Penerapan Microservice Untuk Beberapa Fitur Pada Aplikasi CMS Berbasis Lokal

Ferly Jeremi Purnawan Apiang[#1], Risal[*2]

[#]*Program Studi Sistem Informasi, Universitas Kristen Maranatha*
Jl. Surya Sumantri No.65, Bandung, Indonesia
[1]1973005@maranatha.ac.id
[2]laurentius.risal@it.maranatha.edu

Abstract — **Some companies already have a CMS application within their company. CMS itself stands for Content Management System. CMS can be interpreted as an application to store any information either from the company or other information. The application of microservices for several CMS features involves creating a reporting feature and parameters. The application of the microservice concept to several CMS features uses the JavaScript programming language with the JS node framework. The application of microservices for features in CMS applications is enabled to streamline the performance of each existing employee and can help new developers to understand programming flows that are not too long. The application of this microservice is based on local disks where deploying is not carried out to share APIs with the public.**

Keywords— **Microservice, Content Management System, Node js**

I. PENDAHULUAN

Pada Saat ini teknologi beberbasi web berkembang dengan begitu pesat pada dunia teknologi. Teknologi berbasis web telah sepenuhnya mengubah ide tentang bekerja dengan berbagai informasi yang dimiliki oleh beberapa perusahaan. Kontribusi teknologi berbasis web sangat membantu perusahaan untuk menyimpan informasi-informasi yang dimilikinya. Teknologi yang biasa dimiliki oleh perusahaan untuk menyimpan sebuah informasi dari perusahaannya yaitu CMS atau *content management system.* CMS atau *content management system* adalah alat yang dapat digunakan oleh perusahaan untuk meengelola setiap informasi atau artikel yang dimilikinya. Teknologi dari CMS sendiri bisa dibangun dengan sistem monolitich atau biasa disebut sebagai seluruh code dijadikan kesatuan file besar. Akan tetapi penerapan monolitich membuat developer terutama developer baru akan merasa kebingungan terkait alur dari code yang dibuat. Sehingga pada saat ini banyak perusahaan menerapkan konsep microservice untuk melakukan pengembangan pada content management system yang dimiliki sebuah perusahaan [1].

Konsep microservice menyangkut desain arsitektur untuk membuat sebuah aplikasi yang terdiri dari berbagai unit layanan tersendiri tapi tetap terhubung antara satu dengan lainnya. Setiap unit layanan dalam aplikasi atau teknologi berbasis web menjalankan fungsi berbedam tapi tetap mendukung antar lainnya. Penerapan microservice tersendiri pada bagian ini untuk membuat sebuah service dari aggregator dan adapter. Aggregator sendiri adalah sebuah logic dari code yang akan mengakses setiap adapter yang ada. Serta adapter sendiri adalah sebuah code untuk mengakses setiap tabel database yang dimiliki dari setiap service yang memanggilnya. Bisa diartikan bahwa konsep dari microservice tersendiri adalah untuk membangun aplikasi di dalam aplikasi lainnya [2].

Penerapan dari microservice untuk pengembangan dari fitur pada aplikasi *content management system* (CMS) untuk membuat sebuah kinerja dari perusahaan dapat berjalan lebih cepat dan lebih fleksibel karena sifat dari microservice sendiri bersifat bebas dan developer dapat membuat setiap service sesuai dengan alur kerja dari developer tersebut serta dapat membuat dengan berbagai jenis bahasa yang berbeda. Penerapan konsep microservice ini menggunakan cara kerja agile serta konsep API gatewaye di mana setiap service akan saling menunggu satu dengan lainnya dari setiap urutan service yang dibuat [3].

Penerapan microservice ini akan membuat enam service yang terdiri dari service adapter reporting, service adapter parameter, service adapter txtreferral, service aggregator reporting, service aggregator parameter dan service aggregator txtreferral. Service-service tersebut sebelumnya merupakan code yang berbentuk monolitich di mana akan dilakukan perusahaan menjadi service-service berbentuk microservice. Akan tetapi service yang dibuat tidak sampai naik sampai

staging tetapi berbasis local di mana hanya bisa diakses oleh sebagian orang yang memiliki setiap source code yang diberikan.

A. Rumusan Masalah

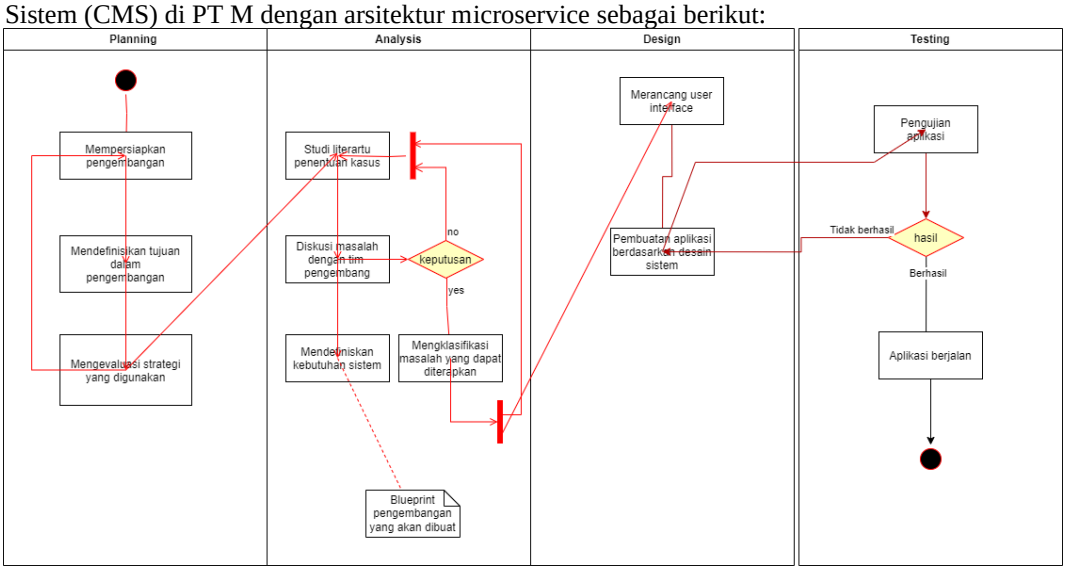
Berdasarkan latar belakang yang sudah dijelaskan maka rumusan masalah dalam penelitian ini adalah bagaimana proses penerapan microservice untuk beberapa fitur pada aplikasi CMS dengan basis local disk.

B. Tinjauan Pustakan

1) *Aplikasi*: Secara umum merupakan alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi bisa dikaitkan sebagai perangkat desktop atau komputer serta perangkat mobile yang siap dipakai bagi user. Aplikasi terdiri dari beberapa jenis kelompok. Terdapat aplikasi desktop, aplikasi web dan aplikasi mobile. Manfaat dari sebuah aplikasi untuk membantu pekerjaan dari pengguna, maka manfaat yang dibuat sebuah aplikasi yaitu untuk membuat alur pekerjaan menjadi lebih efisien. Mempercepat sebuah pekerjaan yang dilakukan secara terorganisir. Sistem umum dibuat sesuai dengan porsi dari pekerjaan yang ada serta dapat menghemat biaya operasi untuk sumber daya manusia yang tidak terlalu dibutuhkan [4] [5].
2) *Microservice*: Kumpulan dari sebuah proses hasil independen dan microservice terdiri dari service-service yang menyediakan blok-blok lebih kecil yang berkomunikasi antara satu service dengan service lainnya untuk membentuk aplikasi kompleks. Manfaat utama dalam mempergunakan microservices adalah agar team developer mampu mengembangkan aplikasi secara cepat dengan membuat komponen-komponen dari aplikasi berjalan secara independen sehingga dapat memenuhi kebutuhan bisnis yang terus menerus berubah. Kekurangan dari microservice yaitu memiliki tes yang rumit umumnya microservice membutuhkan lebih banyak langkah testing. Microservice membutuhkan sistem automation yang cukup tinggi [6].
3) *Content Management System*: Software yang memungkinkan penggunaan membuat dan mengelola website dengan mudah. Aplikasi Content Management Sistem (CMS) memberikan tampilan antarmuka atau user interface di mana pengguna dapat mengatur tampilan, fitur dan isi website dengan praktis. Fungsi dari sebuah Content Management Sistem (CMS) untuk menyediakan fitur manajemen user. Dalam fitur manajemen user terdapat akses yang diberikan dan berbeda-beda tergantung tingkat jabatan dan fungsi dari jabatan tersebut. Serta fungsi yang diterapkan Content Management Sistem (CMS) untuk menyimpan informasi dari perusahaan [7].
4) *Backend Development*: Proses aplikasi atau sistem yang berjalan di mana prosesnya terdapat menambahkan, mengubah serta menghapus data. Backend biasanya mengurus segala jenis proses yang tidak berhubungan langsung dengan pengguna, seperti server dan basis data. Backend development memiliki tugas untuk merancang struktur model data. Membuat kode program untuk aplikasi menjadi lebih aman agar tidak mudah diserang oleh gangguan dari luar seperti hacking, craking, dan lain-lain. Mengembangkan kode program dari sisi client maupun server dan melakukan testing dengan penggunaan framework khusus untuk mempermudah proses penulisan [8] [9].
5) *API Driven Architecture*: Dalam pengembangan sistem menggunakan arsitektur API-Driven, hal yang biasa dilakukan adalah, komunikasi antar service akan dilakukan via API call. Bisa menggunakan RESTful, RPC atau sejenisnya. Kurang lebih komunikasi akan dilakukan secara synchronous. Arsitektur ini bukan pendekatan yang terbaik dikarenakan cara mengakses langsung pada setiap service dengan API call. Setiap service memiliki suatu IP publik yang dapat diakses dari jaringan internet. Terkadang juga dengan satu IP publik tetapi dibedakan port untuk melayani setiap service. Kelebihan dalam menggunakan API driven architecture yaitu terdapat loose coupling. Dengan adanya API gateway dapat menurunkan tingkat ketergantungan client terhadap service-service yang ada. Terlalu banyak round trip atau suatu halaman front end atau mobile app terkadang memerlukan banyak sekali pemanggilan ke beberapa service yang berbeda. Bisa disimpulkan Ketika service dipanggil oleh banyak penggunaan API gateway dapat diturunkan tingkat latency dengan membuat aggregate atau aggregator. Keuntungan selanjutnya dengan memanfaatkan security dalam implement di level api sehingga API service yang diakses hanya bisa dilingkungan privat sehingga tidak terekspose secara langsung dari luar [10][11].

II. ANALISIS DAN RANCANGAN SISTEM

Dalam perancangan perangkat lunak aplikasi ini menerapkan konsep Systems Development Life Cycle (SDLC) agile yang berfungsi untuk menggambarkan tahapan – tahapan yang akan dilakukan dalam perancangan perangkat lunak Content Management Sistem (CMS) di PT M dengan arsitektur microservice sebagai berikut:



Gambar 1. Konsep SDLC

Pada Gambar di atas terdapat sistem perancangan perangkat lunak menggunakan konsep System Development Life Cycle (SDLC) agile. Terdapat planning atau rencana sebelum membuat aplikasi kebutuhan apa yang sedang ingin dikembangkan perusahaan. Analysis atau analisis yang merupakan pencarian kebutuhan pada software ketika sudah mengetahui hal apa saja yang akan dibuat. Selanjutnya design atau desain yang merupakan kebutuhan untuk membuat implementasi dalam bentuk blueprint software untuk membentuk bayangan software apa yang akan dikembangkan. Lalu implementasi yang merupakan pembuatan perangkat lunak dari yang sebelumnya blueprint menjadi interaktif atau bisa digunakan. Serta yang terakhir testing untuk melakukan tes hasil perangkat lunak yang telah dibuat.

A. Perancangan Sistem

Dalam penulisan ini tahapan perencanaan yang dilakukan dalam penerapan microservice untuk beberapa fitur pada aplikasi CMS berbasis local ini. Developer dan tim bisnis memutuskan feature-feature yang sering digunakan oleh admin dalam aplikasi Content Management Sistem (CMS) dan yang digunakan secara berkala. Setelah diputuskan feature-feature yang secara rutin digunakan maka terbentuklah tugas untuk pembuatan service reporting dan parameter dengan menggunakan bahasa pemrograman node.js untuk pembuatan microservice. Service yang dibuat terdiri dari service adapter dan aggregator.

Service adapter dibuat untuk mengakses database dalam pemanggilan data dalam database, pemasukan data dalam database dan pembaharuan data dalam database. Dalam service adapter terdapat service adapter reporting dan adapter parameter. Dalam service adapter reporting memiliki sub service eform, qris dan onboarding. Serta service berikutnya adapter parameter yang di dalamnya terdapat sub service parameter pajak dan Txtreferral.

B. Analisis Masalah

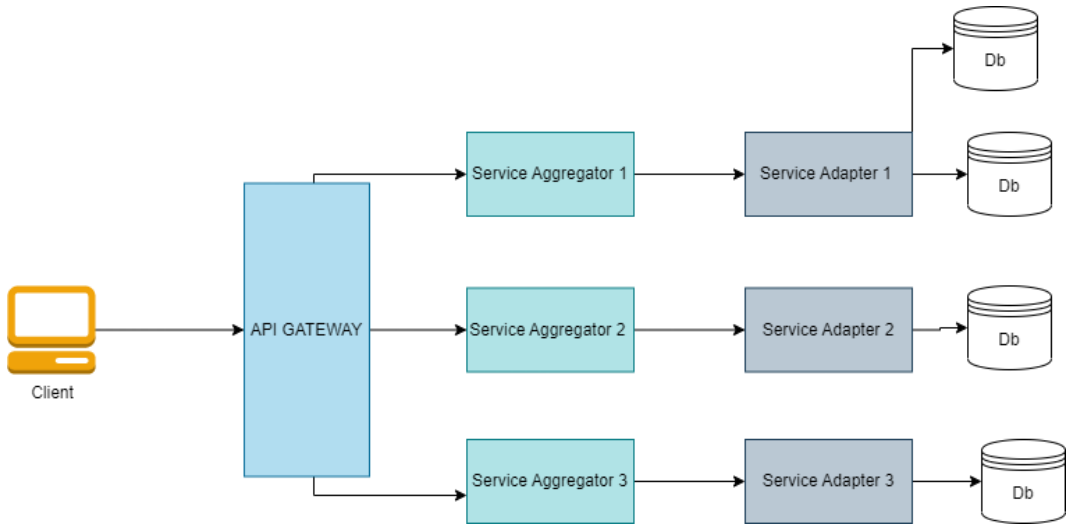
Adanya arsitektur *microservice* membuat pengembangan aplikasi website lebih mudah karena dalam pengembangan perangkat lunak atau aplikasi dibagi menjadi *service-service* yang lebih untuk dijalankan secara independen dan mampu menggabungkan bahasa pemrograman yang berbeda dalam pembuatannya. Hal ini menjadi pertanyaan bagaimana cara perancangan perangkat lunak aplikasi *Content Management Sistem* (CMS) dengan arsitektur *microservice*.

Dari pertanyaan tersebut menjadi sebuah dasar untuk penulisan tugas akhir ini. Untuk mengetahui cara perancangan perangkat lunak aplikasi *Content Management Sistem* (CMS) dengan arsitektur *microservice*.

C. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan untuk memberikan gambaran tentang fungsi dari sistem yang dirancang sehingga dapat mengatasi masalah – masalah yang terjadi, sistem tersebut nantinya mampu *:*

1) Dimengerti oleh developer baru karena penggunaan arsitektur microservice dibuat relatif lebih kecil.
2) Bergerak lebih cepat karena aplikasi dibuat dengan penjalanan secara individual atau sendiri-sendiri terkait service yang akan dijalankan.



Gambar 2. Service API Gateway

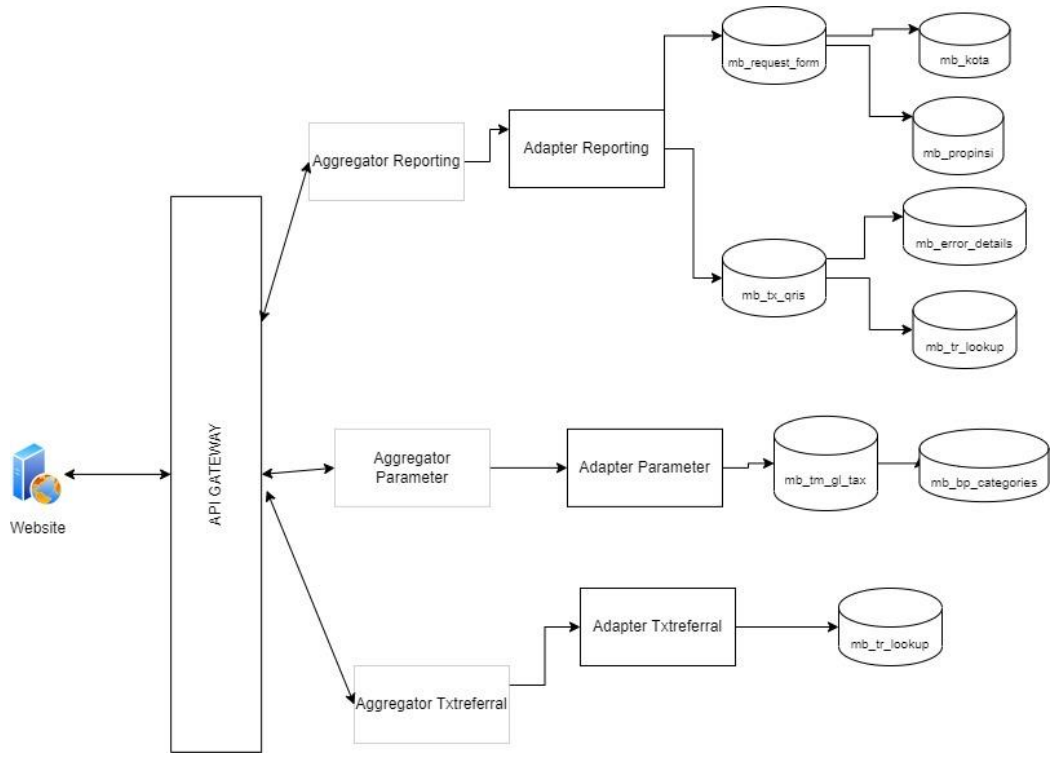
III. IMPLEMENTASI

Implementasi sistem yang merupakan tahapan dari penerapan proses perancangan perangkat lunak Content Management Sistem (CMS) di PT M dengan arsitektur microservice. Proses ini dimulai dari pembuatan database dengan menggunakan SQL untuk penyimpanan data yang dibuat. Lalu menentukan service yang akan dibuat. Service ini terdiri dari dua pola yang pertama terdapat adapter yang difungsikan untuk mengakses semua data dari banyak database yang dibuat. Kedua service aggregator yang difungsikan untuk penyediaan logika pemanggilan untuk bagian depan atau frontend. Kedua service tersebut harus dijalankan secara bersama dari aggregator yang menyediakan request apa saja yang dibutuhkan untuk bagian depan lalu akan memanggil adapter untuk mengakses database dengan memanfaatkan API call yang dibuat. Dalam adapter terdapat tiga service yaitu reporting, parameter dan txtreferral. Aggregator pun memiliki tiga service yaitu reporting, parameter dan txtreferral.

A. Alur Microservice

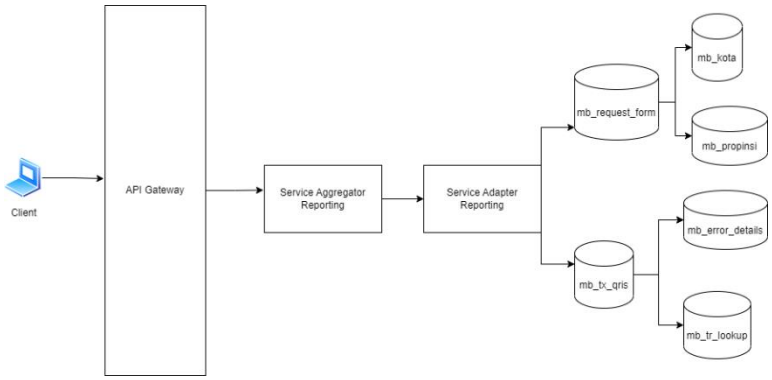
Alur microservice ditugaskan untuk menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap Langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah. Alur ini akan berperan penting dalam memutuskan sebuah langkah atau fungsionalitas dari sebuah proyek yang dijalani dalam pembuatan program.

Alur microservice ini bertujuan dalam rancangan aplikasi berjalan. Alur dari rancangan dibuat terlebih dahulu untuk membuat perancangan lebih mudah dan lebih paham akan alurnya. Berikut merupakan gambar dari rancangan alur yang dibuat dalam pengembangan beberapa fitur menjadi microservice.



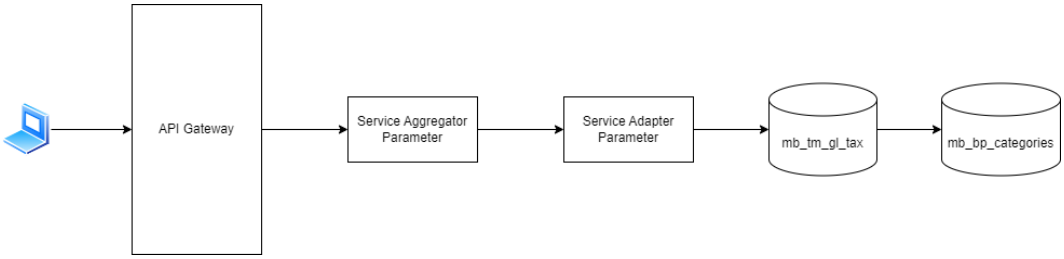
Gambar 3. Alur Fitur Microservice

Gambar di atas menunjukkan alur dari microservice yang digunakan. Penerapannya menggunakan konsep arsitektur API Gateway. Pembuatan dimulai dari pembuatan database. Melanjutkan untuk membuat adapter untuk mengakses setiap tabel yang dibuat. Setelah adapter dibuat melanjutkan pembuatan aggregator untuk mengakses adapter-adapter yang dibuat. User akan mengakses tampilan depan dan tampilan depan akan mengakses API Gateway yang dibuat dan akan berjalan menuju aggregator untuk respons.



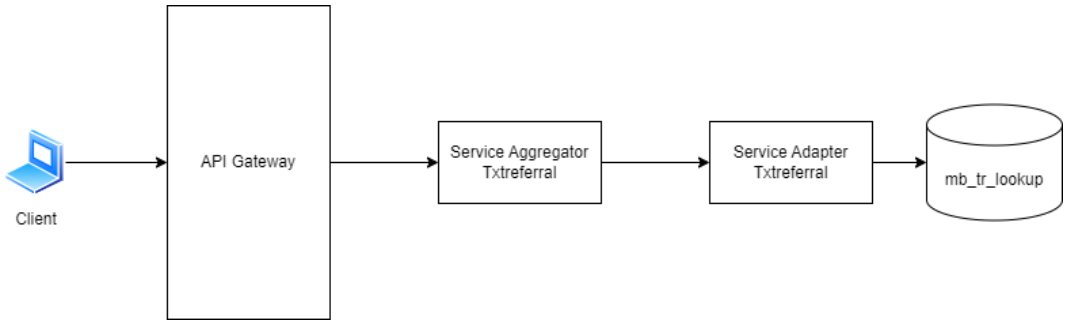
Gambar 4. Microservice Reporting

Pada gambar di atas merupakan alur dari microservice bagian reporting. Berikut merupakan tampilan alur dari microservice parameter.



Gambar 5. Alur Microservice Parameter

Gambar di atas merupakan alur dari microservice parameter. Client akan mengakses api gateway lalu api gateway akan dilakukan validasi dalam service aggregator parameter dan service adapter parameter sebagai penghubung antar database-database. Lalu alur yang terakhir adalah alur microservice txtreferral.



Gambar 6 Alur Microservice Txtreferral

Gambar di atas menunjukan alur dari microservice txtreferral yang pertama client akan mengakses api gateway dari txtreferral lalu service aggregator akan melakukan validasi dari request yang dibutuhkan. Selanjutnya service adapter akan memproses dengan memanggil database yang digunakan txtreferral.

B. Database

Langkah pertama dalam implementasi yaitu pembuatan database untuk membuat sebuah tabel yang bisa digunakan dalam menyimpan semua informasi yang nantinya akan digunakan oleh sistem yang dibuat. Database dibuat dengan mysql dan platform SQLyog. Tabel yang akan dibuat sesuai dengan service. Dalam service reporting yaitu mb_request_form, mb_tx_qris, mb_error_details, mb_kota, mb_propinsi, mb_tr_lookup. Serta dalam service parameter terdapat tabel database mb_tm_gl_tax dan mb_bp_categories.

C. Adapter

Pembuatan adapter merupakan tahapan selanjutnya dalam proses pembuatan aplikasi Content Management Sistem (CMS) berjalan dengan menggunakan sistem microservice. Pola adaptor ini menerjemahkan interface satu ke interface lainnya. Interface sendiri pada pola adapter ini merupakan properti dan metode objek. Pola adapter pada kasus ini merupakan tahapan code yang akan mengakses banyak database yang telah dibuat. Pola microservice yang digunakan dengan konsep API driven arsitektur. API driven arsitetur ini merupakan pola pembuatan microservice dengan pemanfaatan API gateway yang dibuat. Semua code akan dibuat untuk bisa mengakses API yang dibuat. Pembuatannya ini akan mempermudah dalam pengembangan perangkat lunak. Pola adapter ini memungkinkan komponen pemrograman untuk bekerja bersama serta tidak akan terjadi interface yang tidak cocok.

Pada implementasi adapter ini terdapat beberapa service yang dibuat sebagai adapter. Terdapat tiga service adapter yaitu reporting, parameter dan txtreferral. Dari ketiga service tersebut dibuat kategori yang akan mempermudah developer dalam perbaikan karena setiap service yang dibuat memiliki fungsi masing-masing.

D. Aggregator

Tahap aggregator merupakan tahapan dari seluruh logic untuk komunikasi yang akan diberikan dari bagian depan frontend dan diterima serta diberikan respon oleh bagian belakang backend. Pola aggregator memiliki pola desain layanan yang menerima permintaan dari klien atau API gateway dan kemudian mengirimkan permintaan dari beberapa layanan mikro dari bagian belakang atau backend. Membuat penggabungan hasil dan merespons kembali permintaan awal dalam 1 struktur respons. Penerapan aggregator dibuat untuk mengurangi komunikasi yang berlebihan antaran klien dan layanan mikro.

Pada Tahap implementasi aggregator terdapat beberapa service yang dibuat sebagai aggregator. Terdapat tiga service aggregator yaitu aggregator reporting, aggregator parameter dan aggregator txtreferral. Dari ketiga service tersebut dibuat kategori yang akan mempermudah developer dalam perbaikan karena setiap service yang dibuat memiliki fungsi masing-masing.

IV. PENGUJIAN

Pengujian ini lebih menekankan pengujian sistem belakang atau backend. Pengujian ini akan lebih banyak mendapatkan hasil json yang dilakukan dengan aplikasi postman. Pembuatan API yang sudah dilakukan pada saat ini akan dilakukan pengujian jika berhasil dan jika tidak berhasil. Tujuan pengujian untuk membuktikan sistem yang diimplementasikan telah memenuhi spesifikasi dan rancangan yang sudah direncanakan sebelumnya. Hasil dari pengujian akan dimanfaatkan untuk menyempurnakan kinerja dari sistem dan dapat digunakan dalam pengembangan sistem lebih lanjut. Metode yang dilakukan adalah dengan metode fungsional yang merupakan metode untuk menguji setiap komponen dari code yang terdapat pada proyek akhir ini berdasarkan karakteristik dan fungsi masing-masing service yang telah dibuat serta API gateway yang telah dimiliki melalui penyimpanan lokal.

Pengujian akan dilakukan berdasarkan service yang dibuat. Terdapat enam service yang terdiri dari service adapter reporting, service adapter parameter, service adapter txtReferral, service aggregator reporting, service aggregator parameter dan service aggregator txtreferral. Service akan dibagi menjadi dua service inti sesuai pola yaitu adapter dan aggregator.

A. Pengujian Service Reporting

Service reporting memiliki service adapter dan aggregator. Kedua service ini akan menjadi acuan yang penting untuk dijadikan pengujian dan akan mendapatkan hasil dari keberhasilan dan kegagalan apa saja yang bisa terjadi. Dalam kedua service tersebut terdapat sub service yaitu eform dan qris. Pada service reporting akan dijalankan menggunakan terminal dengan penulisan code npm start karena penggunaan node js sebagai framework dan bahasa yang digunakan. Berikut hasil time out jika penjalanan npm start tidak menggunakan mysql yang dijalankan dalam service adapter reporting.

TABEL I

Time Out Adapter

```
  errno: -4078,
    code: 'ECONNREFUSED',
    syscall: 'connect',
    address: '127.0.0.1',
    port: 3306,
    fatal: true
  },
  original: Error: connect ECONNREFUSED 127.0.0.1:3306
      at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1247:16) {
    errno: -4078,
    code: 'ECONNREFUSED',
    syscall: 'connect',
    address: '127.0.0.1',
    port: 3306,
    fatal: true
  }
 }
Server running at http://127.0.0.1:30337
```

Pada gambar di atas *port* berjalan tetapi tidak bisa mengakses *database* karena mysql yang dijalankan tidak terdeteksi aktif. Diperlukan pengaktifan mysql terlebih dahulu. Berikut merupakan tampilan jika mysql di aktifkan.

TABEL II

Npm Success Adapter

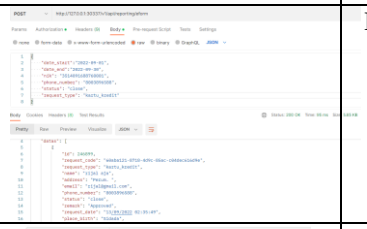
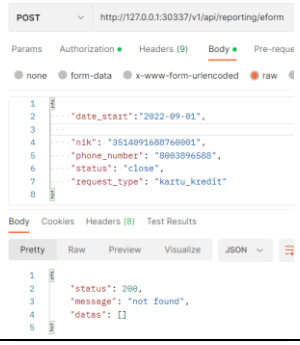
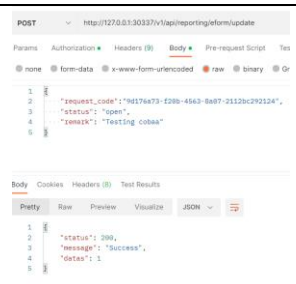
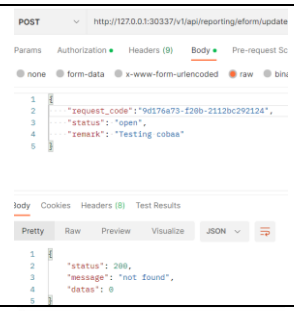
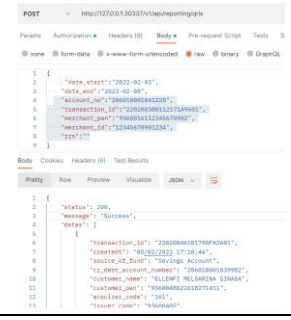
```
PS D:\Kuliah\Aplikasi\Oke\cms-be-adapter-reporting> npm start

> cms-be-adapter-reporting@1.0.0 start
> node ./src/server.js

{
 message: {
  timestamp: '2023-01-09 18:13:57',
  query: 'Executing (default): SELECT 1+1 AS result'
 },
 level: 'info'
}
Connection has been established successfully.
Server running at http://127.0.0.1:30337
```

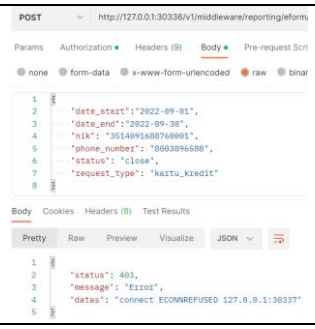
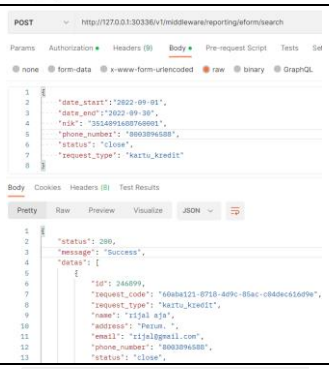
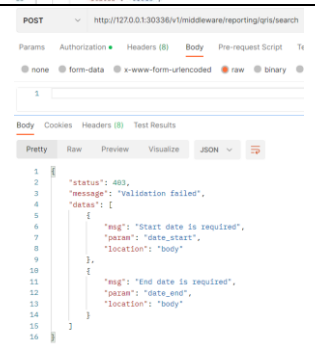
Tampilan success akan memberikan kata-kata *Connection has been established successfully*. Pada bagian adapter reporting aplikasi sudah bisa digunakan jika terdapat tulisan *successfully*. Berikut merupakan tampilan pada aggregator reporting jika dijalankan pada terminal.

e-ISSN : 2443-2229

Hasil dari pengujian reporting yang dilakukan dengan manual ditemukan bahwa sebagai berikut:

TABEL III
Pengujian Service Reporting

Nama Service	Port	Output	Hasil
Get Eform	http://127.0.0.1:30337/v1/api/reporting/eform		Berhasil
Adapter Get Eform	http://127.0.0.1:30337/v1/api/reporting/eform		Tidak Ditemukan
Adapter Update Eform	http://127.0.0.1:30337/v1/api/reporting/eform/update		Berhasil
Adapter Update Eform	http://127.0.0.1:30337/v1/api/reporting/eform/update		Tidak Ditemukan
Adapter Get Qris	http://127.0.0.1:30337/v1/api/reporting/qris		Berhasil

Aggregator Get Eform	http://127.0.0.1:30336/v1/middleware/reporting/eform		Berhasil
AggregaotrGet Eform	http://127.0.0.1:30336/v1/middleware/reporting/eform		NotFound
Aggregator Get Qris	http://127.0.0.1:30336/v1/middleware/reporting/qris		Berhasil

Pada table di atas merupakan hasil dari keseluruhan hasil dari service reporting yang terdiri service adapter reporting dan aggregator reporting. Mengacu pada port hasil berhasil dan tidak ditemukan.

B. Pengujian Service Parameter

Service parameter memiliki service adapter dan aggregator parameter. Service-service tersebut menjadi sebuah jalur yang harus beriringan untuk melakukan tahap pengujian. Dalam kedua service tersebut terdapat sub service yang bernama Gl Pajak. Sama seperti pengujian service reporting pada pengujian service parameter menggunakan pemanggilan npm start untuk kedua service adapter parameter dan aggregator. Berikut merupakan hasil dari penjalaan adapter parameter.

TABEL IV

Adapter Success Parameter

```
PS D:\Kuliah\Aplikasi\Oke\cms-be-adapter-parameter> npm start

> cms-be-adapter-parameter@1.0.0 start
> node ./src/server.js

{
  message: {
    timestamp: '2023-01-09 18:10:17',
    query: 'Executing (default): SELECT 1+1 AS result'
  },
  level: 'info'
```

```
}
Connection has been established successfully.
Server running at http://127.0.0.1:30341
```

Pada parameter adapter memiliki port yaitu 30341 serta dalam pemanggilan aggregator akan memiliki port 30340. Berikut merupakan gambar dari aggregator parameter.

TABEL V

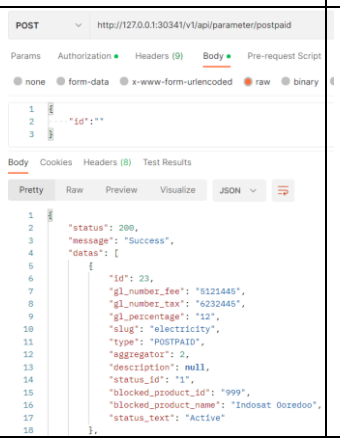
Aggregator Success Parameter

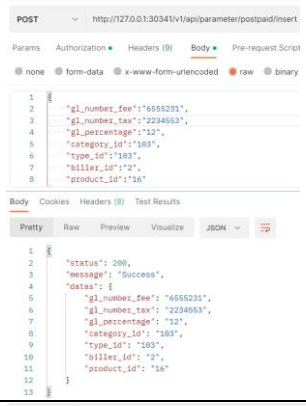
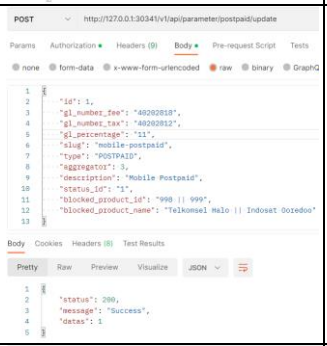
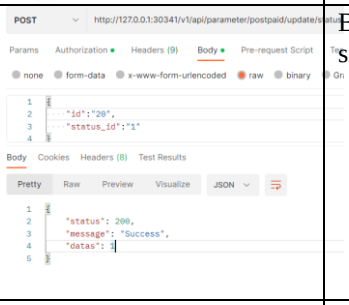
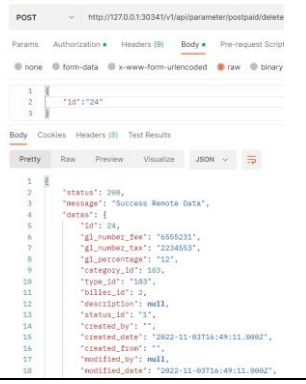
```
PS D:\Kuliah\Aplikasi\Oke\cms-be-aggregator-parameter> npm start

> aggregator-parameter-service@1.0.0 start
> node ./src/server.js

Connection has been established successfully.
Server running at http://127.0.0.1:30340
```

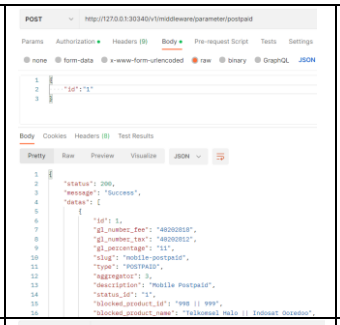
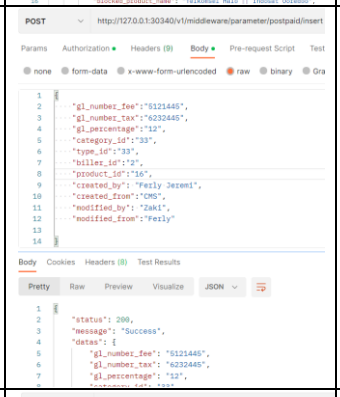
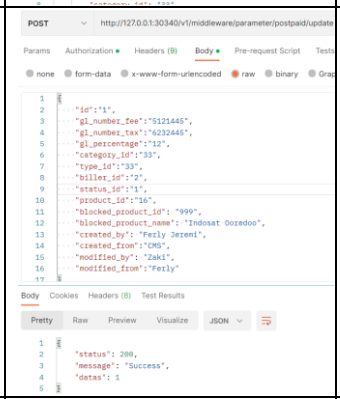
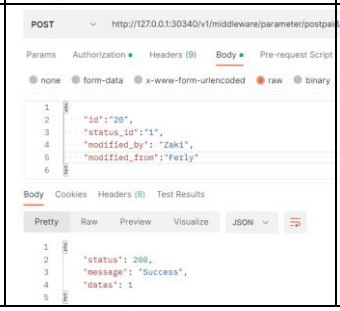
Pada parameter aggregator memiliki port 30340. Jika pada table di atas terdapat server running at http://127.0.0.1:30340 maka aggregator telah berhasil berjalan. Pengujian adapter parameter memiliki lima fungsi yaitu get parameter postpaid URL, update parameter postpaid, insert parameter postpaid, delete parameter postpaid dan update status parameter postpaid. Fungsi pada tiap URL yang akan dipanggil memiliki penggunaan yang berbeda-beda. Pengujian pertama yaitu terkait pengambilan data atau get parameter postpaid URL. URL yang dimiliki untuk pemanggilan sebagai berikut pada adapter http://127.0.0.1:30341/v1/api/parameter/postpaid. Pengujian aggregator parameter memiliki alur yang beriringan dengan adapter karena jika aggregator dijalankan tetapi adapter tidak dijalankan maka hasil akan menjadi time out karena koneksi tidak dapat dilakukan. Aggregator parameter ini memiliki lima fungsi yang sama dengan adapter parameter. Terdapat get parameter postpaid, update parameter postpaid, insert parameter, delete parameter dan update status id parameter. Pembeda aggregator dengan adapter yaitu berdasarkan API jika adapter terdapat v1/api akan tetapi aggregator v1/middleware.

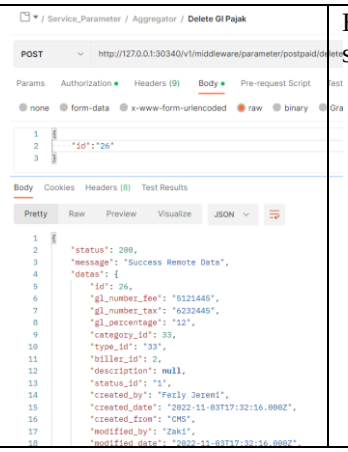
TABEL VI

Aggregator Success Parameter

Nama Service	**Port**	**Output**	**Hasil**
Get Adapter Parameter	http://127.0.0.1:30341/v1/api/parameter/postpaid		Berhasil

Insert Adapter Parameter	http://127.0.0.1:30341/v1/api/parameter/postpaid/insert		Berhasil
Update Adapter Parameter	http://127.0.0.1:30341/v1/api/parameter/postpaid/update		Berhasil
Update StatusAdapter Parameter	http://127.0.0.1:30341/v1/api/parameter/postpaid/update/status		Berhasil
Delete Adapter Parameter	http://127.0.0.1:30341/v1/api/parameter/postpaid/delete		Berhasil

Get Aggregator Parameter	http://127.0.0.1:30340/v1/middleware/parameter/postpaid		Berhasil
Insert Aggregator Parameter	http://127.0.0.1:30340/v1/middleware/parameter/postpaid/insert		Berhasil
Update Aggregator Parameter	http://127.0.0.1:30340/v1/middleware/parameter/postpaid/update		Berhasil
Update Status Aggregator Parameter	http://127.0.0.1:30340/v1/middleware/parameter/postpaid/update/status		Berhasil

Delete Aggregator Parameter	http://127.0.0.1:30340/v1/middleware/parameter/postpaid/delete		Berhasil

V. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Berdasarkan hasil penerapan microservice untuk beberapa fitur pada aplikasi CMS berbasis lokal berjalan sesuai dengan alur dari harapan penulis. Akan tetapi perancangan sistem menggunakan microservice masih dilakukan pada pengembangan berbasis lokal disk atau hanya dapat diakses menggunakan device sendiri tanpa bisa dibagikan kepada orang lain yang ingin menggunakan API dari microservice. Pengembangan pada Content Management Sistem (CMS) jauh lebih mudah dilakukan karena microservice memiliki alur yang lebih kecil jika dibandingkan dengan aplikasi yang berbasis monolithic.

B. *Saran*

Dalam pengembangan sebuah aplikasi teknologi-teknologi yang digunakan memiliki kelebihan ataupun kekurangan. Pengembangan aplikasi CMS dengan konsep microservice memiliki keunggulan yaitu memberikan developer kebebasan dalam pengembangan. Akan tetapi dibalik beberapa keunggulan microservice terdapat kekurangan dalam penggunaan konsep microservice. Pengembangan yang dilakukan oleh microsercive menjadi lebih kompleks. Saran dalam pengembangan yaitu untuk membuat sistem microservice pembuatan fitur masuk ke dalam kategori yang sesuai. Contoh pada service reporting hanya dapat menampilkan data tanpa perlu penghapusan, pemasukan dan pembaharuan data.

DAFTAR PUSTAKA

[1] N. Goroshka, "Abstracts international scientific and practical conference of young scientists," vol. 004, no. 477, pp. 121–122, 2014.
[2] H. Suryotrisongko, "Arsitektur Microservice untuk Resiliensi Sistem Informasi," *Sisfo*, vol. 06, no. 02, pp. 231–246, 2017, doi: 10.24089/j.sisfo.2017.01.006.
[3] J. T. Zhao, S. Y. Jing, and L. Z. Jiang, "Management of API Gateway Based on Micro-service Architecture," *J. Phys. Conf. Ser.*, vol. 1087, no. 3, 2018, doi: 10.1088/1742-6596/1087/3/032032.
[4] M. (2018) Siregar, H. F., Siregar, Y. H., & Melani, "Perancangan Aplikasi Komik Hadist Berbasis Multimedia. JurTI (Jurnal Teknologi Informasi), 2(2), 113-121.," *JurTI (Jurnal Teknol. Informasi)*, vol. 2, no. 2, pp. 113–121, 2018, [Online]. Available: http://www.jurnal.una.ac.id/index.php/jurti/article/view/425
[5] AHMAD SYAIFUDDIN, "Aplikasi Strategi Pemasaran Berbasis Syariah Melalui Pendekatan Bauran Pemasaran Pada Hotel Oase Pekanbaru," *JurTI (Jurnal Teknol. Informasi)*, vol. 3, no. April, pp. 49–58, 2017.
[6] M. Y.Yohakim and Badiyanto, "Implementasi Arsitektur Microservice pada Pembuatan Surat Unit Kegiatan Mahasiswa Informatika dan Komputer Menggunakan Node.JS," *JIKO (Jurnal Inform. dan Komputer)*, vol. 4, no. 2, pp. 71–80, 2019.
[7] C. De Haan, "Website Prototype Design Animation Study Program Based Content Management System Wordpress," *J. Mantik*, vol. 6, no. 36, pp. 1971–1979, 2022.
[8] B. Syaftiaan, N. F. Safira, F. Rizky, and U. Telkom, "Rancang Bangun Backend Aplikasi Jobbie : Pencarian Dan Penyedia," *e-Proceeding Eng.*, vol. 8, no. 6, pp. 12441–12448, 2021.
[9] R. V. Shahir Daya Nguyen Van Duy, Kameswara Eati, Carlos M Ferreira, Dejan Glozic, Vasfi Gucer, Manav Gupta, Sunil Joshi, Valerie Lampkin, Marcelo martins, Shishir Narain, "Microservices from Theory to Practice Creating Applications in IBM Bluemix Using the Microservices Approach," *Ibm*, p. 170, 2015.
[10] C. K. Rudrabhatla, "Comparison of event choreography and orchestration techniques in Microservice Architecture," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 8, pp. 18–22, 2018, doi: 10.14569/ijacsa.2018.090804.
[11] S. Zhelev and A. Rozeva, "Using microservices and event driven architecture for big data stream processing," *AIP Conf. Proc.*, vol. 2172, no. November 2014, pp. 2013–2014, 2019, doi: 10.1063/1.5133587.