

# Hadekha Erfadila Fitra

## Portofolios

### Manajemen Inventory Koperasi

#### Latar Belakang :

Koperasi LNG Academy memiliki divisi Pengembangan Usaha, dimana salah satu kegiatan usaha nya adalah menjual makanan dan minuman ringan kepada Mahasiswa LNG Academy. Model bisnisnya adalah dengan menitipkan makanan dan minuman ringan pada rumah – rumah yang dihuni mahasiswa LNG Academy. Pencatatan dan pelaporan pada awalnya dilakukan secara tradisional dengan menggunakan Excel. Web App ini dibuat untuk menyelesaikan problem tersebut dengan cara pembeli dan penjual memiliki akun untuk transaksi di website. Selain untuk pencatatan dan pelaporan, Web App ini juga digunakan untuk mendapatkan informasi makanan dan minuman yang paling sering dibeli oleh Mahasiswa.

#### Tech Stack :

Golang, Angular, Postgres, dan Heroku

#### Pembelajaran baru yang Didapat :

CRUD dengan PostGres DBAAS, State Management dengan Redux (NgRx), Deploy menggunakan Heroku

#### Komentar :

Web App ini merupakan proyek CRUD pertama yang *diabadikan* di git (pada github.com). Karena saya sendiri masih dalam tahap belajar, proyek ini memiliki banyak kekurangan. Misalnya

1. Setiap endpoint belum mengikuti standar REST API dan tidak ada dokumentasi backend
2. Alur pembuatan aplikasi belum rapih
3. DBAAS dengan ElephantSQL versi gratis memiliki kekurangan hanya bisa 5 konkuren koneksi di waktu bersamaan
4. Role management pada Web App belum rapih
5. JSON Web Token disimpan pada local storage client
6. Belum ada mekanisme jika Token sudah kadaluarsa maka client mendapat token baru

Link Source Code :

[https://github.com/hdkef/kokesma\\_backend](https://github.com/hdkef/kokesma_backend)

[https://github.com/hdkef/kokesma\\_front](https://github.com/hdkef/kokesma_front)

### Hello Hooman

#### Latar Belakang :

saya ingin memahami pembuatan web yang mengimplementasikan WebRTC dan berujung belajar mengenai implementasi WebSocket dengan Golang.

**Tech Stack :**

Golang, Angular, dan Heroku

**Pembelajaran baru yang Didapat :**

penerapan WebSocket, penerapan WebRTC, penerapan concurrency di Golang.

**Komentar :**

Implementasi WebRTC cukup rumit karena saat testing, Web App harus dapat diakses oleh dua device. Selain itu, jika terjadi error, perlu tahu apakah error tersebut berasal dari server WebRTC atau dari WebApp karena server WebRTC yang gratis sendiri banyak yang tidak dapat digunakan.

**Link Source Code :**

[https://github.com/hdkef/hellohuman\\_back](https://github.com/hdkef/hellohuman_back)

[https://github.com/hdkef/hellohuman\\_front](https://github.com/hdkef/hellohuman_front)

## Chirpper

**Latar Belakang :**

saya ingin menerapkan Firebase, menerapkan email verification, dan juga membuat mekanisme CRUD file upload pada backend, frontend karena pada project sebelumnya selalu menggunakan JSON (bukan multipart).

**Tech Stack :**

Golang, Angular, Firebase, dan Heroku

**Pembelajaran baru yang Didapat :**

Implementasi Firebase, Integrasi File Upload Frontend dan Backend, Email verification

**Komentar :**

Ketika proyek ini berjalan, Firebase untuk Golang belum menerapkan fungsi append pada field bertipe array (sedangkan pada bahasa lain sudah ada) sehingga beberapa fitur pada Web App ini belum dibuat, misalnya fitur follow dan unfollow. Lalu untuk fitur email verification sendiri menggunakan server Gmail dan saat itu servernya sering error.

Selain itu WebApp ini juga masih memiliki beberapa kekurangan, misalnya :

1. Pada frontend masih belum dibuat modal loading
2. Respon error dari backend belum diproses ke elemen UI
3. JSON Web Token disimpan pada local storage client

4. Belum ada mekanisme jika Token sudah kadaluarsa maka client mendapat token baru

**Link Source Code :**

[https://github.com/hdkef/chirpper\\_backend](https://github.com/hdkef/chirpper_backend)

[https://github.com/hdkef/chirpper\\_front](https://github.com/hdkef/chirpper_front)

## **The Light CMS**

**Latar Belakang :**

saya ingin membuat Content Management System sendiri dari awal menggunakan Golang dan Angular.

**Tech Stack :**

Golang, Angular, Postgres, dan Heroku

**Pembelajaran baru yang Didapat :**

Pagination dengan efektif dan efisien (front end cache).

**Komentar :**

pembuatan pagination yang efektif dan efisien yang menerapkan caching pada front end lumayan challenging karena adanya implementasi state management NgRx, alias saya perlu menggali lebih dalam mengenai pengelolaan *observable*. Adapun kekurangan dari Web App ini salah satunya adalah JSON web token masih disimpan di local storage client.

**Link Source Code :**

[https://github.com/hdkef/thelight\\_backend](https://github.com/hdkef/thelight_backend)

[https://github.com/hdkef/thelight\\_front](https://github.com/hdkef/thelight_front)

## **Api Lokasi Indonesia**

**Latar Belakang :**

sebelumnya saya mengclone api-lokasi-indonesia dari repository milik *farizdotid* (dibuat dengan bahasa PHP) untuk digunakan pada salah satu Web App. Kemudian, saya memerhatikan repository tersebut dan ternyata mekanisme nya dengan mendecode beberapa file csv. Selain itu, saya juga ingin membuat image dari aplikasi ini ke docker hub.

**Tech Stack :**

Golang dan Docker

**Pembelajaran baru yang Didapat :**

decode csv, docker

**Komentar :**

Golang tidak memiliki generics, sehingga pembuatan model Province, City, District, dan Village masing-masing memiliki fungsi yang berbeda-beda, alias tidak bisa dibuatkan satu fungsi dengan generics yang dapat menerima keempat model tersebut. Adapun kekurangan dari api ini adalah masih belum menerapkan spesifikasi OpenApi sehingga belum rapih.

**Link Source Code :**

<https://github.com/hdkef/api-lokasi-indonesia>

## Api Quran Indonesia

**Latar Belakang :**

Setelah dapat mendecode csv dari pembelajaran sebelumnya, muncul ide untuk membuat api quran Indonesia. Untuk mendapat CSV berisi id,nomor surat, ayat, nama surat, juz, arab dan juga terjemahan dalam bahasa, perlu mendownload dua file csv yang berbeda file csv pertama berisi quran dalam arabic namun tidak memiliki terjemahan bahasa sedangkan file csv kedua terdapat terjemahan bahasa dan tidak ada arabic nya. Kemudian dua file ini akan didecode dan dilakukan looping sedemikian rupa agar membuat file csv baru berisi id,nomor surat, ayat, nama surat, juz, arab dan juga terjemahan

**Tech Stack :**

Golang dan Docker

**Pembelajaran baru yang Didapat :**

decode csv, docker

**Komentar :**

Pembuatan Web App ini rumit diawal karena harus decode dua csv dan membuat file csv baru berupa gabungan data dari kedua csv dengan double looping. Adapun kekurangan dari api ini adalah masih belum menerapkan spesifikasi OpenApi sehingga belum rapih.

**Link Source Code :**

<https://github.com/hdkef/api-quran-indonesia>

## Bloggo Headless CMS

**Latar Belakang :**

Bloggo merupakan Tugas Akhir dari Sanber Bootcamp yang saya ikuti.

**Tech Stack :**

Golang

**Pembelajaran baru yang Didapat :**

decode csv, docker

**Komentar :**

Sebelum mengikuti Bootcamp ini, saya sudah belajar mengenai standar dari REST API dan mencoba membuat dokumentasi menggunakan Swagger dengan spesifikasi OpenAPI. Namun, Swagger pada Golang ini berbeda dari yang saya pelajari karena pembuatan dokumentasinya dengan cara pembuatan komentar komentar khusus dengan decorator @

**Link Source Code :**

<https://github.com/hdkef/Final-Project---BDS-Sanbercode-Golang-Batch-28>

**Kesiniblog.com**

**Latar Belakang :**

saya ingin memahami bagaimana caranya konfigurasi web server (NGINX), konfigurasi server, ssh, dan membiasakan diri menggunakan Linux CLI.