TEKNO Jurnal Teknologi Elektro dan Kejuruan

http://journal2.um.ac.id/index.php/tekno | ISSN 1693-8739 / 2686-4657

Penerapan *Clean Architecture* pada Pengembangan Sistem *Payment Point Online Bank*

Arif Widiasan Subagio[1], Faisal Muttaqin[2]

1. UPN "Veteran" Jawa Timur, Indonesia | 19081010065@student.upnjatim.ac.id
2. UPN "Veteran" Jawa Timur, Indonesia | faisalmuttaqin.if@upnjatim.ac.id

Abstrak

Seiring perkembangan dunia seluler selalu *up-to-date* dimana pemanfaatan teknologinya begitu pesat. Hal ini dapat dilihat dari berbagai macam aplikasi yang ditawarkan, diantaranya pendidikan, *entertainment*, *social media*, *game*, dan bisnis . Saat ini ada juga sebuah sistem yang makin banyak yaitu *Payment Point Online Bank*. Dalam pengembangan sistem *Payment Point Online Bank* ini banyak diterapkan pada aplikasi mobile untuk sisi pengguna, dan untuk sisi admin biasanya ada pada tampilan web atau desktop. Sehingga untuk pembangunan suatu sistem Payment Point Online Bank yang tepat adalah membuat sistem web service. Dan biasanya pengembangan suatu sistem dilakukan oleh lebih dari satu orang. Namun hal ini menjadi masalah karena dengan banyaknya orang yang mengerjakan sistem. Oleh karena itu dibutuhkan suatu aturan prinsip yang diterapkan dalam pengembangan sistem ini, sehingga seluruh orang yang mengembangkan sistem Payment Point Online Bank ini dapat menerapkan hal yang sama agar tidak menimbulkan kebingungan. Dalam penelitian ini akan dilakukan pengembangan sebuah sistem PPOB dengan menerapkan Clean Architecture untuk memudahkan pengembang melakukan kolaborasi antara satu orang dengan orang lain dan sistem tersebut memiliki kerangka yang konsisten sehingga mudah dibaca oleh orang lain. Pengembangan sistem dalam penelitian ini menggunakan Clean Architecture sebagai prinsip utama kerangka software. Untuk bahasa pemrograman yang digunakan adalah Bahasa Go yang dikembangkan oleh Google dengan framework Echo untuk pembuatan web service nya. Xendit digunakan sebagai third-party service untuk melakukan pembayaran dengan metode pembayaran seperti Virtual Account Bank, Pembayaran di Minimarket, dan Dompet Digital. Serta untuk otorisasi menggunakan JSON Web Token dan pengiriman e-mail menggunakan Simple Mail Transfer Protocol (SMTP) service melalui Outlook. Pada pengembangan sistem Payment Point Online Bank ini, setiap kolaborator yang terlibat merasa dimudahkan dalam pengerjaan karena sudah menerapkan clean architecture sebagai standar dasar pada pengembangan sistem PPOB ini.

Kata Kunci

Clean architecture, PPOB, Go, Xendit, API

http://journal2.um.ac.id/index.php/tekno | ISSN 1693-8739 / 2686-4657

1. Pendahuluan

Seiring perkembangan dunia seluler selalu u*p-to-date* dimana pemanfaatan teknologinya begitu pesat. Hal ini dapat dilihat dari berbagai macam aplikasi yang ditawarkan, diantaranya pendidikan, entertainment, social media, game, dan bisnis (Wahyudi dan Rasim,2018). Saat ini ada juga sebuah sistem yang makin banyak yaitu *Payment Point Online Bank*. *Payment Point Online Bank* (PPOB) adalah system pembayaran dalam jaringan (*online)* yang memanfaatkan fasilitas perbankan. Pembayaran yang dimaksud bisa bermacam – macam, mulai dari tagihan listrik, pulsa prabayar, token listrik, hingga saldo dompet digital. PPOB memberi kemudahan kepada pengguna untuk memenuhi kebutuhan aneka pembayaran secara cepat, tepat dan akurat (Yunistira dan Fudholi,2020).

Dalam pengembangan sistem *Payment Point Online Bank* ini banyak diterapkan pada aplikasi *mobile* untuk sisi pengguna, dan untuk sisi admin biasanya ada pada tampilan web atau *desktop*. Sehingga untuk pembangunan suatu sistem *Payment Point Online Bank* yang tepat adalah membuat sistem *web service (Rizqi dan Tolle,2020)*.

Web Service sendiri merupakan sebuah arsitektur perangkat lunak yang menggambarkan berbagai macam informasi atau proses yang dirancang untuk dapat dipanggil atau diakses oleh aplikasi lain. *Web Service* menyediakan *endpoint* yag berupa url untuk diakses melalui sebuah jaringan seprti internet dengan menggunakan format pertukaran data. Proses pertukaran data biasanya menggunakan XML atau JSON (Pratomo dan Kholid,2020). Sehingga sistem *Payment Point Online Bank* ini nantinya akan dijalankan di cloud agar dapat diakses dari sisi pengguna maupun admin dengan mengakses *endpoint* yang dibuat (Priambodo dan Triana,2019). Dalam pengembangan sistem *Payment Point Online Bank* dapat menggunakan berbagai macam Bahasa pemrograman seperti PHP, Python, Javascript, Go, dan masih banyak lagi (Amyra Sheldon,2021). Dan biasanya pengembangan suatu sistem dilakukan oleh lebih dari satu orang. Namun hal ini menjadi masalah karena dengan banyaknya orang yang mengerjakan sistem, ada pula model pengembangan yang berbeda – beda antara satu dengan yang lain. Oleh karena itu dibutuhkan suatu aturan prinsip yang diterapkan dalam pengembangan sistem ini, sehingga seluruh orang yang mengembangkan sistem *Payment Point Online Bank* ini dapat menerapkan hal yang sama agar tidak menimbulkan kebingungan. Untuk itulah diperlukan adanya sebuah *Software Architecture* yang memberi dasar pengembangan dan membuat kerangka pengerjaan *software* yang berkualitas.Banyak pendekatan yang dilakukan untuk membuat suatu *Software Architecture* yang baik, mudah dibaca, dan berkualitas. Untuk alasan inilah sebuah *Software Architecture* dibuat yaitu *Clean Architecture* (Peter Ivanics,2016). Dalam penelitian ini akan dilakukan pengembangan sebuah sistem *Payment Point Online Bank* dengan menerapkan *Clean Architecture* untuk memudahkan pengembang melakukan kolaborasi antara satu orang dengan orang lain dan sistem tersebut memiliki kerangka yang konsisten sehingga mudah dibaca oleh orang lain.

http://journal2.um.ac.id/index.php/tekno | ISSN 1693-8739 / 2686-4657

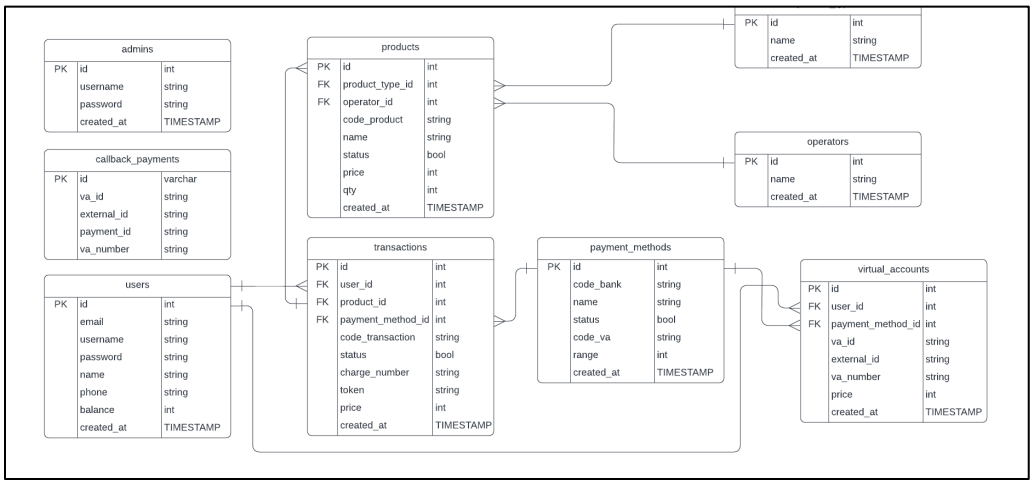
2. Metode

Pengembangan sistem dalam penelitian ini menggunakan *Clean Architecture* sebagai prinsip utama kerangka *software*. Untuk bahasa pemrograman yang digunakan adalah Bahasa Go yang dikembangkan oleh Google dengan framework Echo untuk pembuatan *web service* nya. Xendit digunakan sebagai *third-party service* untuk melakukan pembayaran dengan metode pembayaran seperti *Virtual Account Bank*, Pembayaran di Minimarket, dan Dompet Digital. Serta untuk otorisasi menggunakan *JSON Web Token* dan pengiriman *e-mail* menggunakan *Simple Mail Transfer Protocol* (SMTP) *service* melalui Outlook.

1) Entity Relationship Diagram

Pada tahap awal penelitian akan dibuat sebuah desain *Entity Relationship Diagram*. *Entity Relationship Diagram* atau ERD adalah sebuah diagram struktural yang digunakan untuk merancang sebuah database. Sebuah ERD mendeskripsikan data yang akan disimpan dalam sebuah sistem maupun batasannya. Komponen utama yang terdapat di dalam sebuah ERD adalah *entity set*, *relationship set*, dan juga *constraints* (Mohammed et al., 2015).

Gambar 1 merupakan desain ERD yang digunakan pada sistem *Payment Point Online Bank* yang dikembangkan dengan mempertimbangkan berbagai hal seperti kebutuhan pengguna, kebutuhan admin, dan kebutuhan dengan pihak ketiga untuk metode pembayaran. Database yang digunakan adalah MySQL.



Gambar 1. Entity Relationship Diagram

2) Bahasa Pemrograman Go

Golang (Go *Language)* adalah Bahasa pemrograman bersifat *open source* yang dikembangkan di Google oleh Rob Pie, Robert Griesemer, dan Ken Thompson. Golang mengandalkan kombinasi dari keamanan dan performa. Hingga sekarang Golang banyak digunakan oleh perusahaan besar maupun *startup* yang bergerak di bidang teknologi (Kristanto,Harjoseputro,Samodra. 2020). Golang memiliki banyak kelebihan seperti :

http://journal2.um.ac.id/index.php/tekno | ISSN 1693-8739 / 2686-4657

a.) Mendukung *concurrency* dalam sistem pemrograman dengan sangat baik dan mudah dalam pengaplikasian nya.
b.) *Open source* sehingga terbuka dalam pengembangannya.
c.) Sistem *garbage collection* yang baik sehingga tidak berat saat dijalankan.
d.) Penulisan sintaks yang bersih.
e.) *Reliable* dan cepat dalam pengembangan sistem berskala besar.

3) Echo

Echo merupakan *framework* web pada Bahasa Go yang memiliki performa tinggi, dapat dikembangkan, dan *minimalis* sehingga framework ini cocok digunakan pada sistem skala besar seperti *Payment Point Online Bank.* Keunggulan dari *framework* echo adalah :

a.) *Optimized router.*
b.) Support *HTTP/2.*
c.) Adanya *data rendering* .
d.) *Scaleable.*
e.) Dapat diberi *middleware.*
f.) Terdapat *Automatic TLS* untuk serve HTTPS.
g.) *Data binding* dan *error handling* dapat dikustomisasi.

4) JSON Web Token

JSON Web Token (JWT) merupakan sebuah token berbentuk string JSON yang sangat padat namun bentuknya kecil dan mengandung berbagai informasi yang dibutuhkan untuk melakukan autentifikasi. Token JWT ini dapat dikirim melalui URL, parameter HTTP *POST*, atau dalam *header* HTTP(Edy,Ferdiansyah,Pramusinto,Waluyo. 2019). Fungsi JWT pada sistem *Payment Point Online Bank* ini adalah untuk membedakan token antara pengguna satu dengan yang lain dan mencegah pengguna untuk dapat mengakses *endpoint* yang membutuhkan JWT milik admin.

5) Simple Mail Transfer Protocol

Simple Mail Transfer Protocol (SMTP) adalah mekanisme yang digunakan untuk mengirim *e-mail* antara *host* yang berbeda pada transmisi *internet protocol.* Proses nya adalah *client* SMTP membuka koneksi TCP ke server SMTP pada *remote host* lalu mencoba untuk mengirim *e-mail* pada seluruh koneksi. *Server* SMTP mendengarkan koneksi TCP pada *port* yang spesifik (*port* 25), dan *client* SMTP melakukan proses inisiasi koneksi dengan *port* tersebut (Vladimir V. Riabov, 2005).

E-mail service yang digunakan pada sistem *Payment Point Online Bank* ini adalah Outlook milik Microsoft. Alasan nya adalah SMTP Outlook pada Golang tidak memerlukan autentifikasi yang rumit seperti Gmail sehingga mudah untuk diimplementasikan.

http://journal2.um.ac.id/index.php/tekno | ISSN 1693-8739 / 2686-4657

6) Xendit

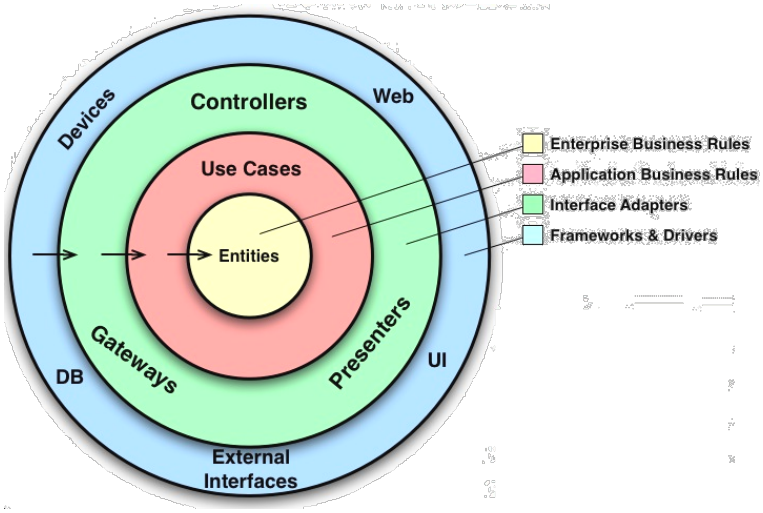
Xendit adalah sebuah perusahaan yang memberikan layanan *payment gateway* yang ada di Indonesia, Filipina, dan Asia Tenggara. Dengan melakukan satu integrasi, dapat melakukan pembayaran pada bisnis anda melalui kartu kredit, debit, transfer bank, *e-wallets,* dan masih banyak lagi (Xendit, 2022).

Keunggulan dari layanan Xendit sebagai pihak ketiga untuk metode pembayaran adalah melalui aplikasi Xendit saja dapat dilakukan konfigurasi untuk berbagai hal yang dilakukan untuk melakukan pembayaran. Lalu seluruh pembayaran yang masuk akan menjadi saldo Xendit sehingga hanya perlu melakukan proses *withdraw* ke bank untuk menerima pembayaran secara *real* (Endarwan dan Setiyadi, 2019).

7) Clean Architecture

Prinsip clean architecture dapat meningkatkan tingkat *maintainability* dari suatu sistem. Hal ini terjadi karena ada pemisahan antara komponen pada suatu sistem menjadi beberapa komponen kecil yang independen dan lebih modular sehingga jika terjadi sebuah bug pada sebuah komponen, pengembang dapat fokus memperbaiki *bug* pada komponen tersebut tanpa memengaruhi komponen lainnya. Penambahan sebuah fitur pada sistem yang menerapkan prinsip *clean architecture* juga sangat mudah dilakukan (Sondha, Sa'adah, Hardiansyah, Rasyid,2020).

Penerapan prinsip clean architecture dalam sebuah sistem dilakukan dengan memisahkan antar *layer* pada sistem tersebut. Sebuah sistem memiliki beberapa *layer* seperti yang digambarkan pada Gambar 2 berikut merupakan bentuk umum dari *clean architecture*. Dengan pembagian komponen tersebut, akan mudah bagi para kolaborator untuk melakukan pengembangan sistem.



Gambar 2. Bentuk umum *clean architecture*

http://journal2.um.ac.id/index.php/tekno | ISSN 1693-8739 / 2686-4657

Pembagian komponen yang umum adalah Entitas, *Use Cases* atau logika sistem, *Controller* dan *Handler*, dan UI atau web. Setiap komponen nantinya akan saling berhubungan dengan memberikan suatu *domain* atau *adapter* agar tiap komponen atau *layer* dapat berhubungan antar satu sama lain.

3. Hasil dan Pembahasan

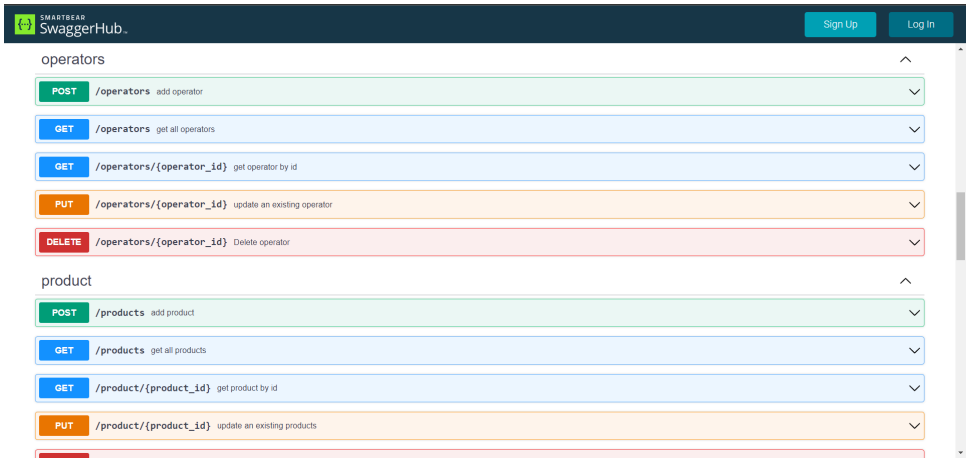
1) Aplikasi Sistem PPOB

Dalam aplikasi sistem *Payment Point Online Bank* ini terdapat 3 hal utama. Yang pertama adalah *Web Service* yang digunakan sebagai penyedia layanan API, tampilan admin berupa web yang dibuat menggunakan VueJS, dan tampilan pengguna berupa aplikasi *mobile* android yang dibuat menggunakan Flutter.

a) Web Service

Web Service untuk sistem PPOB ini menggunakan bahasa Go dengan *framework* Echo untuk pembuatan REST API dengan *endpoint* yang disediakan. Setiap *endpoint* memiliki fungsi dan *request* yang bermacam – macam tergantung fungsi apa yang dibutuhkan nantinya.

Proses *deployment* dari *web service* dilakukan di sebuah layanan penyedia *server* yaitu Amazon Web Service. Tujuan dari *deployment* ini agar sistem dapat berjalan sebagai *web service* secara utuh. Selain itu juga agar API benar – benar dapat digunakan cukup dengan memanggil url dari API yang sudah di-*deploy*. Pada gambar 3 berikut ditunjukkan dokumentasi dari API yang telah dibuat agar memudahkan kolaborator lain.

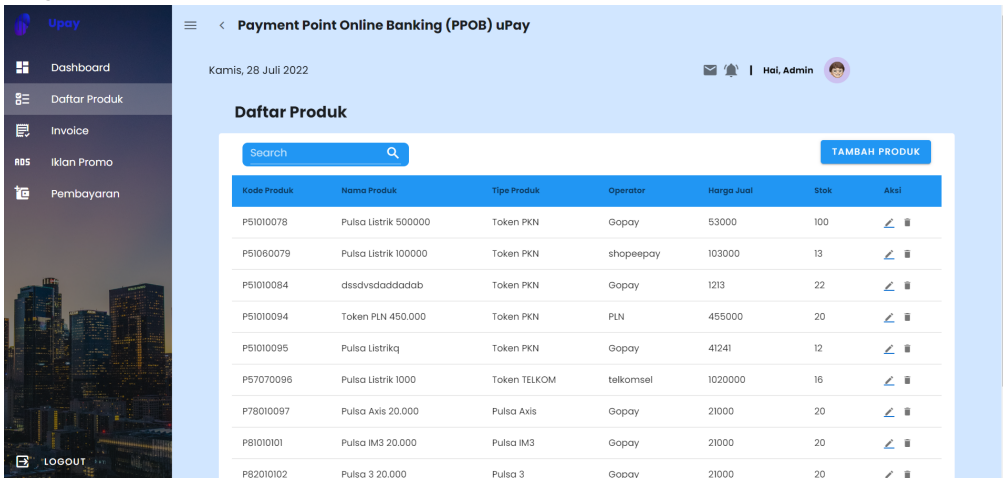


Gambar 3. Dokumentasi API

http://journal2.um.ac.id/index.php/tekno | ISSN 1693-8739 / 2686-4657

b) Sisi Admin

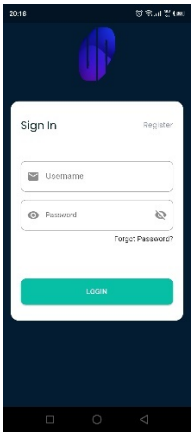
Sisi admin dari sistem PPOB ini berfungsi untuk melakukan berbagai aktifitas administrasi pada sistem PPOB seperti manajemen produk, melihat daftar transaksi, dan manajemen metode pembayaran. Seperti yang terlihat pada gambar 4, pembuatan tampilan sisi admin menggunakan VueJS. Selanjutnya API dipanggil untuk menampilkan atau melakukan berbagai fungsi.



Gambar 4. Tampilan Manajemen Produk

c) Sisi Pengguna

Sisi pengguna dari sistem PPOB ini ditujukan untuk pengguna dimana pengguna bisa melakukan berbagai hal seperti *login,* pendaftaran, melakukan transaksi, dan melihat riwayat transaksi. Seperti yang terlihat pada gambar 5, sisi pengguna dibuat menggunakan flutter lalu aplikasi dapat di-*install* pada perangkat android pengguna.

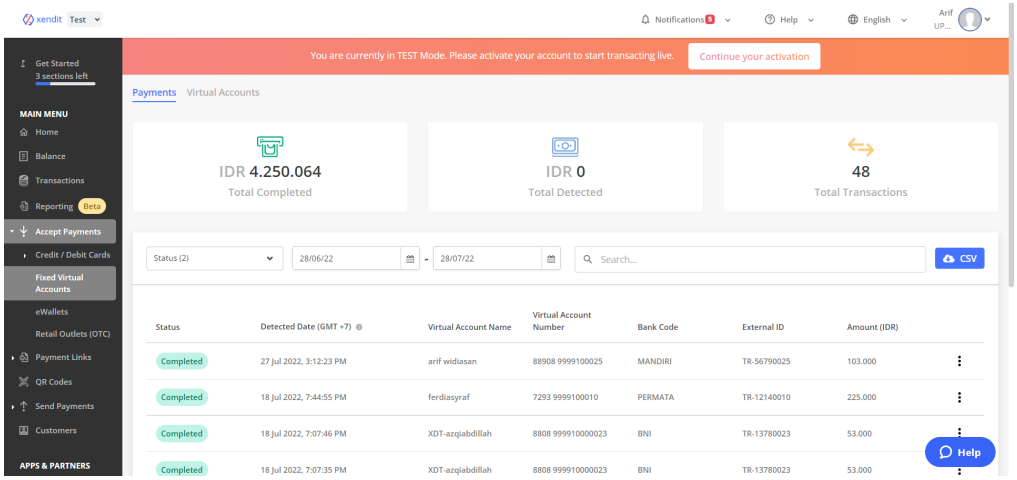


Gambar 5. Tampilan Sisi Pengguna

http://journal2.um.ac.id/index.php/tekno | ISSN 1693-8739 / 2686-4657

2) Integrasi dengan Xendit

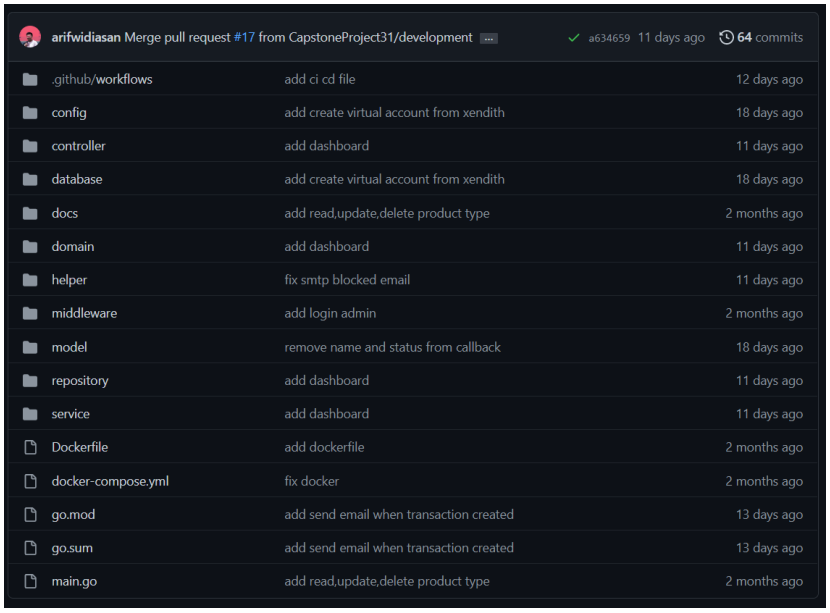
Pada *web service* perlu ditambahkan sebuah *secret key* dari akun xendit yang telah didaftarkan. Tujuan nya adalah agar sistem dapat terintegrasi dengan xendit sebagai *3rd-party* metode pembayaran. Pada gambar 6 ditunjukkan bahwa transaksi yang terjadi pada sistem PPOB akan terekam juga pada *dashboard* xendit.



Gambar 6. Transaksi Xendit

3) Clean Architecture

Pada *web service* sistem *Payment Point Online Bank* ini, seperti yang terlihat pada gambar 7 bahwa *clean architecture* diterapkan dalam pengembangan sistem ini.



Gambar 7. Kode Struktur *Web Service*

http://journal2.um.ac.id/index.php/tekno | ISSN 1693-8739 / 2686-4657

Dapat dilihat bahwa setiap komponen atau *layer* ditaruh pada folder yang berbeda. Folder *config* digunakan untuk konfigurasi dari sistem seperti nama *database*, JWT *secret key*, Xendit *secret key,* dan masih banyak lagi. Pada folder *controller* digunakan untuk handler dari setiap *endpoint* dan fungsi nya. Folder *domain* digunakan sebagai *adapter* atau penghubung dari seluruh *layer*. Model digunakan sebagai tempat pembuatan entitas. *Repository* berguna untuk melakukan perintah pada *database*. Yang terakhir *service* digunakan untuk tempat seluruh *use case* atau logika dari sistem PPOB.

4. Kesimpulan

Pada pengembangan sistem *Payment Point Online Bank* ini, setiap kolaborator yang terlibat merasa dimudahkan dalam pengerjaan karena sudah menerapkan *clean architecture* sebagai standar dasar pada pengembangan sistem PPOB ini. Namun ada beberapa kolaborator yang masih bingung dalam penerapan *clean architecture* ini karena masih belum mengerti konsep dasar dari arsitektur tersebut.

Daftar Rujukan

About xendit: Our mission is to make payments simple. Xendit. (2021, August 9). Retrieved July 28, 2022, from https://www.xendit.co/en/company/

Aflah Taqiu Sondha, Umi Sa'adah, Fadilah Fahrul Hardiansyah, & Maulidan Bagus Afridian Rasyid., 2020. Framework Dan Code Generator Pengembangan aplikasi Android Dengan Menerapkan prinsip clean architecture. Jurnal Nasional Teknik Elektro Dan Teknologi Informasi, 9(4), 327–335. https://doi.org/10.22146/jnteti.v9i4.572

Edy, E., Ferdiansyah, F., Pramusinto, W., & Waluyo, S. (2019). Pengamanan Restful API Menggunakan JWT UNTUK APLIKASI Sales order. Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), 3(2), 106–112. https://doi.org/10.29207/resti.v3i2.860

Endarwan, L., & Setiyadi, A., 2019. CASHLESS PAYMENT APPLICATION PONDOK PESANTREN DARUL FALAH CIHAMPELASCILILIN BASED ON ANDROID.

Harjoseputro, Y., Albertus Ari Kristanto, & Joseph Eric Samodra. (2020). Golang and NSG implementation in REST API based third-party sandbox system. Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), 4(4), 745–750. https://doi.org/10.29207/resti.v4i4.2218

High performance, minimalist go web framework. Echo. (n.d.). Retrieved July 28, 2022, from https://echo.labstack.com/

Ivanics, P., 2016. An Introduction to Clean Software Architecture. Department of Computer Science, University of Helsinki.

Mohammed, M. A., Muhammed, D. A. kareem, & Abdullah, J. M., 2015. Practical Approaches of Transforming ER Diagram into Tables. International Journal of Multidisciplinary and Scientific Emerging Research, 4(2), 1106–1110.

Pratomo, B. D., & Haryono, K., 2020. Perancangan RESTful Web Service Satuan Kredit Partisipasi di Universitas Islam Indonesia. SEMINAR NASIONAL Dinamika Informatika 2020 Universitas PGRI Yogyakarta, 74–77.

Priambodo, R., & Triana, Y. S., 2019. Payment point online bank (PPOB) mobile hybrid Dengan Koneksi Billers Berbasis HTTP Dan RESTful Services Untuk USAHA MIKRO. Jurnal Ilmiah FIFO, 11(1), 97. https://doi.org/10.22441/fifo.2019.v10i1.010

Rizqi, M., Tolle, H., & Hanggara, B., 2020. Pengembangan Modul Web Service Pada Situs PPOB Untuk Transaksi B2B (Studi Kasus PT XYZ). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 4, no. 10, p. 3373-3382, sep. 2020. ISSN 2548-964X.

Riabov, V. V., 2005. SMTP (Simple Mail Transfer Protocol).

Sheldon, A. (2021, July 28). Top 10 best web application development languages. Medium. Retrieved July 28, 2022, from https://becominghuman.ai/top-10-best-web-application-development-languages-8204aad91bc4

Wahyudi, A. M., & Rasim. APLIKASI PAYMENT POINT ONLINE BANK (PPOB) MENGGUNAKAN WEB SERVICE DI PT. SUKSES MITRA MANDIRI BERBASIS ANDROID.

Yunistira, A., & Fudholi, D. H., 2020. Analisis Penerapan Model Business Intelligence Pada APLIKASI payment point online banking Dalam Meningkatkan strategi pemasaran (Studi Kasus: Aplikasi Apotikkuota). Jurnal Ilmu Komputer Dan Agri-Informatika, 7(1), 1–10. https://doi.org/10.29244/jika.7.1.1-10