

# Implementasi Arsitektur enterprise Pola *Vending Machine* pada Teknologi *Microservices*

Muhammad Iqbal  
Jurusan Informatika  
Universitas Islam Indonesia  
Yogyakarta, Indonesia  
17523092@students.uii.ac.id

Teduh Dirgahayu  
Jurusan Informatika  
Universitas Islam Indonesia  
Yogyakarta, Indonesia  
teduh.dirgahayu@uui.ac.id

Hanson Prihantoro Putro  
Jurusan Informatika  
Universitas Islam Indonesia  
Yogyakarta, Indonesia  
hanson@uui.ac.id

**Abstrak**— Salah satu tujuan dari penerapan Arsitektur enterprise adalah menciptakan kesesuaian antara bisnis dan teknologi informasi bagi kebutuhan organisasi. Di sisi lain, *microservices* merupakan teknologi yang sering digunakan belakangan ini berkat skalabilitasnya. Literatur yang membahas rancangan Arsitektur enterprise yang diimplementasikan menggunakan teknologi *microservices* masih sangat terbatas. Begitu juga tingkat keberhasilannya yang masih rendah merupakan permasalahan yang dihadapi pada penelitian ini. Maka dari itu penelitian ini bertujuan membangun kerangka implementasi pola Arsitektur enterprise menggunakan teknologi *microservices*. Dalam pengembangan penelitian ini menggunakan metode yang digunakan secara berulang dan berada pada tiap ranah lapisan Arsitektur enterprise yang disesuaikan ke dalam kasus yang diangkat. Rancangan ini diharapkan dapat mempermudah pengembangan sistem enterprise serta meningkatkan peluang keberhasilannya.

**Kata Kunci**—arsitektur enterprise, *microservices*, pola *vending machine*

## I. PENDAHULUAN

Arsitektur enterprise adalah sebuah sarana yang digunakan dalam mewujudkan kesesuaian teknologi informasi dengan bisnis yang dijalankan dalam sebuah organisasi. Arsitektur enterprise merepresentasikan tujuan dari *stakeholder* yang mencakup informasi, kegunaan, lokasi organisasi dan parameter kinerja. Arsitektur enterprise menggambarkan rencana untuk mengembangkan suatu sistem atau sekumpulan sistem [1]. Secara umum arsitektur enterprise sangat diperlukan, contohnya dalam menyelaraskan kondisi nyata perusahaan terhadap tujuan manajemen, mengurangi perubahan yang mungkin terjadi dalam pengembangan sistem, mengurangi *timeframe*, dan mengurangi kebutuhan sumber daya [5]. Karena rencana awal telah menetapkan kesesuaian antara teknologi informasi dan bisnis yang akan dijalankan.

Dalam pembuatan arsitektur enterprise, dimulai dengan memodelkan keseluruhan sudut pandang yang berkaitan [1]. Serta dilakukannya pengelompokan sudut pandang berdasarkan ranah arsitektur menggunakan kerangka kerja yang akan digunakan [6]. Kerangka kerja yang akan digunakan nantinya akan menjadi patokan dalam pembuatan arsitektur enterprise. Ada berbagai macam kerangka kerja yang dapat dimanfaatkan. Contohnya, TOGAF dan Zachman [5].

Arsitektur enterprise memiliki beberapa ranah arsitektur yang tersusun ke dalam sejumlah lapisan yang antar kerangka tidak memiliki kepastian akan jumlah dan ranah arsitektur penyusunnya. Makalah ini menggunakan ranah arsitektur yang ada pada TOGAF yaitu: arsitektur bisnis, arsitektur data, arsitektur aplikasi, dan juga arsitektur teknologi. Dikarenakan arsitektur yang lebih sederhana, TOGAF juga diharapkan

dapat memperluas tingkat penerimaan terhadap kerangka implementasi yang ditawarkan dengan menggunakan beberapa pemetaan [6].

Pengembangan sistem enterprise dapat dilakukan dengan menggunakan *microservices* salah satunya. Dalam pengembangan perubahan bisa terjadi pada proses bisnis maupun beban komputasi. *Microservices* menerapkan gunaulang (*reuse*) pada pengembangan sistem enterprise sehingga memudahkan sistem untuk dikembangkan lebih cepat dan sistem dapat menyesuaikan perubahan kebutuhan [11]. Sebagai contoh, sebuah toko telah memperoleh *customer* melalui pemasaran klasik seperti penyebaran pamflet dan iklan di media cetak. Namun pada saat sekarang *customer* lebih banyak memperhatikan pemasaran melalui internet daripada memperhatikan pamflet ataupun iklan di media cetak. Hal ini membuat pemasaran produk melalui pamflet dan iklan di media cetak menjadi kurang efektif [6]. Maka dari itu, perlunya aplikasi untuk mendukung proses bisnis baru melalui internet *vending machine* (toko *online*). Selain berdampak terhadap proses bisnis yang ada, dengan adanya proses bisnis yang baru ini akan berujung pada peningkatan penjualan. Hal ini yang harus dipertimbangkan saat menerapkan proses bisnis baru. *Vending machine* merupakan pola yang digunakan untuk menyediakan produk maupun jasa melalui katalog untuk diperjualbelikan melalui internet.

Perancangan arsitektur di tiap ranah dilakukan secara bertahap mulai dari arsitektur bisnis, arsitektur data, arsitektur aplikasi, hingga arsitektur teknologi. Perancangan arsitektur dilakukan secara bertahap untuk meyakinkan bahwa kebutuhan bisnis enterprise dibantu sepenuhnya oleh sistem teknologi informasi enterprise mulai dari ranah arsitektur data, aplikasi hingga teknologi. Hal ini dilakukan agar kebutuhan bisnis enterprise didukung sepenuhnya oleh sistem yang akan dibangun. Tetapi perlu diperhatikan bahwa sebaiknya arsitektur enterprise dibebaskan dari pilihan teknologi yang akan digunakan, Hal ini akan mempengaruhi arsitektur enterprise terhadap perkembangan dan perubahan *platform* [8].

Dalam hal teknologi, arsitektur teknologi *microservices* merupakan salah satu teknologi yang sering digunakan belakangan ini berkat skalabilitasnya. Karakteristik tertentu dari arsitektur *microservices* adalah merangkai aplikasi menjadi layanan-layanan kecil yang bersifat independen, berkomunikasi melalui satu sama lain dengan baik, serta menggunakan mekanisme yang ringan untuk melayani tujuan bisnis. Komunikasi layanan satu sama lain bergantung pada kebutuhan aplikasi yang digunakan. Selain itu, berkat skalabilitasnya biaya dalam proses pengembangan *microservices* akan relatif lebih murah [3]. *Microservices* sangat dibutuhkan dalam pengembangan kasus *vending machine* (toko *online*) yang kompleks dan sulit untuk di

*maintenance*. *Microservices* mampu beradaptasi dengan perubahan terutama pada kompleksitas dan *maintenance*. Seperti kendala yang ditemukan pada penelitian serupa yang mengatakan bahwa sebagian besar beralih dari *monolithic* ke *microservices* dikarenakan masalah utama yaitu: kompleksitas, dan juga sulitnya *maintenance monolithic*.

Sebuah survei Oreilly [9] mengatakan bahwa dalam adopsi *microservices* sekitar 28% organisasi responden telah menggunakan *microservices* setidaknya kurun waktu 3 tahun dan sebanyak 62% responden telah menggunakan *microservices* selama satu tahun lebih. Sebanyak 10% melaporkan bahwa adopsi *microservices* mereka telah sukses total, sedangkan mayoritas (54%) melaporkan bahwa setidaknya Sebagian besar berhasil. Sejumlah pengadopsi mengatakan bahwa kompleksitas yang menjadi tantangan terbesar bagi mereka [9].

Maka dari itu penelitian ini bertujuan menggali lebih dalam mengenai deskripsi implementasi pola arsitektur enterprise pada teknologi *microservices*. Dengan menggunakan pola *vending machine* sebagai acuan, serta menggunakan ranah arsitektur yang ada pada TOGAF. Penelitian ini diharapkan dapat memudahkan dalam mengembangkan sistem enterprise dan meningkatkan peluang keberhasilan.

## II. KAJIAN PUSTAKA

### A. Arsitektur Enterprise

Pendekatan arsitektur enterprise pada perusahaan dapat membantu menyesuaikan sumber daya bisnis dan TI. Selain itu, Arsitektur enterprise juga dapat menyesuaikan prinsip-prinsip dasar dan metodologi umum yang mengatur seluruh sistem informasi pada saat proses pengembangan. Dalam hal ini, kerangka kerja arsitektur enterprise adalah cara mudah dalam mendukung metodologi dan peran terpisah yang memfasilitasi dan menerapkan metodologi ini sesuai kebutuhan [5]. Berikut ini merupakan ranah arsitektur yang ada pada TOGAF [6]:

#### 1) Arsitektur Bisnis

Arsitektur bisnis yang mendeskripsikan proses bisnis organisasi. Arsitektur ini, merupakan penggerak bagi komponen lain dari arsitektur enterprise.

#### 2) Arsitektur Data/Informasi

Arsitektur data/informasi yang merepresentasikan data/informasi sebagai aset untuk mendukung bisnis.

#### 3) Arsitektur Aplikasi

Arsitektur aplikasi yang mendeskripsikan aplikasi-aplikasi utama yang akan digunakan dalam mengelola data dan diperlukan dalam mendukung bisnis.

#### 4) Arsitektur Teknologi

Arsitektur teknologi yang mendeskripsikan *platform* teknologi sebagai penyedia lingkungan bagi aplikasi-aplikasi tersebut.

### B. Pola Arsitektur Vending Machine

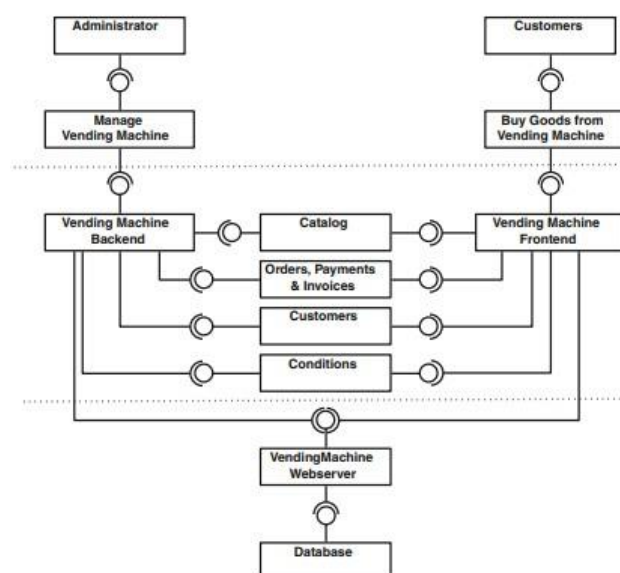
Pola arsitektur *vending machine* merupakan pola untuk membantu pemecahan masalah dalam penyajian produk atau layanan dari katalog untuk diperjualbelikan melalui *platform* internet (toko berbasis *web*). Baik pada *Business-to Consumer* (B2C) atau *Business-to-Business* (B2B). Pola ini memungkinkan *customer* dapat menelusuri katalog online,

memperoleh informasi harga produk, detail produk, dan memasukkan produk yang diinginkan kedalam keranjang pembelian *virtual*. *Customer* difasilitasi dalam proses pembayaran hingga nantinya produk di proses oleh penjual [6]. Berdasarkan definisi tersebut, berikut beberapa kemampuan dari pola arsitektur *vending machine* [6]:

- Mengelola dan menyajikan katalog artikel dengan info produk kepada *customer*.
- Menjual atau memberi segala jenis produk atau jasa yang dapat dijual dan dibeli (termasuk yang dapat diunduh seperti artikel, perangkat lunak, atau dokumen elektronik).
- Memberi *customer* kemungkinan untuk meletakkan produk ke dalam keranjang *virtual*.
- Memandu *customer* melalui proses pembayaran sesuai dengan pengiriman dan kondisi pembayaran yang ditentukan oleh kebutuhan bisnis.
- Memberi *customer* kemampuan untuk mengelola profil dan pesanan mereka.
- Membantu dalam manajemen *order* dan pembayaran.
- Memberi pemilik *vending machine* sarana untuk menganalisis dan melacak aktivitas di *platform online*.

#### 1) Pandangan Holistic

Pandangan ini memuat proses bisnis beserta aktor dan teknologi yang digunakan dalam membangun arsitektur *vending machine*.

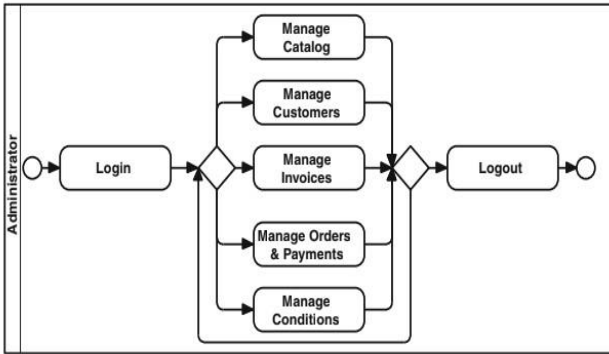


Gambar 1. Pandangan holistic [6]

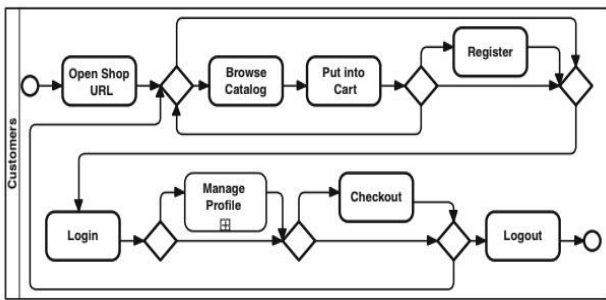
Gambar 1 menunjukkan bahwa kotak yang berada di tengah (*catalog*, *order*, *customer*, dan *conditions*) merupakan data dan lapisan aplikasi. *Administrator* dapat mengaksesnya melalui kotak yang *vending machine backend*, sedangkan *customer* dapat mengakses hal yang sama melalui kotak *vending machine frontend*. Masing-masing bagian menyediakan antarmuka dan layanan khusus yang dapat diakses melalui *frontend* dan *backend*.

## 2) Proses Bisnis

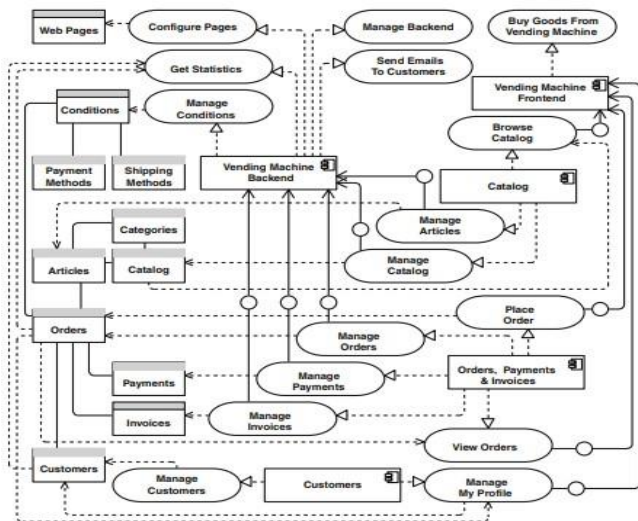
Gambar 2 dan 3 menunjukkan proses bisnis yang memuat keseluruhan aktivitas yang dibutuhkan dalam mengelola *vending machine* [6]:



Gambar 2. Proses bisnis administrator [6]



Gambar 3. Proses bisnis customer [6]



Gambar 4. Struktur data dan aplikasi [6]

Gambar 4 menunjukkan objek data terpenting di sisi kiri dan koneksi antar komponen aplikasi di sisi kanan. Setiap aktivitas proses bisnis mempunyai layanan yang sesuai dengan lapisan aplikasi. *customer* hanya dapat mengakses dua layanan yakni: *manage my profile* dan *buy goods from vending machine* yang memungkinkan operasi penulisan data objek, *place order*, dan *manage my profile* (alamat, kata sandi dan nama pengguna). Pengelolaan dalam *vending machine* juga termasuk *configures pages* yang memungkinkan membuat dan mengubah konten web, seperti halaman informasi, jam buka, serta informasi kontak. *Administrator*

juga bisa mendapatkan statistik tentang kinerja *vending machine* (contoh: jumlah penjualan produk dalam suatu bulan, serta informasi keuangan yang akan diteruskan ke aplikasi finansial). Dan dalam struktur data dan aplikasi katalog sering kali disusun dalam kategori. Suatu produk mungkin berbeda kategori, seperti, sebuah sepatu akan berada dalam kategori model sepatu yang berbeda. serta terdapat beberapa Ketentuan yang dipilih oleh *customer* seperti apa metode pengiriman dan pembayaran, yang akan digunakan dalam proses pengiriman dan pembayaran.

## C. Teknologi Microservices

Tujuan utama teknologi *microservices* adalah memecah dan merangkai aplikasi menjadi layanan-layanan kecil. setiap layanan memiliki tanggung jawab masing-masing, tetapi dibuat tetap saling berkomunikasi. Sistem pada *microservices* juga didistribusikan secara independen yang hasilnya setiap layanan dapat beradaptasi dengan setiap perubahan kebutuhan [4].

Dalam setiap upaya penerapan teknologi tentunya tidak terlepas dari kelemahan dan kelebihan pada saat digunakan. Berikut merupakan beberapa kelemahan dalam penerapan teknologi *microservices*, kesulitan dalam membuat komunikasi antar layanan dan mengatasi kegagalan di sebagian layanan serta menerapkan permintaan yang menjangkau banyak layanan yang membutuhkan koordinasi yang baik antar layanan. Serta penggunaan memori yang jauh lebih tinggi dan biaya yang di keluarkan menjadi lebih besar. Sedangkan kelebihan yang diungkapkan langsung oleh responden pada survei yang dilakukan Oreilly. Mengatakan bahwa mayoritas (45%) responden senang dengan fleksibilitas fitur. Diikuti dengan responden (di bawah 45%) mengatakan bahwa *microservices* merespons dengan cepat terhadap perubahan teknologi dan kebutuhan bisnis. Lanjut di bawah 44% yang mengatakan bahwa manfaat dari skalabilitas keseluruhan yang lebih baik. Serta manfaat yang paling sedikit di kutip (15%) adalah biaya pengembangan yang lebih rendah [9].

Teknologi *microservices* sendiri memiliki 2 struktur komposisi. Yaitu koreografi dan orkestrasi. Yang dimaksud orkestrasi merupakan proses bisnis tunggal yang dieksekusi secara terpusat dengan mengkoordinasikan hubungan antar layanan, sedangkan koreografi merupakan keseluruhan layanan yang berpartisipasi dengan menentukan pertukaran pesan dan aturan interaksi dengan dua atau lebih keluaran serta menggunakan pendekatan desentralisasi dalam komposisi layanan [11].

## D. Penelitian Serupa

Kajian Pustaka ini mengkaji 4 penelitian serupa yang menggunakan *microservices*. Penelitian [12] membahas desain dan Implementasi *microservices* studi kasus pada layanan *taking order* ( aplikasi *e-commerce* PT. Xyz ). Sulitnya bekerja dengan pengembang lain dikarenakan masih menggunakan arsitektur *monolithic* menjadi alasan utama beralih ke *microservices*. Penelitian [13] membahas mengenai *refactoring* arsitektur *microservices* pada aplikasi absensi PT. Graha Usaha Teknik penelitian ini dilakukan karena sulitnya *maintenance* menggunakan *monolithic*. Penelitian [14] membahas mengenai implementasi arsitektur *microservices* pada aplikasi *online travel* tourinc untuk mengatasi kompleksitas yang besar dan memilih beralih ke *microservices*. Penelitian [15] membahas aplikasi pengelolaan skripsi di STMIK AKAKOM menggunakan

arsitektur *microservices* dengan *node.js* agar *maintenance* yang dilakukan lebih mudah.

Dari seluruh penelitian serupa, belum ada pengimplementasian *microservices* yang didasari oleh arsitektur enterprise. Maka dari itu, penelitian ini bermaksud mengimplementasikan *microservices* menggunakan pola yang didasari oleh arsitektur.

### III. METODOLOGI PENELITIAN

#### A. Kajian Pustaka

Dalam kajian pustaka terdapat 3 konten utama yang terdiri dari arsitektur enterprise, Pola arsitektur *vending machine*, dan teknologi *microservices*. Seluruh bagian dari konten utama dan *vending machine* dikaji dan digali lebih dalam melalui beberapa literatur resmi serta buku [6].

#### B. Perancangan Arsitektur Enterprise Vending Machine

Dalam melakukan perancangan arsitektur enterprise *vending machine* digunakan pola yang berdasar pada buku [6]. Pola tersebut kemudian dimodifikasi sesuai kebutuhan mengikuti kasus *vending machine* (toko *online*) terutama pada ranah arsitektur data untuk menyesuaikan kebutuhan proses bisnis yang mengarah kepada teknologi *microservices*.

#### C. Pengembangan layanan API

Layanan API menggunakan pola arsitektur enterprise dari *vending machine* yang sudah di disesuaikan dengan kebutuhan *microservices*. Dalam pengembangan API layanan penerapan pola-pola *microservices* dilakukan, yaitu: pola saga untuk menjaga konsistensi data antar layanan, serta menggunakan pola orkestra yang memiliki kelonggaran antar layanan yang membuat perkembangan lebih cepat serta konsisten dan efisien.

#### D. Implementasi

Beberapa teknologi yang digunakan dalam implementasi *vending machine*.

##### 1) Node.js

Pengembangan *backend* pada *vending machine* dilakukan dengan menggunakan *Node.js* yang merupakan *platform* yang bekerja pada sisi *server* untuk menjalankan kode *JavaScript* sebelum ditampilkan di *browser*. Cocok digunakan untuk komunikasi *real time* dari *client* dan *server* arsitektur bisnis yang mendeskripsikan proses bisnis organisasi.

##### 2) MongoDB

pengembangan struktur basis data pada *vending machine* dilakukan dengan menggunakan *mongoDB*. *MongoDB* merupakan basis data yang menggunakan sistem non relasional. Sistem penyimpanan data pada *mongoDB* menggunakan dokumen yang terstruktur seperti *JSON*.

##### 3) React.js

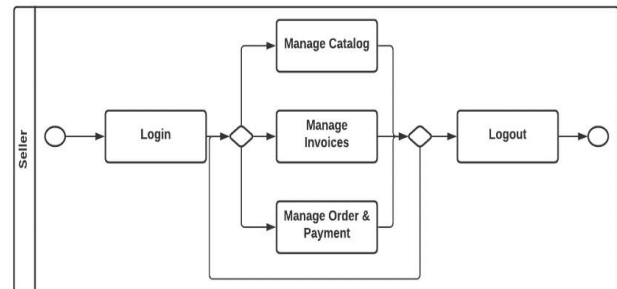
Pengembangan antarmuka pada *vending machine* dilakukan dengan menggunakan *React.js*. *React.js* merupakan sebuah *library JavaScript* yang sifatnya *opensource*. *React* membuat *User Interfaces* interaktif agar dengan mudah membuat desain yang sederhana untuk setiap *state* di dalam aplikasi.

## IV. HASIL

### A. Arsitektur Aktual

Dalam proses implementasi terdapat penambahan proses bisnis aktor yaitu *seller* untuk manajemen produk yang dibuat oleh *seller* dan manajemen *order* dari produk yang dibuat oleh *seller*. Serta terdapat sebuah layanan yang tidak diimplementasikan yaitu *Send email to customer*.

#### 1) Proses Bisnis



Gambar 4. Proses bisnis *seller*

Gambar 5 memuat bagian penambahan proses bisnis aktor yaitu *seller*. Fungsi dari *seller* sendiri adalah manajemen khusus produk yang telah *seller* buat beserta *order* yang masuk.

#### 2) Aplikasi dan Layanan

##### a) Manage Product

Layanan ini menggabungkan layanan *manage catalog* dan *manage articles* karena fungsinya yang saling berkaitan, dimana di dalam sebuah *catalog product* biasanya telah tersedia harga produk, jumlah produk yang tersedia serta deskripsi dari sebuah produk. Deskripsi produk termasuk ke dalam *articles*.

##### b) Manage Order

layanan ini menggabungkan layanan *manage order* dan *manage invoices* karena fungsi yang digunakan sama dengan *manage order* yaitu melihat keseluruhan informasi *order*.

### B. Basis Data

Dari pola yang sudah diberikan dari awal terjadi beberapa penggabungan beberapa objek data dalam basis data. Berikut TABEL I yang menyediakan uraian beberapa basis data yang masing-masing berdiri sendiri di tiap layanannya.

TABEL I. BASIS DATA DI TIAP LAYANAN

Layanan	Basis data
Manage Product	Product
Browse Product	
Get Statistic	
Place Order	Order
Manage Order	
View Order	User
Manage User	
Manage My Profile	

#### a) Product

Basis data ini berisikan keseluruhan informasi produk mulai dari nama, harga, gambar, berat, kategori merek, jumlah stok serta deskripsi produk.

#### a) Order

Basis data ini berisikan seluruh informasi pemesanan mulai dari *invoices*, *payment*, *payment method* serta *shipping method*.

#### a) User

Basis data ini berisikan informasi keseluruhan pengguna mulai dari *admin*, *customer* dan juga *seller*.

### C. Layanan API

Terdapat 8 layanan API yang dihasilkan. Seluruh layanan diimplementasikan menggunakan Node.js dan Axios untuk saling berkomunikasi. Berikut 8 layanan tersebut.

#### 1) Layanan Manage Product

Layanan ini ditujukan untuk memajemen seluruh produk yang memiliki fungsi yaitu:

- a) *Create product*,
- b) *View product*,
- c) *Edit product*,
- d) *Delete product*.

#### 2) Layanan Place Order

Layanan ini memasukkan produk yang akan dipesan ke dalam keranjang beserta alamat pemesan dan pemilihan metode pembayaran yang akan digunakan. Fungsi *pada* layanan *place order* adalah *create order*.

#### 3) Layanan Manage Order

Pada bagian layanan ini ditujukan untuk memeriksa informasi order apakah transaksi telah dilakukan oleh pemesan produk, apabila produk tersebut telah dilunasi maka *admin/seller* terkait dapat mengirimkan produk, serta pengguna mendapatkan informasi bahwa produk yang dipesan telah dikirim. Berikut fungsi dari layanan *manage order*:

- a) *Pay via client id paypal*,
- b) *Update payment*,
- c) *Update deliver order*.

#### 4) Layanan View order

Layanan ini digunakan pengguna untuk melihat pesanan yang telah dibuatnya. *Admin* dapat melihat dan menghapus keseluruhan pesanan yang telah dibuat oleh pengguna. Berikut fungsi dari layanan *view order*:

- a) *View all order*,
- b) *Delete order*.

#### 5) Layanan Manage User

Layanan ini ditujukan untuk *admin* memajemen pengguna ataupun mengubah status pengguna menjadi *admin*. Berikut fungsi dari layanan *manage customer*:

- a) *View all User*,
- b) *Delete user*,
- c) *Update status user*.

#### 6) Layanan Manage My Profile

Layanan ini ditujukan untuk menyimpan segala informasi akun yang dibutuhkan pengguna mulai dari nama, *email*, hingga *password*, apabila pengguna belum mempunyai akun maka pengguna harus membuat akunnya terlebih dahulu. Apabila pengguna telah memiliki akun pengguna hanya perlu masuk menggunakan akun yang telah ada. Berikut fungsi dari layanan *manage my profile*:

- a) *Sign in*,
- b) *Register*,
- c) *View Info User*,
- d) *Edit Info User*.

#### 7) Layanan Browse Product

Layanan ini untuk memudahkan pengguna dalam mencari produk yang diinginkan sesuai dengan nama, penjual, harga, kategori, maupun rating produk. Fungsi dari layanan *browse product* adalah *filter product*.

#### 8) Layanan Get Statistic

Untuk mengetahui kualitas produk yang dijual maka ulasan dari tiap produk diperlukan melalui *customer* yang telah memesan produk terlebih dahulu. Serta informasi mengenai jumlah penjualan produk yang masuk per bulan dapat dilihat pada layanan ini. Berikut fungsi yang ada dalam layanan *get statistic*:

- a) *Review Product*,
- b) *View sales*.

### D. Antarmuka

Antarmuka dikembangkan menggunakan React.js serta menggunakan state untuk menyimpan keadaan pada suatu tempat untuk menjaga agar apa yang telah di input pada halaman sebelumnya tidak hilang. Antarmuka berisi interaksi antar layanan. Berikut merupakan antarmuka dari tiap layanan.

Antarmuka dari *manage product* yang ditunjukkan pada Gambar 6 digunakan untuk menambahkan dan merubah informasi produk. Antarmuka dari layanan *place order*, ditunjukkan pada Gambar 7 antarmuka ini ditujukan untuk membuat pesanan dengan memasukkan informasi yang dibutuhkan penjual, yaitu: produk dan jumlah produk yang dipesan, alamat pengiriman, metode pengiriman, metode pembayaran, dan menampilkan *invoice*. Antarmuka dari layanan *manage order*, ditunjukkan pada Gambar 8 antarmuka ini ditujukan untuk *admin* memajemen pengiriman dan melihat pembaruan pembayaran. Antarmuka dari layanan *view order*, ditunjukkan pada Gambar 9 antarmuka ini ditujukan untuk melihat keseluruhan order.

Antarmuka dari layanan *manage user*, ditunjukkan pada Gambar 10 antarmuka ini ditujukan untuk *admin* memajemen keseluruhan pengguna. Antarmuka dari layanan *manage my profile*, ditunjukkan pada Gambar 11 antarmuka ini digunakan untuk melihat dan merubah informasi pengguna. Antarmuka dari layanan *browse product*, ditunjukkan pada Gambar 12 antarmuka ini digunakan pengguna untuk melihat produk berdasarkan harga, kategori produk, dan produk terbaru.

ID	NAME	PRICE	WEIGHT	CATEGORY	BRAND	ACTIONS
6061b6c28f8613b6c054ee	All Star	300	1	Sneakers	Converse	Edit Delete
605f7945bd23e43d96b2f2b	Superstar	320	2	Sneakers	Adidas	Edit Delete
6058695bd23e43d96b2f2b	Tubular Shadow	405	3	Sports	Adidas	Edit Delete
6058633bd23e43d96b2f27	Cortez	380	3	Sports	Nike	Edit Delete
60585ee5bd23e43d96b2f26	Air Jordan Blue	500	4	Sports	Nike	Edit Delete
605859f5bd23e43d96b2f25	Proto Low Reissue 1	400	2	Sneakers	Compass	Edit Delete
605856e5bd23e43d96b2f24	Gazelle High Blue Sky	250	2	Sneakers	Compass	Edit Delete
60585375bd23e43d96b2f23	Vintage 98 High Grey	250	2	Sneakers	Compass	Edit Delete
60584865bd23e43d96b2f22	Flocking Compass Versi 1	450	3	Sneakers	Compass	Edit Delete

Gambar 6. Manage product

**Shipping**  
 Name: asd  
 Address: Jalan kalurang, Yogyakarta, 78113, Diy  
 Method: JNE

**Payment**  
 Method: PayPal

**Order Summary**  
 Item: \$320.00  
 Shipping: \$4.12  
 Tax: \$0.32  
 Total: \$324.44

**Order Item**  
 Superstar  
 Item: QTY 1 x Price \$320 = \$320  
 Shipping: Weight 2kg x \$2.06 = \$4.12

Gambar 7. Place order

**Order Summary**  
 Items: \$300.00  
 Shipping: \$2.06  
 Tax: \$0.30  
 Order Total: \$302.36

**Shipping**  
 Name: asd  
 Address: Jalan kalurang, Yogyakarta, 78113  
 Method: JNE  
 Not Delivered

**Payment**  
 Method: PayPal  
 Paid at 2021-05-31T06:33:23.244Z

**Order Items**  
 All Star  
 Item: QTY 1 x Price \$300 = \$300  
 Shipping: Weight 1kg x \$2.06 = \$2.06

Gambar 8. Update deliver order and update payment

ID	USER	DATE	TOTAL	PAID	DELIVERED	ACTIONS
605964e89688b31481e95b	604c2f6a2cce1e0d80b4c88e	2021-04-04	386.56	2021-05-04	No	Details Delete
605a9518834e6134468b071	604c2f6a2cce1e0d80b4c88e	2021-04-05	302.36	2021-04-05	2021-04-05	Details Delete
60b21481fcb0de1c385588c	604c2f6a2cce1e0d80b4c88e	2021-05-29	508.74	No	No	Details Delete
60b2159afcb0de1c385588e	604c2f6a2cce1e0d80b4c88e	2021-05-29	324.44	No	No	Details Delete
60b4708b37542a36a8ca298e	604c2f6a2cce1e0d80b4c88e	2021-05-31	302.36	2021-05-31		Details Delete

Gambar 9. View order

ID	NAME	EMAIL	IS SELLER	IS ADMIN	ACTIONS
604c2f6a2cce1e0d80b4c88e	Iqbal	admin@gmail.com	YES	YES	Edit Delete
604c2f6a2cce1e0d80b4c88f	Hanif	user@gmail.com	NO	NO	Edit Delete
605460422ce04e0c8b0b0c5	Hanif	Hanif@gmail.com	YES	YES	Edit Delete
60589a8d3e01141e14e00	Apis	apis@gmail.com	NO	NO	Edit Delete
6058a98146762571c1cc3c	qwerty	qwerty@gmail.com	YES	YES	Edit Delete
605813c394e44e48a4b05	Compass	Compass@gmail.com	YES	NO	Edit Delete
6058155394e44e48a4b07	Nike	Nike@gmail.com	YES	NO	Edit Delete
605816a394e44e48a4b08	Addas	Addas@gmail.com	YES	NO	Edit Delete
60587e45bd23e43d96b2f2c	Converse	Converse@gmail.com	YES	NO	Edit Delete

Gambar 10. Manage user

**User Profile**

Name: Iqbal

Email: admin@gmail.com

Password: Enter password

confirm Password: Enter confirm password

Gambar 11. View and edit info user

6 Results

Sort by: Newest Arrivals

**Categories**  
 Any  
 Sneakers  
 Sports

**Price**  
 Any  
 \$10 to \$100  
 \$100 to \$1000

**Avg. Customer Review**  
 ★★★★★ & up  
 ★★★★☆ & up  
 ★★★☆☆ & up  
 ★★☆☆☆ & up  
 ★☆☆☆☆ & up

**All Star**  
 ★★★★★ 1 reviews  
 \$300

**Superstar**  
 ★★★★★ 1 reviews  
 \$320

**Proto Low Reissue 1**  
 ★★★★★ 0 reviews  
 \$400

Gambar 12. Filter Product

### E. Pengujian

Pengujian dilakukan untuk mengetahui apakah proses bisnis berjalan dengan baik. Pengujian proses bisnis dilakukan berdasarkan sudut pandang pengguna masing-masing yaitu *Admin*, *seller* dan juga *customer*. Pengujian dilakukan terhadap semua fungsional layanan API sesuai dengan teknologi *microservices*. Apabila keluaran dari tiap fungsionalitas belum menemui harapan, maka dilakukannya perbaikan dan diuji ulang hingga tiap keluaran fungsionalitas layanan API dirasa sesuai dengan harapan dan tujuan *microservices*.

Berdasarkan hasil pengujian yang dilakukan untuk mengecek API tiap layanan dan bentuk data menggunakan postman sudah sesuai dengan harapan. Begitu pula dengan pengujian antarmuka dengan menggunakan *web browser*, keluaran dan juga interaksi dari antarmuka telah sesuai dengan proses bisnis dan sesuai dengan harapan.

## V. PEMBAHASAN

Rancangan yang setara dengan pola arsitektur enterprise *vending machine* dibutuhkan saat mengembangkan aplikasi berbasis *microservices*. Untuk memastikan arsitektur dirancang secara bertahap untuk meyakinkan bahwa kebutuhan bisnis enterprise dibantu sepenuhnya oleh sistem teknologi informasi enterprise mulai dari ranah arsitektur data, aplikasi hingga teknologi. Untuk mendukung kebutuhan bisnis enterprise oleh sistem yang akan dibangun. Yang menghasilkan pemodelan di tiap ranah terperinci. Terutama pemilihan teknologi menggunakan *microservices* yang mempermudah pengembangan. Karena tiap layanan dapat dikembangkan tanpa mengganggu layanan lain, serta tidak ada hambatan untuk menggunakan teknologi baru. Hal ini mampu memberikan dampak positif terutama dalam kasus yang diangkat berupa toko *online*. Yaitu dapat meningkatkan penjualan, dan mempermudah *customer* dalam membeli barang, serta terbentuknya proses bisnis lebih efisien.

## VI. KESIMPULAN

Setelah dilakukan pengujian implementasi pola arsitektur enterprise *vending machine* dalam kasus konkret berupa toko *online* pada teknologi *microservices* dapat disimpulkan sebagai berikut:

- Rancangan aplikasi yang setara dengan pola arsitektur enterprise *vending machine* dibutuhkan saat mengembangkan aplikasi berbasis *microservices*.
- Pengembangan pemodelan pola arsitektur enterprise *vending machine* dapat menghasilkan arsitektur bisnis, data, serta teknologi yang terperinci.
- Pola arsitektur *vending machine* dapat membantu proses bisnis baru dalam menjual ataupun membeli produk di internet. Serta meningkatkan penjualan dan mempermudah *customer* dalam membeli barang.

## DAFTAR PUSTAKA

- [1] Osvalds, Gundars. (2001). Definition of Enterprise ArchitectureCentric Models for the Systems Engineer. INCOSE International Symposium. 11. 10.1002/j.2334-5837.2001.tb02278.x.
- [2] H. Shah and M. El Kourdi, "Frameworks for Enterprise Architecture," in *IT Professional*, vol. 9, no. 5, pp. 36-41, Sept.-Oct. 2007, doi: 10.1109/MITP.2007.86.
- [3] Huston Tom, 2015, What are microservice?, <http://smartbear.com/allresources/articles/what-is-microservices-architecture/>.
- [4] James Lewis, and Martin Fowler. "Microservices." *Martinfowler.com*, 25 Mar. 2014, [www.martinfowler.com/articles/microservices.html](http://www.martinfowler.com/articles/microservices.html).
- [5] H. Shah and M. El Kourdi, "Frameworks for enterprise architecture," *IT Prof.*, vol. 9, no. 5, pp. 36-41, 2007, doi: 10.1109/MITP.2007.86.
- [6] T. Perroud and R. Inversini, *Enterprise Architecture Patterns*. 2013.
- [7] Brambilla, Marco. (2018). Modeling, Modeling, Modeling: From Web to Enterprise to Crowd to Social. 10.1007/978-3-319-61893-7\_14.
- [8] C. Richardson, "Developing event-driven microservices with event sourcing and CQRS svcc 2015.key."
- [9] "Microservices Adoption in 2020 – O'Reilly." <https://www.oreilly.com/radar/microservices-adoption-in-2020/> (accessed May 27, 2021).
- [10] (2006). Mobile Information Systems-Infrastructure and Design for Adaptivity and Flexibility (the MAIS approach). *Recherche*. 67. 02. 10.1007/3-540-31008-8.
- [11] T. Cerny, "Aspect-oriented challenges in system integration with microservices, SOA and IoT," *Enterp. Inf. Syst.*, vol. 13, no. 4, pp. 467-489, 2019, doi: 10.1080/17517575.2018.1462406.
- [12] S. D. Putra and K. Saputra, "Desain Dan Implementasi Microservices Studi Kasus Pada Layanan Taking Order ( Aplikasi E-Commerce Pt Xyz)," 2019.
- [13] R. Mufrizal and D. Indarti, "Refactoring Arsitektur Microservice Pada Aplikasi Absensi PT. Graha Usaha Teknik," *J. Nas. Teknol. dan Sist. Inf.*, vol. 5, no. 1, pp. 57-68, 2019, doi: 10.25077/teknosi.v5i1.2019.5768.
- [14] M. D. Rafiqi, E. Subyantoro, and D. K. W, "Implementasi Arsitektur Microservice Pada Aplikasi Online Travel Tourinc," no. 1, pp. 1-10.
- [15] I. Y. B. Purnama, Heri, "APLIKASI PENGELOLAAN SKRIPSI DI STMIK AKAKOM YOGYAKARTA MENGGUNAKAN ARSITEKTUR MICROSERVICE DENGAN Node.js," pp. 1-28, 2010, [Online]. Available: <http://perpus.akakom.ac.id/>.