. ECMAScript 6 (ES6) atau ECMAScript 2015 merupakan standar scripting language yang dirilis pada tahun 2015. Disini ES6 mengenalkan beberapa fitur baru seperti let, cost, arrow function, classes, default parameter values, template literals, destructuring, assignments, dan masih banyak lagi.

Gambar 2. 1 JavaScript Es6+

2.1.2 MANFAAT BELAJAR JAVASCRIPT

JavaScript adalah bahasa pemrograman yang banyak digunakan dalam pengembangan website, aplikasi, dan game. Dengan menguasai bahasa pemrograman ini, Anda bisa membuat tampilan website yang menarik atau mengembangkan game online berbasis web yang populer. Selain itu, JavaScript juga memungkinkan penggunanya untuk berkolaborasi dengan bahasa pemrograman lainnya dan sangat fleksibel untuk diproses dalam beragam bentuk ekstensi data pada laman suatu web.

Ada beberapa manfaat yang dapat diperoleh dari mempelajari bahasa pemrograman Javascript, antara lain:

1. Membuat website lebih menarik dengan menambahkan elemen interaktif seperti slideshow foto, gambar animasi, pengisian poling, dan masih banyak lainnya.
2. Menciptakan aplikasi mobile.
3. Mengembangkan game berbasis web browser.
4. Menjalankan web server,
5. Mudah dipelajari dan dipahami
6. Fleksibel digunakan
7. Beban server lebih ringan.
8. Bisa digunakan untuk semua kebutuhan
9. Memiliki komunitas yang aktif.

2.2 TENTANG GO

Go atau Golang merupakan bahasa pemrograman yang dibuat pada tahun 2009 oleh pihak Google. Tujuan dibuatnya Golang ialah untuk menyempurnakan bahasa pemrograman yang sudag ada seperti Phyton, C, dan masih banyak lagi. Golang dirancang dengan mempertimbangkan aspek kesederhanaan, mudah dipelajari, sederhana, efisien, dan dapat diketik secara statis. Golang juga menggabungkan beberapa property bagus dari framework lama seperti Phyton dan C agar dapat digunakan dalam satu bahasa pemrograman yang sama. Ada beberapa keunggulan yang membuat bahasa pemrograman ini diminati oleh banyak Perusahaan terutama *startup*.

Gambar 2. 2 Logo Go

2.2.1 MANFAAT BELAJAR GO

Ada beberapa manfaat yang dapat diperoleh dari mempelajari bahasa pemrograman Go, antara lain:

1. Mudah Dipelajari. Dalam perbandingan dengan bahasa pemrograman lain, sintaks Golang lebih sederhana dan mudah dipelajari. Hal ini disebabkan oleh sintaks yang lebih kecil dan tidak memerlukan waktu yang lama untuk mencari istilah-istilah yang sulit dimengerti. Kemudahan ini bahkan dapat dirasakan oleh programmer yang menggunakan gaya sintaks berbeda sekalipun.
2. Lebih Cepat. Dalam bahasa pemrograman Go (Golang), kode sumber dikompilasi menjadi kode mesin, sehingga dapat berjalan lebih cepat daripada bahasa pemrograman lain yang menggunakan virtual runtime. Selain itu, Go juga memiliki API yang memungkinkan pengguna untuk mengompilasi program dalam hitungan detik, sehingga mempercepat proses pengembangan perangkat lunak.
3. Memiliki Concurrency. Concurrency adalah kemampuan program untuk membagi tugas menjadi bagian-bagian yang lebih kecil dan dapat berfungsi secara independen. Meskipun tidak banyak bahasa pemrograman yang mendukung concurrency, bahasa pemrograman Golang memiliki kemampuan ini dengan model yang lebih mudah digunakan.
4. Memiliki Garbage Collector. Menurut sumber yang saya temukan, pengelolaan memori pada bahasa pemrograman Go lebih mudah dibandingkan bahasa pemrograman lain seperti C dan C++. Bahasa pemrograman Go memiliki fitur garbage collector yang memungkinkan pengumpulan objek-objek yang teralokasi secara dinamis dalam satu tempat. Hal ini memudahkan pengelolaan memori pada bahasa pemrograman Go.

2.2.2 KEUNGGULAN BAHASA GO

Ada beberapa keunggulan dari Bahasa Go, berikut adalah beberapa kelebihan dari bahasa pemrograman Golang:

1. Memiliki Goroutines: Golang memiliki Goroutines yang mampu membuat manajemen memori bekerja dengan efektif. Untuk setiap new thread yang Anda buat di Java, pada umumnya Anda akan menggunakan memori sebesar 1 Mb. Namun ketika Anda menggunakan Golang, memori yang dibutuhkan sangat kecil yaitu sebesar 2 Kb. Hal ini tentu sangat berguna ketika Anda memiliki multiple thread yang harus dijalankan.
2. Bekerja dengan cepat: Meskipun Golang dirancang untuk menghasilkan aplikasi dengan performa yang tinggi, namun struktur dan sintaks yang dimiliki cukup sederhana. Hal ini dapat memudahkan Anda untuk menulis kode dengan cepat. Golang juga mampu memproses kompilasi dengan cepat, sehingga proses pengembangan dapat selesai dengan waktu yang lebih singkat.
3. Sederhana dan mudah dipelajari: Go sangat mudah untuk dimengerti. Sintaks yang digunakan sederhana sehingga mudah diakses oleh pemula. Di dalamnya juga tidak terdapat berbagai fungsi komplek yang harus dipelajari. Selain itu, dokumentasi yang dimiliki juga terstruktur dengan baik dan rapi. Jika Anda sudah mahir menggunakan C++ atau C#, Anda pasti bisa mempelajari bahasa Golang dengan cepat karena sintaksis yang digunakan mirip dengan bahasa pemrograman C.
4. Memiliki fungsi garbage collection: Bahasa pemrograman Golang mempunyai fungsi garbage collection yang dapat membantu manajemen memori bekerja dengan baik. Fungsi ini memiliki pengaruh signifikan untuk menjaga performa serta membuat concurrency bekerja efisien.
5. Memiliki dukungan dari Google: Google memiliki salah satu infrastruktur cloud terbesar di dunia dan terus ditingkatkan secara masif.

2.3 APA ITU API

API, atau Application Programming Interface, adalah suatu jembatan perangkat lunak yang memungkinkan dua aplikasi atau sistem berbeda untuk saling berbicara dan bekerja sama. API berfungsi sebagai panduan yang menetapkan aturan dan protokol agar pengembang perangkat lunak dapat menggunakan fitur atau layanan dari suatu aplikasi, platform, atau sistem tanpa harus tahu seluk-beluk implementasinya. API bisa berupa kumpulan fungsi, prosedur, atau protokol komunikasi yang memungkinkan pengembang untuk mengakses dan memanfaatkan bagian-bagian tertentu dari suatu sistem atau aplikasi. Penggunaan API sangat umum dalam pembuatan perangkat lunak karena memungkinkan integrasi antar aplikasi dan menyederhanakan proses pengembangan solusi kompleks dengan memanfaatkan layanan atau data yang sudah ada. Sebagai contoh, API web memungkinkan pengembang untuk mengambil data dari server web atau mengirim data ke server tersebut, membuka pintu bagi berbagai jenis aplikasi dan layanan untuk berinteraksi secara efisien.

2.4 APA ITU GITHUB

GitHub adalah tempat penyimpanan populer bagi proyek pengembangan perangkat lunak yang menggunakan Git sebagai dasar. Platform ini memungkinkan pengembang untuk secara sentral menyimpan, mengelola, dan berbagi kode sumber proyek mereka. GitHub memiliki peran kunci dalam memfasilitasi kolaborasi dan kerja tim dalam pengembangan perangkat lunak, menyediakan alat kolaborasi, sistem kontrol versi yang handal, dan integrasi dengan berbagai layanan pengembangan lainnya.

Gambar 2. 3 Logo Github

2.4.1 APA ITU GITHUB API

GitHub API, atau Application Programming Interface, adalah antarmuka pemrograman yang memberikan serangkaian titik akhir HTTP untuk berinteraksi dengan berbagai data dan komponen yang terdapat di platform GitHub. Melalui GitHub API, pengembang memiliki kemampuan untuk secara programatik mengakses, membuat, dan mengelola repositori, isu, pull request, serta informasi pengguna. API ini mendukung metode HTTP seperti GET, POST, PUT, dan DELETE untuk membaca, membuat, memperbarui, atau menghapus sumber daya di GitHub.

GitHub API memungkinkan pengembang untuk mengintegrasikan fitur-fitur GitHub ke dalam aplikasi atau alat pengembangan yang mereka kembangkan. Penggunaan API ini dapat melibatkan otomatisasi tugas-tugas sehari-hari, pemantauan perubahan di repositori, manajemen isu, dan berbagai fungsi lainnya. Sebagai contoh, dengan menggunakan GitHub API, tim pengembang dapat membuat alat otomatis untuk merilis perangkat lunak, melacak status build secara otomatis, atau menghasilkan laporan analisis statis kode.

GitHub API menggunakan token otentikasi API sebagai langkah keamanan untuk memastikan akses yang aman. Tiap permintaan API harus dilengkapi dengan token otentikasi yang sesuai dengan izin yang diberikan oleh pemilik repositori atau organisasi. Ini tidak hanya meningkatkan kendali akses, tetapi juga memastikan bahwa hanya pengguna yang sah yang memiliki hak untuk mengakses atau mengubah data di GitHub.

Dengan demikian, GitHub API menjadi bagian yang sangat penting dalam ekosistem GitHub, memberikan kesempatan kepada pengembang untuk sepenuhnya memanfaatkan fitur-fitur yang ditawarkan oleh platform tersebut, dan meningkatkan produktivitas serta efisiensi dalam pengembangan perangkat lunak secara kolaboratif

2.5 Google Cloud Function

Google Cloud Function merupakan lingkungan eksekusi tanpa server yang memungkinkan pengguna untuk untuk menghubungkan dan membangun layanan cloud. Dengan Cloud Functions, pengguna dapat menulis fungsi sederhana dan khusus yang hanya terpasang pada peristiwa yang dihasilkan infrastruktur dan layanan cloud pengguna. Fungsi pengguna dapat dipicu ketika peristiwa yang sedang diamati dijalankan.

Google Cloud Platform (GCP) adalah salah satu penyedia layanan komputasi awan terkemuka yang ditawarkan oleh Google di seluruh dunia. GCP menawarkan berbagai layanan dan infrastruktur cloud untuk membantu organisasi mengelola dan memanfaatkan sumber daya teknologi informasi mereka dengan lebih efisien. Platform ini mencakup berbagai layanan, termasuk komputasi awan, penyimpanan data, basis data, kecerdasan buatan (AI), Internet of Things (IoT), dan sejumlah layanan lainnya.

GCP juga menyediakan layanan kecerdasan buatan dan machine learning melalui Cloud AI, Vision AI, Natural Language Processing, dan AI Platform. Ini memudahkan pengembang untuk membangun aplikasi cerdas dan analitis. Layanan komputasi awan dalam Google Cloud Platform (GCP) mencakup Compute Engine (VMs), Kubernetes Engine (pengelolaan kontainer), dan App Engine (platform untuk membangun dan mendeploy aplikasi). Adapun penyimpanan data di GCP, melibatkan Cloud Storage untuk menyimpan objek, Cloud SQL untuk basis data SQL, Cloud Bigtable untuk basis data NoSQL, dan berbagai layanan lainnya.

Gambar 2. 4 Logo Google Cloud Function

2.6 MongoDB

MongoDB adalah salah satu produk database noSQL Open Source yang menggunakan struktur data JSON untuk menyimpan datanya. MongoDB adalah merupakan database noSQL yang paling populer di internet. MongoDB sering dipakai untuk aplikasi berbasis Cloud, Grid Computing, atau Big Data (Saputra : 2016).
MongoDB adalah sebuah basis data yang tidak relasional dan bersifat open source. Basis data ini menggunakan konsep manajemen database berorientasi dokumen (document-oriented) yang dibuat menggunakan pemograman C++. MongoDB sudah dikembangkan oleh Iogen sejak Oktober 2007, namun baru dipublikasikan mulai Februari 2009. Performa pada MongoDB sudah mencapai 4 kali lebih cepat dibandingkan dengan MySQL. Karena MongoDB ini tergabung sebagai modul PHP, MongoDB mudah diaplikasikan dan memiliki performa yang tinggi.

Berbeda dengan RDBMS seperti MySQL, MongoDB tidak menggunakan tabel, melainkan menggunakan koleksi dan dokumen. Koleksi dianggap sebagai sebuah directory (folder) sedangkan dokumen sendiri dianggap sebagai file (berkas) dalam directory (folder) tersebut. Perbandingan pada RDBMS seperti MySQL, pada koleksi diibaratkan dengan tabel, sedangkan dokumen diibaratkan dengan baris dalam tabel tersebut. Baris pada MongoDB ini tidak sama dengan yang ada pada RDBMS, dokumen pada MongoDB dapat memiliki beda atribut dengan dokumen yang lainnya walaupun ada pada satu koleksi.

Gambar 2. 5 Logo MongoDB

2.7 HTML 5

HTML5 adalah standar terbaru yang mendefinisikan bahasa HTML. HTML5 memperkenalkan elemen, atribut, perilaku baru, dan seperangkat teknologi yang lebih besar yang memungkinkan pengembang web untuk membuat situs web yang lebih beragam dan interaktif. Dalam HTML5, pengembang web dapat memanfaatkan fitur seperti audio, video, dan animasi tanpa memerlukan plug-in tambahan

Gambar 2. 6 Logo HTML5

2.8 CSS

Menurut Abdulloh (2018:45), CSS adalah dokumen web yang berfungsi untuk mengatur elemen HTML dengan berbagai properti yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan. Dalam kata lain, CSS memungkinkan pengguna untuk memisahkan antara konten dan presentasi dari sebuah halaman web. Cascading Style Sheets (CSS) adalah bahasa pemrograman yang digunakan untuk memperindah dan mengatur tampilan halaman web. CSS dikembangkan oleh W3C, organisasi yang bertanggung jawab atas pengembangan teknologi internet.

Gambar 2. 7 Logo CSS

2.9 POSTMAN

Postman adalah alat yang digunakan untuk pengujian API dan membantu pengembang membedah dan menguji API yang dibuat oleh orang lain atau menguji yang Anda buat sendiri. Berikut ini beberapa fitur utama dari Postman:

1. Postman adalah alat yang membantu pengembang dalam membangun, menguji, dan memodifikasi API. Postman menawarkan berbagai fitur dan fungsi penting yang memudahkan pengembangan aplikasi.
2. Versi gratis dan berbayar: Postman tersedia untuk diunduh secara gratis dan juga memiliki versi berbayar dengan fitur yang lebih professional.
3. Otomatisasi: Postman memungkinkan pengguna untuk menyimpan permintaan dan mengatur rangkaian pengujian untuk mengotomatisasi pengujian.
4. Lingkungan pengujian: Postman memungkinkan pengguna untuk membuat lingkungan pengujian yang berbeda melalui pembuatan/penggunaan variabel.
5. Interface yang sederhana: Postman menawarkan interface yang sederhana dan ramah pengguna, memudahkan pengguna untuk mengirim permintaan HTTP yang dapat dipilih metode, seperti GET, POST, PUT, dan PATCH.

Postman sangat sering digunakan dalam dunia pendidikan dan pengembangan perangkat lunak karena membantu mengujicobakan API dengan cepat dan efisien. Selain itu, Postman juga dapat digunakan untuk mencegah plagiat dalam proses pengembangan perangkat lunak, dengan memastikan bahwa kode yang dibuat oleh pengembang tidak salah atau bergantung pada kode yang diterima dari sumber lain.

Gambar 2. 8 Logo Postman

BAB III

PERANCANGAN WEGOTOUR

3.1 Deskripsi WeGoTour

WeGoTour adalah sebuah aplikasi yang memungkinkan pengguna untuk memesan tiket secara online ke tempat-tempat wisata yang menarik. Aplikasi ini dibuat dengan tujuan memudahkan pengguna dalam melakukan pemesanan tiket. Dengan WeGoTour, pengguna dapat memilih dari berbagai destinasi wisata yang tersedia dan memesan tiket dengan mudah. WeGoTour juga menyediakan informasi tentang tempat-tempat wisata yang populer dan menarik, sehingga pengguna dapat memilih destinasi wisata yang sesuai dengan minat mereka. Dalam hal ini, WeGoTour adalah solusi yang tepat bagi mereka yang ingin merencanakan liburan yang menyenangkan dan mudah.

3.2 Fitur-Fitur WeGoTour

Dengan tampilan UI yang sederhana dan mudah dipahami oleh pengguna dari berbagai kalangan. Selain itu, WeGoTour juga menyediakan informasi tentang tempat-tempat wisata yang populer dan menarik, sehingga pengguna dapat memilih destinasi wisata yang sesuai dengan minat mereka. Dalam hal memesan tiket, pengguna dapat melakukannya dengan mudah dan cepat melalui tampilan yang sederhana.

3.3 Keuntungan Menggunakan WeGoTour

- Kami menawarkan beberapa rekomendasi tempat wisata yang menarik dan memukau di Indonesia.
- Kami menyediakan layanan pemesanan tiket tempat wisata yang mudah dan cepat, sehingga Anda dapat menikmati liburan tanpa khawatir.
- Kami berusaha untuk memberikan layanan yang efektif dan efisien agar Anda dapat merencanakan liburan Anda dengan mudah dan nyaman.

3.4 MERANCANG STRUKTUR DATABASE DENGAN MONGODB

Dalam desain konsep aplikasi WeGoTour, Bab ini membahas tentang merancang struktur database menggunakan MongoDB. MongoDB adalah basis data NoSQL yang fleksibel dan memberikan pendekatan yang ideal untuk menyimpan dan mengelola data yang beragam dan dinamis dalam konteks pemesanan tiket

- **Skema MongoDB**

  Dalam perancangan skema MongoDB untuk WeGoTour, sangat di perhatikan untuk kebutuhan pengguna aplikasi. Skema ini mencakup:

  1. Informasi Pengguna
  2. Informasi Admin
  3. Kumpulan Data Tiket
  4. Kumpulan Data Transaksi

3.5 DESAIN ANTARMUKA PENGGUNA DENGAN BOOTSTRAP

Desain UI/UX yang dipilih harus mencerminkan nilai-nilai aplikasi, yaitu kemudahan penggunaan, kejelasan informasi, dan estetika yang menarik. Bootstrap CSS dipilih sebagai framework desain karena kemampuannya untuk memberikan kontrol yang tinggi dan fleksibilitas dalam styling antarmuka.

3.6 KEBUTUHAN-KEBUTUHAN YANG DIPERLUKAN

Pada bagian ini, kita akan membahas tentang kebutuhan dan logika dalam mengintegrasikan bahasa Golang dan JavaScript untuk pembuatan web-service. Bagian pertama ini akan menjelaskan persyaratan yang diperlukan untuk mengikuti tutorial pembuatan web aplikasi Wegotour ini.

3.6.1 INSTALASI GO

Selanjutnya, kita perlu menginstal Golang. Dapat diunduh dan install Golang melalui link resmi The Go Programming Language. Golang, yang dibuat oleh Google, diprediksi akan menjadi bahasa yang sangat populer di masa depan. Menguasai dasar-dasar Golang akan sangat bermanfaat bagi kalian dalam mengikuti buku ini, karena akan mempermudah pemahaman dan pengaplikasian materi yang diajarkan.

3.6.2 JAVASCRIPT

Selanjutnya, untuk JavaScript, tidak diperlukan proses instalasi seperti bahasa pemrograman lainnya. Kamu dapat langsung membuat file dengan ekstensi .js dan mulai menulis kode JavaScript. Namun, perlu diingat bahwa jika kamu ingin menggunakan fitur-fitur khusus dari ori.js atau React.js, mungkin akan memerlukan instalasi tambahan sesuai dengan kebutuhan proyek tersebut.

3.6.3 AKUN GITHUB

Selanjutnya, penting bagi kita untuk membuat akun di GitHub. GitHub memiliki peran yang sangat penting dalam pengembangan proyek, karena memungkinkan kita untuk menyimpan proyek-proyek kita di repositori GitHub. Selain itu, GitHub juga menyediakan berbagai platform deploy yang memudahkan para pengembang untuk melakukan deploy backend atau frontend proyek mereka. Contohnya, GitHub Pages adalah salah satu fitur deploy frontend yang disediakan oleh GitHub secara gratis. Selain itu, terdapat juga platform lain seperti 000Webhost, Netlify, dan Heroku yang terintegrasi dengan GitHub.

Untuk mendaftar, kamu dapat mengunjungi situs resmi GitHub atau menggunakan link yang disediakan. Selain itu, diharapkan kamu sudah menguasai penggunaan git, karena git merupakan salah satu aspek penting dalam menggunakan GitHub.

3.6.4 AKUN GOOGLE CLOUD FUNCTION(GCF)

Selanjutnya, kita perlu memiliki akun Google Cloud Platform (GCP) untuk melakukan deploy backend Golang kita. GCP adalah platform komputasi awan yang digunakan untuk mempublikasikan atau deploy aplikasi backend kita. Untuk menggunakan GCP, biasanya diperlukan biaya, oleh karena itu untuk dapat menggunakan nya harus memerlukan biaya dan kebetulan pada GCP terdapat promo untuk pemakaian pertama sehingga bisa mendapatkan akses yang dapat digunakan untuk melakukan deploy aplikasi backend. Kamu dapat mendaftar akun GCP dengan mengunjungi situs web resmi mereka atau menggunakan link yang disediakan, yaitu Google Cloud Platform | Google Cloud.

3.6.5 AKUN MONGODB

Tahap selanjutnya adalah membuat akun di ini MongoDB Atlas: Cloud Document Database | MongoDB sebagai tempat penyimpanan database kita. Untuk membuat akun, kamu dapat mengunjungi link MongoDB Atlas: Cloud Document Database | MongoDB atau mengakses website resmi mereka. Selain itu, kamu juga perlu mengunduh dan menginstal MongoDB melalui link MongoDB Compass Download (GUI) | MongoDB atau mengunjungi website resmi mereka. Penting untuk diingat bahwa baik pembuatan akun di MongoDB Atlas maupun instalasi MongoDB itu sendiri dapat dilakukan secara gratis untuk studi kasus kita.

3.6.6 GOLANG SEBAGAI BACKEND

Bahasa pemrograman Golang akan digunakan sebagai bagian backend dalam aplikasi. Golang dianggap sebagai bahasa backend yang kuat dan cepat karena kemampuannya dalam mengkonversi kode program menjadi bentuk biner. Keunggulan Golang dibandingkan dengan bahasa lainnya adalah kemampuannya dalam menghasilkan performa yang tinggi.

Pada tahap selanjutnya, membuat kode program package backend yang akan digunakan, Repositori ini dapat ditemukan di GitHub Wegotour, github.com/wegotour

3.6.7 JAVASCRIPT SEBAGAI FRONTEND

Selanjutnya Javascript, kita akan menggunakan Javascript untuk Front-End pada Aplikasi kita yang dimana Javascript digunakan untuk sebagai fungsi dari API yang sudah di deploy dari Back-End, Tujuan kita ini Mulai dari Get Data dari API juga akan melakukan Post Data API.

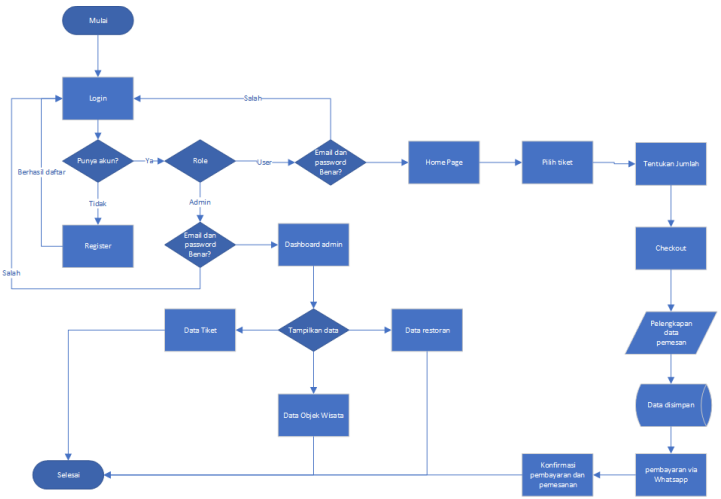
3.6.8 HTML SEBAGAI FRONTEND

Selanjutnya HTML, kita akan menggunakan HTML untuk Front-End pada Aplikasi kita yang dimana HTML ini digunakan untuk sebagai jembatan dari JavaScript yang telah di buat.

3.7 PERANCANGAN APLIKASI YANG AKAN DIBANGUN

Pada bagian ini, kita akan membahas perancangan pada Wegotour, terdapat Flowmap, Use Case dan Sequence yang akan di relasikan pada web aplikasi Wegotour ini.

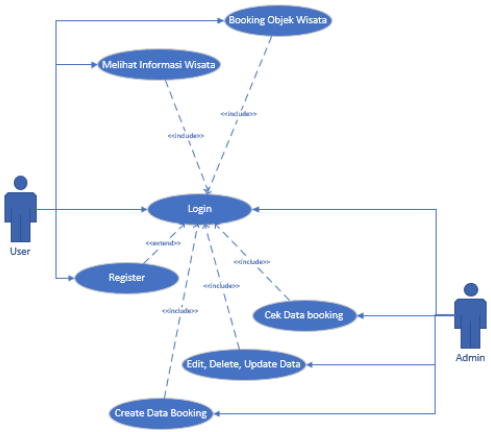
3.7.1 FLOWMAP



Gambar 3. 1 Flowmap Perancangan Aplikasi Yang Akan Dibangun

Aplikasi ini akan dimulai dengan pengguna membuka web WEGOTOUR, kemudian akan dikonfirmasi apakah sudah mempunyai akun atau belum. Apabila belum maka akan diarahkan ke form register, namun apabila sudah, maka akan diarahkan ke login. Kemudian akan diperiksa apakah berhasil login atau tidak. Apabila tidak maka akan kembali ke halaman login awal, apabila berhasil maka akan diarahkan menuju dashboard aplikasi. Pengguna dapat langsung melakukan pemesanan tiket, lalu jika sudah memilih pesan pada destinasi wisata pengguna akan langsung diarahkan ke halaman pengisian data untuk melakukan pemesanan.

3.7.2 USE CASE DIAGRAM



Gambar 3. 2 Use Case Aplikasi Yang Akan Dibangun

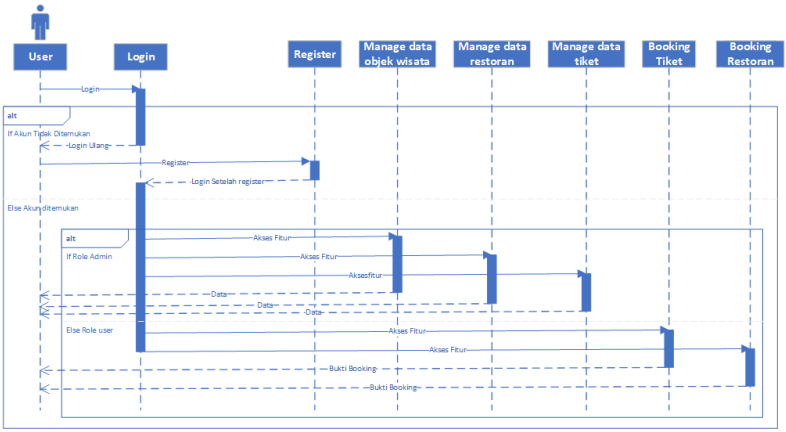
Admin

Disini admin dapat melakukan data tambahan pada aplikasi seperti menambah data objek wisata mengelola database, dan memeriksa data booking pengguna.

User

User disini berperan sebagai pengguna yang akan melakukan pemesanan tiket objek wisata dan pengecekan tempat wisata.

3.7.3 SEQUENCE DIAGRAM



Gambar 3. 3 Sequence Aplikasi

BAB IV

INTEGRASI FRONTEND BACKEND

4.1 INTEGRASI

Dalam pembahasan integrasi untuk aplikasi ini, penting untuk memahami peran kunci dari Frontend dan Backend yang akan digunakan.

Untuk Backend kita akan menggunakan Golang, Golang dikenal dengan kecepatan eksekusinya dan tingkat efisiennya dalam mengelola konkurensi. Pada aplikasi ini, Golang akan digunakan untuk:

- Membangun Web Services: Golang sangat cocok untuk membuat backend dan layanan web, memberikan kinerja tinggi dan efisiensi.
- Integrasi: Golang akan menangani integrasi utama dalam server, termasuk operasi CRUD (Create, Read, Update, Delete).

Dalam konteks integrasi, Javascript akan berperan dalam:

- Pengembangan Front-end: Memastikan antarmuka pengguna dapat berinteraksi dengan pengguna secara dinamis.
- Integrasi dengan API: Javascript akan digunakan untuk berkomunikasi dengan Web Services yang dibangun menggunakan Golang, memungkinkan pertukaran data antara server dan client.

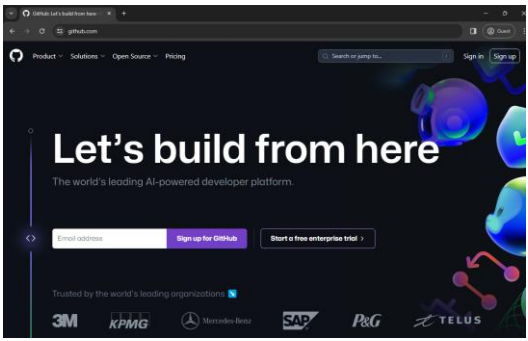
Dengan demikian, Frontend dan Backend akan berkolaborasi secara sinergis untuk menciptakan aplikasi yang handal, efisien, dan responsif. Integrasi keduanya menjadi fondasi yang kuat untuk membangun aplikasi Wegotour yang dapat memberikan pengalaman pengguna yang baik dan performa yang optimal.

4.2 GITHUB

Pada bagian ini , akan menjelaskan bagaimana menggunakan github agar bisa melanjutkan pengerjaan berikutnya.

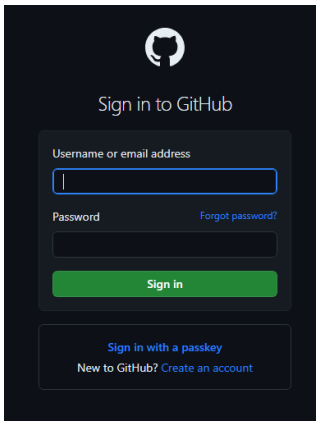
4.2.1 LOGIN GITHUB

Pertama ketik github.com pada browser anda dan sign in



Gambar 4. 1 Tampilan Utama GitHub

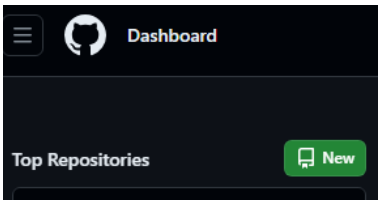
Kemudian, jika anda tidak memiliki akun GitHub kalian bisa registrasi terlebih dahulu dengan menuju halaman Sign Up, namun jika anda sudah memiliki akun GitHub bisa langsung menuju halaman Sign In.



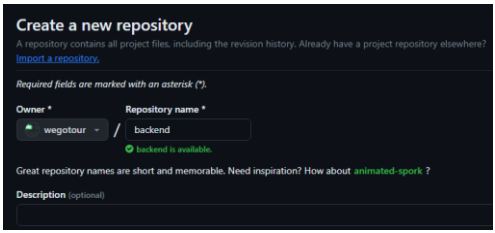
Gambar 4. 2 Tampilan Login GitHub

Pastikan Username dan password yang dimasukan adalah username yang benar, agar dapat melanjutkan ke langkah selanjutnya.

4.2.2 MEMBUAT REPOSITORY DI GITHUB



Gambar 4. 3 Tampilan Home GitHub

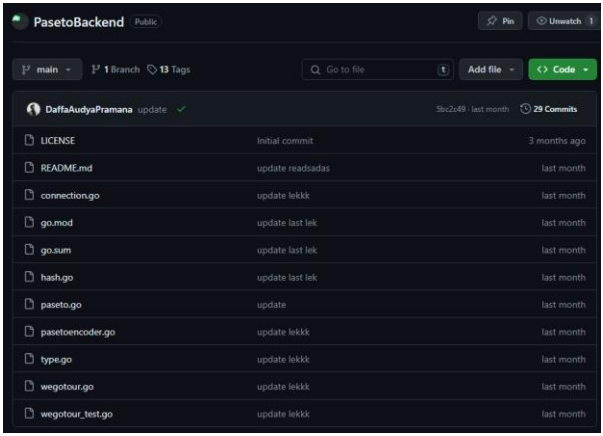


Gambar 4. 4 Tampilan Membuat Repositori GitHub

Selanjutnya, setelah anda login, akan menuju halaman Dashboard kemudian New, untuk membuat repositori baru pada GitHub.

4.3 PACKAGE GOLANG

Langkah selanjutnya kita akan membuat Package



Gambar 4. 5 Tampilan isi Repo dengan Package

Gambar diatas merupakan package yang berada di repositori pada aplikasi Wegotour bernama PasetoBackend.

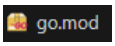
Baik sekarang kita akan membuat nya dengan langkah langkah seperti berikut:

1. Tahap awal melibatkan pencarian library Go yang akan digunakan dalam praktik kali ini.
2. Library yang akan digunakan adalah library Go-Github yang tersedia pada url ini go-github.
3. Kemudian buat Golang project menggunakan aplikasi code editor yang biasa anda gunakan (Disini Kami menggunakan VSCODE).
4. Kemudian go mod init project kalian *go mod init* (akun github), (namarepo).

```
PS C:\Semester 5\Proyek 3\Wegotour\PasetoBackend> go mod init github.com/wegotour/PasetoBackend
```

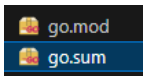
Gambar 4. 6 Go Mod Init

5. Jika sudah maka akan muncul file yang bernama go.mod.



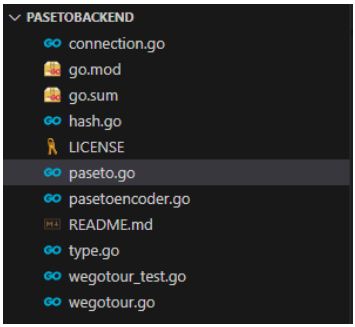
Gambar 4. 7 File go.mod

6. Tahap selanjutnya ketik perintah go mod tidy kemudian akan tampil file baru yang bernamakan go.sum.



Gambar 4. 8 File go.sum

7. Kemudian buatlah file sesuai kebutuhan.



Gambar 4. 9 Isi File pada Package PasetoBackend

8. Pada gambar diatas yang digunakan untuk membuat CRUD pada aplikasi Wegotour pemesanan tiket online, berupa 3 file yakni wegotour.go, wegotour_test.go, type.go.
9. Membuat struct pada type.go sebelum memanggilnya di wegotour.go dan wegotour_test.go.

```
type Ticket struct {
    ID          primitive.ObjectID `bson:"_id,omitempty" `
    Nomorid     int                `json:"nomorid" bson:"nomorid"`
    Title       string             `json:"title" bson:"title"`
    Description string             `json:"description" bson:"description"`
    Image       string             `json:"image" bson:"image"`
    Status      bool               `json:"status" bson:"status"`
}
```

Gambar 4. 10 Kode Struct Ticket pada type.go

Type Ticket disini untuk pembuatan CRUD pada Pemesanan Ticket.

10. Tahap selanjutnya membuat function Create pada file wegotour.go untuk struct dari yang telah dibuat tadi.

```
// ticket function
func insertTicket(mongoconn *mongo.Database, collection string, ticketdata Ticket) interface{} {
    return atdb.InsertOneDoc(mongoconn, collection, ticketdata)
}
```

Gambar 4. 11 Function Insert

11. Kemudian membuat function delete.

```
func DeleteTicket(mongoconn *mongo.Database, collection string, ticketdata Ticket) interface{} {
    filter := bson.M{"nomorid": ticketdata.Nomorid}
    return atdb.DeleteOneDoc(mongoconn, collection, filter)
}
```

Gambar 4. 12 Function Delete

12. Tahap Ketiga buat function update dari data yang telah tersedia.

```
func UpdatedTicket(mongoconn *mongo.Database, collection string, filter bson.M, ticketdata Ticket) interface{} {
    updatedFilter := bson.M{"nomorid": ticketdata.Nomorid}
    return atdb.ReplaceOneDoc(mongoconn, collection, updatedFilter, ticketdata)
}
```

Gambar 4. 13 Function Update

13. Setelah itu membuat function mengambil semua data yang telah dibuat.

```
func GetAllTicket(mongoconn *mongo.Database, collection string) []Ticket {
    ticket := atdb.GetAllDoc[[]Ticket](mongoconn, collection)
    return ticket
}
```

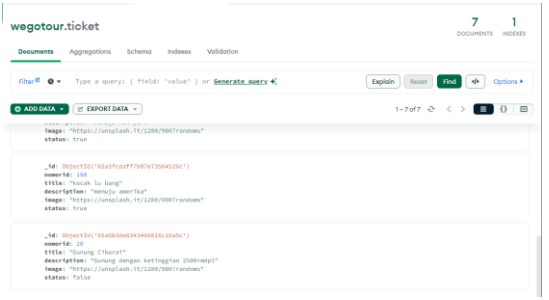
Gambar 4. 14 Function Read

14. Pada tahap ke lima ini, kode yang akan kita buat merupakan function test ticket untuk memeriksa apakah struct dan function yang telah kita buat apakah berjalan atau tidak, menampilkan data collection ke mongodb yang bernama ticket.

```go
// Test Insert Ticket
run test | debug test
func TestInsertTicket(t *testing.T) {
    mconn := SetConnection("MONGOSTRING", "wegotour")
    var ticketdata Ticket
    ticketdata.Nomorid = 20
    ticketdata.Title = "Gunung Cikurai"
    ticketdata.Description = "Gunung dengan ketinggian 2500+mdpl"
    ticketdata.Image = "https://unsplash.it/1200/900?randoms"
    CreateNewTicket(mconn, "ticket", ticketdata)
}
```

Gambar 4. 15 Function test Insert Ticket

15. Selanjutnya kita jalankan command “go test” di terminal dan kita bisa melihat hasil dari function diatas yang dijalankan dari command “go test”. Berikut hasil nya pada gambar di bawah ini.



Gambar 4. 16 Database MongoDB Ticket

4.3.1 PUBLISH PACKAGE GO

Pada bagian ini kita mempublish dari package yang telah dibuat pada bagan sebelumnya, masukkan command seperti di gambar berikut agar package yang telah dibuat terpublish di pkg.go.dev.

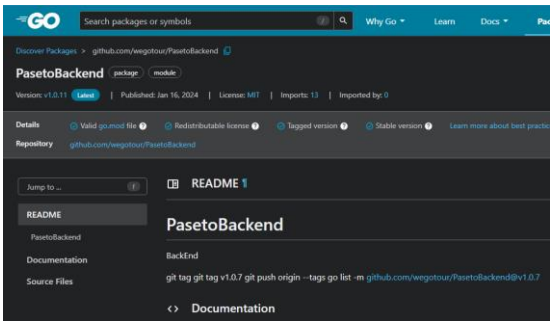
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  TEST RESULTS  PORTS
PS C:\Semester 5\Proyek 3\Wegotour\PasetoBackend> git tag
v.1.0.6
v0.0.1
v1.0.0
v1.0.1
v1.0.10
```

Gambar 4. 17 Memeriksa Versi Terakhir Tag Package

```
PS C:\Semester 5\Proyek 3\Wegotour\PasetoBackend> git tag v1.0.11
PS C:\Semester 5\Proyek 3\Wegotour\PasetoBackend> git push --tags
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/wegotour/PasetoBackend.git
 * [new tag]         v1.0.11 -> v1.0.11
PS C:\Semester 5\Proyek 3\Wegotour\PasetoBackend> go list -m github.com/wegotour/PasetoBackend@v1.0.11
github.com/wegotour/PasetoBackend v1.0.11
```

Gambar 4. 18 Update Versi Package Golang

Kemudian untuk memeriksa package tersebut sudah rilis atau belum kita bisa mengunjungi website pkg.go.dev, setelah itu dilanjutkan dengan url repo kita, lalu jika sudah rilis maka akan tampil isi function dan struct lainnya yang telah kita kerjakan tadi. berikut contoh jika package telah rilis.



Gambar 4. 19 Halaman Package Go

Catatan: Kita perlu mengatur email dan username pada Terminal Git dan menambah field berisi proxy go pada env di device kita agar berhasil saat proses publish package tersebut.

4.4 JAVASCRIPT

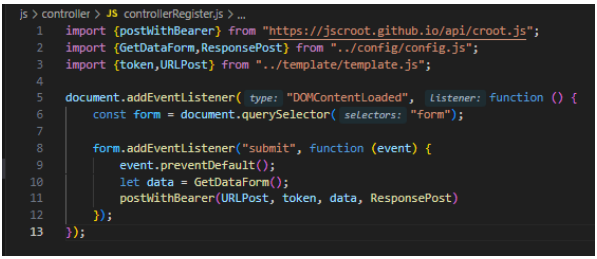
Selanjutnya membuat Javascript untuk dipanggil di HTML nanti, Javascript yang harus disiapkan antara lain:

4.4.1 REGISTER USER

```
template > JS template.js > ...
1   // template js
2
3   //url register
4   export let URLPost = `https://asia-southeast2-wegotour-403712.cloudfunctions.net/wegotour-registeruser`
5
6   //url login
7   export let URLLogin = `https://asia-southeast2-wegotour-403712.cloudfunctions.net/wegotourlogin`
8
9   //url changepass
10  export let URLChgPass = 'https://asia-southeast2-wegotour-403712.cloudfunctions.net/wegotourchangepassworduser'
11
12  export let token = 'token';
```

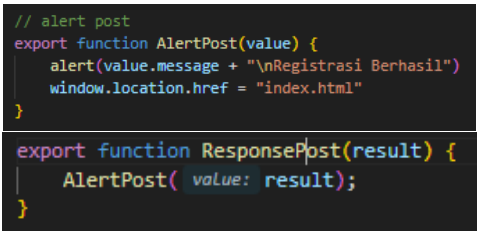
Gambar 4. 20 Template.js untuk menampung URL Google Cloud Function

1. Pertama kita membuat file js pada folder js kalian dan berikan nama sesuai kebutuhan, disini kami memberi nama dengan template.js.



Gambar 4. 21 ControllerRegister.js untuk membuat controller register

2. Kedua kita membuat Controller js untuk memanggil template.js yang telah dibuat tadi.

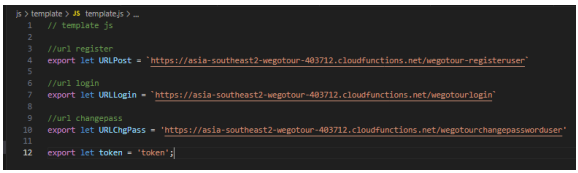


Gambar 4. 22 Membuat function untuk register user

3. Ketiga kita membuat function untuk register, disini kami memberikan nama file config.js
4. Javascript untuk Register pun sudah selesai, sekarang kita menuju tahap berikutnya.

4.4.2 LOGIN USER

1. Sama halnya seperti Register User, URL Google Cloud Function di satukan dengan Register untuk User.



Gambar 4. 23 JS untuk menampung URL Google Cloud Function

2. Kedua kita membuat ControllerLogin.js untuk memanggil URL GCF yang berada di template.js.

```
js > controller > JS controllerLogin.js > ...
1   import {postWithToken} from "https://jscroot.github.io/api/croot.js";
2   import {PostLogin,ResponseLogin} from "../config/config.js";
3   import {URLLogin} from "../template/template.js";
4   import {token} from '../template/template.js';
5
6   document.addEventListener("DOMContentLoaded", function () {
7       const form = document.getElementById("formlogin");
8       form.addEventListener("submit", function (event) {
9           event.preventDefault();
10          let data = PostLogin();
11          postWithToken(URLLogin, 'Authorization', 'Bearer ' + token, data, ResponseLogin);
12      });
13  });
```

Gambar 4. 24 ControllerLogin.js untuk membuat controller login user

3. Ketiga kita membuat function untuk login, disini kami memberikan nama file config.js

```
js > config > JS config.js > ...
1   import {
2       setCookieWithExpireHour
3   } from 'https://jscroot.github.io/cookie/croot.js';
4
5   //token api
6   export function getTokenFromAPI() {
7       const tokenUrl = "https://asia-southeast2-wegotour-403712.cloudfunctions.net/wegotourlogin";
8       fetch(tokenUrl)
9           .then(response => response.json())
10          .then(tokenData => {
11              if (tokenData.token) {
12                  userToken = tokenData.token;
13                  console.log('Token dari API:', userToken);
14              }
15          })
16          .catch(error => console.error('Gagal mengambil token:', error));
17  }
18
```

Gambar 4. 25 Membuat function dengan token untuk login user

4. Tahap Ke empat membuat function PostLogin yang telah kita buat tadi pada file controllerLogin.js, fungsinya untuk memanggil function yang berada di file tersebut.

```
// post login
export function PostLogin() {
    const username = document.getElementById("username").value;
    const password = document.getElementById("password").value;
    const role = document.getElementById("role").value;

    const data = {
        username: username,
        password: password,
        role: role
    };
    return data;
}
```

Gambar 4. 26 Pemanggilan function PostLogin

5. Pada tahap ke lima tidak lupa untuk menambahkan function ResponsePostLogin.

```
export function ResponseLogin(result) {
    ResponsePostLogin( response: result)
}
```

```
// response post login
function ResponsePostLogin(response) {
    if (response && response.token) {
        console.log('Token User:', response.token);
        setCookieWithExpireHour('Login', response.token, 2);
        window.location.href = 'https://wegotour.my.id/dashboard/';
        Swal.fire({
            icon: 'success',
            title: 'Login Successful',
            text: 'You have successfully logged in!',
          });
} else {
  Swal.fire({
            icon: 'error',
            title: 'Login Failed',
            text: 'Invalid email or password. Please try again.',
          });
}
}
```

Gambar 4. 27 Function ResponseLogin dan direct halaman

4.4.3 CHANGE PASSWORD UNTUK USER

Disini Change Password sangat diperlukan karena User terkadang lupa dengan password akunnya, maka dari itulah javascript ini dibuat.

1. File template.js untuk menampung URL GCF.

```
// template js

//url register
export let URLPost = `https://asia-southeast2-wegotour-403712.cloudfunctions.net/wegotour-registeruser`

//url login
export let URLLogin = `https://asia-southeast2-wegotour-403712.cloudfunctions.net/wegotourlogin`

//url changepass
export let URLChgPass = `https://asia-southeast2-wegotour-403712.cloudfunctions.net/wegotourchangepassworduser`

export let token = 'token';
```

Gambar 4. 28 URL GCF Change Password

2. Membuat controller untuk change password.

```
import {postWithBearer} from "https://jscroot.github.io/api/croot.js";
import {GetDataForm,ResponseUpdate} from "../config/config.js";
import {token,URLChgPass} from "../template/template.js";

document.addEventListener("DOMContentLoaded",  listener: function () {
    const form = document.querySelector("form");

    form.addEventListener("submit", function (event) {
        event.preventDefault();
        let data = GetDataForm();
        postWithBearer(URLChgPass, token, data, ResponseUpdate)
    });
});
```

Gambar 4. 29 Controller dari ChangePass.js

3. Membuat function untuk memanggil js yang telah dibuat sebelumnya.

```
//get data
export function GetDataForm() {
    const username = document.querySelector("#username").value;
    const password = document.querySelector("#password").value;
    const role = document.querySelector("#role").value;
    console.log(password)

    const data = {
        username: username,
        password: password,
        role: role
    };
    return data
}
```

Gambar 4. 30 Pemanggilan GetDataForm pada ChangePass.js

4. Membuat Alert bahwa data telah diubah.

```
// alert update
export function AlertUpdate(value) {
    alert(value.message + "\nUpdate Berhasil")
    window.location.href = "index.html"
}
```

Gambar 4. 31 Alert bahwa data diubah dan direct halaman

5. Memanggil Alert Update.

```
export function ResponseUpdate(result) {
    AlertUpdate( value: result);
}
```

Gambar 4. 32 Membuat function ResponseUpdate dan Alert

4.4.4 LOGIN ADMIN

1. Sama halnya seperti JS User, namun bedanya disini ada folder baru yang bernamakan jsadmin.

```
jsadmin > template > JS template.js > URLLogin
1   export let URLLogin = `https://asia-southeast2-wegotour-403712.cloudfunctions.net/wegotourloginadminbaru`
2
3   export let token = 'token';
```

Gambar 4. 33 template.js pada js admin

2. Membuat controllerLogin.js untuk login pada admin.

```
jsadmin > controller > JS controllerLogin.js > ...
1  import {postWithToken} from "https://jscroot.github.io/api/croot.js";
2  import {PostLogin,ResponseLogin} from "../config/config.js";
3  import {URLLogin} from "../template/template.js";
4  import {token} from '../template/template.js';
5
6  document.addEventListener( type: "DOMContentLoaded",  listener: function () {
7      const form = document.getElementById("formlogin");
8      form.addEventListener( type: "submit", function (event) {
9          event.preventDefault();
10         let data = PostLogin();
11         postWithToken(URLLogin, 'Authorization', 'Bearer ' + token, data, ResponseLogin);
12     });
13 });
```

Gambar 4. 34 Tampilan controllerLogin.js untuk login admin

3. Membuat file config.js untuk mengatur function login admin.

```
jsadmin > config > JS config.js > ...
1  import {
2    setCookieWithExpireHour
3  } from 'https://jscroot.github.io/cookie/croot.js';
4
5  //token API
6  export function getTokenFromAPI() {
7      const tokenUrl = "https://asia-southeast2-wegotour-403712.cloudfunctions.net/wegotourloginadminbaru";
8      fetch(tokenUrl)
9          .then(response => response.json())
10         .then(tokenData => {
11             if (tokenData.token) {
12                 userToken = tokenData.token;
13                 console.log('Token dari API:', userToken);
14             }
15         })
16         .catch(error => console.error('Gagal mengambil token:', error));
17 }
```

Gambar 4. 35 Memanggil function controllerLogin.js

4. Membuat function untuk memanggil pada file-file sebelumnya yang telah dibuat.

```
jsadmin > config > JS config.js > ...
33
34 // post login
35 export function PostLogin() {
36     const email = document.getElementById("email").value;
37     const password = document.getElementById("password").value;
38     const role = document.querySelector("#role").value;
39
40     const data = {
41         email: email,
42         password: password,
43         role: role
44     };
45     return data;
46 }
47
48 // response post login
49 function ResponsePostLogin(response) {
50     if (response && response.token) {
51         console.log('Token User:', response.token);
52         setCookieWithExpireHour('login', response.token, 2);
53         window.location.href = 'https://wegotour.my.id/dashboardadmin/';
54         Swal.fire({
55             icon: 'success',
56             title: 'Login Successful',
57             text: 'You have successfully logged in!',
58           });
59 } else {
60   Swal.fire({
61             icon: 'error',
62             title: 'Login Failed',
63             text: 'Invalid email or password. Please try again.',
64           });
65 }
66 }
67
68 export function ResponseLogin(result) {
69     ResponsePostLogin( response: result)
70 }
```

Gambar 4. 36 Function pada config.js login Admin

4.4.5 CRUD ADMIN

4.4.5.1 JAVASCRIPT CREATE

Pada tahap awal pembuatan JS Create diperlukan file yang Bernama controllerInsertTicket.js denga isi seperti gambar dibawah

Gambar 4. 37 Javascript controllerInsertTicket.js bagian Create

Pada source code diatas dapat beraneka macam jenis javascript yang digunakan pada CRUD Admin, pada bagian Admin ini kita sangat protektif untuk data data pengguna yang telah memasukkan data sehingga disini kami memakai header login atau token login untuk admin sendiri.

4.4.5.2 JAVASCRIPT READ

Pada tahap Read/Get ini diberi nama controllerTicket.js seperti gambar dibawah.

```
jsadmin > controller > JS controllerTicket.js > deleteTicket
const getTokenFromCookies = (cookieName) => {
  const cookies = document.cookie.split( separator: ';')
  for (const cookie of cookies) {
    const [name, value] = cookie.trim().split( separator: '=')
    if (name === cookieName) {
      return value
    }
  }
  return null
}

const getAllTicket = async () => {
  const token = getTokenFromCookies('Login')

  if (!token) {
    Swal.fire({
      icon: 'warning',
      title: 'Authentication Error',
      text: 'You are not logged in.',
    }).then(() => {
      window.location.href = 'https://wegotour.my.id/loginadmin.html'
    })
    return
  }

  const URLGetAllTicket = 'https://asia-southeast2-wegotour-403712.cloudfunctions.net/wegotour'

  const myHeaders = new Headers()
  myHeaders.append( name: 'Login', value: token)

  const requestOptions = {
    method: 'GET',
    headers: myHeaders,
    redirect: 'follow',
  }

  try {
    const response = await fetch( input: URLGetAllTicket, init: requestOptions)
    const data = await response.json()

    if (data.status) {
      displayTicketData(data.data, 'TicketDataBody')
    } else {
      Swal.fire({
        icon: 'error',
        title: 'Error',
        text: data.message,
      })
    }
  } catch (error) {
    console.error( data[0]: 'Error:', data[1]: error)
  }
}

const deleteTicket = async (nomorId) => {
  const token = getTokenFromCookies('Login')

  if (!token) {
    showAlert('Header Login Not Found', 'error')
    return
  }
```

Gambar 4. 38 Javascript controllerTicket.js bagian Read

Sama halnya seperti Create js, tetap menggunakan header yakni token login yang hanya dimiliki oleh admin itu sendiri.

4.4.5.3 JAVASCRIPT UPDATE

Pada tahap Update ini kami memberikan nama controllerUpdateTicket.js seperti gambar dibawah.

Gambar 4. 39 controllerUpdateTicket.js bagian Update

4.4.5.4 JAVASCRIPT DELETE

Pada tahap Delete ini kami menyatukannya dengan js Read, yaitu controllerTicket.js seperti gambar dibawah.

Gambar 4. 40 controllerTicket.js bagian Delete

Pada bagian Delete ini kode lumayan panjang karena menurut kami sendiri bagian update js ini lah yang membuat kami kewalahan dikarenakan lumayan rumit.

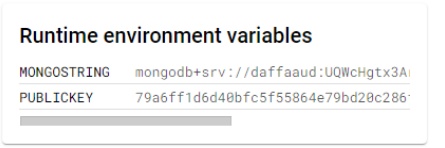
4.5 MEMBUAT FUNCTION GOOGLE CLOUD FUNCTION PADA GOOGLE CLOUD PLATFORM

Selanjutnya membuat Function di GCP untuk memanggil tiap function yang telah dibuat dan terpublish, diingat kembali bahwa package telah tertampil di pkg.go.dev yang dimana function nya sudah tampil. Berikut langkah - langkahnya :

1. Pertama kita create function pada GCP dan berikan nama sesuai kebutuhan, disini kami memberi nama dengan insertdataticket.

Google Cloud Platform insertdataticket

2. Selanjutnya menambahkan variable pada function yang akan dibuat tadi pada insertdataticket tadi. Terdapat 2 variable yaitu MONGOSTRING dan PUBLICKEY, pada mongostring di isi dengan connnection string pada database mongodb yang telah tersedia.



Environment Variables GCF insertdataticket

3. Memanggil function yang telah kita buat, pertama kita mengisi import yang yang diperlukan beserta url repo github yang berisi package yang telah kita buat. Seperti gambar berikut.

```
package gcf

import (
    "fmt"
    "github.com/wegotour/PasetoBackend"
    "net/http"

    "github.com/GoogleCloudPlatform/functions-framework-go/functions"
)
```

Pemanggilan Package Go Yang Telah Di Rilis

4. Setelah memanggil package kita, kali ini kita memanggil function yang akan digunakan dan menambahkan CORS bila

5. kita ingin menggunakan function tersebut pada web yang akan digunakan.

```
package gcf

import (
    "fmt"
    "github.com/wegotour/PasetoBackend"
    "net/http"

    "github.com/GoogleCloudPlatform/functions-framework-go/functions"
)

func init() {
    functions.HTTP("insertdata", insertdataticket)
}

func insertdataticket(w http.ResponseWriter, r *http.Request) {
    // Set CORS headers for the preflight request
    if r.Method == http.MethodOptions {
        w.Header().Set("Access-Control-Allow-Origin", "https://wegotour.my.id")
        w.Header().Set("Access-Control-Allow-Methods", "POST")
        w.Header().Set("Access-Control-Allow-Headers", "Content-Type,Authorization, Token, Login")
        w.Header().Set("Access-Control-Max-Age", "3600")
        w.WriteHeader(http.StatusNoContent)
        return
    }
    // Set CORS headers for the main request.
    w.Header().Set("Access-Control-Allow-Origin", "https://wegotour.my.id")
    w.Header().Add("Content-Type", "application/json")
    fmt.Fprintf(w, PasetoBackend.GCFInsertTicket("PUBLICKEY", "MONGOSTRING", "wegotour", "admin", "ticket", r))
}
```

Kode CORS Google Cloud Platform insertdataticket

Setelah itu tekan tombol Deploy dan tunggu sampai proses deployment selesai.

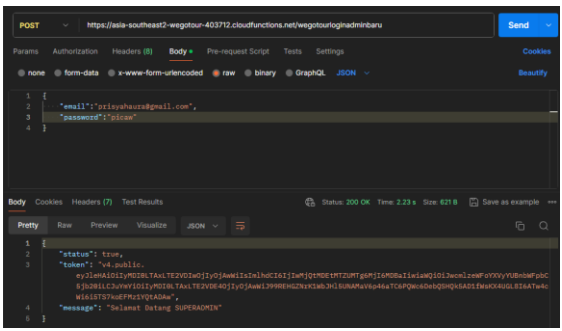
BAB V
IMPLEMENTASI

5.1 IMPLEMENTASI CLOUD FUNCTION PADA POSTMAN

Akhir dari semua deployment yang telah dilakukan sebelumnya kita akan menjalakan melalui postman untuk mengetahui hasil dari jalannya function yang telah dipanggil melalui Cloud Function. Berikut hasil tiap function nya.

5.5.1 CLOUD FUNCTION LOGIN-ADMIN

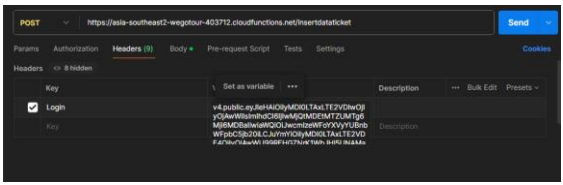
Pertama tama kita masukan link GCF Login. Disini kita mencoba untuk memasukan link gcf yang login admin, karena untuk menambahkan data ticket harus melakukan login terlebih dahulu. Setelah itu masukan email dan password pada body, raw, JSON, dan akan message "Selamat Datang SUPERADMIN"



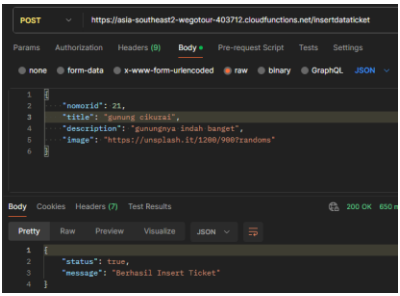
Gambar 5. 1 Login Admin Postman

5.5.2 CLOUD FUNCTION INSERTDATATICKET

Kita akan mencoba insertdataticket pada function yang telah dibuat sebelumnya, perlu di ingat bahwa sebelum menambahkan data kita harus melakukan login untuk mendapatkan token/akses dan menambah pada bagian header di postman untuk melakukan pengujian.



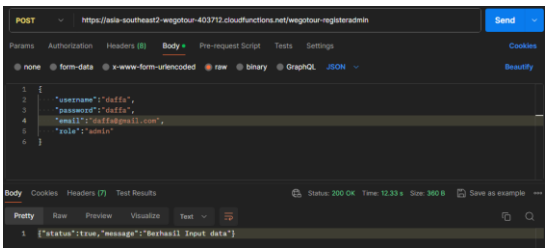
Gambar 5. 2 Memasukkan Key dan Value Pada Header Login Postman



Gambar 5. 3 Berhasil Input Ticket

5.5.3 CLOUD FUNCTION REGISTRASI

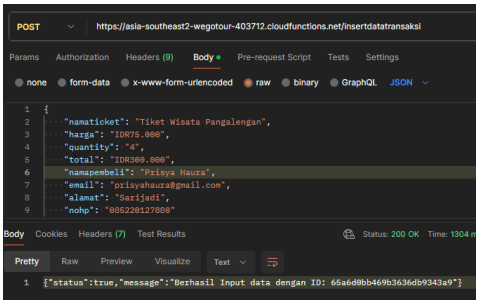
Disini kita akan mencoba test function registrasi di Postman, pada registrasi disini mencoba melakukan register pada admin dan masukan Kembali pada bagian body-ray-json lalu Send.



Gambar 5. 4 Registrasi Admin Postman

5.5.4 CLOUD FUNCTION TRANSAKSI

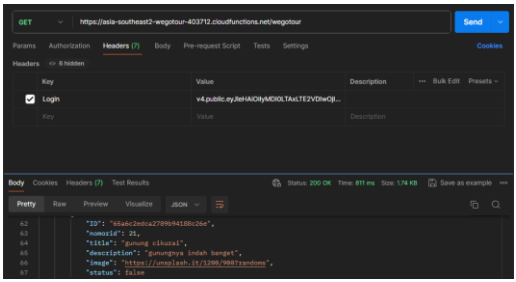
Disini kita akan mencoba test function insertdatatransaksi pada Postman, pada insert transaksi disini mencoba melakukan register pada user dan masukan Kembali pada bagian body-ray-json lalu Send.



Gambar 5. 5 Insert Transaksi Postman

5.5.5 CLOUD FUNCTION GET ALL-TICKET

Selanjutnya melakukan test pada get all data ticket, kita memasukan header yakni Key di isi dengan Login dan Value di isi dengan token user yang didapatkan dari login.

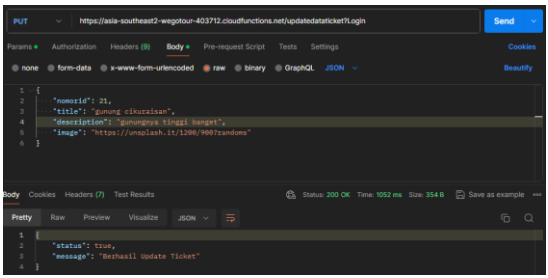


Gambar 5. 6 Get All Ticket Postman

Jika berhasil maka akan tampil seperti gambar di atas, output pada body akan memunculkan seluruh data yang telah di input dan yang sudah ada di collection ticket dan hanya admin yang dapat melihat seluruh data ticket.

5.5.6 CLOUD FUNCTION UPDATE DATA TICKET

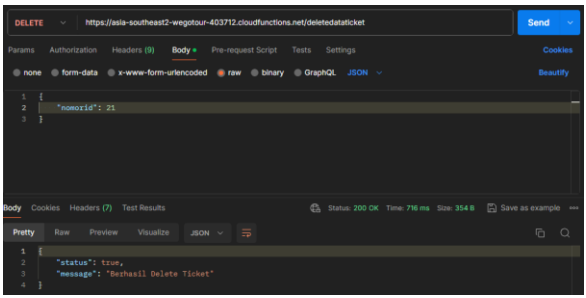
Selanjutnya uji coba pada update data ticket, memasukkan terlebih dahulu header seperti sebelumnya ke Header, kemudian Key dan pada Value di isi kembali dengan Token Admin, karena hanya admin yang dapat meng update data ticket dari collection ticket.



Gambar 5. 7 Update Ticket Postman

5.5.7 CLOUD FUNCTION DELETE DATA TICKET

Selanjutnya pada delete ticket, berfungsi untuk menghapus data ticket dari collection ticket dan di test pada token admin dan memasukan body-raw-JSON dan mengisi syntax dengan nomorid. Jika berhasil maka akan tampil seperti gambar berikut.

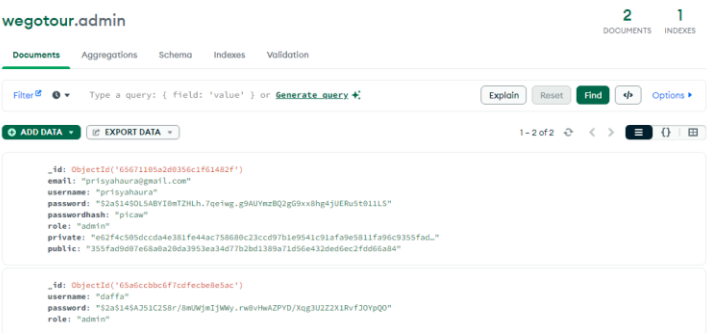


Gambar 5. 8 Delete Data Ticket Postman

5.2 PERBANDINGAN DATABASE

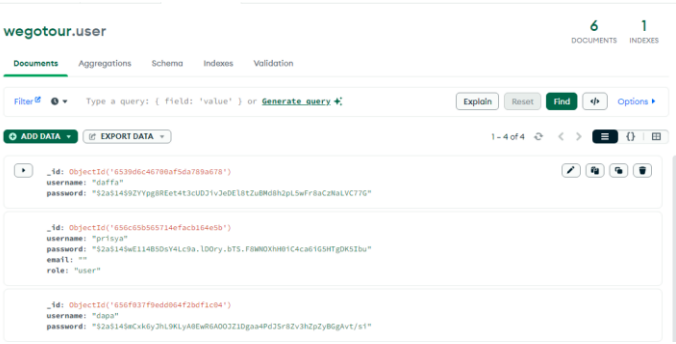
Setelah menjalankan cloud function di postman, dapat di lihat hasil nya dalam Database nya yaitu MongoDB sebagai perbandingan nya. Berikut gambar perbandingan.

1) Database Akun Admin



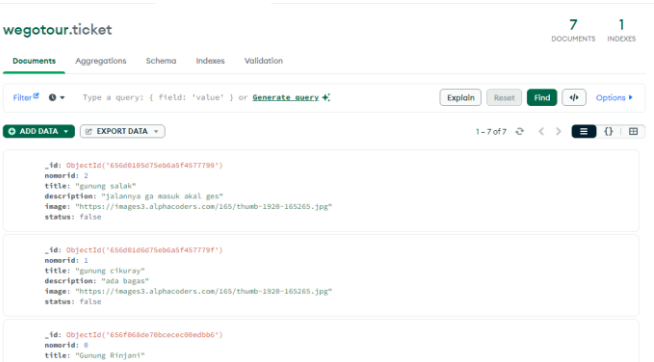
Gambar 5. 9 Database Admin MongoDB

2) Database Akun User



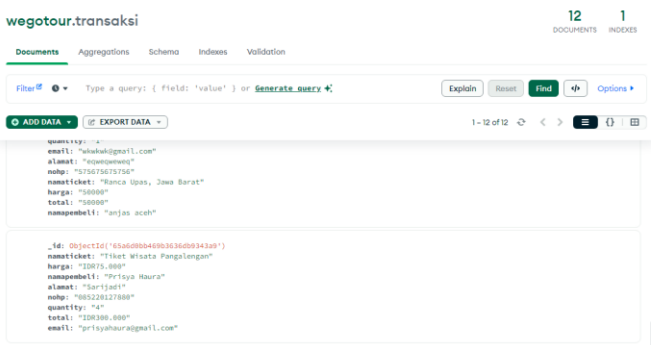
Gambar 5. 10 Database User MongoDB

3) Database Ticket



Gambar 5. 11 Database Ticket MongoDB

4) Database Transaksi



Gambar 5. 12 Database Transaksi MongoDB

BAB VII

Kesimpulan

Berdasarkan apa yang penulis telah kerjakan disini, penulis merasa dengan adanya aplikasi pemesanan tiket tempat wisata secara online ini dapat memudahkan para penikmat wisata dalam melakukan pemesanan tiket objek wisata, walaupun penulis disini sadar masih banyak kekurangan dalam pembuatan serta perancangan aplikasi, tapi penulis berharap dengan selesainya aplikasi ini dapat memudahkan banyak penggunanya.

DAFTAR PUSTAKA

https://pji.uma.ac.id/index.php/2021/12/23/apa-itu-javascript/

http://eprints.polsri.ac.id/8822/3/BAB%202.pdf

https://elibrary.unikom.ac.id/id/eprint/2916/8/12.10115044_MOCHAMAD%20FIQRI_BAB%202.pdf

http://eprints.polsri.ac.id/6838/3/File%20III.pdf

https://rachmat.id/articles/bahasa-pemrograman-golang

https://eprints.utdi.ac.id/8138/3/3_135410027_BAB_II.pdf

https://www.mistercoding.com/post-set/basic-bahasa-pemrograman-go/pengenalan-golang

https://www.softwareseni.co.id/blog/mengenal-go-bahasa-pemrograman-yang-kian-populer-softwareseni-software-house-jakarta

https://medium.com/himit-pens/mengenal-go-language-golang-dan-mengapa-harus-mempelajarinya-e9bd1ddecd72

https://repo.unsrat.ac.id/2977/1/JURNAL_GUMOLUNG_SAMUEL_13021106026.pdf

https://journal.amikmahaputra.ac.id/index.php/JIT/article/download/64/55/

https://jurnal.likmi.ac.id/Jurnal/7_2009/Memanfaatkan_Cascading_BM_.pdf

https://repo.unsrat.ac.id/2910/1/E-Jurnal-Andy_Antonius_Setiawan-13021106091_%28Repaired%29.pdf

https://repository.bsi.ac.id/repo/files/304897/download/File_14-BAB-2.pdf

https://cloud.google.com/functions/docs/concepts/overview

https://www.researchgate.net/profile/Swathi-Kaluvakuri/publication/376841685_Demystifying_Google_Cloud_A_Comprehensive_Review_of_Cloud_Computing_Services/links/6596d6290bb2c7472b32b95c/Demystifying-Google-Cloud-A-Comprehensive-Review-of-Cloud-Computing-Services.pdf

https://link.springer.com/chapter/10.1007/978-3-319-75178-8_34

GLOSARIUM

A

Table: Struktur data yang digunakan untuk menyusun informasi.

Attribute: Kolom yang berada dalam suatu tabel atau entitas dari struktur atau database.

B

Button: Tombol.

Backend: Kode program yang tidak bisa di akses oleh user.

C

Cloud: Penyimpanan melalui layanan internet.

CRUD: Create, Read, Update, Delete.

Create: Membuat.

Commit: Yang mendekteksi perubahan yang terjadi.

Cache: Proses penyimpanan sementara data.

Command: Sebuah perintah.

Code editor: Tempat menyusun, mengedit, menghapus code.

D

Deploy: Kurang lebih sama seperti hosting.

Delete: Menghapus.

Drag: Menarik.

Drop: Lepas.

Database: Kumpulan data yang dikelola berdasarkan ketentuan tertentu, saling berhubungan sehingga mudah dalam pengelolaannya.

Debug: Menguji hasil dari aplikasi apakah berhasil atau ada kendala dalam aplikasi.

Dependensi: Merapikan Struktur code project file dan folder dalam kasus ini kita untuk merapikan file dan folder dalam Bahasa Golang.

E

Error: Kode atau Dependensi Bermasalah.

ENV: Environment.

F

Frontend: Bagian depan dari aplikasi seperti tampilan aplikasi.

Function: Fungsi-fungsi code yang telah dibuat.

Fundamental: Sifat mendasar

Field: Attribute atau kolom.

G

Get: Mendapatkan.

H

Hosting: Mempublikasikan web atau suatu aplikasi ke server internet atau jaringan online.

I

Integrasi: Menyatukan.

Import: Menggunakan, bisa seperti menggunakan library yang telah disediakan oleh bahasa tersebut atau dapat dari orang lain.

J

K

L

Library: Sebuah kerangka code yang bisa kita manfaatkan untuk aplikasi kita.

M

Merge: Menggabungkan.

Master: Cabang di Git yang lebih sering digunakan.

Main: Sebuah cabang di Git.

N

O

Open source: Suatu software atau kode pemrograman komputer yang dipublikasikan secara umum pada orang-orang.

P

Push: Mendorong atau mengirim sebuah project ke suatu repository.

Programmer: Sebuah profesi.

Pull request: Suatu permintaan untuk menggabungkan (merge) kode yang kita modifikasi dengan repositori utama atau repositori lain.

Post: Mengunggah.

Q

R

Repository: Tempat penyimpanan di Github.

Read: Membaca/Mengambil.

S

Source code: Merupakan contoh kode yang telah disediakan oleh programmer.

Skill: Kemampuan.

Sintaks: Penulisan Kode.

Solve: Memperbaiki Kode yang bermasalah.

T

U

Upload: Mengunggah.

Update: Memperbaharui.

V

Verifikasi: Menyetujui.

W

X

Y

Z

INDEKS

A

B

C

D

G

I

K

L

M

P

T

Buku ini berisi panduan untuk pembaca yang ingin meningkatkan pengetahuan mereka dalam bidang pemrograman. Panduan ini membahas cara membangun aplikasi pemesanan tiket bernama Wegotour yang berbasis web dengan menggunakan API dan tiga bahasa pemrograman.

Buku ini ditulis dalam bahasa Indonesia dan disajikan dengan cara yang nyaman, ramah, dan interaktif. Selain itu, buku ini juga dilengkapi dengan penjelasan yang lengkap.