INTEGRASI TRANSAKSI PEMBELIAN PAKET DAN SALDO PADA PERUSAHAAN XYZ MELALUI PERANCANGAN API

Prambudhi Wibowo Pandji[1*], Mira Musrini Barmawi[1]
[1]Institut Teknologi Nasional Bandung
Email: prambudhipandji@gmail.com
Received 06 09 2023 | Revised 13 09 2023 | Accepted 13 09 2023

ABSTRAK

Pada era digitalisasi, komunikasi sekarang dilakukan melalui media internet yaitu situs web yang menggunakan protokol HTTP. Terdapat sebuah protokol yang membantu komunikasi antar aplikasi yaitu API yang dapat mengakses fungsi atau data dari aplikasi, sistem operasi, dan layanan lainnya. Perusahaan XYZ merupakan perusahaan yang bergerak di bidang perbankan dan memiliki aplikasi mobile banking untuk mendukung aktivitas transaksi nasabah. Perusahaan tersebut melakukan integrasi bisnis dengan pihak ketiga untuk mendukung proses bisnisnya dimana terdapat perbedaan ruang lingkup dari perusahaan dan pihak ketiga namun dapat diintegrasikan melalui integrasi sistem informasi karena menyatukan sistem informasi dan membantu dalam bertukar informasi. Integrasi dibantu oleh framework Spring untuk perancangan REST API dimana tahapan yang dilakukan terdiri dari analisis, perancangan desain komunikasi, dan perancangan API. Setelah itu pengujian dilakukan untuk melihat integrasi menggunakan API yang menghasilkan integrasi dapat tercapai karena protokol HTTP dari pihak ketiga memberikan respon dari permintaan API yaitu kode “0000” dengan keterangan “SUKSES”.

Kata kunci: Integrasi Sistem Informasi, API, REST API, Spring Framework.

ABSTRACT

In the era of digitalization, communication is now carried out through internet media, namely websites that use the HTTP protocol. There is a protocol that helps communication between applications, namely APIs that can access functions or data from applications, operating systems, and other services. XYZ Company is a company engaged in banking and has a mobile banking application to support customer transaction activities. The company conducts business integration with third parties to support its business processes where there are differences in the scope of the company and third parties but can be integrated through information system integration because it unites information systems and helps exchange information. Integration is assisted by the Spring framework for REST API design where the stages carried out consist of analysis, communication design design, and API design. After that, testing was carried out to see the integration using the API which resulted in integration being achieved because the HTTP protocol from the third party responded to the API request, namely the code "0000" with the caption "SUCCESS".

Keywords: Information System Integration, API, REST API, Spring Framework.

1. PENDAHULUAN

Pada era digitalisasi, komunikasi sekarang dilakukan melalui media internet dengan menggunakan web. Web merupakan salah satu media pada internet yang menggunakan suatu protokol saat mengakses yaitu HTTP (Hypertext Transfer Protocol). Protokol tersebut digunakan untuk media komunikasi berupa request dan response antara client dan server **(Zabar & Novianto, 2015)**.

Terdapat sebuah protokol yang membantu komunikasi antar aplikasi yaitu API (Application Programming Interface). Dengan API, perancang aplikasi dapat mengakses fungsi atau data dari aplikasi, sistem operasi, dan layanan lainnya **(Miftahul Huda, 2019)**. REST (Representational State Transfer) API merupakan salah satu arsitektur API yang digunakan untuk melakukan komunikasi antar aplikasi. Berbeda dengan arsitektur sebelumnya yaitu SOAP (Simple Object Access Protocol) yang menggunakan transfer data melalui XML (Extensible Markup Language) dengan format yang lebih padat dan lebih besar karena menggunakan kalimat atau kode yang banyak sehingga transfer data membutuhkan waktu yang lama, REST menggunakan format JSON (JavaScript Object Notation) dengan penulisan yang ringkas sehingga menghasilkan format yang lebih kecil dan mudah untuk didistribusikan karena cepat.

Perusahaan XYZ merupakan perusahaan yang bergerak di bidang perbankan dan memiliki aplikasi mobile banking untuk mendukung aktivitas transaksi nasabah. Pada perusahaan tersebut ditemukan bahwa perusahaan ingin melakukan integrasi bisnis dengan pihak ketiga untuk mendukung proses bisnisnya yaitu nasabah dapat membeli paket dan menambahkan denom atau denominasi dari akun pada layanan yang diberikan oleh pihak ketiga melalui aplikasi mobile banking.

Perbedaan ruang lingkup dari perusahaan dan pihak ketiga dapat diintegrasikan melalui integrasi sistem informasi karena menyatukan sistem informasi dan membantu dalam bertukar informasi dan meningkatkan proses secara real-time juga membantu organisasi untuk berjalan lebih efesien sehingga memberi keunggulan yang kompetetif **(Juha & Leena, 2010).** Integrasi tersebut dibantu oleh framework Spring yang membantu perancangan REST API sehingga komunikasi dari aplikasi mobile banking Perusahaan XYZ dapat mencapai layanan pihak ketiga yang menyediakan melalui protokol HTTP.

Fokus dari paper adalah melakukan perancangan REST API yang membantu integrasi sistem informasi dengan integrasi transaksi pembelian paket dan saldo pada Perusahaan XYZ dengan pihak ketiga yang melakukan komunikasi untuk integrasi melalui protokol HTTP sehingga nasabah dapat melakukan transaksi seperti membeli paket dan menambahkan denom pada pihak ketiga melalui aplikasi mobile banking perusahaan XYZ.

2. KAJIAN PUSTAKA

2.1 API

API (Application Programming Interface) memberikan suatu intisari dari masalah dan menentukan bagaimana client harus berinteraksi dengan komponen perangkat lunak yang memberikan solusi terhadap masalah tersebut, komponen tersebut merupakan pustaka perangkat lunak yang biasanya dibagikan sehingga dapat digunakan pada beberapa aplikasi **(Reddy, 2011)**. Kemudian terdapat istilah API endpoint yang memiliki definisi menurut Zimmermann, et, al. Pada tahun 2021 dan pada tahun 2020 pada publikasi **(Singjai & Zdun, 2022)** yaitu endpoint merupakan sisi penyedia di ujing saluran komunikasi dan sebuah penentuan lokasi dari API endpoint sehingga dapat diakses oleh client API dan perubahan pada endpoint dapat mempengaruhi arsitektur dari API sisi client.

2.2 REST API

REST (Representational State Transfer) memungkinkan sistem permintaan dapat mengkases dan memodifikasi kalimat yang ditampilkan pada Web Service, kalimat tersebut dapat berupa JSON, URI, XML, dan HTTP **(Rulloh et al., 2017)**. REST API merupakan satu dari beberapa antarmuka dari web yang dapat mengakses data tanpa menggunakan user interface. Dengan

menggunakan REST API, sistem dapat berkomunikasi dengan mengirim dan menerima data secara sederhana. Titik akhir (endpoint) pada API berfungsi memetakan sumber daya pada basis data pada suatu aplikasi **(Rizki & Adil, 2018)**.

2.3 HTTP

HTTP (Hypertext Transfer Protocol) adalah sebuah protokol yang digunakan untuk media komunikasi berupa request dan response antara client dan server **(Zabar & Novianto, 2015)**. Client dapat berupa sistem, komputer, atau suatu proses yang melakukan request data atau layanan kepada server sedangkan server adalah sistem atau proses yang menyediakan data atau layanan dan memberikan response sesuai permintaan client. HTTP mengirimkan data dengan sebuah metode yang biasa digunakan seperti "GET", "POST", "PUT", dan "DELETE".

2.4 Framework Spring

Spring adalah sebuah kerangka kerja yang memiliki sifat open source dengan menggunakan Bahasa Java dengan infrastruktur komprehensif dalam membangun aplikasi dengan mudah dan cepat **(Sonatha, 2020)**. Kemudian framework ini memiliki sub-framework yang dikenal dengan nama Spring Boot yang membantu dalam pengembangan aplikasi berbasis java yang dapat berdiri dengan sendirinya dan dapat langsung dengan mudah untuk dijalankan karena hanya membutuhkan sedikit konfigurasi dari aplikasi Spring dan sudah terdapat built-in server.

2.5 JSON

JavaScript Object Notation (JSON) adalah format data semi struktur yang ringan dan popular dengan berbasis tipe data pemrograman bahasa JavaScript. Format ini menjadi format utama yang digunakan untuk bertukar data pada dunia internet pada beberapa tahun belakang **(Lv et al., 2018)**. Struktur data ini sudah ditetapkan pada beberapa bahasa pemrograman untuk bertukar data dan bekomunikasi antar sistem karena ringan, penulisan tidak berlebihan berbanding menggunakan format XML (Extensible Markup Language) yang membutuhkan paramater kemudian JSON juga memudahkan saat dibaca oleh pengembang maupun sistem **(Saputra & Nabil, 2021)**.

2.6 JWT

JSON Web Token (JWT) adalah suatu token yang memiliki tiga bagian terdiri dari header, payload, dan signature berupa string yang digunakan sebagai autentikasi dan pertukaran informasi **(M. B. Jones et al., 2015)**. Token memiliki 2 jenis yaitu Bearer Token dan Holder-of-Key dan berdasarkan tujuannya terdapat 2 skema yaitu Identity Token dan Access Token. Bearer Token adalah token yang dapat digunakan oleh siapa saja yang memilikinya tanpa menunjukkan kepunyaan dari key kripto tersebut berbeda dengan Holder-of-Key yang membutuhkan kepemilikan dari key kripto tersebut. Identity Token menyajikan hasil dari autentikasi yang dapat bertahan pada memori dan digunakan untuk meminta permintaan Access Token dengan jangka waktu yang pendek untuk mengakses layanan sedangkan Access Token adalah digunakan untuk mengakses layanan tertentu karena hanya digunakan sebagai akses suatu layanan **(Zheng & Jiang, 2014)**.

2.7 Oauth2

Open Authorization 2 adalah suatu protokol otorisasi untuk aplikasi dapat mengakses sumber daya dengan melalui protokol HTTP pada suatu aplikasi oleh pengguna **(Fatman, 2020)**. Protokol ini dirancang untuk mengatasi permasalahan autentikasi pada client-server **(M. Jones & Hardt, 2012)**. Komponen pada protokol ini terdiri dari Resource Owner yang berisikan sumber daya pada sistem yang dilindungi oleh pengguna dan dapat memberikan akses pada pihak ketiga, Client yaitu adalah sistem yang membutuhkan akses pada sumber daya yang dilindungi oleh Resource Owner dan membutuhkan Access Token yang sesuai, Auhorization Server adalah tempat dimana penerimaan permintaan dari Client berupa Access Token dan memberikan akses dengan sepengetahuan Resource Owner, dan terakhir Resource Owner yang berfungsi untuk memvalidasi Access Token dari Client dan menyerahkan sumber daya yang diminta.

2.8 ORM

Object Relational Mapping (ORM) adalah suatu konsep yang digunakan untuk memetakan model relasional ke model objek berupa basis data dapat menjadi kelas – kelas objek atau objek

– objek kelas berubah menjadi suatu model basis data **(Krisnanto & Kristianto, 2013)**. Konsep ini menjadi penghubung antara aplikasi berorientasi objek dengan basis data yang memiliki perbedaan **(Riyanto et al., 2018)**. Pada aplikasi java, media yang digunakan untuk pemetaan objek menjadi model relasional adalah dengan menggunakan Java Persistence API (JPA) dan bantuan dari Hibernate. Hibernate merupakan framework java yang digunakan sebagai alat untuk mendukung ORM berfungsi sebagai rmenyimpan objek menjadi model relasional.

3. METODOLOGI

Metodologi terdiri dari 3 tahapan yaitu Analisis yang terdiri dari 2 sub – proses yaitu Analisis Permasalahan dan Analisis Proses Bisnis, Perancangan Desain Komunikasi, dan Perancangan API. Tahapan tersebut dapat dilihat melalui Gambar 1 dan untuk penjelasan dari tiap tahapan dideskripsikan pada sub bab berikutnya.

Gambar 1. Metodoloogi Penelitian

3.1 Analisis

Tahapaan analisis terdiri dari 2 buah proses yaitu analisis permasalahan dan proses bisnis. Analisis permasalahan dilakukan untuk mengetahui layanan yang disediakan oleh pihak ketiga agar implementasi integrasi sistem informasi sesuai proses bisnis untuk mencapai integrasi transaksi pembelian paket dan saldo antar kedua belah pihak. Kemudian analisis proses bisnis dilakukan untuk mengetahui rangkaian kegiatan untuk mencapai layanan yang telah disediakan sehingga terdapat penyesuaian aktivitas guna menghasilkan layanan dari bisnis tersebut.

3.2 Perancangan Desain Komunikasi

Tahapan ini merancang desain komunikasi berdasarkan masalah yang ditemukan dan penyesuaian proses bisnis yang telah dilakukan pada tahapan sebelumnya melalui diagram sehingga memperlihatkan interaksi antar 2 sistem yang melakukan komunikasi yaitu sistem aplikasi perbankan dengan pihak ketiga pada sistem aplikasi perbankan perusahaan XYZ.

3.3 Perancangan API

Perancangan API dengan merancang REST API dengan framework Spring berdasarkan perancangan diagram desain komunikasi dari integrasi antar 2 sistem. Perancangan diawali dengan memetakan arsitektur dari API yang terdiri dari beberapa komponen untuk mendukung integrasi kemudian merancang dari function atau API sehingga membuat komunikasi berupa permintaan dan respon.

4. PERANCANGAN

4.1 Analisis Permasalahan

Perusahaan XYZ memiliki aplikasi perbankan yang menyediakan layanan bagi nasabah melakukan transaksi pada telepon genggam dimilikinya. Kemudian perusahaan ingin melakukan integrasi transaksi dengan salah satu mitra agar pengguna atau nasabah dapat melakukan transaksi penambahan saldo denom dan pembelian paket yang dinamakan sebagai Top Up Denom dan Beli Paket.

Top Up Denom merupakan layanan untuk mengisi jumlah saldo pada akun pelanggan yang digunakan untuk membeli sebuah paket sehingga pelanggan dapat berlangganan pada salah satu paket yang disediakan. Saldo yang ditambahkan merupakan denom atau sebuah nilai yang sebenarnya dari produk sehingga pelanggan jika ingin melakukan penambahan saldo, pelanggan membayarkan sejumlah nilai yang berbeda dari jumlah denom.

Selanjutnya Beli Paket merupakan layanan untuk membeli paket yang berisi layanan acara televisi dengan program yang bervariasi seperti paket bola, film, anak, dan paket lainnya. Setiap paket yang ada memiliki batasan waktu sehingga pelanggan harus memperbarui langganannya dengan membeli paket lainnya atau menambahkan saldonya untuk membeli kembali paket acara televisi yang sama.

Akun pelanggan merupakan akun pelanggan yang sudah berlangganan dengan layanan yang disediakan oleh pihak ketiga sebagai media transaksi untuk transaksi pembelian paket. Supaya pengguna aplikasi perbankan dapat menikmati pembelian paket dengan menggunakan akun pelanggannya dibutuhkan integrasi antar sistem satu sama lain sehingga aplikasi dapat melakukan transaksi pembelian paket dan penambahan saldo menggunakan akun khusus nasabah tersebut.

Integrasi antar sistem dibantu oleh API yang berfungsi sebagai jembatan penghubung antar sistem atau aplikasi yang memiliki ruang lingkup yang berbeda dengan cara kerja pertamanya adalah sistem atau aplikasi mengakses API kemudian API mengirimkan request yang merupakan sebuah permintaan yang sebelumnya ditentukan oleh pengguna pada aplikasi sehingga permintaan tersebut dapat diteruskan ke penerima yang merupakan server sehingga server dapat menemukan data yang sebelumnya diminta dan data tersebut dijadikan sebagai response yang disampaikan oleh API ke aplikasi.

4.2 Analisis Proses Bisnis

Proses bisnis adalah rangkaian aktivitas dari satu atau lebih input dan menciptakan suatu output yang memiliki nilai diinginkan oleh pelanggan **(Hammer & Champy, 1993)**. Pada integrasi transaksi antara perusahaan XYZ dengan pihak ketiga, transaksi yang dapat dilakukan adalah melakukan penambahan saldo denom dan pembelian paket. Pada proses bisnis ini merupakan representasi jika tahapan perancangan API sudah diimplementasikan ke dalam sebuah sistem oleh karena itu proses bisnis yang dipetakan merupakan hasil analisis pemetaan to-be yang memperlihatkan bagaimana API dapat membantu bisnis untuk melakukan penambahan saldo denom dan pembelian paket pada aplikasi perbankan perusahaan XYZ.

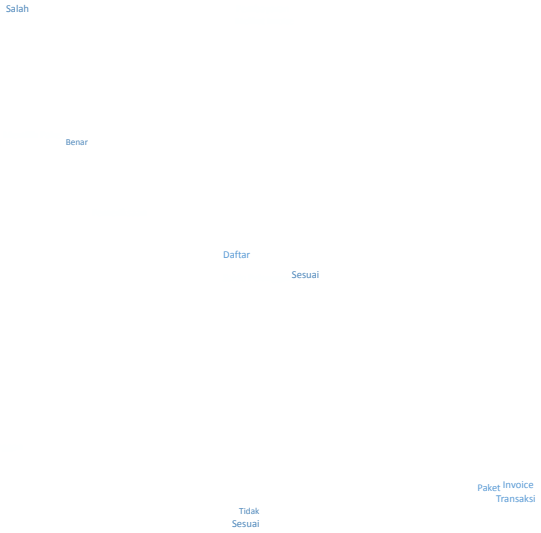
4.2.1 Proses Bisnis Penambahan Saldo Denom

Proses bisnis penambahan saldo denom adalah transaksi penambahan saldo pada akun pelanggan yang sudah berlangganan dengan layanan pihak ketiga. Jika ingin menambahkan saldo, pelanggan dapat melakukan beberapa tahapan yang diantaranya adalah memilih menu pembayaran tagihan, memasukan id pelanggan, memasukan jumlah denom, dan melakukan pembayaran. Jika id pelanggan yang dimasukkan salah maka pelanggan harus memasuki id pelanggan yang benar dan jika jumlah denom yang dimasukan tidak berkisar antara Rp. 50.000 dan Rp. 1.000.000 maka pelanggan harus memasukan nominal yang sesuai dengan kisaran atau jumlah minimal dan maksimal tersebut. Pelanggan dapat melihat detail informasi terkait akunnya berupa saldo yang dimiliki, nama pelanggan, id pelanggan, dan nomor STB (Set Top Box) dan setelah melakukan pembayaran pelanggan dapat melihat informasi dari transaksi yang sudah berhasil. Proses bisnis tersebut dapat dilihat melalui Gambar 2.

Gambar 2. Proses Bisnis Penambahan Saldo Denom

4.2.2 Proses Bisnis Pembelian Paket

Proses bisnis pembelian paket adalah transaksi untuk melakukan pembelian paket yang tersedia pada layanan pihak ketiga. Tahapan yang dilakukan untuk membeli sebuah paket diantaranya adalah memasukan id pelanggan, melihat daftar paket, memilih paket, melakukan pembayaran, dan melihat invoice yang merupakan daftar dengan pernyataan dari tagihan yang sudah dibayar. Jika id pelanggan tidak valid maka pelanggan harus memasukan id yang benar kemudian jika saldo akun pengguna tidak mencukupi dari harga paket yang ingin dibeli maka sistem akan menolak transaksi tersebut dan mengirimkan pemberitahuan berupa saldo tidak mencukupi terakhir pelanggan dapat melihat invoice jika telah melakukan pembayaran. Deskripsi pembelian paket dapat dilihat melalui pemetaan Gambar 3.



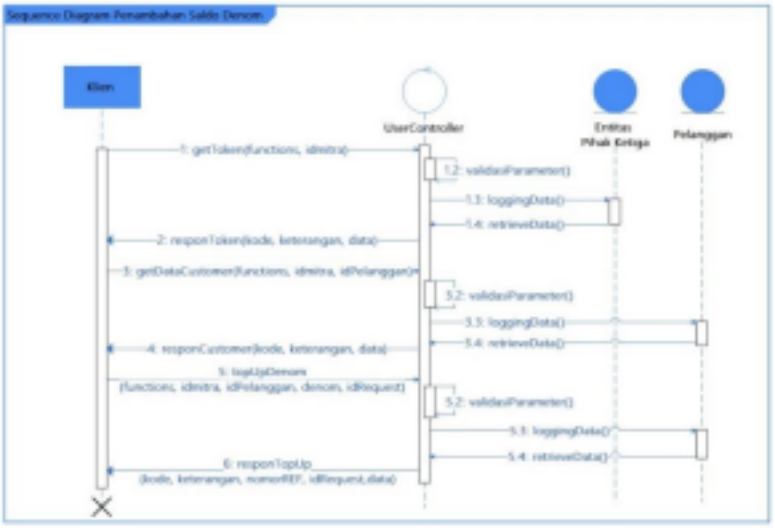
Gambar 3. Proses Bisnis Pembelian Paket

4.3 Perancangan Desain Komunikasi

Perancangan desain komunikasi dipetakan menggunakan Sequence Diagram dengan memperlihatkan interaksi antar objek dengan memperlihatkan pesan atau perintah antar objek saat berinteraksi dan memperlihatkan pengiriman dan penerimaan pesan dengan waktu yang disebut sebagai lifeline. Desain komunikasi memperlihatkan perusahaan XZY berinteraksi dengan pihak ketiga yaitu Klien melalui UserController pada sistem yang telah dikonfigurasi untuk berkomunikasi dengan protokol HTTP pihak ketiga.

Alur komunikasi untuk penambahan saldo denom adalah dengan melakukan permintaan/request token pada protokol HTTP pihak ketiga. Token disini merupakan sebuah kombinasi secara acak dari angka dan huruf yang terenkripsi yang digunakan sebagai akses untuk menggunakan fungsi lainnya seperti getDataCustomer(). Klien sebagai pelaku utama meminta hak akses dengan mengirimkan message menggunakan parameter functions dan idmitra sebagai pengenal yang nantinya akan divalidasi apakah parameter yang diterima merupakan valid. Jika valid maka data yang dikirimkan akan digunakan sebagai logging atau pengarsipan data berupa kejadian, error, dan status dari message ke entitas pihak ketiga setelah itu akan dikirimkan kembali ke pelaku utama sebagai respon atau jawaban dari controller yang sebagai mengatur arus informasi berupa kode HTTP, keterangan atau deskripsi dari kode HTTP yang diterima, dan data yang berupa akses token dan waktu valid atau batas waktu yang dipunya untuk akses dari token tersebut.

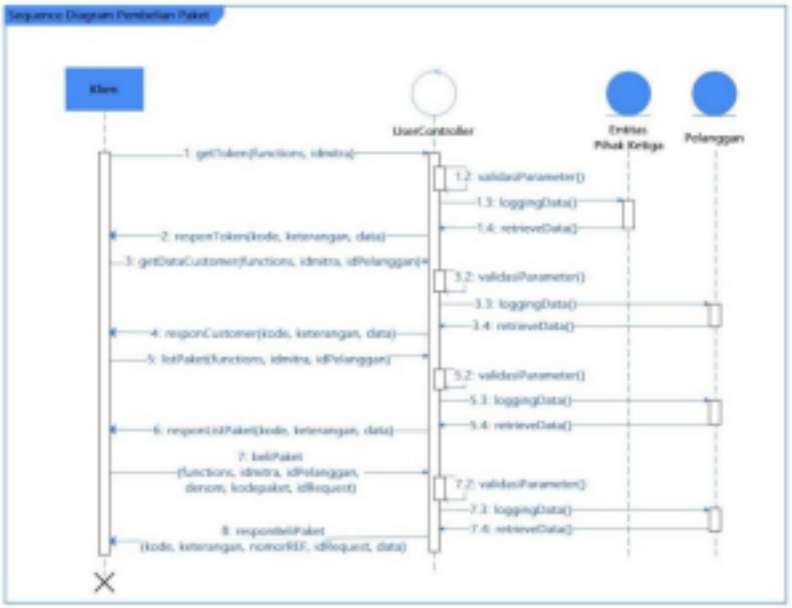
Setelah mendapatkan token akses dari pihak ketiga, klien dapat melakukan fungsi lainnya untuk melakukan penambahan saldo denom seperti getDataCustomer() dan topUpDenom(). Fungsi getDataCustomer() digunakan untuk meminta data pelanggan dari pihak ketiga sehingga saat melakukan penambahan saldo, saldo yang ditambahkan adalah saldo dari pemilik atau pelanggan dari layanan pihak ketiga dengan meminta permintaan (request) menggunakan parameter functions, idmitra, idPelanggan setelah itu controller melakukan validasi sama seperti sebelumnya dengan melihat apakah parameter sesuai dengan yang diminta sehingga controller dapat memberikan jawaban (response) yang sesuai dengan request. Data logging juga disimpan dengan entitas yang berbeda yaitu pelanggan kemudian controller memberikan response berupa kode HTTP, keterangan kode HTTP, dan data dengan rinciannya terdapat nomor Set Top Box (STB), nama pelanggan, id dari pelanggan, dan saldo yang dimiliki. Selanjutnya untuk fungsi topUpDenom() Klien mengirimkan parameter denom yang digunakan untuk memasukkan nominal saldo yang ingin ditambah dan idRequest yang digunakan untuk pemberian identitas yang unik setiap request penambahan saldo dan setiap detail atau data yang divalidasi dan dikirimkan disimpan pada entitas pelanggan. Deskripsi dari desain komunikasi penambahan saldo denom dapat dilihat pada Gambar 4.



Gambar 4. Desain Komunikasi Penambahan Saldo Denom

Pada transaksi Beli Paket, Klien membutuhkan token kembali karena token yang digunakan pada transaksi sebelumnya akan hangus atau sudah terpakai sehingga alur pertama untuk pembelian paket dengan menggunakan fungsi getToken() dan controller akan melakukan validasi dari parameter fungsi tersebut dan data dari fungsi yang dikirimkan akan disimpan sebagai logging di entitas pihak ketiga kemudian jika parameter valid, controller mengirimkan response berupa akses token kepada Klien.

Kemudian fungsi getDataCustomer() digunakan untuk melihat detail informasi dari akun pelanggan layanam pihak ketiga sehingga sebelum melihat layanan daftar paket yang tersedia dengan menggunakan fungsi listPaket(), Klien dapat melihat sisa saldo yang dimiliki dan dapat memilih harga paket yang sesuai dengan saldo dimiliki. Untuk fungsi listPaket() data yang dikirimkan sebagai response adalah data terkait data pelanggan dan daftar paket yang terdiri dari harga, waktu atau batas penggunaan paket pada tanggal tertentu, id paket, nama paket, dan frekuensi atau berapa hari paket tersebut berlaku dalam hitungan hari. Setelah menggunakan fungsi tersebut, fungsi selanjutnya adalah beliPaket() yang menggunakan parameter kodepaket, denom, dan idRequest yang digunakan untuk memenuhi transaksi saat pembelian paket dan mencocokan saldo denom yang dipunyai oleh akun dengan harga paket yang dibeli. Deskripsi tersebut dipetakan melalui sequence diagram yang dapat dilihat pada Gambar 5.

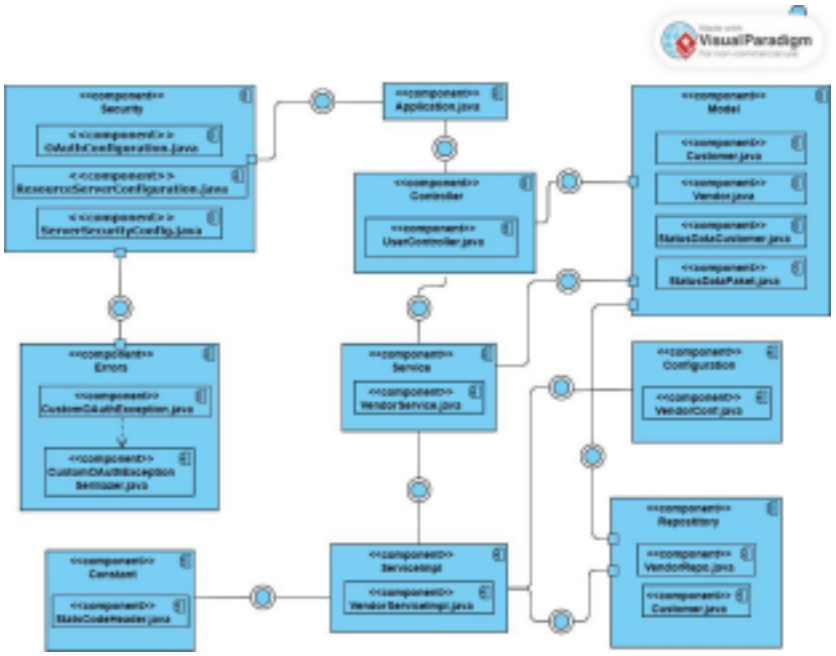


Gambar 5. Desain Komunikasi Pembelian Paket

4.4 Perancangan API

API yang dibangun menggunakan framework Spring terutama menggunakan Spring Boot. Framework ini dapat diinisiasikan secara otomasi menggunakan Spring Initializr yang sudah disediakan pada website Spring. Dalam project Spring Boot, terdapat beberapa arsitektur yang mendukung pembuatan REST API yang saling berhubungan satu sama lain sehingga terdapat komponen yang memiliki ketergantungan satu sama lain.

API yang dirancang menggunakan Spring Boot memiliki beberapa sumber daya yang memudahkan perancang untuk melakukan konfigurasi baik dari sisi client, server, dan security dengan menggunakan konfigurasi Resource Server, Server Security, dan Otentikasi yang dibantu oleh konfigurasi OAuth2. Dari konfigurasi tersebut terdapat beberapa komponen yang dirancang menjadi komponen utuh sehingga terbentuknya API yang digambarkan menggunakan component diagram seperti pada Gambar 6.



Gambar 6. Arsitektur API

5. IMPLEMENTASI

Implementasi API adalah dengan menentukan API sebagai media integrasi antar Klien dengan pihak ketiga berdasarkan perancangan desain komunikasi dan API dengan arsitekturnya kemudian API akan diuji untuk melihat apakah API berhasil melakukan integrasi dengan pihak ketiga.

5.1 API Get Token

Integrasi transaksi antara Klien dengan pihak ketiga diawali dengan menggunakan token yang disediakan oleh pihak ketiga. Dengan token tersebut, Klien dapat melanjutkan transaksi seperti melakukan penambahan saldo denom dan pembelian paket. Untuk permintaan akses token dari pihak ketiga, API akan menerima HTTP request dengan header dan data raw pada body dengan data tertentu. Kemudian setelah request diterima, maka response yang diterima akan dibuat objek yang akan menjadi response dari request tersebut.

5.2 API Get Data Customer

API setelah mendapatkan akses token dari pihak ketiga, Klien dapat mengakses API yang disediakan oleh pihak ketiga salah satunya yaitu mengambil data pelanggan. Data tersebut dapat terlihat jika pengguna memasukkan id pelanggannya sehingga pelanggan dapat melihat id, nama, nomor STB (Set Top Box), dan saldo yang dimiliki oleh akun pelanggan tersebut. Data ini digunakan untuk memvalidasi dari akun pelanggan layanan pihak ketiga
yang sudah berlangganan agar setiap melakukan penambahan saldo atau pembelian paket maka akun yang dituju adalah akun yang diambil datanya atau mendapatkan akses data tersebut.

5.3 API Top Up Denom

API Top Up Denom dibentuk supaya pelanggan layanan dari pihak ketiga dapat melakukan penambahan saldo denom. Fungsi ini dapat menambahkan jumlah saldo sesuai dengan ketentuan yang telah ditetapkan yaitu berkisar Rp. 50.000 sampai dengan Rp. 1.000.000. Jika tidak memenuhi minimal dan maksimal dari saldo yang ingin ditambah maka request akan ditolak dan akan mengembalikan response gagal karena jumlah denom tidak memenuhi ketentuan yang ada. Jika transaksi denom berhasil maka API akan memproses id request terlebih dahulu agar setiap transaksi hanya dapat terjadi satu kali dalam satu saat jika id request terdapat kesamaan maka API akan merespon dengan gagal akses API karena terdapat duplikasi pada id request. Jika transaksi berhasil maka API akan mengembalikan nomor referensi yang menandakan bahwa

transaksi tersebut berhasil dilakukan.

5.4 API Daftar Paket

API ini berfungsi untuk memperlihatkan semua daftar paket yang tersedia pada layanan pihak ketiga. Paket tersebut berupa daftar sehingga dapat dipergunakan untuk diperlihatkan pada antarmuka saat pengguna ingin memilih paket yang akan dibeli. Sebelum memperlihatkan daftar paket tersebut, pengguna harus memasukkan nomor STB agar sistem melakukan verifikasi akun pengguna jika valid maka sistem akan memperlihatkan daftar paket tersebut. Daftar tersebut berisikan id paket, nama paket, lama paket, harga paket, dan masa paket berlaku.

5.5 API Beli Paket

API ini digunakan setelah pengguna mengakses API yang memperlihatkan daftar paket sehingga pengguna dapat melakukan transaksi dengan memilih paket yang tersedia. Dengan API ini, pengguna dapat melanjutkan pembayaran dengan syarat bahwa saldo akun layanan pihak ketiganya memenuhi nominal dengan paket yang akan dibeli jika tidak memenuhi maka pengguna harus melakukan penambahan saldo denom terlebih dahulu jika mencukupi maka API akan memberikan response berupa masa aktif paket, nomor referensi yang menandakan transaksi berhasil, dan status transaksi.

5.6 Pengujian Integrasi API

Pengujian dilakuan setiap API yang telah dibuat melalui perancangan dan implementasi dengan tujuan supaya setiap API dapat melakukan integrasi sistem informasi antar 2 sistem sehingga transaksi dapat tercapai. Indikator pengujian berhasil adalah saat API mengembalikan nilai "kode" dengan status "0000" dan "keterangan" dengan status "SUKSES". Pengujian dengan mengirimkan parameter sesuai yang diterima oleh protokol pihak ketiga agar mengirimkan respon dari permintaan yang menggunakan parameter tersebut. Hasil pengujian dengan parameter yang dikirimkan dapat dilihat melalui Tabel 1.

Tabel 1. Pengujian API dengan Respon dari Pihak Ketiga

Nama API	Parameter	Respon Pihak Ketiga
API Get Token	functions, idmitra.	**"**Kode**"**:"0000", **"**Keterangan**"**: "SUKSES".
API Get Data Customer	functions, idmitra, idPelanggan.	**"**Kode**"**:"0000", **"**Keterangan**"**: "SUKSES".
API Top Up Denom	functions, idmitra, idPelanggan, denom, idRequest.	**"**Kode**"**:"0000", **"**Keterangan**"**: "SUKSES".
API Daftar Paket	functions, idmitra, idPelanggan.	**"**Kode**"**:"0000", **"**Keterangan**"**: "SUKSES".
API Beli Paket	functions, idmitra, idPelanggan, denom, kodepaket, idRequest.	**"**Kode**"**:"0000", **"**Keterangan**"**: "SUKSES".

6. KESIMPULAN

Integrasi transaksi pada perusahaan XYZ yang melakukan integrasi sistem informasi dengan pihak ketiga melalui protokol HTTP dapat tercapai melalui pemetaan proses bisnis yang akan diintegrasikan kemudian merancang desain komunikasi dari API untuk mengakses protokol HTTP pihak ketiga kemudian melakukan perancangan REST API dengan framework Spring Boot yang membantu perancangan API sehingga saat API digunakan untuk integrasi menggunakan parameter yang dikenali oleh protokol pihak ketiga, protokol tersebut memberikan respon kode "0000" dengan keterangan "SUKSES" yang berarti API dikenali dan integrasi tercapai.

DAFTAR RUJUKAN

Fatman, Y. (2020). Implementasi Metode Open Authorization (OAUTH2) Untuk Pengelolaan Data Dosen di Universitas Islam Nusantara. Ainet : Jurnal Informatika, 2(1), 10–18. https://doi.org/10.26618/ainet.v2i1.3212

Hammer, M., & Champy, J. (1993). Reengineering the Corporation: A Manifesto for Business Revolution. Harper Business. https://books.google.co.id/books?id=VpYgWyc16twC

Jones, M. B., Bradley, J., & Sakimura, N. (2015). JSON Web Token (JWT). RFC, 7519, 1–30.

Jones, M., & Hardt, D. (2012). The OAuth 2.0 Authorization Framework: Bearer Token Usage. Ietf Rfc, RFC 6750(9), 1689–1699.

Juha, P., & Leena, P. (2010). Application Integration and Semantic Integration in Electronic Prescription Systems. International Journal of Computer Science Issues, 7.

Krisnanto, N. D., & Kristianto, B. (2013). Analisis Perbandingan Performa Object Relational Mapping dan Query SQL pada Sistem Informasi Pegawai MG Sport Musik dengan Framework CodeIgniter. 1–26.

Lv, T., Yan, P., & He, W. (2018). Survey on JSON Data Modelling. Journal of Physics: Conference Series, 1069(1). https://doi.org/10.1088/1742-6596/1069/1/012101

Miftahul Huda, M. K. (2019). Open Systems Interconnection: Lapisan Fisik. bisakimia. https://books.google.co.id/books?id=sSi0DwAAQBAJ

Reddy, M. (2011). API Design for C++. Elsevier Science. https://books.google.co.id/books?id=IY29LylT85wC

Riyanto, N. R. D., Nurfaizah, N., Bratakusuma, T., Utomo, F. S., & Hermanto, N. (2018). Implementasi Object Relational Mapping (ORM) Pada Aplikasi Point of Sale Berbasis Android. Citisee, November 2018.

Rizki, K., & Adil, A. (2018). Implementasi Google Maps API Berbasis Android untuk Lokasi Fasilitas Umum di Kabupaten Sumbawa. Jurnal MATRIK, 17(2), 34–44. https://doi.org/10.30812/matrik.v17i2.87

Rulloh, A., Mahmudah, D. E., & Kabetta, H. (2017). Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android. 1(2).

Saputra, M. H. K., & Nabil, L. M. (2021). Penerapan Arsitektur Microservice Pada Sistem Tata Kelola Matakuliah Proyek Politeknik Pos Indonesia. Teknik Informatika, 13(3), 22–28.

Singjai, A., & Zdun, U. (2022). Conformance assessment of Architectural Design Decisions on API endpoint designs derived from domain models. Journal of Systems and Software, 193, 111433. https://doi.org/10.1016/j.jss.2022.111433

Sonatha, Y. (2020). Pengembangan Sistem Informasi Taman Pendidikan Al Qur'an Menggunakan Framework Java Spring. JATISI (Jurnal Teknik Informatika Dan Sistem Informasi), 7(1), 81–90. https://doi.org/10.35957/jatisi.v7i1.184

Zabar, A. A., & Novianto, F. (2015). Keamanan Http Dan Https Berbasis Web Menggunakan Sistem Operasi Kali Linux. Komputa : Jurnal Ilmiah Komputer Dan Informatika, 4(2), 69–74. https://doi.org/10.34010/komputa.v4i2.2427

Zheng, K., & Jiang, W. (2014). A token authentication solution for hadoop based on kerberos pre-authentication. DSAA 2014 - Proceedings of the 2014 IEEE

International Conference on Data Science and Advanced Analytics, October 2014, 354–360. https://doi.org/10.1109/DSAA.2014.7058096