

# BAB 1

## PENDAHULUAN

---

### 1.1 Latar Belakang

Saat ini banyak sekali perusahaan-perusahaan yang memanfaatkan teknologi metamesta dalam pelayanan maupun aktifitas internal mereka. Langkah ini yang diambil oleh PT. Bank Negara Indonesia Persero Tbk dalam memperluas jangkauan perusahaan dalam memberikan pelayanan penuh kepada nasabah bank. Nantinya pengguna mampu mengenal produk yang ada pada BNI lewat digitalisasi pada proyek metamesta ini.

Disinilah tugas bagi Divisi Pengembangan Digital dalam melakukan mengembangkan *metaverse*. Divisi ini bertugas dalam mengembangkan aplikasi, website, ataupun layanan digital BNI lainnya. Salah satunya adalah proyek metamesta BNITopia yang sedang dikembangkan ini. Proyek ini bertujuan untuk mengenalkan produk BNI kepada nasabah lewat digitalisasi dalam dunia metamesta. Dalam pembagian divisinya, terdapat 2 divisi diantaranya Divisi Metaverse dan Divisi 2D/3D Model. Dimana Divisi Metaverse bertugas dalam pengembangan dunia virtual metamesta serta Divisi 2D/3D Model bertugas dalam pengembangan *assets* yang akan digunakan pada dunia metamesta nantinya. Penulis sendiri masuk kedalam sub divisi yang ada pada Divisi Metaverse yaitu Divisi Backend. Dengan *task* yang diberikan kepada Penulis berupa pengembangan *REST API* pada Model AppUser dan Portal.

### 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah yang dapat disimpulkan adalah belum adanya sistem *REST API* Model AppUser dan Portal untuk menampung data pengguna serta data portal pada metamesta BNITopia.

### 1.3 Tujuan

Tujuan dari proyek backend REST API BNITopia ini adalah untuk membangun web servis untuk saling berintegrasi. Dengan adanya Model AppUser dan Model Portal sistem mampu menampung data pengguna dan data koordinat portal.

## 1.4 Batasan Masalah

Pada laporan praktik magang ini terdapat beberapa ruang lingkup yaitu sebagai berikut:

1. Penulis mendapatkan tugas sebagai *backend developer*.
2. Konsep dan aturan *REST API* yang dibuat berdasarkan ketentuan perusahaan.
3. Ada beberapa *API* yang ada, diantaranya yang penulis kerjakan :
  - a. Model AppUser.
  - b. Model Portal.

## 1.5 Definisi Operasional

Dengan *REST API*, aplikasi dapat berkomunikasi antara web servis yang dibuat serta bertukar data satu sama lain. Seperti halnya BNITopia yang Tim Metaverse kembangkan nantinya akan mampu berkomunikasi satu sama lain terhadap *REST API* yang Penulis buat.

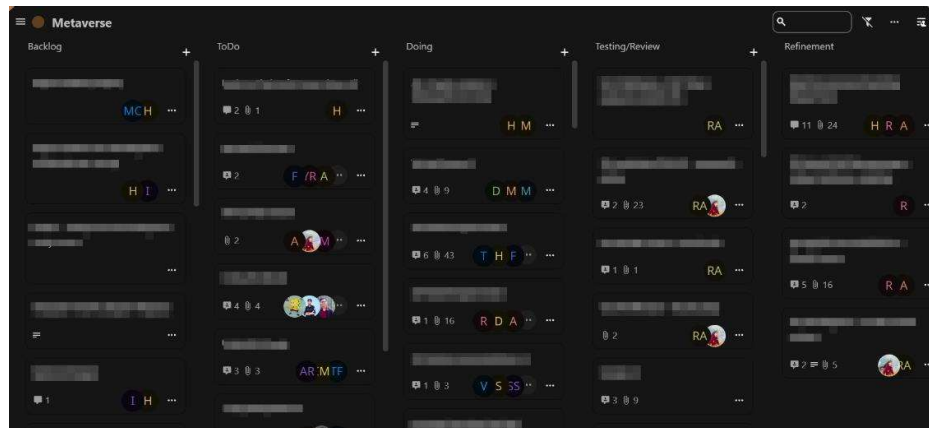
*Backend* merupakan bagian dari sebuah sistem komputer atau aplikasi yang berfungsi untuk mengelola dan memproses data serta menjalankan logika bisnis dari suatu aplikasi. Secara umum, sebuah aplikasi atau situs web terdiri dari dua komponen utama: *frontend* (tampilan antarmuka pengguna) dan *backend* (bagian yang tersembunyi dari sistem yang berjalan di belakang layar).

AppUser merupakan sebutan tabel pengguna dalam pengembangan *backend* REST API BNITopia yang terdiri dari kata *application user*. Tabel AppUser ini nantinya menampung data diri pengguna yang terdiri dari nama, email, password dsb.

Portal adalah sebutan bagi sebuah fitur yang menjembatani pengguna antara fitur Space yang merupakan dunia virtual dari BNITopia. Nantinya, pengguna dapat berpindah dari Space satu dengan yang lainnya menggunakan fitur Portal ini. Portal juga akan menampung data koordinat objek 3D yang dihasilkan di dunia virtual untuk nantinya disimpan pada tabel portal di basis data.

## 1.6 Metode Pengerjaan

Pada proses pengembangannya Penulis mengikuti model pengembangan yang ada dan ditetapkan oleh BNI. Pada Tim Metaverse sendiri menggunakan metodologi *Agile* dimana semua kebutuhan dan pengerjaan didasarkan pada proses pengerjaan berulang.



**Gambar 1.1 Penerapan metode Kanban**

Penerapan metodologi Agile Kanban dibagi menjadi beberapa bagian diantaranya *backlog*, *todo*, *doing*, *testing/review*, *refinement* dan *done*. Dimana semua proses didasari pada *card-card* atau proses yang berkesinambungan.

1. Backlog

Proses *Backlog* merupakan proses pengumpulan kebutuhan yang ada disertai identifikasi masalah yang sesuai dengan kebutuhan sistem. Masalah yang teridentifikasi nantinya mampu menghasilkan tujuan dari *task* yang akan dikerjakan.

2. Todo

Pada proses ini semua hasil identifikasi masalah serta kebutuhan sistem didapat dan disimpulkan untuk menjadi *task* yang nantinya dapat dikerjakan oleh individual tim.

3. Doing

Pekerjaan *task* yang diberikan sesuai individu maupun tim nantinya dilakukan bersama-sama dengan saling berintegrasi antara *task* yang diberikan. Di dalam proses *Doing* seseorang bisa melakukan konsultasi ke proyek manager atau mentor terkait mengenai *task* apabila terdapat beberapa perubahan maupun kendala dalam pengerjaan.

4. Testing/Review

Pada testing dilakukan pengetesan pada *task* yang nantinya akan di *review* oleh proyek manager atau mentor terkait. Beberapa *task* yang belum sesuai atau terdapat *error* akan dikerjakan kembali pada bagian *Doing*. Dan proses ini akan terus berkelanjutan hingga *task* tersebut bersih dari *error*.

5. Refinement

Proses refinement merupakan bagian penting yang bertujuan untuk memperbaiki dan meningkatkan kualitas *Backlog*. Proses ini melibatkan anggota tim dalam melakukan

perbaikan, penambahan atau pengurangan pada *Backlog* agar lebih tepat dan sesuai dengan tujuan serta kebutuhan.

#### 6. Done

*Task* yang sudah selesai akan masuk kedalam bagian *Done* untuk nantinya sebagai informasi bagi tim/individu lain apabila mempunyai keterkaitan terhadap kebutuhan *task* yang sudah selesai.

### 1.7 Jadwal Pengerjaan

Pada proses pengerjaan dilakukan selama 4 bulan dari Oktober 2022 hingga Januari 2023. Dibagi menjadi 6 tahap pengerjaan diantaranya pengumpulan, analisis, perancangan, implementasi, pengujian dan penyesuaian sehingga dapat di jelaskan pada Tabel 1.1 berikut.

Tabel 1.1 Jadwal Pengerjaan API

No	Kegiatan	Waktu Pelaksanaan															
		Bulan 1				Bulan 2				Bulan 3				Bulan 4			
		m 1	m 2	m 3	m 4	m 1	m 2	m 3	m 4	m 1	m 2	m 3	m 4	m 1	m 2	m 3	m 4
1	Backlog																
2	Todo																
3	Doing																
4	Testing/Review																
5	Refinement																
6	Done																