

Pengembangan Backend REST API Model AppUser dan Portal Metamesta BNITopia Dengan Java SpringBoot di PT Bank Negara Indonesia (Persero) Tbk.

1st Raynaldi Faizal Biqri
Prodi SI Terapan Teknologi Rekayasa
Multimedia
Fakultas Ilmu Terapan
Telkom University
Bandung, Indonesia
hellobiqri@student.telkomuniversity.ac
.id

2nd Dr. Ismail, S.Si., M.T.
Prodi SI Terapan Teknologi Rekayasa
Multimedia
Fakultas Ilmu Terapan
Telkom University
Bandung, Indonesia
ismailrusli@telkomuniversity.ac.id

Dr. Duddy Soegiarto, S.T., M.T.
Prodi SI Terapan Teknologi Rekayasa
Multimedia
Fakultas Ilmu Terapan
Telkom University
Bandung, Indonesia
duddysu@telkomuniversity.ac.id

Abstrak — Pada laporan Proyek Akhir ini Penulis mendapatkan tugas menjadi backend developer dalam pengembangan REST API dimana saat ini BNI sedang melakukan pengembangan teknologi kearah Metaverse. Dalam pengembangannya, BNI memerlukan REST API Model AppUser dan Model Portal pada proyek BNITopia. REST API ini nantinya akan digunakan pada BNITopia dalam fitur dan layanan, dimana penulis berfokus pada sisi model AppUser dan Portal. Menurut Masse, 2011 sebuah API mengungkapkan sejumlah data dan fungsi untuk memfasilitasi interaksi antara program komputer dan memungkinkan mereka saling bertukar informasi [1]. Penulis menggunakan metodologi Agile saat pengembangan web servis. Terdapat total 10 endpoint yang dihasilkan dari pengembangan REST API ini dimana disetiap model menghasilkan 5 endpoint.

Kata kunci— REST API, Backend, BNITopia

I. PENDAHULUAN

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.

II. KAJIAN TEORI

A. Profil Perusahaan

PT. Bank Negara Indonesia Tbk (BNI) adalah sebuah bank komersial yang berpusat di Jakarta, Indonesia. BNI didirikan pada tanggal 5 Juli 1946 dan menjadi bank pertama yang didirikan dan dimiliki oleh pemerintah Indonesia. BNI awalnya didirikan sebagai Bank Negara Indonesia Serikat dan kemudian berganti nama menjadi Bank Negara Indonesia pada tahun 1949 [2].

BNI menawarkan berbagai produk dan layanan perbankan seperti tabungan, deposito, pinjaman, kartu kredit, transfer, dan lain sebagainya. BNI juga memiliki layanan perbankan online dan mobile banking untuk memudahkan nasabah dalam melakukan transaksi perbankan. Sesuai visi bank BNI Menjadi Lembaga Keuangan yang Terunggul dalam Layanan dan Kinerja secara berkelanjutan. Untuk mencapai misi visi [3] tersebut BNI menetapkan misinya yang terdiri dari:

- 1) Memberikan layanan prima dan solusi digital kepada seluruh nasabah selaku mitra bisnis.
- 2) Memperkuat layanan internasional untuk mendukung kebutuhan mitra bisnis global.
- 3) Meningkatkan nilai investasi yang unggul bagi investor.
- 4) Menciptakan kondisi terbaik bagi karyawan sebagai tempat kebanggaan untuk berkarya dan berprestasi.

- 5) Meningkatkan kepedulian dan tanggung jawab kepada lingkungan dan masyarakat.
- 6) Menjadi acuan pelaksanaan kepatuhan dan tata kelola perusahaan yang baik bagi industri.

B. Landasan Teori

REST API merupakan suatu cara bagi aplikasi untuk berkomunikasi dengan server. Ketika sebuah aplikasi mengirim permintaan data (*request*) ke server dan server memberikan sebuah (*response*) hasilnya kembali ke aplikasi disinilah *REST API* bekerja. *REST API* menggunakan protokol HTTP untuk berkomunikasi dimana setiap permintaan terdapat metode membaca (GET), membuat (POST), memperbarui (PUT), atau menghapus (DELETE).

Menurut Roy Fielding terdapat 6 kategori dalam web arsitektur [4] diantaranya:

- 1) *Client Server*, entitas terpisah yang berkomunikasi melalui jaringan. Klien mengirim permintaan ke server dan server memproses permintaan tersebut dan mengirimkan respons.
- 2) *Uniform Sentence REST*, menggunakan kumpulan metode yang terdefinisi dengan baik dan terstandarisasi untuk berinteraksi dengan sumber daya. Metode ini meliputi GET, POST, PUT, DELETE, dan lain-lain. Antarmuka ini konsisten di seluruh sumber daya yang berbeda dan memungkinkan klien memahami dan berinteraksi dengan server.
- 3) *Layered System REST*, mendukung arsitektur berlapis, di mana beberapa lapisan perantara (seperti penyeimbang beban, penyimpanan sementara, atau gateway) dapat ada antara klien dan server. Setiap lapisan menyediakan fungsionalitas khusus tanpa mempengaruhi perilaku keseluruhan sistem.
- 4) *Cache REST*, memungkinkan penyimpanan sementara respons di sisi klien atau server perantara. Penyimpanan sementara meningkatkan kinerja dan mengurangi beban pada server dengan melayani permintaan yang berulang dari penyimpanan sementara daripada memprosesnya ulang.
- 5) *Stateless Server*, tidak mempertahankan status klien apa pun. Sebaliknya, klien menyertakan semua informasi yang diperlukan dalam setiap permintaan agar server dapat memprosesnya. Ini memungkinkan skalabilitas dan menyederhanakan desain server.
- 6) *Resource-Based*, sumber daya merupakan konsep kunci dalam REST. Setiap sumber daya memiliki identifikasi unik (URI), dan klien berinteraksi dengan sumber daya ini melalui antarmuka yang seragam. Sumber daya dapat mewakili data atau fungsionalitas apa pun yang diekspos oleh server.

III. METODE

Model AppUser merupakan sebuah kelas entitas yang nantinya menjadi sebuah tabel. Tabel ini nantinya digunakan untuk menampung data pengguna. Pada Model Portal juga nantinya digunakan untuk menampung data koordinat portal.

A. Fungsional Model AppUser

Pada fungsional Model AppUser terdapat beberapa fungsi yang menyangkut tentang pengguna kedalam sistem. Dengan mempunyai pengguna berinteraksi dengan sistem maka ada beberapa kebutuhan fungsional yang dibutuhkan diantaranya:

- 1) Fungsi Sign In pengguna untuk dapat masuk kedalam sistem.
- 2) Fungsi Sign Up pengguna untuk dapat melakukan registrasi data pengguna.
- 3) Fungsi Get All User dimana sistem mampu mengambil seluruh data pengguna.
- 4) Fungsi Update Profile dimana pengguna mampu memperbaharui data diri.
- 5) Fungsi Delete User dimana sistem mampu menghapus data pengguna apabila dibutuhkan.

B. Fungsional Model Portal

Pada fungsional Model Portal terdapat beberapa fungsi yang menyangkut tentang koordinat portal kedalam sistem. Pada aplikasi Unity nantinya portal akan menghasilkan informasi koordinat yang digunakan untuk menyimpan portal. Berikut kebutuhan fungsional diantaranya:

- 1) Fungsi Get All Portal digunakan untuk mengambil semua data portal.
- 2) Fungsi Create Portal digunakan untuk membuat data portal.
- 3) Fungsi Update Portal digunakan untuk memperbaharui data portal.
- 4) Fungsi Find By Id digunakan untuk mencari portal berdasarkan id.
- 5) Fungsi Delete Portal digunakan untuk menghapus portal.

C. Endpoint

Dengan begitu pada Model AppUser terdapat 5 endpoint yang bisa dirancang untuk memenuhi kebutuhan pengguna diantaranya sebagai berikut:

- 1) Sign In, pada proses Sign In dengan *endpoint* /api/auth/signin terdapat proses pengecekan pada 2 attribute yaitu *username* dan *password* pengguna.
- 2) Sign Up, pada proses Sign Up dengan *endpoint* api/auth/signup terdapat proses pembuatan data pengguna baru.
- 3) Get All User, pada proses Get All User dengan *endpoint* /api/v1/user/ digunakan untuk mengambil seluruh data pengguna.
- 4) Update Profile User, pada proses Update Profile User dengan *endpoint* /api/v1/user/update/{id} dibutuhkan parameter id pengguna.
- 5) Delete User, pada proses Delete User dengan *endpoint* /api/v1/user/delete/{id} dengan id sebagai parameter dari id pengguna.

Dengan begitu pada Model Portal terdapat 5 *endpoint* yang bisa dirancang untuk memenuhi kebutuhan pengguna diantaranya sebagai berikut:

- 1) Get All Portal, pada proses Get All Portal dengan endpoint `/api/v1/portal/` digunakan untuk mengambil semua data portal.
- 2) Create Portal, pada proses Create Portal dengan endpoint `/api/v1/portal/create` digunakan oleh pengguna untuk membuat portal.
- 3) Update Portal, pada proses Update Portal dengan endpoint `/api/v1/portal/update` terdapat id portal pada body request yang diperlukan untuk proses identifikasi id portal.
- 4) Find By Id, pada proses Find By User dengan endpoint `/api/v1/portal/{id}` digunakan untuk mencari portal yang berdasarkan id.
- 5) Delete Portal, pada proses Delete Portal dengan endpoint `/api/v1/portal/{id}` dibutuhkan id portal sebagai parameter data portal.

D. MRSC

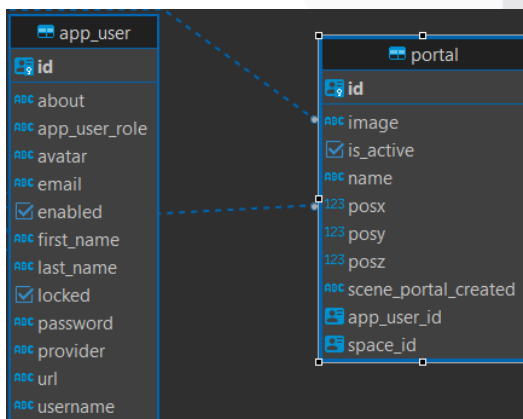
Pada proses pengerjaan *REST API* dibagi menjadi empat bagian. Setiap tahapan bagian pada Gambar 1 MRSC(A) saling berkesinambungan satu dengan yang lainnya. *Source code* di implementasikan melalui patern tahapan yaitu *model*, *repository*, *service*, dan *controller* [5]. Disetiap tahapan tersebut merepresentasikan fungsi yang berbeda. Hasil pelaksanaan pengerjaan nantinya berupa web servis *REST API* yang akan digunakan oleh Tim Metaverse pada aplikasi Unity.



GAMBAR 1
(MRSC)

IV. HASIL DAN PEMBAHASAN

Setelah dilakukan perancangan pada Model AppUser dan Portal dihasilkan 2 buah tabel yang mempunyai relasi one to many, dimana satu pengguna mampu memiliki banyak portal yang bisa dilihat pada Gambar 2(ER Diagram) berikut.



GAMBAR 2
(ER Diagram)

Proses coding disesuaikan dengan tahapan pattern MRSC seperti setiap endpoint terdapat pada tahapan Controller, setiap logika bisnis terdapat pada tahapan Service dan juga setiap class terdapat pada tahapan Model.

Pengujian *API* dilakukan pada aplikasi Postman untuk dapat mengetahui keberhasilan *API* yang dibuat. Terdiri dari *method*, *URL*, *body* dan *response* dimana masing-masing bagian. *Method* berperan sebagai pengirim tipe permintaan yang akan dikirimkan oleh aplikasi klien ke server. *URL* merupakan alamat unik yang digunakan untuk mengidentifikasi lokasi dan sumber daya yang ingin diakses di internet atau dalam *API*. *Body* berisi data yang akan dikirim ke server, terutama digunakan pada metode *POST* atau *PUT* untuk mengirim data ke server. *Response* biasanya berisi status tanggapan terhadap permintaan yang dikirimkan melalui Postman atau aplikasi klien lainnya.

V. KESIMPULAN

Dengan total 10 endpoint yang ada telah berhasil dijalankan. Dimana hasil pengujian endpoint dilakukan secara lokal dengan waktu response time yang bervariasi. Setelah itu Tim Metaverse mampu mengaplikasikan fitur pengguna dan juga portal dari endpoint yang ada.

REFERENSI

- [1] M. Masse, REST API Design Rulebook, Sebastopol: O'Reilly Media, 2011.
- [2] PT. Bank Negara Indonesia Tbk, "BNI," Bank Negara Indonesia, 2021. [Online]. Available: <https://www.bni.co.id/id-id/perseroan/tentang-bni/sejarah>. [Accessed 4 Juni 2023].
- [3] PT. Bank Negara Indonesia Tbk, "BNI," Bank Negara Indonesia, [Online]. Available: <https://www.bni.co.id/id-id/perseroan/tentang-bni/visi-misi>. [Accessed 15 Agustus 2023].
- [4] R. T. Fielding, "Architectural Styles and," University of California, Irvine, 2000.
- [5] T. Collings, "Controller-Service-Repository," 11 Agustus 2021. [Online]. Available: <https://tom-collings.medium.com/controller-service-repository-16e29a4684e5>. [Accessed 15 Agustus 2023].