



# Final Project Requirements Backend Engineer Track

Pelatihan intensif selama 5 bulan, setara dengan 20 SKS untuk mempersiapkan tenaga ahli digital.

## Introduction

Perusahaan fashion ecommerce dengan pangsa pasar "Indonesian Young Urbans" -- pemuda-pemudi dengan rentang umur 15-35 tahun -- berdiri di Indonesia sejak awal tahun 2016 dengan nama Fashion Campus. Perusahaan ini menyajikan katalog brand-brand lokal hingga internasional yang digandrungi anak muda. Oleh karena banyak bekerja sama dengan brand lokal, setelah beroperasi selama satu tahun lebih, mereka berhasil memperoleh cukup banyak return customers dengan pengguna aktif 10.000 per Bulan Juni 2022 dan menerima lebih dari 100.000 pesanan setiap bulannya.

Sejak pandemi menyerang pada tahun 2020, Fashion Campus melihat potensi pada perkembangan belanja digital karena lebih banyak waktu bagi masyarakat dalam mengakses internet. Lisna dan Wira, sebagai salah satu Tim Marketing, diminta untuk melakukan riset pasar. Hasilnya, mereka menemukan bahwa selama pandemi ini juga muncul tren baru di kalangan target market Fashion Campus. Ternyata, "Indonesian Young Urbans" mulai banyak melakukan praktik thrifting atau jual beli pakaian bekas. Dari penemuan ini, Lisna dan Wira mengajukan usulan ke bagian Tim Business Development untuk mengembangkan bisnis model penjualan pakaian bekas yang masih layak pakai.

Pengembangan bisnis model oleh tim Business Development ini nantinya akan dibantu oleh empat tim, yaitu tim Data Science, Tim UI/UX, Tim Artificial Intelligence dan Tim Backend. Lisna dan Wira juga sudah menyampaikan ke tim yang terlibat bahwa waktu pengembangan tidak banyak. Seluruh tim diharapkan dapat mempresentasikan hasil temuannya kepada jajaran petinggi Fashion Campus sesuai dengan timeline yang diberikan.

## What must the backend team do?

Karena saat ini semua serba digital dan Fashion Campus akan menjalankan strategi dalam *multichannel marketing*, tim Product Fashion Campus berencana untuk mengembangkan web app baru untuk bisnisnya. Dengan berbagai pertimbangan dan permintaan dari konsumen, tim Product meminta beberapa fitur ada di web tersebut. Tim Backend bertugas membantu tim Product untuk membuat web application yang sesuai dengan fitur-fitur yang diminta. Selain itu, API yang telah dibuat oleh tim Backend nantinya akan dihubungkan dengan frontend yang telah disediakan oleh tim Frontend. Tim Backend ga sendirian! Karena dalam misi ini, tim Backend akan berkolaborasi dengan tim AI untuk memasukkan beberapa fitur tertentu.

Karena akan ada campaign yang harus segera dijalankan, tim Backend harus bergerak cepat untuk menyelesaikan misi ini! Supervisor Fashion Campus untuk tim Backend sudah menentukan timeline pengerjaan web ini sehingga pengerjaan diharapkan selesai tepat waktu.

## Assessment Criteria

Berikut adalah kriteria penilaian yang akan digunakan pada Final Proeject Backend Engineer Track Startup Campus batch 3 beserta dengan bobotnya:

1. Semua API berhasil diimplementasikan (55%)
2. Berhasil melakukan deployment menggunakan Google Cloud Ecosystem (30%)
3. Presentasi: (15%)

Aspek yang dinilai saat presentasi adalah sebagai berikut:

1. Substansi - seluruh requirements tercakup dalam presentasi
2. Komunikasi - lugas, on point
3. Visual - mudah dibaca dan dipahami
4. Penguasaan materi terutama pada saat Q&A dengan penilai

# Timeline Pengerjaan

## Week 1

Pembentukan tim, info outcome, kasus, milestone, rubrik penilaian, showcase.

## Week 2-3

- Menerima handover requirements dari tim SC
- Mendefinisikan API yang akan dibuat
- Mulai membuat API yang diperlukan dalam web app

## Week 4-6

- Melanjutkan pembuatan API sesuai dengan requirements
- Menghubungkan API yang telah dibuat dengan frontend yang telah disediakan oleh tim SC
- Menghubungkan web app dengan server pada Google Cloud Platform (GCP)

## Week 7

- Menerima handover dari tim AI untuk fitur “search with image”
- Update deployment web application ke dalam server

## Week 8

Finalisasi deployment dan persiapan selection to final project showcase.

## Showcase

Presentasi dari best team, penyerahan hadiah, dan informasi graduation.

# ENDPOINTS

Berikut requirements berupa API endpoints yang perlu dilengkapi oleh peserta. Terdapat total 28 endpoints yang memiliki bobot nilai bervariasi dengan total 29 poin nilai. Keberhasilan penyelesaian endpoints akan masuk ke dalam kriteria penilaian "API".

## UNIVERSAL

Terdapat 1 (satu) endpoint universal yang perlu dibuat oleh peserta, yaitu endpoint untuk mengambil atau memunculkan gambar dalam platform.

**For any error messages**, you should return with this response.

Response:

Key	Example
message	error, user already exists

### 1. Get Image

Bobot nilai: 1 poin

Metode: GET

URL: /image/{image\_name.extension}

Response: image file

## HOME

Pada menu Home, terdapat 2 (dua) endpoints yang harus dikerjakan oleh peserta, yaitu sebagai berikut.

### 2. Get Banner

Bobot nilai: 0.5 poin

Metode: GET

URL: /home/banner

Response:

Key	Example
data	<pre>{   "data": [     {       "id": "uuid",       "image": "/something/image.png",       "title": "lorem ipsum blablabla"     }   ] }</pre>

## HOME

### 3. Get Category

Bobot nilai: 0.5 poin

Metode: GET

URL: /home/category

Response:

Key	Example
data	<pre>{   "data": [     {       "id": "uuid",       "image": "/something/image.png",       "title": "Category A"     }   ] }</pre>

## AUTHENTICATION

Untuk melakukan *authentication*, terdapat 2 (dua) endpoints yang harus dikerjakan oleh peserta, yaitu Sign-up dan Sign-in sebagai berikut:

### 4. Sign-up

Bobot nilai: 1 poin

Metode: POST

URL: /sign-up

Request Body

Key	Example
name	Raihan Parlaungan
email	raihan@gmail.com
phone_number	081380737126
password	password1234

Response

Key	Example
message	success, user created

## 5. Sign-in

Bobot nilai: 1 poin

Metode: POST

URL: /sign-in

Request Body

Key	Example
email	raihan@gmail.com
password	password1234

Response

Key	Example
user_information	{ "name": "Raihan Parlaungan", "email": "raihan@gmail.com", "phone_number": "08138073126" "type": "buyer" ( there are 2 types of user, "buyer" and "seller" }
token	jwt_token
message	Login success

## PRODUCT LIST

Pada menu Product List, terdapat 4 (empat) endpoints yang perlu diselesaikan oleh peserta, sebagai berikut:

### 6. Get Product List

Bobot nilai: 1 poin

Catatan: Hanya menunjukkan produk yang 'available' → produk belum dihapus dan 'category' masih available., dan total\_rows adalah berapa banyak total hasil yang didapatkan dengan semua filter yang ada

Metode: GET

URL: /products

Query Params

Key	Example
page	1
page_size	100
sort_by	Price a_z, Price z_a
category	Id category a, id category b
price	0,10000
condition	used
product_name	name

## 6. Get Product List

Response

Key	Example
data	<pre>{   "data": [     {       "id": "uuid",       "image": "/something/image.png",       "title": "Item A",       "price": 15000     }   ] }</pre>
total_rows	10

## 7. Get Category

Bobot nilai: 0.5 poin

Metode: GET

URL: /categories

Response

Key	Example
data	<pre>{   "data": [     {       "id": "uuid",       "title": "Category A"     }   ] }</pre>

## 8. Search Product by Image

Bobot nilai: 1 poin

Metode: POST

URL: /products/search\_image

Query params



## 8. Search Product by Image

Request body:

Key	Example
image	Base_64 image

Response

Key	Example
category_id	uuid

## PRODUCT DETAIL PAGE

Pada halaman product detail, akan ada 2 (dua) endpoints yang perlu diselesaikan oleh peserta, sebagai berikut:

### 9. Get Product Details

Bobot nilai: 1 poin

Metode: GET

URL: /products/{id}

Response:

Key	Example
data	<pre>{   "id": "uuid",   "title": "Nama product",   "size": ["S", "M", "L"],   "product_detail": "lorem ipsum",   "price": 1000,   "images_url": ["/image/image1", "/image/image2"], }</pre>

## 10. Add to Cart

API untuk menambahkan produk menuju menu 'cart' pada user

Bobot nilai: 2 poin

Catatan:

- Hanya bisa di panggil oleh user yang sudah login, jika belum login, return error message
- Items di cart bersifat unique berdasarkan item\_id dan size
- Jika menambahkan item yang mempunyai id dan size yang sama ke dalam cart, akan menambahkan quantity item di cart tersebut

Metode: POST

URL: /cart

Headers:

Key	Example
Authentication	Jwt token

Request body

Key	Example
id	item_id
quantity	10
size	M

Response

Key	Example
message	Item added to cart

## CART

Pada section Cart, terdapat 5 (lima) endpoints yang harus diselesaikan oleh peserta.

Catatan: semua request dari endpoint yang terdapat dalam section "Cart" akan memiliki headers sebagai berikut:

Headers:

Key	Example
Authentication	Jwt token

## 11. Get User's Carts

Bobot nilai: 0.5 poin

Metode: GET

URL: /cart

Response

Key	Example
data	<pre>[   {     "id": "uuid",     "details": {       "quantity": 100,       "size": "M"     },     "price": 10000,     "image": "/url/image.jpg",     "name": "Product a"   } ]</pre>

## 12. Get User's Shipping Address

Bobot nilai: 0.5 poin

Metode: GET

URL: /user/shipping\_address

Response

Key	Example
data	<pre>{   "id": "uuid",   "name": "address name",   "phone_number": "082713626",   "address": "22, ciracas, east jakarta",   "city": "Jakarta" }</pre>

### 13. Get Shipping Price

Pada endpoint berikut, peserta perlu membuat endpoint untuk menghitung shipping price

Bobot nilai: 2 poin

Catatan:

- Hanya bisa diakses oleh user yang sudah login
- Hanya bisa diakses oleh user yang sudah memiliki cart
- Dihitung based on cart user yang sekarang
- Akan ada 2 jenis shipping method:
  - Regular:
    - Jika total harga item < 200: Shipping price merupakan 15% dari total harga item yang dibeli
    - Jika total harga item >= 200: Shipping price merupakan 20% dari total harga item yang dibeli
  - Next Day:
    - Jika total harga item < 300: Shipping price merupakan 20% dari total harga item yang dibeli
    - Jika total harga item >= 300: Shipping price merupakan 25% dari total harga item yang dibeli

Metode: GET

URL: /shipping\_price

Headers

Key	Example
Authentication	Jwt token

Response

Key	Example
data	<pre>[   {     "name": "regular",     "price": 30000   },   {     "name": "next day",     "price": 20000   } ]</pre>

## 14. Create Order

Endpoint ini dibuat untuk memproses pesanan yang dilakukan user.

Bobot nilai: 1.5 poin

Catatan:

- Hanya bisa diakses oleh user yang sudah login
- Akan mengurangi balance user berdasarkan total harga
- Jika balance kurang dari total harga, return error
- Akan menghapus semua cart yang ada di user tersebut

Metode: POST

URL: /order

Request body:

Key	Example
shipping_method	"Same day"
shipping_address	{ "name": "address name", "phone_number": "082713626", "address": "22, ciracas, east jakarta", "city": "Jakarta" }

Response

Key	Example
message	Order success

## 15. Delete Cart Item

Menghapus salah satu produk dari menu "Cart".

Bobot nilai: 1 poin

Metode: DELETE

URL: /cart/cart\_id

Response

Key	Example
message	Cart deleted

## PROFILE PAGE

Terdapat 5 (lima) endpoints baru yang perlu diselesaikan oleh peserta, yaitu 1) User Details; 2) Change Shipping Address; 3) Top-up Balance; 4) Get User Balance; 5) User Orders.

Selain itu, pada halaman ini, peserta juga perlu memanggil kembali endpoint yang sebelumnya sudah dibuat: "Get User Shipping Address".

Semua endpoints dalam Profile Page harus memiliki headers berikut.

### Headers

Key	Example
Authentication	Jwt token

### 16. User Details

Bobot Nilai: 1 poin

Metode: GET

URL: /user

Response

Key	Example
data	<pre>{   "name": "Raihan Parlaungan",   "email": "raihan@gmail.com",   "phone_number": "08138073126" }</pre>

### 17. Change Shipping Address

Bobot nilai: 1 poin

Metode: POST

URL: /user/shipping\_address

Response

Key	Example
name	address_name
phone_number	082713626
address	22, ciracas, east jakarta,
city	Jakarta

### 18. Top-up Balance

Menambahkan balance pada akun user yang digunakan untuk pembelian.

Bobot nilai: 1 poin

Metode: POST

URL: /user/balance

Query params:

Key	Example
amount	10000

Response

Key	Example
message	Top Up balance success

### 19. Get User Balance

Menampilkan current balance yang dimiliki oleh user.

Bobot nilai: 1 poin

Metode: GET

URL: /user/balance

Response:

Key	Example
data	{ "balance": 10000 }

#### 19.5 Endpoint tambahan dalam halaman Profile:

[Get User Shipping Address](#)

## 20. User Orders

Berupa informasi mengenai pesanan yang telah dilakukan oleh pengguna.

Bobot nilai: 1 poin

Catatan: "price" dari products disini adalah harga asli dari product \* jumlah product yang di checkout.

Metode: GET

URL: /order

Response:

Key	Example
data	<pre>[   {     "id": "uuid",     "created_at": "Mon, 22 august 2022",     "products": [       {         "id": "uuid",         "details": {           "quantity": 100,           "size": "M"         },         "price": 10000,         "image": "/url/image.jpg",         "name": "Product a"       }     ],     "shipping_method": "same day",     "status": "waiting",     "shipping_address": {       "name": "address name",       "phone_number": "082713626",       "address": "22, ciracas, east jakarta",       "city": "Jakarta"     }   } ]</pre>



## ADMIN PAGE

Terdapat total 4 (empat) endpoints baru yang perlu dibuat oleh peserta dalam Admin Page beserta 2 (dua) endpoints tambahan yang telah dibuat pada section sebelumnya.

Catatan: semua yang ada di dalam admin page hanya bisa diakses oleh user dengan level admin dan memiliki headers sebagai berikut.

Headers

Key	Example
Authentication	Jwt token

### 21. Get Orders

Bobot Nilai: 1 poin

Catatan: When query\_params "is\_admin" is true, return all orders from all user, with additional key:

- Created\_at
- User\_id
- email

Metode: GET

URL: /orders

Query params:

Key	Example
sort_by	Price a_z, Price z_a
page	1
page_size	25
is_admin	True(boolean)

## 21. Get Orders

Response:

Key	Example
data	<pre>{   "data": [     {       "id": "uuid",       "title": "Nama product",       "size": [         "S",         "M",         "L"       ],       "created_at": "Tue, 25 august 2022",       "product_detail": "lorem ipsum",       "email": "raihan@gmail.com",       "images_url": [         "/image/image1",         "/image/image2"       ],       "user_id": "uuid",       "total": 1000     }   ] }</pre>

## 22. Create Product

Menambahkan sebuah product melalui admin page.

Catatan: produk unique berdasarkan: name, category, condition

Bobot nilai: 2 poin

Metode: POST

URL: /products

Request Body:

Key	Example
product_name	Product 1
description	Lorem ipsum
images	[image_1, image_2, image_3]
condition	new
category	category_id
price	10000

Response:

Key	Example
message	Product added

## 23. Update Product

Melakukan perubahan pada item product yang sebelumnya telah dibuat.

Catatan: produk unique berdasarkan: name, category, condition.

Bobot nilai: 1 poin

Metode: PUT

URL: /products

Request Body

Key	Example
product_name	Product 1
description	Lorem ipsum
images	[image_1, image_2, image_3]
condition	new
category	category_id
price	10000
product_id	product_id

## 23. Update Product

Response:

Key	Example
message	Product updated

## 24. Delete Product

Menghapus item product yang sebelumnya telah dibuat.

Bobot nilai: 1 poin

Catatan: hanya berupa “soft delete” jadi semua orders/pesanan yang terdapat product tersebut tidak terhapus juga.

Metode: DELETE

URL: /products/:product\_id

Response:

Key	Example
message	Product deleted

## 24.5 Endpoint tambahan dalam Admin Page

[Get Category](#)

## 25. Create Category

Catatan: category is unique by name

Bobot nilai: 1 poin

Metode: POST

URL: /categories

Request body:

Key	Example
category_name	Category A

Response:

Key	Example
message	Category added

## 26. Update Category

Bobot nilai: 1 poin

Catatan: category unique by name

Metode: PUT

URL: /categories/{category\_id}

Request body:

Key	Example
category_name	Category A
category_id	category_id

Response:

Key	Example
message	Category updated

## 27. Delete Category

Bobot nilai: 1 poin

Catatan: hanya berupa “soft delete” jadi semua orders/pesanan yang terdapat product tersebut tidak terhapus juga.

Metode: DELETE

URL: /categories/{category\_id}

Response:

Key	Example
message	Category deleted

## 28. Get Total Sales

Bobot nilai: 1 poin

Metode: GET

URL: /sales

Response:

Key	Example
data	<pre>{   "data": {     "total": 100000   } }</pre>