

## PROJECT-BASE TASK ALTERRA ACADEMY

### Objektif

Tujuan dari tugas ini adalah untuk menguji kemampuan kamu dalam bidang Software Engineering khususnya BackEnd Development. Detail ekspektasi yang diharapkan dari tugas secara detail dijelaskan dibawah.

### Latar Belakang

ALTA Store adalah sebuah toko ritel yang menjual berbagai macam produk secara offline. Produk yang dijual diantaranya adalah buku, barang elektronik, alat olahraga, dan lain-lain. ALTA Store memiliki pelanggan dari berbagai daerah di Indonesia, sehingga CEO ALTA Store memutuskan untuk mengembangkan aplikasi web agar pelanggan dapat membeli produk-produk yang dijual secara online.

Sebagai seorang BackEnd Engineer tugas kamu adalah membantu mewujudkan keinginan dari CEO untuk membuat aplikasi web ALTA Store. Kamu tidak perlu membuat semua bagian dari aplikasi ini, Tidak perlu membuat FrontEnd, Melainkan fokus ke BackEnd (RESTful API) saja.

Kamu dapat mengembangkan aplikasi dengan memulai dari fitur-fitur yang prioritas dulu. Berikut fitur prioritas untuk memenuhi kriteria MVP (Minimum Viable Product):

- Customer dapat melihat **list products** dan bisa berdasarkan **product category, dan name (filter)**
- Customer dapat memasukan **product** ke **shopping cart**
- Customer dapat melihat **list product** yang telah ditambahkan pada **shopping cart**
- Customer dapat menghapus **list product** pada **shopping cart**
- Customer dapat melakukan **checkout** dan melakukan **transaksi pembayaran**
- Admin menambahkan product dan category
- Admin dapat mengganti status checkout / pembayaran
- **Login** dan **Register** Customer

### Output Yang Diharapkan

#### System Design

- Buatlah README yang bagus dalam project yang kamu kerjakan ini!
- RESTful API Design

- RESTful API adalah metode untuk membuat API sesuai dengan metode HTTP. Buatlah RESTful API Design menggunakan OpenApi Spec (<https://swagger.io/docs/specification/about/>)
- Atau kamu dapat menggunakan README untuk dokumentasi RESTful API.
- Entity Relationship Diagram (ERD)
  - ERD adalah suatu bentuk diagram yang menjelaskan hubungan antar objek-objek data yang mempunyai hubungan antar relasi. Kamu dapat menggunakan tools (<https://app.diagrams.net/>) untuk membuat ERD.

## BackEnd Development

- Implementasikan Git Workflow / Trunk Base sesuai dengan best practice yang ada dalam dunia Software Development. Selalu terapkan Git Workflow dalam tahapan pengembangan dan penambahan fitur. Buat penamaan **branch** dan **commit message** yang sesuai dalam setiap penambahan fitur baru!
- Implementasikan Database dari ERD yang telah kamu buat sebelumnya ke aplikasi RDBMS seperti MySQL (boleh juga menggunakan database lain seperti Postgres, dll)
- Implementasikan RESTful API yang sebelumnya telah kamu buat ke aplikasi web backend yang bisa digunakan secara nyata menggunakan **JWT Authentication**.
- Implementasikan Unit Testing untuk BackEnd Application dengan code coverage mencapai  $\geq 50\%$  untuk folder bisnis/usecase (Optimal nya  $> 90\%$ )
- Gunakan clean architecture

## Deployment

- Build kedalam Docker Images dan Upload ke registry docker (docker hub)
- Deploy aplikasi BackEnd ke Cloud Provider AWS
- Gunakan CI/CD Github Action

## Nilai tambah :

- Menambahkan database nosql mongodb - option logging
- Gunakan Open API (ex: Sandbox Midtrans)
- Gunakan microservices
- Gunakan komunikasi antar service menggunakan message broker (mqtt / kafka)
- Kamu dapat menambahkan fitur lain diluar fitur utama yang disebutkan!
- Kamu dapat mengintegrasikan API kamu yang buat dengan API sesuai kebutuhan kamu, seperti api pembayaran, api pengiriman barang, dsb.

## Kriteria Penilaian

### System Design

- README Documentation
- RESTful API Design
- Entity Relationship Diagram

### BackEnd Development

- Git Workflow (*branch name, commit message*)
- Database Implementation
- RESTful API (Kelengkapan Fitur)
- Clean Architecture (Uncle Bob / Hexagonal)
- Unit Testing Coverage

### Deployment

- Docker
- Computer Service (AWS)
- CI/CD (Github Action)

## Reference

- <https://blog.readme.com/how-to-use-openapi-and-swagger-spec-for-documentation/>
- <https://github.com/jamescooke/restapidocs/blob/master/examples/README.md>
- <https://app.swaggerhub.com/apis-docs/sepulsa/RentABook-API/1.0.0#/>
- <https://www.lucidchart.com/pages/er-diagrams>