

**APLIKASI PESAN ANTAR MAKANAN MENGGUNAKAN
KONSEP ONLINE FOOD SERVICE DI WILAYAH POLITEKNIK
POS INDONESIA
(MODUL: BACK-END WEB API)**

LAPORAN TUGAS AKHIR

Diajukan untuk memenuhi kelulusan matakuliah Tugas Akhir

Pada Program Studi DIII Teknik Informatika

Disusun Oleh :

Nama : MUHAMMAD AFIEF FARISTA

NPM : 1143012



**PROGRAM DIPLOMA III TEKNIK INFORMATIKA
POLITEKNIK POS INDONESIA
BANDUNG
2017**

DAFTAR ISI

DAFTAR ISI
BAB I PENDAHULUAN	I-1
1.1 Latar Belakang.....	I-1
1.2 Identifikasi Masalah	I-2
1.3 Tujuan.....	I-2
1.4 Ruang Lingkup	I-3
1.5 Sistematika Penulisan	I-3
BAB II LANDASAN TEORI.....	II-1
2.1 Definisi Food Service	II-1
2.2 Definisi Politeknik Pos Indonesia	II-1
2.3 Definisi Aplikasi.....	II-2
2.4 Definisi <i>Laravel</i>	II-2
2.5 Definisi MySQL	II-3
2.6 Definisi Application Programming Interface (API)	II-4
2.7 Definisi Business Process Modelling Notation	II-5
2.8 Definisi Pengujian Sistem	II-5
2.9 Definisi Analisis Sistem	II-6
2.10 Definisi Desain Sistem	II-6
BAB III ANALISIS DAN PERANCANGAN	III-1
3.1 Analisis Sistem	III-1
3.1.1 Analisis Sistem Berjalan.....	III-2
3.1.2 Analisis Sistem yang akan Dibangun	III-4

3.2 Perancangan Sistem.....	III-25
3.2.1 Component Diagram	III-25
3.2.2 Deployment Diagram	III-27
BAB IV IMPLEMENTASI DAN PENGUJIAN.....	IV-1
4.1 Lingkungan Implenentasi	IV-1
4.2 Pengujian dan Hasil Pengujian	IV-3
BAB V PENUTUP	V-1
5.1 Kesimpulan.....	V-1
5.2 Saran	V-1
DAFTAR PUSTAKA	

DAFTAR GAMBAR

Gambar 2.1 Cara kerja <i>API</i> [5]	II-4
Gambar 3.1 BPMN sistem yang sedang berjalan	III-3
Gambar 3.2 BPMN Sistem yang akan dibangun	III-6
Gambar 3.3 <i>Use Case Diagram FOD</i>	III-9
Gambar 3.4 <i>Class Diagram FOD</i>	III-11
Gambar 3.5 <i>Sequence Diagram FOD</i>	III-14
Gambar 3.6 <i>Collaboration Diagram FOD</i>	III-17
Gambar 3.7 <i>Activity Diagram FOD</i>	III-20
Gambar 3.8 Statechart Diagram FOD	III-23
Gambar 3.9 Component diagram FOD	III-25
Gambar 3.10 Deployment diagram FOD	III-28

DAFTAR TABEL

Tabel 3.1 Tabel keterangan BPMN sistem sedang berjalan	III-4
Tabel 3.2 Tabel keterangan BPMN	III-7
Tabel 3.3 Tabel keterangan Use Case Diagram	III-10
Tabel 3.4 Tabel keterangan Class Diagram	III-12
Tabel 3.5 Tabel keterangan Sequence Diagram	III-15
Tabel 3.6 Tabel keterangan Collaboration Diagram	III-18
Tabel 3.7 Tabel keterangan Activity Diagram FOD	III-21
Tabel 3.8 Tabel keterangan statechart diagram FOD	III-24
Tabel 3.9 Tabel keterangan Component diagram FOD	III-26
Tabel 3.10 Tabel keterangan deployment diagram FOD	III-29
Tabel 4.1 Tabel spesifikasi perangkat lunak untuk server-side application	IV-2
Tabel 4. 2 Tabel spesifikasi perangkat keras server	IV-2
Tabel 4.3 Daftar url yang dapat diakses oleh pihak front-end	IV-4

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam kehidupan manusia, teknologi dan informasi berkembang setiap saat. Teknologi dan informasi tersebut diterapkan dalam berbagai aspek dalam kehidupan manusia, terutama di bidang industri. Alhasil, manusia dapat memanfaatkan teknologi dan informasi tersebut kapan saja dan di mana saja selama terhubung melalui suatu media. Salah satu dari media tersebut adalah internet.

Transaksi data pesan antar makanan di Politeknik Pos Indonesia merupakan proses pengolahan terhadap data pesan antar makanan yang terjadi di perguruan tinggi tersebut. Saat ini, transaksi data tersebut dilakukan secara manual tanpa bantuan suatu teknologi terkomputerisasi sehingga memiliki rasio redundansi data yang besar. Adapun data yang dimaksud adalah data historis dari siapa yang memesan dan membeli, siapa yang mengantarkan pesanan tersebut, berapa ongkos kirimnya, dan kapan terjadinya proses pesan antar makanan tersebut.

Terkait dengan transaksi data, proses pesan antar makanan membutuhkan pengelolaan data secara langsung (*realtime*). Walaupun pengelolaan banyak data dalam satu waktu yang sama tidak mustahil namun hal tersebut merupakan suatu tugas yang dilakukan oleh suatu sistem terkomputerisasi sehingga terhindar dari kekeliruan pengelola dalam pengelolaan data atau disebut sebagai *human error*.

Berdasarkan uraian di atas, dengan menggunakan tema pesan antar makanan secara online, akan dibuat ***Aplikasi Pesan Antar Makanan Menggunakan Konsep Online Food Delivery Di Wilayah Politeknik Pos Indonesia (Modul: Back-end Web API)***. Aplikasi ini diberi kode nama FOD yang mana merupakan singkatan dari *Food Online Delivery*. Adapun aplikasi tersebut merupakan salah satu teknologi

terkomputerisasi yang dibuat untuk memberikan solusi terhadap permasalahan yang dimaksud.

1.2 Identifikasi Masalah

Berdasarkan latar belakang pada sub-bab sebelumnya, adapun identifikasi masalah mengenai topik yang akan dibahas adalah sebagai berikut:

1. belum ada teknologi terkomputerisasi yang memiliki fungsi utama untuk memberikan pelayanan pesan antar makanan di wilayah Poltekpos;
2. transaksi data dalam proses bisnis pesan antar makanan di Politeknik Pos Indonesia saat ini belum menggunakan sistem terkomputerisasi sehingga memiliki redundansi data dalam rasio yang besar, baik itu data transaksi, data identitas pelanggan, data identitas pelayan, ataupun data identitas kurir;
3. pemesanan manual dalam jumlah yang banyak akan menghasilkan antrian pesanan yang panjang, adapun penerimaan pesanan yang dilakukan secara manual tanpa menggunakan sistem terkomputerisasi saat ini tidak dilakukan secara paralel sehingga proses bisnis bersifat *single tasking*.

1.3 Tujuan

Berdasarkan latar belakang dan identifikasi masalah pada sub-bab sebelumnya, maka tujuan yang akan dicapai dengan topik yang akan dibahas adalah sebagai berikut:

1. menciptakan suatu teknologi terkomputerisasi yang memiliki fungsi khusus untuk dapat memberikan pelayanan pesan antar makanan kepada konsumen untuk wilayah Poltekpos;
2. memberikan teknologi *back-end API* terhadap manajemen data dalam bisnis pesan antar makanan, adapun data yang dimaksud adalah data transaksi, data pengguna, data kurir, dan data pelanggan.
3. membuat suatu sistem terkomputerisasi yang dapat memberikan solusi terhadap antrian panjang dalam proses bisnis pesan antar makanan di wilayah Poltekpos

dengan menciptakan suatu aplikasi yang dapat melakukan manajerial yang secara relatif bersifat *multitasking*.

1.4 Ruang Lingkup

Berikut adalah batasan-batasan yang perlu diperhatikan dalam perancangan Aplikasi Pesan Antar Makanan Menggunakan Konsep Online Food Service di Wilayah Politeknik Pos Indonesia:

1. aplikasi *Back-end API* ini digunakan oleh *front-end application* yang mendukung sistem operasi pada umumnya seperti: Android, iOS, Windows, ataupun MacOS, yang artinya *Back-end API* ini bersifat *accessible* selama itu terhubung dengan internet;
2. data-data yang akan dikelola meliputi: data *users* berupa data konsumen dan data produsen, dan data makanan/minuman;
3. adapun konsumen yang dimaksud terbatas untuk wilayah Politeknik Pos Indonesia;
4. *back-end API* dapat diakses dengan menggunakan sebuah *web browser*;
5. pengembangan menggunakan konsep dasar *back-end* dan API;
6. tidak ada aktor yang dapat mengakses API, API hanya dapat diakses oleh aplikasi *front-end*.

1.5 Sistematika Penulisan

Dalam penyusunan maupun pembuatan laporan ini telah terbagi menjadi beberapa bab beserta sub-judul yang ada di dalamnya dengan tujuan untuk mempermudah dalam pencarian data atau informasi, serta membuktikan bahwa laporan ini diselesaikan dengan sistematis dan terstruktur. Berikut ini adalah pembagian bab yang terdapat dalam penyusunan laporan ini.

Bab I Pendahuluan, dalam pendahuluan hal-hal yang dibahas yaitu latar belakang yang berisi ulasan ringkas mengenai keadaan atau kondisi yang ada atau telah

terjadi, dan kekurangan dari sistem/aplikasi yang diamati sehingga muncul topik yang diambil. Selain itu membahas identifikasi masalah yang berisi berbagai masalah yang sudah dikenali dan akan diberikan solusinya melalui fungsi dari sistem atau aplikasi yang dibuat. Pada bab ini juga membahas tentang tujuan dari mengapa sistem atau aplikasi ini dibuat, serta ruang lingkup yang berisi batasan-batasan atau cakupan yang akan dibangun atau dibuat, dan juga sistematika penulisan yang di dalamnya menjelaskan isi dari laporan proyek.

Bab II Landasan Teori, berisi pembahasan mengenai aplikasi, *back-end*, dan API. Kemudian uraian tentang teori yang berasal dari beberapa sumber terpercaya dan pengertian program aplikasi yang mendukung terbentuknya “Aplikasi Pesan Antar Makanan Menggunakan Konsep Food Service di Wilayah Politeknik Pos Indonesia”.

Bab III Analisis dan Perancangan, yang membahas tentang proses dalam menentukan kebutuhan sistem atau aplikasi yang akan dibuat pada saat membangun maupun pada saat implementasi. Adapun perancangan yang membahas bagaimana sistem atau aplikasi tersebut akan dibuat dengan membuat diagram alur atau *flowchart*.

Bab IV Implementasi dan Pengujian, bab ini berisi tentang implementasi dari sistem atau aplikasi yang dibuat dengan merinci komponen-komponen pendukung berupa program, lingkungan implementasi, tampilan antarmuka/interface, petunjuk pemakaian, dan petunjuk aplikasi. Pada pengujian, berisi cara untuk mengetahui apakah sistem atau aplikasi yang dibuat sesuai dengan rancangan dan menuliskan hasil ujinya.

Bab V Kesimpulan dan Saran, bab ini berisi tentang kesimpulan yang didapatkan dari sistem atau aplikasi yang dibuat serta saran-saran yang berisi halaman atau tujuan maupun kondisi implementasi yang berkaitan dengan sistem atau aplikasi yang dibuat demi mendapatkan hasil yang sempurna ataupun tercapainya tujuan dari sistem dan aplikasi tersebut.

BAB II

LANDASAN TEORI

2.1 Definisi Food Service

Food and beverage service adalah “*food flow*” (dari pembelian makanan (*purchasing*) untuk layanan kepada pelanggan) terutama berkaitan dengan pengiriman dan penyajian makanan untuk pelanggan, setelah selesainya produksi pangan. Pada umumnya melibatkan transportasi jika ada pemisahan fasilitas produksi dan pelayanan; [3]

Pelayanan makanan dan minuman adalah salah satu bagian di hotel atau restoran yang memberikan pelayanan makanan dan minuman kepada tamu hotel atau tamu restoran. Pelayanan makanan dan minuman ini biasa disebut dengan tata hidangan atau *food and beverages services*; [11]

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *food service* adalah suatu seni dalam penyajian makanan yang pada umumnya melibatkan transportasi untuk memberikan pelayanan makanan dan minuman kepada pengguna layanan tersebut serta memberikan kepuasan terhadap pengguna layanan tersebut.

2.2 Definisi Politeknik Pos Indonesia

Politeknik Pos Indonesia merupakan institusi pendidikan tinggi yang didirikan oleh Yayasan Pendidikan Bhakti Pos Indonesia (YPBPI), pada tanggal 5 Juli 2001 berdasarkan SK Mendiknas No. 56/D/O/2001; [4]

Politeknik Pos Indonesia merupakan lembaga pendidikan tinggi vokasional dengan program Diploma III (Ahli Madya) dan Diploma IV (Sarjana Terapan) dimana lulusannya telah dipersiapkan untuk dapat langsung berperan dalam tugas-tugas operasional dan manajerial di industri (bisnis) logistik; [4]

Berdasarkan uraian tersebut, dapat disimpulkan bahwa Politeknik Pos Indonesia merupakan institusi pendidikan tinggi yang memiliki fokus untuk menyiapkan lulusan agar dapat langsung berperan dalam tugas-tugas operasional dan manajerial di dunia industri.

2.3 Definisi Aplikasi

Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi merupakan suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna; [1]

Adapun definisi lain dari aplikasi yaitu aplikasi merupakan komponen yang berguna melakukan pengolahan data maupun kegiatan-kegiatan seperti pembuatan dokumen atau pengolahan data. Aplikasi adalah bagian PC yang berinteraksi langsung dengan user. Aplikasi berjalan di atas sistem operasi, sehingga agar aplikasi bisa diaktifkan, kita perlu melakukan instalasi sistem operasi terlebih dahulu; [6]

Dari definisi-definisi di atas dapat disimpulkan bahwa aplikasi merupakan suatu salah satu jenis perangkat lunak yang memiliki fungsi untuk dapat memanfaatkan fungsi dari perangkat keras sesuai dengan fungsi dari perangkat keras itu sendiri dengan baik dan benar.

2.4 Definisi *Laravel*

Laravel adalah sebuah Framework *PHP* 5.3 yang bersifat *opensource* yang ditulis oleh Taylor Otwell dengan lisensi dibawah MIT License. *Laravel* dibuat untuk membantu para *developer* khususnya dalam membuat sebuah *web* dengan sintak yang sederhana, elegan, ekspresif dan menyenangkan; [9]

Laravel adalah aplikasi *web* dengan sintak yang ekspresif dan elegan. Dengan *Laravel*, tugas-tugas umum *developer* dapat dikurangi pada sebagian besar proyek-proyek *web* seperti *routing*, *session* dan *caching*. Disamping itu, *Laravel* berusaha

menggabungkan pengalaman-pengalaman development dalam bahasa lain, seperti *Ruby on Rails*, *ASP.NET*, *MVC* dan *Sinatra*. [10]

Dari definisi-definisi di atas dapat disimpulkan bahwa *Laravel* merupakan *framework PHP* yang dirancang untuk membuat *web* dengan sintak yang sederhana, elegan, ekspresif dan menyenangkan.

2.5 Definisi MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan database sebagai sumber dan pengolahan datanya; [1]

Basis Data (*Database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Basis data digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer. Basis data juga dapat berarti kumpulan data yang saling berhubungan yang disimpan secara bersama dengan sedemikian rupa tanpa pengulangan (*redundansi*) data yang tidak perlu, untuk memenuhi berbagai kebutuhan pengguna basis data. Pada dasarnya, suatu basis data tersusun atas komponen-komponen berikut ini; [8]

- a. *File*; merupakan kumpulan berbagai informasi (*records, fields, data values*) yang berhubungan dan juga tersimpan di dalam ruang penyimpanan kedua (*secondary storage*). Dalam konsepnya, file memiliki beberapa tipe yaitu *numeric, character, dan binary*;
 - b. *Record*; merupakan kumpulan *field* yang sangat lengkap, dan biasanya dihitung dalam satuan baris (*rows*);
 - c. *Field*; merupakan tempat atau kolom yang terdapat dalam suatu tabel dan dapat diisi dengan menggunakan value yang dikehendaki;
 - d. *Data value*; merupakan nilai dari suatu data yang dapat digunakan sebagai masukan dalam *field*, biasanya ukurannya (*length*) dalam satuan *byte* atau *bit*;
2. *MySQL* adalah sebuah implementasi dari sistem manajemen basis-data relasional (*RDBMS*) yang didistribusikan secara gratis di bawah lisensi *GPL (General Public License)*. Setiap pengguna dapat secara bebas menggunakan *MySQL*,

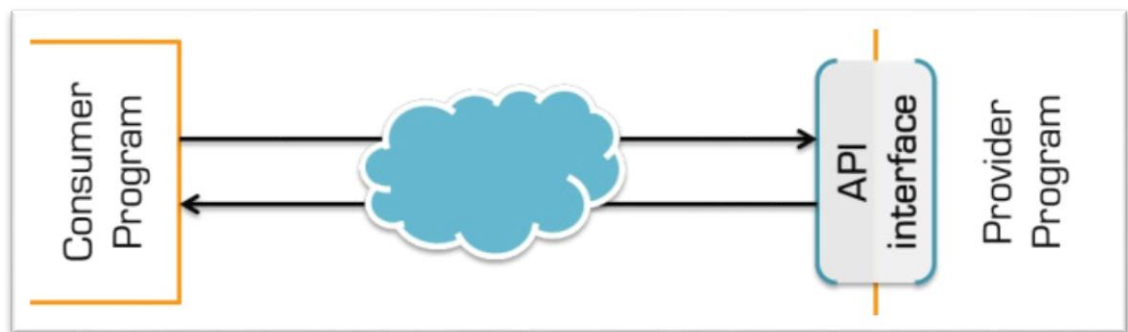
namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. [8]

Berdasarkan uraian tersebut, dapat disimpulkan bahwa 1) Basis Data (*Database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga satu atau lebih program aplikasi dapat menggunakan atau mengolah kembali informasi tersebut, dan 2) *MySQL* adalah satu dari sekian banyaknya *Database Management System (DBMS)* yang digunakan untuk membangun aplikasi *web*.

2.6 Definisi Application Programming Interface (API)

API itu terdapat fungsi-fungsi/perintah-perintah untuk menggantikan bahasa yang digunakan dalam *system calls* dengan bahasa yang lebih terstruktur dan mudah dimengerti oleh *programmer*; [5]

API (Application Programming Interface) memungkinkan pengguna untuk dapat menggunakan sumber daya dari aplikasi lain tanpa perlu mengetahui bagaimana aplikasi itu dibuat. [12]



Gambar 2.1 Cara kerja *API* [5]

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *API* adalah sekumpulan perintah yang dapat digunakan oleh pengguna ataupun *programmer* sehingga memungkinkan suatu aplikasi untuk dapat mengambil data dari aplikasi lainnya tanpa perlu mengetahui seperti apa aplikasi lainnya itu dibuat.

2.7 Definisi Business Process Modelling Notation

BPMN memudahkan para pengembang teknis dalam menerapkan teknologi yang akan melakukan proses-proses tersebut, hingga kepada orang-orang bisnis yang akan mengelola dan memantau proses mereka; [7]

Maksud dari *BPMN* adalah untuk membakukan model bisnis proses dan notasi dalam menghadapi banyak notasi pemodelan dan sudut pandang yang berbeda; dalam melakukannya, *BPMN* akan menyediakan cara sederhana mengkomunikasikan informasi proses bisnis pengguna, proses pelaksana, pelanggan, dan pemasok. [7]

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *BPMN* adalah suatu alat (perangkat lunak) yang berisi sekumpulan notasi yang mudah dipahami oleh semua pengguna bisnis, dari awal proses bisnis tersebut mulai, hingga akhir dari bisnis tersebut.

2.8 Definisi Pengujian Sistem

Pengujian sistem merupakan suatu investigasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari produk atau layanan yang sedang diuji (under test). Pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara obyektif dan independen, yang bermanfaat dalam operasional bisnis untuk memahami tingkat risiko pada implementasinya. Teknik-teknik pengujian mencakup, namun tidak terbatas pada, proses mengeksekusi suatu bagian program atau keseluruhan aplikasi dengan tujuan untuk menemukan bug perangkat lunak; [8]

Beberapa bentuk pengujian sistem yang umum untuk digunakan yaitu Black-Box Testing dan White-Box Testing. Black-Box Testing merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, penguji (tester) dapat mendefinisikan kumpulan kondisi input dan melakukan pengujian pada spesifikasi fungsional program. Sedangkan White-Box Testing adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Berbeda dengan Black-Box Tesing yang melakukan pengujian terhadap

sistem secara langsung, White-Box Testing melakukan pengujian terhadap algoritma sistem. [6]

Berdasarkan uraian tersebut dapat disimpulkan bahwa 1) pengujian sistem merupakan suatu investigasi yang dilakukan untuk mendapatkan informasi terhadap sistem yang akan diuji, dan 2) dua dari banyaknya bentuk pengujian sistem adalah *Black-Box Testing* dan *White-Box Testing*.

2.9 Definisi Analisis Sistem

Analisis sistem adalah fase pengembangan sistem yang menentukan sistem informasi apa yang harus dilakukan untuk memecahkan masalah yang sudah ada dengan mempelajari sistem dan proses kerja untuk mengidentifikasi kekuatan, kelemahan, dan peluang untuk perbaikan; [3]

Analisis sistem adalah penguraian suatu sistem informasi yang sudah utuh kedalam bagian-bagian komponennya dengan tujuan dapat mengidentifikasi dan mengevaluasi berbagai macam permasalahan maupun hambatan yang terjadi pada sistem sehingga nantinya dapat dilakukan perbaikan atau pengembangan; [8]

Berdasarkan uraian tersebut, dapat disimpulkan bahwa ANSI merupakan fase pengembangan sistem untuk mendapatkan solusi setelah melakukan identifikasi terhadap permasalahan yang terjadi pada sistem.

2.10 Definisi Desain Sistem

Desain sistem merupakan tahap setelah analisis dari siklus pengembangan sistem, yaitu tahap dimana dilakukan pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi, menggambarkan bagaimana suatu sistem dibentuk; [8]

Desain sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang harus diselesaikan; tahap ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem, sehingga setelah

instalasi dari sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem. [6]

Berdasarkan uraian tersebut, dapat disimpulkan bahwa desain sistem merupakan penggambaran atas analisis yang telah dilakukan pada tahap analisis sistem, penggambaran tersebut akan memudahkan pengembang dalam menentukan bagaimana suatu sistem akan dibentuk.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis melakukan suatu penelitian dengan cara observasi dan survei secara langsung. Dengan adanya analisis dapat membantu peneliti menyelesaikan permasalahan yang terjadi agar dapat ditemukan jawaban dari permasalahan tersebut dan dengan dilakukannya analisis peneliti dapat mengetahui proses-proses yang terjadi dalam aplikasi.

Pada saat proses analisis, adapun hasil analisis yang didapat adalah beragam. Beberapa analisis yang didapat tersebut kemudian digunakan untuk desain sistem dan pemodelan sistem. Pemodelan sistem tersebut kemudian akan digunakan oleh *developer* untuk memulai mengerjakan aplikasi sesuai dengan pemodelan sistem yang dibuat.

Sistem yang akan dibangun dalam proses bisnis Aplikasi Pesan Antar Makanan sesuai dengan sub-modul yang tertera yaitu menggunakan *Back-end API*. Adapun analisis sistem yang akan dibangun berupa analisis proses bisnis dengan menggunakan *Business Process Modelling Notation (BPMN)*. *BPMN* berikut menjelaskan alur dari proses bisnis yang sedang berjalan dan akan dibangun.

3.1.1 Analisis Sistem Berjalan

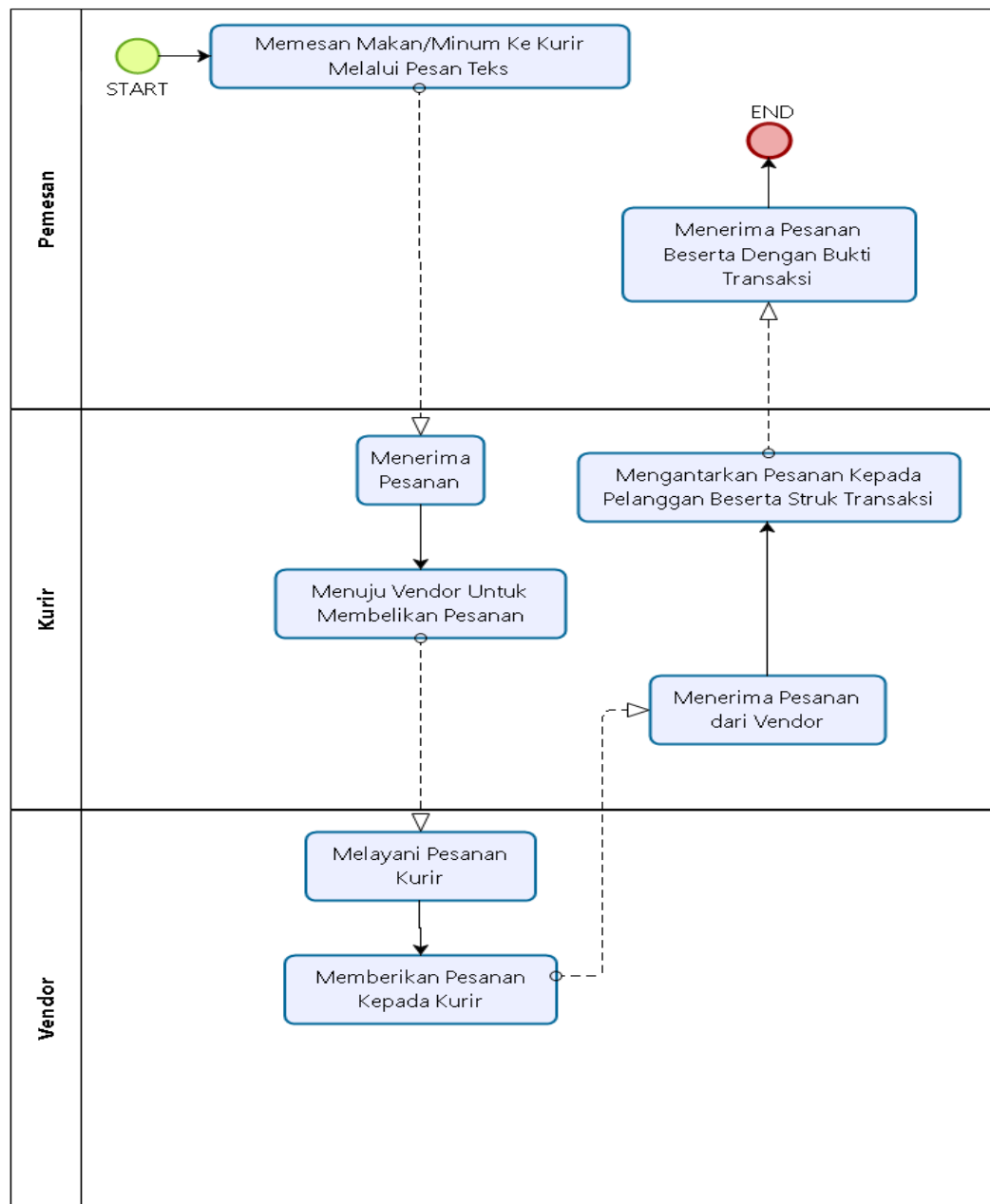
Sistem berjalan dari bisnis pesan antar makanan di wilayah Politeknik Pos Indonesia merupakan hal yang saat ini dirasakan oleh khalayak umum terutama masyarakat Politeknik Pos Indonesia dan sekitarnya.

Adapun sistem berjalan tersebut yaitu dimulai dari pelanggan yang memesan makanan kepada pelayan, dalam *case* tersebut jika pelayan menawarkan terlebih dahulu dengan menanyakan kepada pelanggan “mau pesan apa?” maka pada saat itu juga pelanggan dapat memesan makanan tanpa harus menghampiri toko. Namun jika pelayan tidak menawarkan terlebih dahulu, mau tidak mau pelanggan harus datang menghampiri toko untuk memesan pesanan yang dikehendaki. Itu adalah proses pemesanan, adapun setelah itu pemesan/pelanggan menunggu di meja makan yang telah disediakan. Proses pembayaran makanan dapat dilakukan di awal sebelum memesan makanan atau setelah pemesan/pelanggan memakan makanan yang telah dipesannya, pada kenyataannya, keduanya tidak dipermasalahkan oleh pihak toko.

Uraian tersebut menjelaskan seperti apa sistem berjalan dari bisnis pesan antar makanan di Politeknik Pos Indonesia, uraian tersebut kemudian akan dilakukan pemodelan desain dengan menggunakan pemodelan *BPMN (Business Process Modelling Notation)* dengan menggunakan aplikasi *Bizagi Modeller*.

3.1.1.1 Analisis Prosedur

Berikut adalah analisis prosedur dari bisnis pesan antar makanan di wilayah Politeknik Pos Indonesia.

Gambar 3.1 *BPMN* sistem yang sedang berjalan

Keterangan:

Tabel 3.1 Tabel keterangan *BPMN* sistem sedang berjalan

No	Deskripsi
1	Prosedur dimulai dengan pemesan/pelanggan memesan makanan/minuman ke kurir.
2	Kurir kemudian menerima pesanan tersebut dan menuju ke <i>vendor</i> untuk membeli makanan/minuman sesuai dengan pesanan pelanggan.
3	Di sisi <i>vendor</i> , <i>vendor</i> melayani kurir, menerima pesanan kurir dan memberikannya kepada kurir setelah kurir memberikan sejumlah uang sesuai dengan harga dari pesanan.
4	Setelah itu, dengan segera kurir mengantarkan pesanan tersebut kepada pelanggan/pemesan.
5	Pelanggan/pemesan dapat menerima makanan/minuman setelah memberikan sejumlah uang sesuai dengan harga dari makanan/minuman yang dipesan oleh pelanggan/pemesan tersebut.

3.1.2 Analisis Sistem yang akan Dibangun

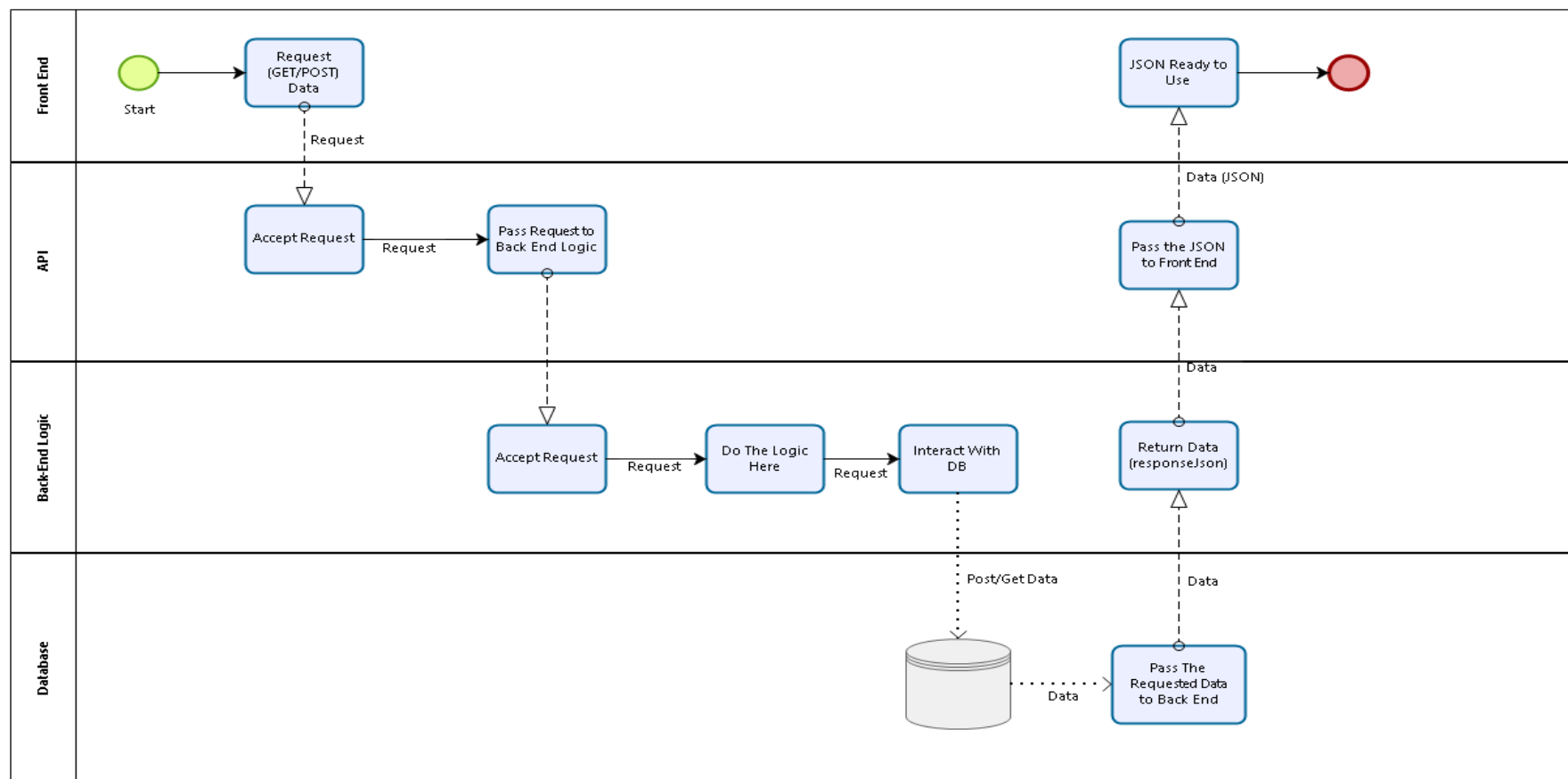
Analisis sistem yang akan dibangun terhadap bisnis pesan antar makanan yaitu menggunakan konsep *online food service* atau pesan antar makanan secara *online*. Pemodelan menggunakan aplikasi yang sama yaitu *Bizagi Modeller* dengan membuat *Business Process Modelling Notation* di dalamnya.

Aplikasi *back-end* menyediakan *API* yang akan digunakan oleh tiga aktor, yaitu pelanggan, kurir, dan admin. Aktor pelanggan melakukan pesanan yang kemudian data pesanan tersebut akan dikirimkan ke *back-end API* untuk kemudian diteruskan/digunakan datanya oleh aktor kurir. Aktor kurir akan lebih lanjut memproses data tersebut, jika kurir mengambil pesanan, maka kurir akan memberikan data kepada *back-end API* bahwa satu pesanan tersebut sedang dalam proses pengiriman oleh kurir tersebut. Admin memiliki peran untuk mendaftarkan penjual atau *vendor*, selain itu, admin memiliki fungsi untuk mengaktifkan atau memberhentikan seorang *user* yang menyalahgunakan aplikasi *FOD*.

Oleh karena itu, *back-end* memiliki peran vital dalam bisnis pesan antar makanan dan aplikasi *FOD*, karena pertukaran data (data transaksi, data pelanggan, data kurir, data admin) antara *front-end* dengan basis data (*database*) dapat terjadi ketika ada *back-end API* yang menjembatannya.

3.1.2.1 Analisis Prosedur

Berikut adalah analisis prosedur sistem yang akan dibangun dari *Food Online Delivery* beserta dengan keterangannya.



Gambar 3.2 BPMN Sistem yang akan dibangun

Keterangan:

Tabel 3.2 Tabel keterangan *BPMN*

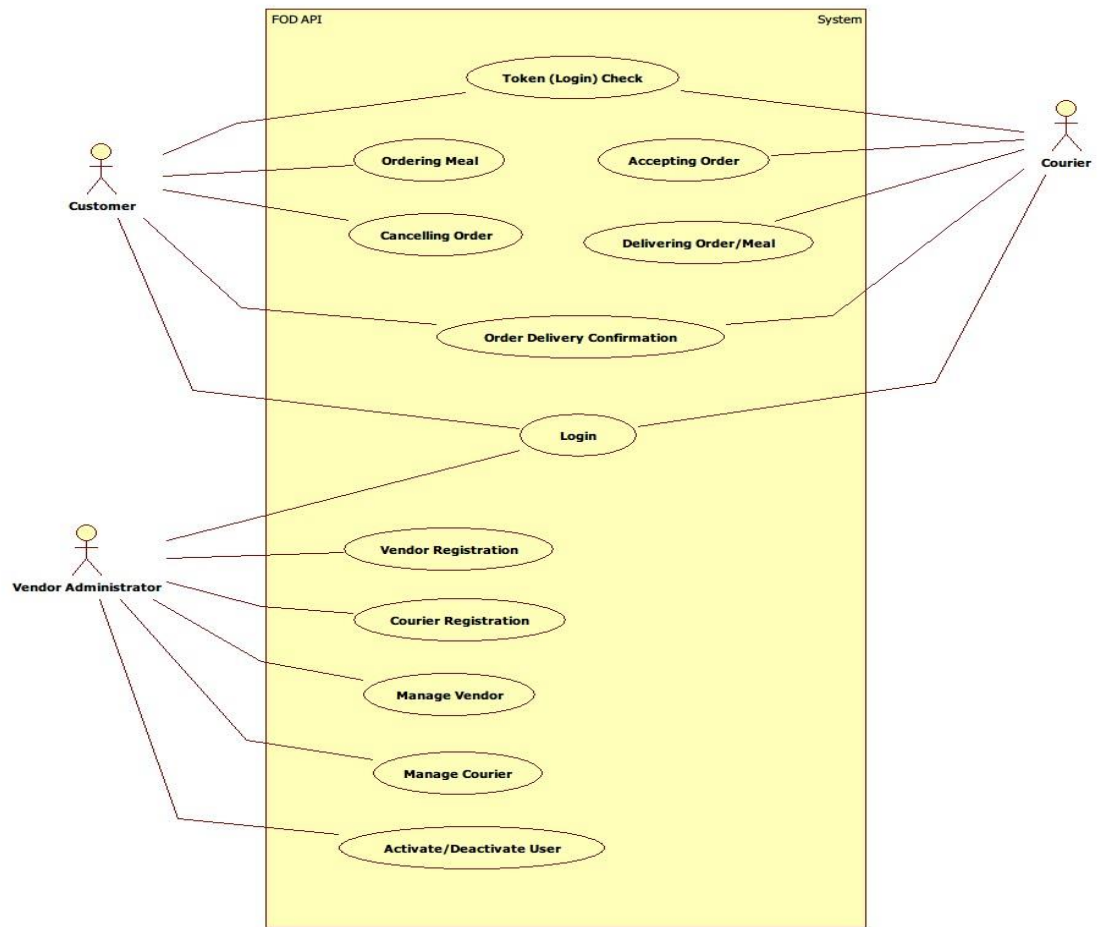
No	Deskripsi
1	Konsep API adalah membuat responsive dan RESTful back-end yang dapat diakses oleh front-end, tanpa mempedulikan seperti apa perangkat front-end yang digunakan.
2	<i>Front-end</i> mengakses <i>API</i> , dengan memberikan <i>Http Request</i> ke <i>API</i> .
3	<i>Jika Http Request yang dikirim oleh front-end adalah POST, maka request dari front-end akan disertai dengan key atau parameter.</i>
4	<i>Jika Http Request yang dikirim oleh front-end adalah GET, maka request dari front-end tidak disertai dengan key atau parameter.</i>
5	<i>API menerima request tersebut lalu mengirimkannya ke controller.</i>
6	<i>Controller kemudian mengerjakan 'logic' dari request tersebut sesuai dengan API url yang diakses.</i>
7	<i>Controller mengirimkan request tersebut ke service, kemudian service mengirimkannya lagi ke repository, di repository inilah terdapat 'logic' yang akan memproses request tersebut.</i>
8	<i>Repository berinteraksi dengan database sesuai dengan API url yang diakses.</i>
9	<i>Kemudian, repository akan memberikan return data ke service, lalu ke controller, kemudian controller ke API, data yang dikirimkan oleh controller ke API berupa JSON; JSON inilah yang kemudian dapat digunakan oleh front-end tanpa mempedulikan seperti apa perangkat front-end yang digunakan.</i>

3.1.2.2 Use Case

Terdapat sebelas *use case*, *use cases* tersebut terbagi menjadi 3 kelompok, yaitu *use case* untuk kurir, *use case* untuk pelanggan, dan *use case* untuk *administrator*.

Use case untuk administrator yaitu 1) *login*, 2) *vendor registration*, 3) *courier registration*, 4) *manage vendor*, 5) *manage courier*, dan *activate/deactivate user*. *Use case* untuk pelanggan yaitu: 1) *login*, 2) *token check*, 3) *ordering meal*, 4) *cancelling order*, dan 5) *order delivery confirmation*. Adapun *use case* untuk kurir sama dengan *use case* untuk pelanggan karena kedua pihak tersebut sama-sama melakukan konfirmasi pengiriman, yang membedakan adalah bahwa pelanggan mengirimkan *order* dan kurir menerima *order*, pelanggan dapat membatalkan *order* dan kurir dapat mengirimkan *order*.

Pada umumnya, deklarasi satu *use case* meliputi satu *sequence diagram*, satu *collaboration diagram*, satu *statechart diagram*, dan satu *activity diagram*. Namun, karena proses yang terdapat di *back-end API* itu adalah sama semua apapun data yang dikelolanya, sehingga pembuatan diagram tersebut cukup satu saja dengan tujuan tidak adanya pengulangan pemodelan analisis sistem yang akan dibangun, hal ini juga mempermudah bagi pihak *developer* untuk mengembangkan aplikasi. Oleh karena itu, berikut adalah *use case* dari sistem yang akan dibangun untuk *Food Online Delivery*.



Gambar 3.3 Use Case Diagram FOD

Keterangan:

Tabel 3.3 Tabel keterangan *Use Case Diagram*

No	Deskripsi
1	Terdapat 3 aktor dalam aplikasi pesan antar makanan yang akan dibangun, yaitu <i>customer</i> , <i>vendor administrator</i> , dan <i>courier</i> .
2	Terdapat 11 total <i>use case</i> dalam aplikasi pesan antar makanan yang akan dibangun.
3	<i>Use case login</i> dan <i>token check</i> dapat dilakukan oleh semua aktor dan termasuk salah satu <i>use case</i> yang wajib dilalui oleh aktor tersebut.
4	Jika dikelompokkan, <i>use case</i> yang dimiliki oleh <i>customer</i> yaitu: 1) <i>login</i> , 2) <i>token check</i> , 3) <i>ordering meal</i> , 4) <i>cancelling order</i> , 5) <i>order delivery confirmation</i> , dan 6) <i>tracking order</i> .
5	Dan <i>use case</i> yang dimiliki oleh aktor <i>courier</i> yaitu: 1) <i>login</i> , 2) <i>token check</i> , 3) <i>accepting order</i> , 4) <i>delivering order/meal</i> , dan 5) <i>order delivery confirmation</i> .
6	Dan <i>use case</i> yang dimiliki oleh aktor <i>vendor administrator</i> yaitu: 1) <i>login</i> , 2) <i>token check</i> , dan 3) <i>manage meal data</i> .

3.1.2.3 Class Diagram

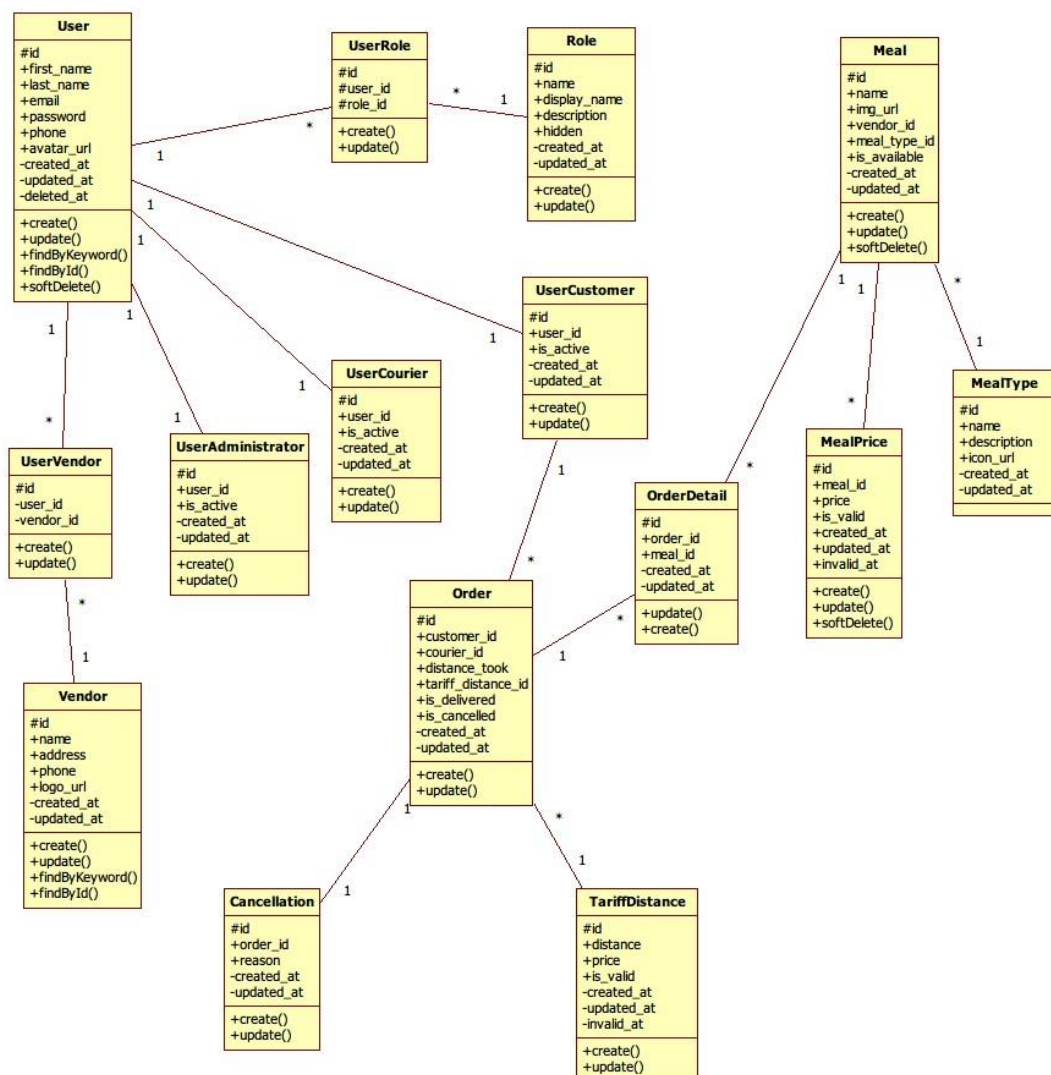
Class diagram food online delivery menggunakan konsep *repository pattern*, adapun konsep tersebut menggunakan konsep polimorfisme dimana dibuat suatu *class interface* dan terdapat suatu *class* yang mengimplementasikan *interface* tersebut.

Adapun *class* tersebut terdiri atas 15 *class*, *class* tersebut adalah sebagai berikut.

1. User (berisi data user)
2. User Administrator (berisi data user sebagai administrator)
3. User Courier (berisi data user sebagai kurir)
4. User Customer (berisi data user sebagai pelanggan)
5. User Role (berisi data user beserta dengan rolenya sebagai apa)
6. User Vendor (berisi data user terdaftar di vendor yang mana)
7. Vendor (berisi data vendor)
8. Order (berisi data pesanan)
9. Order Detail (berisi detail dari data pesanan)

10. Meal (berisi data makanan)
11. Meal Price (berisi data harga dari data makanan)
12. Meal Types (berisi data tipe dari data makanan)
13. Role (berisi daftar peran/aktor yang tersedia)
14. Cancellation (berisi data pembatalan pesanan)
15. Tariff Distance (berisi data tarif atau biaya pengiriman)

Berikut adalah *class diagram* dari *food online delivery*.



Gambar 3.4 *Class Diagram FOD*[deklarasikan]

Keterangan:

Tabel 3.4 Tabel keterangan *Class Diagram*

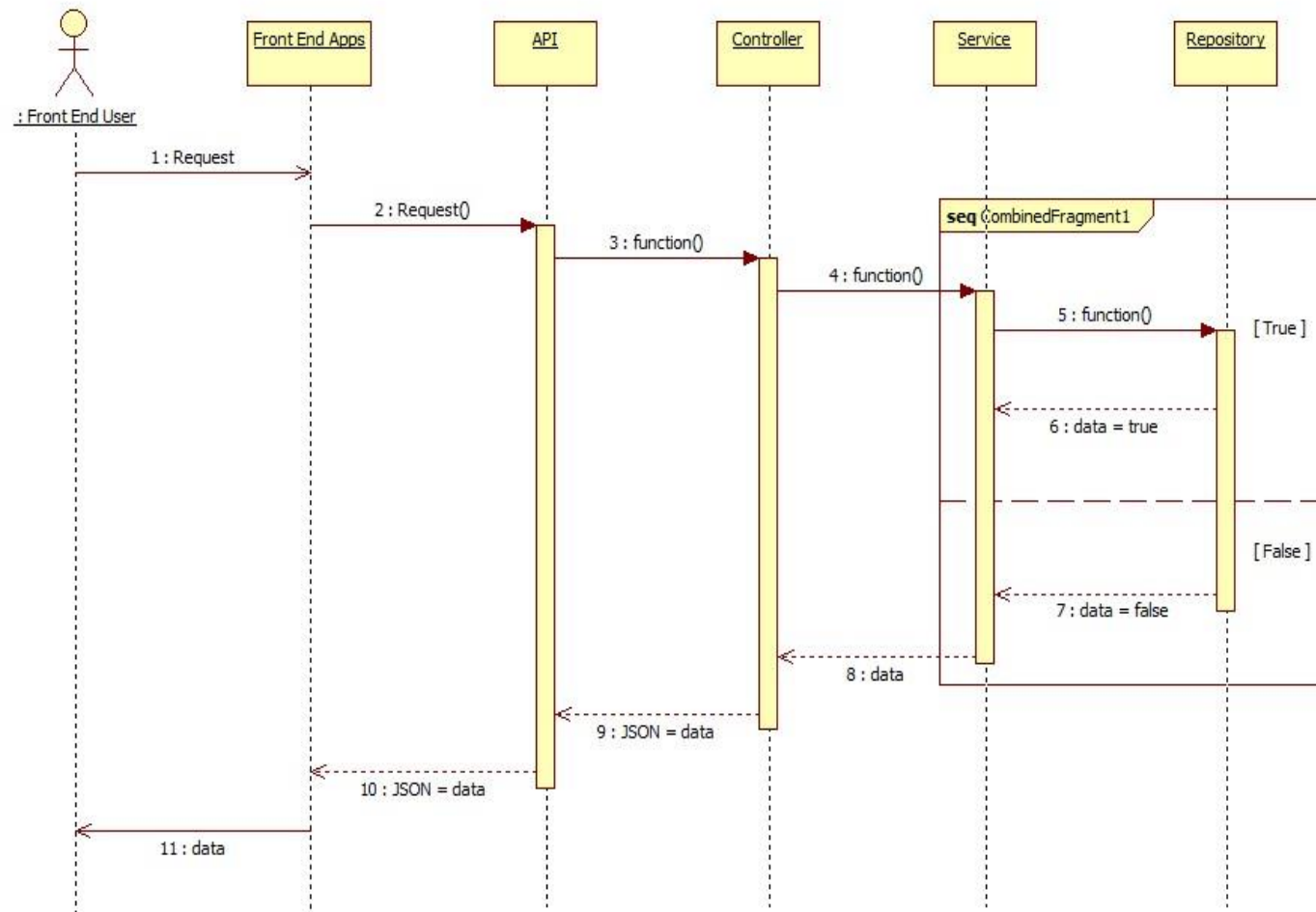
No	Deskripsi
1	<i>Class diagram</i> yang dimaksud disebut juga sebagai <i>entity</i> dalam <i>object oriented programming (OOP)</i> .
2	Terdapat total 14 <i>entity</i> , yaitu: 1) <i>user</i> , 2) <i>vendor</i> , 3) <i>role</i> , 4) <i>user customer</i> , 5) <i>user role</i> , 6) <i>user courier</i> , 7) <i>user vendor</i> , 8) <i>meal</i> , 9) <i>meal price</i> , 10) <i>meal type</i> , 11) <i>cancellation</i> , 12) <i>order</i> , 13) <i>order detail</i> , dan 14) <i>tariff distance</i> ;

3.1.2.4 *Sequence Diagram*

Sequence diagram dari *use case* apapun, untuk proses dalam *back-end*, akan memiliki alur yang sama, yang membedakan adalah data yang dikirimkannya.

Adapun alur proses yang dimaksud adalah sebagai berikut, aplikasi *front-end* mengirimkan data ke *API*, lalu *API* mengirimkan data ke *logic* di *back-end*, kemudian *back-end* mengelola data tersebut sesuai dengan metode yang digunakan, jika *front-end* menggunakan metode *POST* maka *back-end* melakukan penyimpanan data terhadap basis data, sedangkan jika menggunakan metode *GET* maka *back-end* melakukan *get-data* terhadap basis data. Seperti apapun metode yang digunakan, *back-end* akan mengirimkan suatu *return* atau data yang diberikan karena telah menggunakan *function* atau *method* yang terdapat dalam *back-end*, adapun *return* tersebut pasti berupa suatu data, baik data tersebut berupa *array*, berupa *boolean*, ataupun berupa *integer*.

Berikut adalah *sequence diagram* dari *food online delivery*.



Gambar 3.5 Sequence Diagram FOD

Keterangan:

Tabel 3.5 Tabel keterangan *Sequence Diagram*

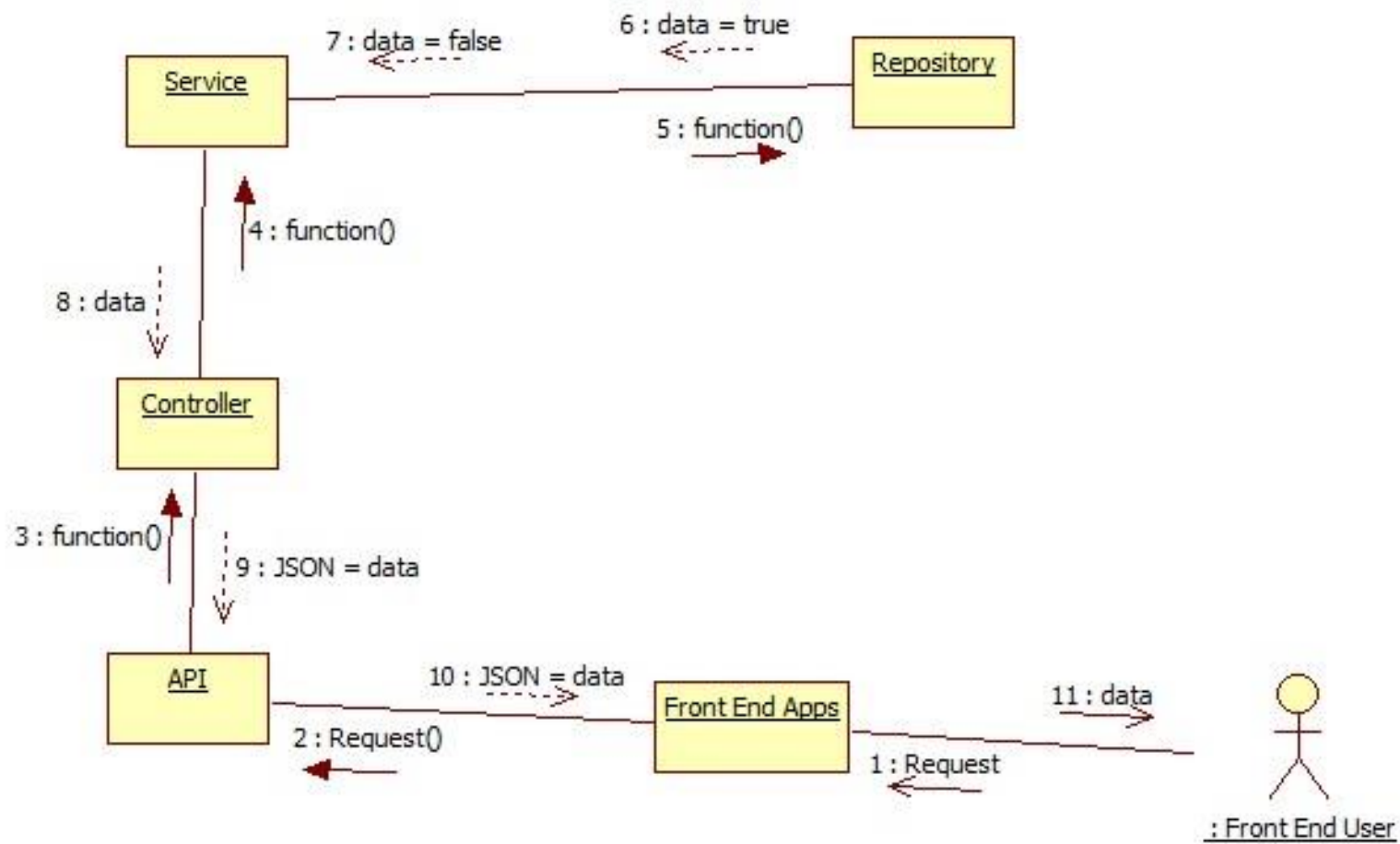
No	Deskripsi
1	Seperti apapun <i>use casenya</i> , konsep <i>API</i> adalah dimana <i>front-end side</i> mengirimkan <i>HTTP Request</i> berupa <i>GET</i> atau <i>POST</i> .
2	Back-end logic kemudian menerima request tersebut dan mengerjakan <i>logic/fungsi</i> sesuai dengan <i>API URL</i> yang diakses.
3	Adapun konsep <i>back-end</i> yang digunakan adalah menggunakan <i>Repository Pattern</i> , dimana <i>logic</i> dibagi menjadi 3 sesi, yaitu <i>Controller</i> , <i>Service</i> , dan <i>Repository</i> .
4	Jika terdapat <i>error</i> , <i>return</i> yang akan dikembalikan adalah data <i>false message</i> , pesan beragam sesuai dengan <i>error</i> yang didapatkan.

3.1.2.5 *Collaboration Diagram*

Collaboration diagram dari *food online delivery* sama halnya dengan *sequence diagram* dari *food online delivery*, memiliki alur yang sama seperti apapun prosesnya, yang membedakan adalah datanya.

Adapun prosesnya, dimodelkan tidak secara *timeline* seperti *sequence diagram*, dimulai dari aplikasi front end yang mengirimkan data ke *API*, *API* memberikan ke *back-end*, kemudian *back-end* mengelolanya dan memberikan respon balik kepada *API* yang kemudian *API* memberikan respon tersebut kepada *front end*.

Adapun *collaboration diagram* yang dimaksud adalah sebagai berikut.



Gambar 3.6 Collaboration Diagram FOD

Keterangan:

Tabel 3.6 Tabel keterangan *Collaboration Diagram*

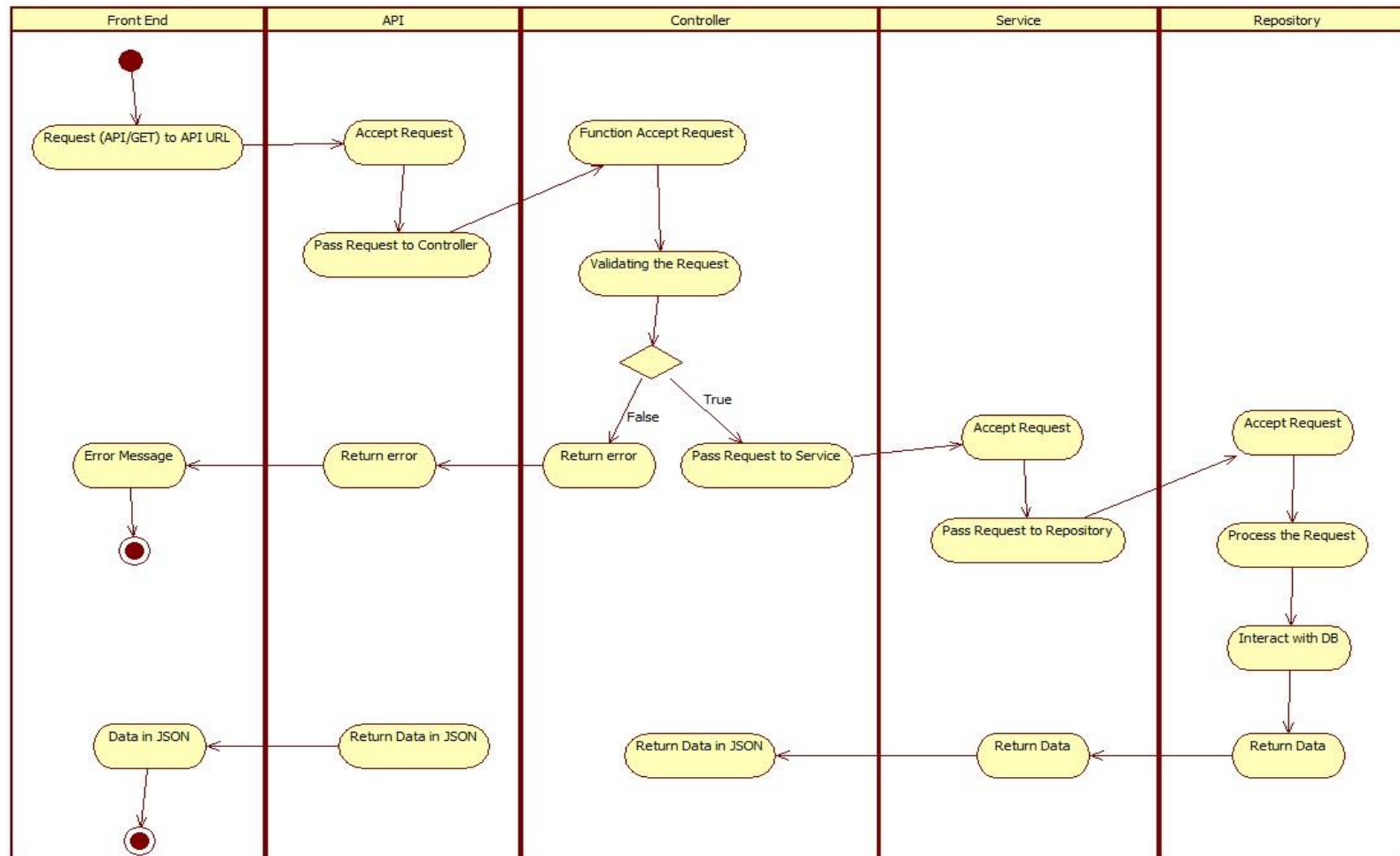
No	Deskripsi
1	<i>Collaboration diagram</i> merupakan <i>sequence diagram</i> yang dijabarkan tidak secara <i>timeline</i> .
2	<i>Collaboration diagram</i> menunjukkan interaksi antar objek kepada <i>programmer</i> sehingga dapat memahami interaksi tersebut untuk kemudian dilakukan implementasi dalam bahasa pemrograman.

3.1.2.6 *Activity Diagram*

Activity diagram memiliki kesamaan dengan *flowmap* atau *BPMN*, hanya saja *activity diagram* dimodelkan secara *UML (Unified Modelling Language)*.

Activity diagram dari *food online delivery* menggambarkan proses bisnis dari sistem yang akan dibangun secara “urutan aktifitas”, dari mulai *start* sampai dengan *end*. Adapun alurnya adalah dimulai dari aplikasi *front-end* yang mengirimkan data atau mengakses *API*, kemudian *API* menyerahkannya kepada *back-end*, lalu *back-end* kemudian mengolah data tersebut sesuai dengan permintaan dari *front-end*, seperti apapun data yang dikirimkan atau diminta, prosesnya di *back-end* akan memiliki prosedur yang sama.

Adapun *activity diagram* yang dimaksud adalah sebagai berikut.



Gambar 3.7 Activity Diagram FOD

Keterangan:

Tabel 3.7 Tabel keterangan *Activity Diagram FOD*

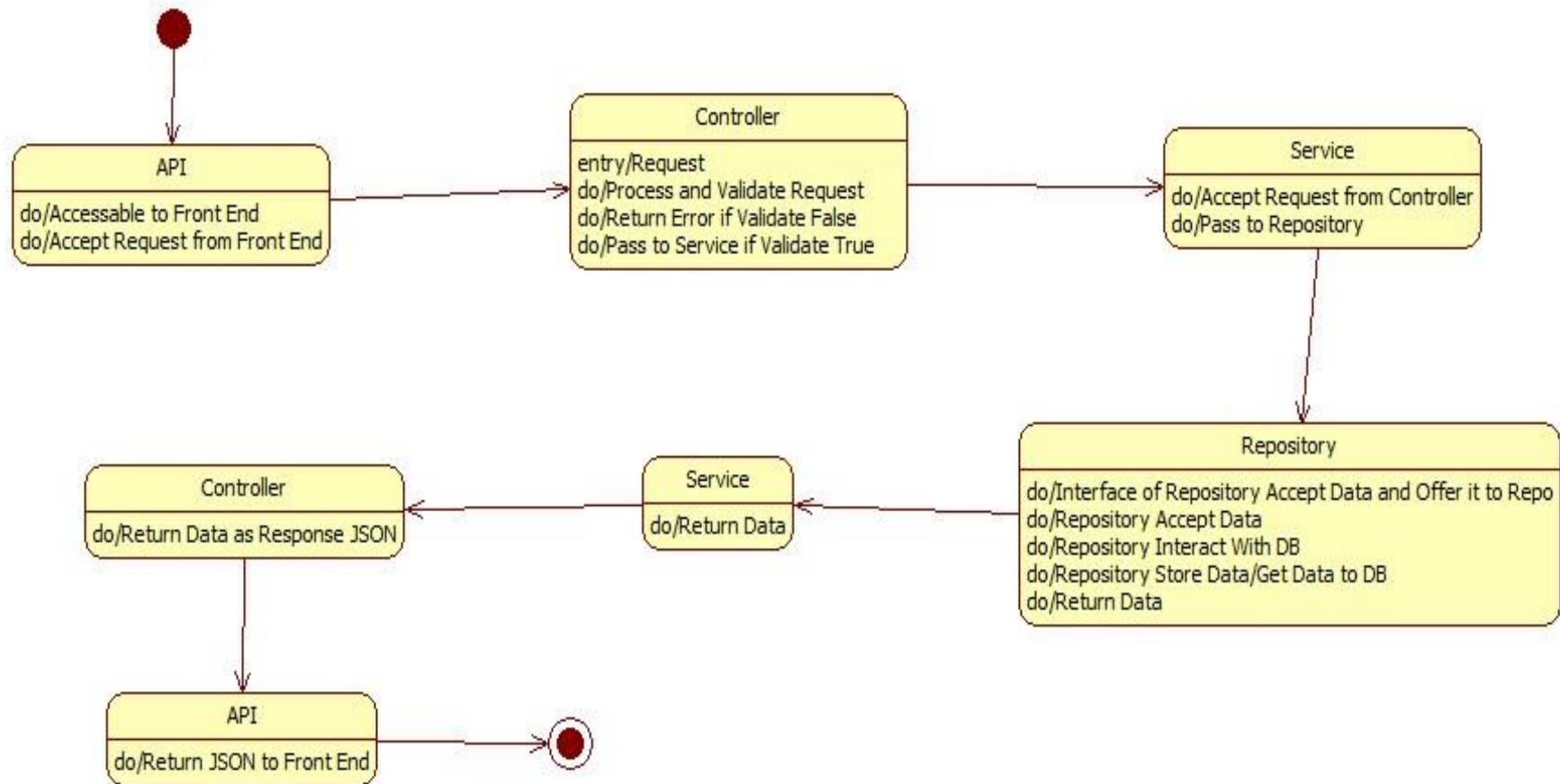
No	Deskripsi
1	Sebagai <i>start</i> , pengguna aplikasi <i>front-end</i> akan mengirimkan <i>request</i> , baik itu <i>POST</i> ataupun <i>GET</i> . <i>API</i> dapat menerima <i>request</i> tersebut tidak peduli seperti apa <i>platform</i> yang digunakan oleh <i>front-end</i> .
2	<i>API</i> menerima <i>request</i> tersebut lalu mengirimkannya ke <i>back-end logic</i> , yaitu oleh <i>controller</i> .
3	<i>Controller</i> melakukan validasi terhadap <i>request</i> tersebut, umumnya bila melakukan proses <i>upload</i> , bilamana ekstensi dari <i>file</i> yang di- <i>upload</i> tidak sesuai dengan persyaratan <i>file upload</i> yang telah ditentukan maka <i>controller</i> akan dengan seketika memberikan <i>return false</i> berupa <i>error message</i> .
4	Jika <i>request</i> tersebut <i>valid</i> , maka akan dikirimkan ke <i>service</i> .
5	<i>Service</i> kemudian akan mengirimkannya ke <i>repository</i> .
6	Setelah <i>repository</i> menerima <i>request</i> tersebut, <i>repository</i> akan memproses <i>request</i> tersebut dengan berinteraksi dengan <i>database</i> sesuai dengan <i>API URL</i> yang diakses.
7	Jika <i>POST</i> maka <i>repository</i> akan mengirimkan <i>request</i> tersebut ke basis data dan menyimpannya.
8	Jika <i>GET</i> maka <i>repository</i> akan mengambil data dari basis data sesuai dengan <i>API URL</i> yang diakses.
9	Setelah proses interaksi tersebut, <i>repository</i> akan <i>return data</i> ke <i>service</i> , <i>service</i> akan <i>return data</i> ke <i>controller</i> , <i>controller</i> akan <i>return data</i> berupa <i>response JSON</i> ke <i>API</i> , dan <i>API</i> akan mengirimkannya ke <i>front-end side</i> untuk diolah lebih lanjut.

3.1.2.7 *Statechart Diagram*

Statechart diagram menggambarkan alur dari proses bisnis *food online delivery* dengan memberikan keterangan pada setiap *statenya* apa saja yang dilakukan oleh sistem pada *state* tersebut.

Terdapat 7 state tidak termasuk dengan *start* dan *end*, *state* tersebut dimulai dari *API* mengirimkan data ke *controller*, *controller* kemudian mengirimkan lagi data tersebut ke *service*, *service* kemudian mengirimkannya ke *repository*, *repository* kemudian berinteraksi dengan basis data, setelah *repository* berinteraksi dengan basis data, *repository* akan memberikan *return data* ke *service*, *service* kemudian memberikannya ke *controller*, *controller* kemudian memberikannya ke *API*, *API* kemudian memberikan respon tersebut ke aplikask *front-end* yang mengaksesnya.

Adapun alur tersebut dimodelkan dengan diagram sebagai berikut.



Gambar 3.8 Statechart Diagram FOD

Keterangan:

Tabel 3.8 Tabel keterangan *statechart diagram FOD*

No	Deskripsi
1	<i>Statechart diagram</i> merupakan bentuk detail proses dari <i>use case</i> .
2	Setiap <i>use case</i> memiliki masing-masing <i>statechart diagram</i> .
3	Adapun karena <i>logic</i> yang terdapat dalam <i>back-end API</i> beserta dengan konsep <i>repository pattern</i> , setiap proses memiliki detail proses yang sama.
4	Adapun detail proses tersebut yaitu: kirim request berupa POST/GET ke API URL, kemudian API URL mengirimkannya ke controller, controller kemudian melakukan validasi, bila tidak valid maka return error, jika valid maka request dikirimkan ke service. <i>Service</i> kemudian mengirimkannya ke <i>repository</i> dimana <i>logic code</i> yang sebenarnya berada. <i>Repository</i> kemudian berinteraksi dengan basis data, kemudian <i>return data</i> sampai ke <i>controller</i> lalu ke <i>API</i> berupa <i>response JSON</i> .

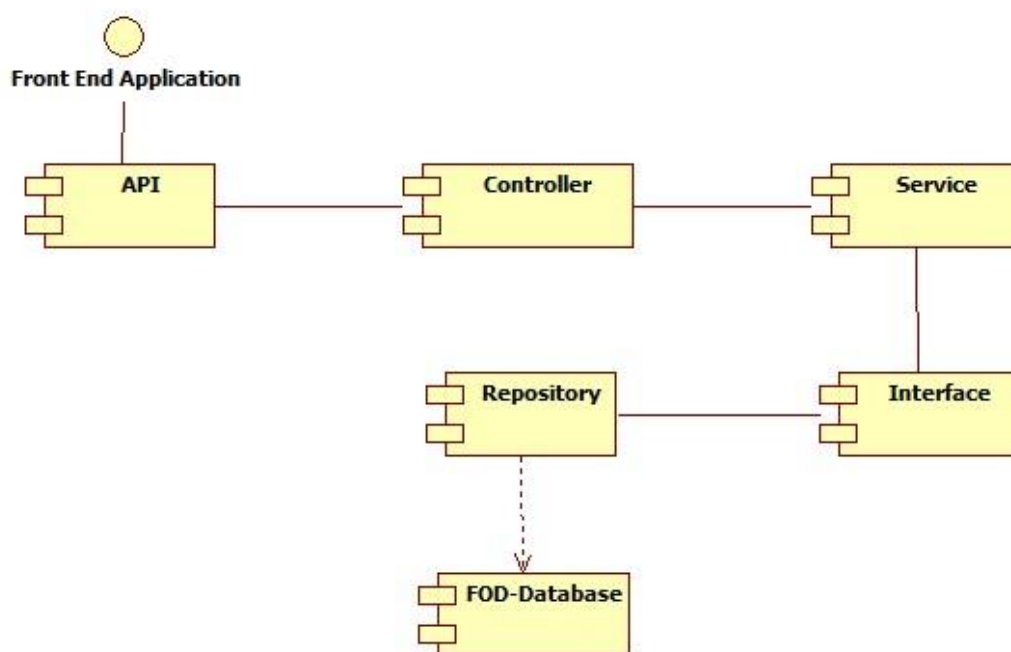
3.2 Perancangan Sistem

Perancangan sistem merupakan tahap pemodelan sistem dimana sistem tersebut nantinya ketika diluncurkan atau digunakan oleh khalayak akan seperti apa jadinya.

Adapun perancangan sistem untuk *food online delivery* menggunakan konsep sederhana yaitu konsep *client-server*, di mana aplikasi *front-end* melakukan transfer data dengan *back-end* melalui *web services* atau *API*.

Berikut adalah component diagram yang terdapat dalam aplikasi *food online delivery*.

3.2.1 Component Diagram



Gambar 3.9 Component diagram FOD

Keterangan:

Tabel 3.9 Tabel keterangan *Component diagram FOD*

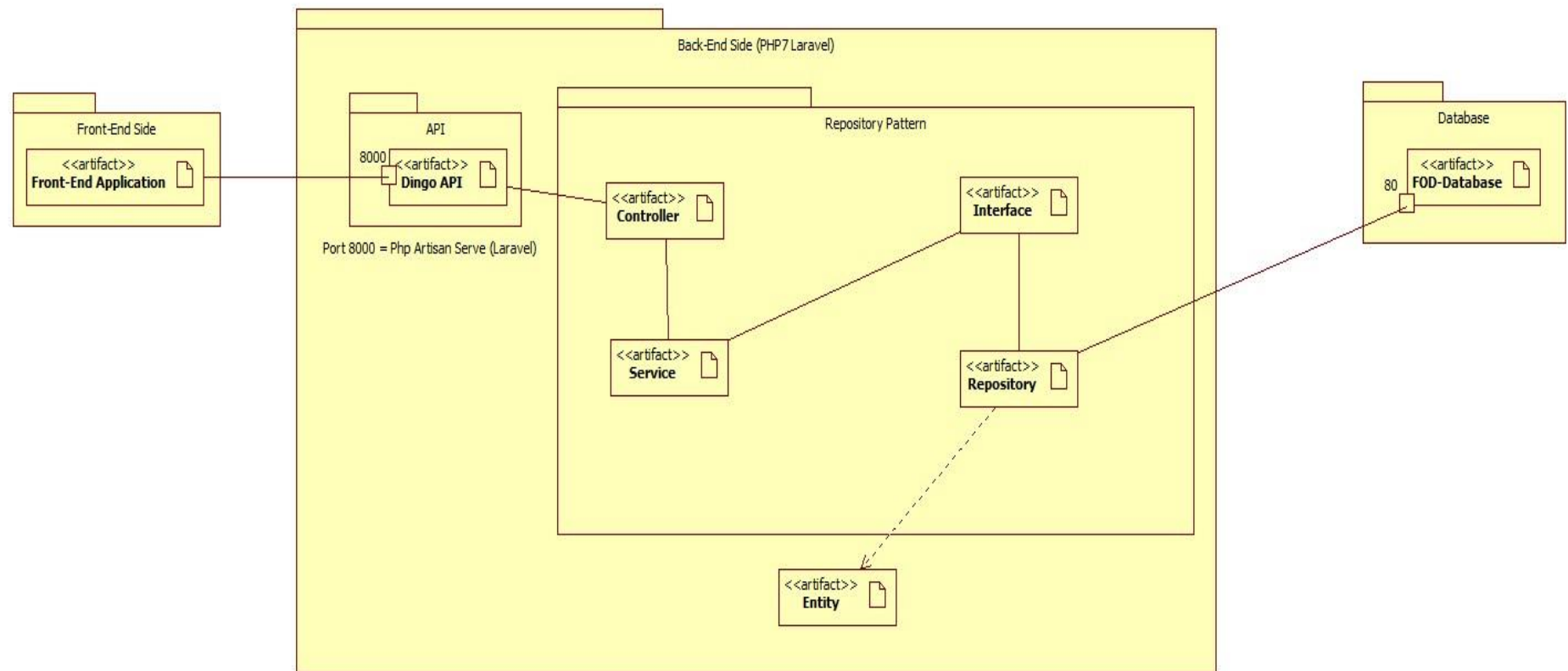
No	Deskripsi
1	Terdapat enam komponen utama, yaitu: <i>API</i> , <i>Controller</i> , <i>Service</i> , <i>Interface</i> , <i>Repository</i> , dan <i>Database</i> .
2	Komponen <i>API</i> yaitu sebagai ‘jembatan’ antara aplikasi <i>front-end</i> dengan aplikasi <i>back-end</i> .
3	Komponen <i>Controller</i> dapat disebut sebagai komponen dimana data yang dikirimkan oleh <i>API</i> akan divalidasi.
4	Komponen <i>Service</i> sebagai jembatan antara <i>Controller</i> dengan <i>Repository</i> .
5	Komponen <i>Interface</i> merupakan ‘gerbang’ dimana untuk mengakses <i>Repository</i> , <i>Service</i> harus terlebih dulu mengakses <i>Interface</i> .
6	Komponen <i>Repository</i> merupakan komponen dimana <i>logic code</i> disimpan, di sini <i>Repository</i> berinteraksi dengan basis data. Di sini juga, <i>entity</i> dideklarasikan.

3.2.2 Deployment Diagram

Deployment diagram food online delivery, sesuai dengan uraian yang telah disebutkan pada sub-bab tentang *component diagram*, menggunakan konsep *client-server*.

Pada sisi *client* tentu terdapat aplikasi *client* yang akan mengakses data pada aplikasi *server* atau *back-end web API*. Aplikasi *client* dapat terhubung dengan *back-end* melalui *port* yang ditentukan, pada *deployment* nantinya akan menggunakan port 8000. *Dingo API* merupakan salah satu *library* laravel yang dipilih dan digunakan sebagai *API*. Dalam *repository pattern* terdapat *Controller*, *Service*, *Interface*, dan *Repository* yang terhubung dengan *Entity*, *Repository* juga berinteraksi langsung dengan basis data.

Berikut adalah *deployment diagram* dari *food online delivery*.

Gambar 3.10 *Deployment diagram FOD*

Keterangan:

Tabel 3.10 Tabel keterangan *deployment diagram FOD*

No	Deskripsi
1	Terdapat tiga <i>side</i> , yaitu <i>front-end side</i> , <i>back-end side</i> , dan basis data;
2	<i>Front-end side</i> mendapatkan data yang diperlukan dari <i>API</i> melalui <i>port</i> 8000. Adapun <i>port</i> tersebut dapat dikonfigurasi secara <i>manual</i> sesuai dengan kebutuhan.
3	<i>API</i> berinteraksi langsung dengan <i>back-end side</i> .
4	<i>Back-end side</i> terdiri atas empat komponen, yaitu: <i>controller</i> , <i>service</i> , <i>interface</i> , dan <i>repository</i> .
5	<i>Repository</i> merupakan satu-satunya komponen yang dapat berinteraksi langsung dengan <i>repository</i> melalui <i>port</i> 80.
6	<i>Entity</i> dideklarasikan di <i>repository</i> .

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Lingkungan Implementasi

Lingkungan implementasi *food online delivery* meliputi *SRS (System Requirements Specification)*, kebutuhan perangkat lunak pendukung (*third party software*), dan kebutuhan perangkat keras untuk *back-end web API* yang akan digunakan. Di akhir penjelasan akan diberikan hasil pengujian terhadap *url API*.

4.1.1 Kebutuhan Perangkat Lunak

No	Kode	Fungsi
1	FOD-01	<i>API</i> dapat diakses secara <i>private</i> , hanya beberapa aplikasi dan atau aktor tertentu yang dapat mengaksesnya, adapun aktor tersebut hanya 3 yaitu admin, kurir, dan pelanggan. Adapun yang dimaksud dengan tersedia <i>API</i> pada deskripsi di bawah artinya dapat diakses oleh ketika aktor tersebut.
2	FOD-02	Tersedia <i>API</i> login, adapun logout termasuk ke dalam <i>destroy session</i> dilakukan di <i>front-end side application</i> .
3	FOD-03	Tersedia <i>API</i> untuk registrasi setiap aktor, registrasi admin, registrasi kurir, dan registrasi pelanggan.
4	FOD-04	Tersedia <i>API</i> untuk melakukan ubah data identitas diri berlaku untuk ketiga aktor tersebut.
5	FOD-05	Tersedia <i>API</i> untuk mengirimkan <i>order</i> atau pesanan.
6	FOD-06	Tersedia <i>API</i> untuk membatalkan pesanan.

4.1.2 Aplikasi Pendukung

1. Perangkat Lunak

Perangkat lunak yang dimaksud adalah perangkat lunak pendukung dimana aplikasi *food online delivery* dapat digunakan dengan perangkat lunak pendukung yang dimaksud. Adapun perangkat lunak yang dibutuhkan dalam lingkungan implementasi adalah sebagai berikut.

Tabel 4.1 Tabel spesifikasi perangkat lunak untuk *server-side application*

No	Nama	Fungsi
1	XAMPP (w/ Apache dan MySQL)	Penyedia layanan <i>localhosting</i> Apache dan MySQL <i>Database Service</i>
2	Google Chrome (Version 55.0.2883.87 m (64-bit))	<i>Browser</i> yang digunakan untuk mengakses dan menggunakan aplikasi
3	Composer, Php Dependencies Manager	Selain sebagai manajemen <i>library</i> yang digunakan oleh <i>back-end</i> , juga digunakan sebagai <i>local hosting</i> pada <i>port 80</i> dengan menggunakan <i>command line</i> “ <i>php artisan serve</i> ”
4	Postman	Digunakan untuk melakukan <i>testing</i> terhadap <i>back-end API</i>

2. Perangkat Keras

Perangkat keras yang dimaksud adalah perangkat keras pendukung yang digunakan bersamaan dengan aplikasi *food online delivery*. Perangkat keras tersebut digunakan untuk melakukan pengujian terhadap aplikasi *food online delivery*. Spesifikasi kebutuhan perangkat keras yang digunakan adalah beragam namun pada saat pengujian, spesifikasi yang digunakan adalah menyesuaikan dengan laptop penguji, adapun salah satu laptop penguji yang digunakan sebagai *server testing* adalah sebagai berikut.

Tabel 4. 2 Tabel spesifikasi perangkat keras server

No	Nama Perangkat Keras	Spesifikasi
1	Processor	Intel Core i3 4010U 1.9 GHz
2	HDD	500 GB
3	RAM	10 GB RAM DDR3L
4	Video Card (VGA)	NVidia GT 820M
5	Monitor	1366 x 768 Pixels
6	Wireless Network Adapter	Atheros Wireless Network Adapter
7	Network Card untuk LAN	Network Card <i>support</i> kabel LAN (UTP/STP)

4.2 Pengujian dan Hasil Pengujian

Pengujian menggunakan *local hosting* dengan domain *local.fod.com*. Berikut adalah daftar *API Url* yang akan dilakukan pengujiannya sehingga *front-end side* dapat menggunakan *API Url* tersebut untuk melakukan transaksi data. Adapun seluruh *API Url* berhasil digunakan oleh pihak *front-end* tanpa ada cacat sedikitpun, sehingga ditandai dengan hasil 'sukses'.

4.2.1 Pengujian API URL

Berikut adalah pengujian terhadap *back-end API url*. Masing-masing dari *url* memberikan *return* yang berbeda namun memiliki prosedur pengolahan data yang sama (data transaksi, data user pelanggan, data user kurir, dan data user admin).

Tabel 4.3 Daftar *url* yang dapat diakses oleh pihak *front-end*

No	Nama	API Url	Aktor	Metode Http	Fungsi	Hasil
1	Login	http://local.fod.com/api/login	User	POST	<i>User login</i> , digunakan oleh pelanggan, kurir, ataupun admin untuk melakukan login	Sukses
2	User Index	http://local.fod.com/api/user/index	User	GET	Get all user atau index dari user	Sukses
3	User Get	http://local.fod.com/api/user/{user_id}	User	GET	get satu user berdasarkan user id	Sukses
4	User Find	http://local.fod.com/api/user/find	User	POST	Cari user berdasarkan keyword, pencarian berdasarkan first_name dan atau last_name	Sukses
5	User Update	http://local.fod.com/api/user/{user_id}/update	User	POST	Update user atau ubah identitas diri user	Sukses
6	Customer Create (Registration)	http://local.fod.com/api/user/customer/create	Customer	POST	Registrasi user dengan role customer	Sukses

No	Nama	API Url	Aktor	Metode Http	Fungsi	Hasil
7	Customer Index	http://local.fod.com/api/user/customer/index	Customer	GET	Index dari user dengan role customer	Sukses
8	Customer Get	http://local.fod.com/api/user/customer/{customer_id}	Customer	GET	Get customer berdasarkan customer id	Sukses
9	Customer Find	http://local.fod.com/api/user/customer/find	Customer	POST	Cari customer berdasarkan nama customer	Sukses
10	Customer Update	http://local.fod.com/api/user/customer/{customer_id}/update	Customer	POST	Ubah identitas diri customer	Sukses
11	Courier Create (Registration)	http://local.fod.com/api/user/courier/create	Courier	POST	Buat kurir atau registrasi kurir	Sukses
12	Courier Index	http://local.fod.com/api/user/courier/index	Courier	GET	Index dari kurir atau get all kurir	Sukses
13	Courier Get	http://local.fod.com/api/user/courier/{courier_id}	Courier	GET	Get satu kurir berdasarkan id kurir	Sukses

No	Nama	API Url	Aktor	Metode Http	Fungsi	Hasil
14	Courier Find (By Keyword)	http://local.fod.com/api/user/ courier/find	Courier	POST	Cari kurir berdasarkan nama kurir	Sukses
15	Courier Update	http://local.fod.com/api/user/ courier/{courier_id}/update	Courier	POST	Ubah data berdasarkan id kurir	Sukses
16	Vendor Create	http://local.fod.com/api/vend or/create	Vendor	POST	Buat vendor atau vendor registrasi	Sukses
17	Vendor Index	http://local.fod.com/api/vend or/index	Vendor	POST	Index dari vendor atau get semua vendor	Sukses
18	Vendor Get	http://local.fod.com/api/vend or/{vendor_id}	Vendor	GET	Get satu vendor berdasarkan vendor id	Sukses
19	Vendor Update	http://local.fod.com/api/vend or/{vendor_id}/update	Vendor	POST	Ubah data vendor berdasarkan vendor id	Sukses
20	Meal Create	http://local.fod.com/api/meal /create	Meal	POST	Buat meal / makanan	Sukses
21	Meal Index	http://local.fod.com/api/meal /index	Meal	GET	Index dari meal atau get semua meal	Sukses

No	Nama	API Url	Aktor	Metode Http	Fungsi	Hasil
22	Meal Get	http://local.fod.com/api/meal/{meal_id}	Meal	GET	Get satu meal berdasarkan id meal	Sukses
23	Meal Update	http://local.fod.com/api/meal/{meal_id}/update	Meal	POST	Ubah data meal berdasarkan id meal	Sukses
24	Meal Category Create	http://local.fod.com/api/meal/category/create	Meal Category	POST	Buat kategori meal	Sukses
25	Meal Category Index	http://local.fod.com/api/meal/category/index	Meal Category	GET	Index dari meal / semua makanan	Sukses
26	Meal Category Get	http://local.fod.com/api/meal/category/{meal_category_id}	Meal Category	GET	Get satu kategori makanan berdasarkan id	Sukses
27	Meal Category Update	http://local.fod.com/api/meal/category/{meal_category_id}/update	Meal Category	POST	Ubah kategori makanan berdasarkan id	Sukses

No	Nama	API Url	Aktor	Metode Http	Fungsi	Hasil
28	Order Create	http://local.fod.com/api/order/create	Order	POST	Buat order	Sukses
29	Order Index	http://local.fod.com/api/order/index	Order	GET	Index dari order	Sukses
30	Order Get	http://local.fod.com/api/order/{order_id}	Order	GET	Get satu order berdasarkan id	Sukses
31	Order Update	http://local.fod.com/api/order/{order_id}/update	Order	POST	Update order berdasarkan id	Sukses
32	Cancellation Create	http://local.fod.com/api/order/{order_id}/cancel	Cancellation	POST	Buat pembatalan order	Sukses
33	Cancellation Index	http://local.fod.com/api/order/cancel/index	Cancellation	GET	Daftar dari pembatalan order	Sukses
34	Cancellation Get	http://local.fod.com/api/order/cancel/{cancel_id}	Cancellation	GET	Ambil satu data pembatalan order berdasarkan id	Sukses
35	Cancellation Update	http://local.fod.com/api/order/cancel/{cancel_id}/update	Cancellation	POST	Ubah data pembatalan order berdasarkan satu id	Sukses

4.2.2 Pengujian Kondisional

Maksud dari pengujian kondisional adalah kondisi atau syarat prasyarat yang diperlukan untuk mengakses setiap *api url* yang ada, baik itu menggunakan metode *POST* ataupun *GET*. Berikut adalah tabel pengujian kondisional untuk *back-end API food online delivery*.

No	Nama	API Url	Aktor	Metode Http	Kondisi	Hasil
1	Login	http://local.fod.com/api/login	User	POST	Kondisi <i>login</i> yaitu. 1. Jika validasi <i>username</i> dan <i>password</i> benar maka <i>return true</i> 2. Jika validasi <i>username</i> dan <i>password</i> salah maka <i>return false</i>	Sukses
2	User Index	http://local.fod.com/api/user/index	User	GET	Tidak ada kondisi, data akan diambil semuanya secara berurut, adapun <i>pagination</i> dilakukan oleh <i>front-end application</i>	Sukses
3	User Get	http://local.fod.com/api/user/{user_id}	User	GET	Jika <i>user id</i> yang dimaksud tidak ada maka <i>return false</i> , selebihnya <i>return true</i>	Sukses
4	User Find	http://local.fod.com/api/user/find	User	POST	Jika tidak ada user yang memiliki nama sesuai dengan <i>keyword</i> yang dimasukkan maka <i>return false</i> , selebihnya <i>return true</i>	Sukses

No	Nama	API Url	Aktor	Metode Http	Kondisi	Hasil
5	User Update	http://local.fod.com/api/user/{user_id}/update	User	POST	Jika terdapat salah satu parameter yang tidak valid seperti salah penulisan <i>email</i> atau memasukkan email yang telah ada maka <i>return false</i> , nomor handphone yang dimasukkan pun bersifat <i>unique</i> sehingga antar <i>user</i> tidak akan ada yang memiliki nomor handphone yang sama. Selebihnya <i>return true</i> .	Sukses
6	Customer Create (Registration)	http://local.fod.com/api/user/customer/create	Customer	POST	Terdapat validasi di setiap parameternya, jika salah satu parameter tidak valid seperti <i>username</i> , <i>email</i> , atau <i>phone number</i> telah dipakai maka <i>return false</i> , selebihnya jika data tersebut adalah benar dan asli maka <i>return true</i> .	Sukses

No	Nama	API Url	Aktor	Metode Http	Kondisi	Hasil
7	Customer Index	http://local.fod.com/api/user/customer/index	Customer	GET	Tidak ada kondisi.	Sukses
8	Customer Get	http://local.fod.com/api/user/customer/{customer_id}	Customer	GET	Jika <i>customer id</i> yang dimaksud tidak ada maka <i>return false</i> .	Sukses
9	Customer Find	http://local.fod.com/api/user/customer/find	Customer	POST	Jika pencarian <i>customer</i> berdasarkan <i>keyword</i> yang dimasukkan tidak ada maka <i>return false</i> .	Sukses
10	Customer Update	http://local.fod.com/api/user/customer/{customer_id}/update	Customer	POST	Jika setiap parameter dari data identitas <i>customer</i> tidak valid, seperti nomor handphone, email, ataupun username, maka <i>return false</i> .	Sukses
11	Courier Create (Registration)	http://local.fod.com/api/user/courier/create	Courier	POST	Setiap parameter harus berdasarkan data yang asli. Jika data palsu maka <i>return false</i> . Adapun data palsu yang dimaksud adalah <i>user</i> tersebut menggunakan data yang sama atau data milik orang lain. Data yang sama yang dimaksud adalah bahwa data tersebut pernah ada di basis data <i>food online delivery</i> . Selebihnya jika pengisian formulir valid maka <i>return true</i> .	Sukses

No	Nama	API Url	Aktor	Metode Http	Kondisi	Hasil
12	Courier Index	http://local.fod.com/api/user/courier/index	Courier	GET	Tidak ada kondisi.	Sukses
13	Courier Get	http://local.fod.com/api/user/courier/{courier_id}	Courier	GET	Jika kurir dengan kurir id yang dimaksud tidak ditemukan maka return false.	Sukses
14	Courier Find (By Keyword)	http://local.fod.com/api/user/courier/find	Courier	POST	Jika pencarian kurir berdasarkan <i>keyword</i> yang dikehendaki tidak ditemukan maka <i>return false</i> .	Sukses
15	Courier Update	http://local.fod.com/api/user/courier/{courier_id}/update	Courier	POST	Jika data kurir yang dimasukkan tidak valid, adapun yang dimaksud dengan tidak valid adalah jika kurir memasukkan <i>username</i> , <i>email</i> , ataupun nomor handphone yang dimiliki oleh <i>user</i> lainnya maka <i>return false</i> . Sebaliknya jika validasi sah maka <i>return true</i> .	Sukses
16	Vendor Create	http://local.fod.com/api/vendor/create	Vendor	POST	Jika <i>email</i> yang dikehendaki telah digunakan oleh <i>user</i> atau <i>vendor</i> lain maka <i>return false</i> .	Sukses
17	Vendor Index	http://local.fod.com/api/vendor/index	Vendor	POST	Tidak ada kondisi.	Sukses

No	Nama	API Url	Aktor	Metode Http	Kondisi	Hasil
18	Vendor Get	http://local.fod.com/api/vendor/{vendor_id}	Vendor	GET	Jika vendor_id tidak ditemukan maka <i>return false</i> .	Sukses
19	Vendor Update	http://local.fod.com/api/vendor/{vendor_id}/update	Vendor	POST	Jika data yang baru terdapat data yang telah digunakan oleh <i>user</i> lainnya, seperti <i>email</i> , <i>username</i> , dan nomor handphone maka <i>return false</i> .	Sukses
20	Meal Create	http://local.fod.com/api/meal/create	Meal	POST	Tidak ada kondisi.	Sukses
21	Meal Index	http://local.fod.com/api/meal/index	Meal	GET	Tidak ada kondisi.	Sukses

No	Nama	API Url	Aktor	Metode Http	Kondisi	Hasil
22	Meal Get	http://local.fod.com/api/meal/{meal_id}	Meal	GET	Jika <i>meal id</i> tidak ditemukan maka <i>return false</i> .	Sukses
23	Meal Update	http://local.fod.com/api/meal/{meal_id}/update	Meal	POST	Tidak ada kondisi.	Sukses
24	Meal Category Create	http://local.fod.com/api/meal/category/create	Meal Category	POST	Tidak ada kondisi.	Sukses
25	Meal Category Index	http://local.fod.com/api/meal/category/index	Meal Category	GET	Tidak ada kondisi.	Sukses
26	Meal Category Get	http://local.fod.com/api/meal/category/{meal_category_id}	Meal Category	GET	Jika kategori id tidak ditemukan maka <i>return false</i>	Sukses
27	Meal Category Update	http://local.fod.com/api/meal/category/{meal_category_id}/update	Meal Category	POST	Tidak ada kondisi.	Sukses

No	Nama	API Url	Aktor	Metode Http	Kondisi	Hasil
28	Order Create	http://local.fod.com/api/order/create	Order	POST	<i>User id</i> yang dimasukkan harus <i>user id</i> yang telah terdaftar, adapun <i>user</i> yang terlibat adalah <i>customer</i> dan <i>courier</i> .	Sukses
29	Order Index	http://local.fod.com/api/order/index	Order	GET	Tidak ada kondisi.	Sukses
30	Order Get	http://local.fod.com/api/order/{order_id}	Order	GET	Jika <i>order id</i> tidak ditemukan maka <i>return false</i>	Sukses
31	Order Update	http://local.fod.com/api/order/{order_id}/update	Order	POST	Tidak ada kondisi, karena <i>order update</i> hanya dapat melakukan perubahan terhadap keterangan.	Sukses
32	Cancellation Create	http://local.fod.com/api/order/{order_id}/cancel	Cancellation	POST	<i>Order id</i> harus merupakan <i>order</i> yang telah ada dan memiliki status nol (<i>is cancelled</i>)	Sukses
33	Cancellation Index	http://local.fod.com/api/order/cancel/index	Cancellation	GET	Tidak ada kondisi.	Sukses
34	Cancellation Get	http://local.fod.com/api/order/cancel/{cancel_id}	Cancellation	GET	Jika <i>cancel id</i> tidak ditemukan maka <i>return false</i> .	Sukses
35	Cancellation Update	http://local.fod.com/api/order/cancel/{cancel_id}/update	Cancellation	POST	Tidak ada kondisi.	Sukses

BAB V

PENUTUP

5.1 Kesimpulan

Adapun kesimpulan yang didapat adalah sebagai berikut.

1. Aplikasi *Online Food Service* menciptakan suatu teknologi terkomputerisasi yang memiliki fungsi khusus untuk dapat memberikan pelayanan pesan antar makanan kepada konsumen untuk wilayah Poltekpos; [back end api][melayani query]
2. Aplikasi *Online Food Service* memberikan solusi terhadap besarnya rasio redundansi yang terdapat dalam transaksi data dengan menciptakan suatu teknologi terkomputerisasi yang dapat mengelola transaksi tersebut berupa *RESTful API*;
3. Aplikasi *Online Food Service* membuat suatu sistem terkomputerisasi yang dapat memberikan solusi terhadap antrian panjang dalam proses bisnis pesan antar makanan di wilayah Poltekpos dengan menciptakan suatu aplikasi yang dapat melakukan manajerial yang secara relatif bersifat *multitasking*.

5.2 Saran

Adapun saran untuk aplikasi *marketing-tools* adalah sebagai berikut.

1. *Online food service* akan dikembangkan lagi menjadi suatu aplikasi yang lebih fleksibel, artinya dapat dikembangkan oleh banyak *developer* yang memiliki keahlian dalam bahasa pemrograman yang beragam. Dari sudut pandang instansi seperti Politeknik Pos Indonesia, dapat digunakan untuk kebutuhan suatu instansi tersebut dalam melakukan pesan antar makanan; [detailkan]
2. Pengembangan aplikasi pesan antar makanan akan dilanjutkan kembali sehingga diperlukan dokumentasi pemrograman, nantinya dokumentasi tersebut akan menggunakan *Swagger* yang pada umumnya digunakan oleh banyak *developer*;

3. Nantinya, aplikasi *online food service* akan diterapkan pada instansi lainnya, dimulai dari kampus selain Politeknik Pos Indonesia, dan beberapa instansi lainnya di wilayah Bandung. Adapun hal tersebut akan diterapkan bilamana sudah tidak ada *bug/error* pada aplikasi *online food service*; [kampus/instansi lainnya]

DAFTAR PUSTAKA

1. Zaki, Ali, dan SmitDev Community. 2013. *36 Menit Belajar Komputer PHP dan MySQL*. Jakarta: Elex Media Komputindo.
2. Badan Pengembangan dan Pembinaan Bahasa. 2016. *Kamus Besar Bahasa Indonesia*. Jakarta: Gramedia Pustaka Utama. Edisi Ke Lima.
3. Fauziah, Gina Resty. 2015. *Analisis Potensi Ekonomi Kreatif Berbasis Ekowisata Di Pulau Tidung Kepulauan Seribu (Studi Kasus: Kerajinan Tangan dan Kuliner Khas)*. (Online). (<http://repository.upi.edu/19401/>, diakses 10 Juli 2017).
4. Politeknik Pos Indonesia. 2017. *Tentang dan Sejarah Politeknik Pos Indonesia*. (Online). (<http://poltekpos.ac.id/sejarah/>, diakses 10 Juli 2017).
5. Susanto, Rani. 2017. *Web Service*. (Online). (<http://repository.unikom.ac.id/52260/>, diakses 10 Juli 2017).
6. Hartono, Jogiyanto. 2013. *Sistem Teknologi Informasi Bisnis (Pendekatan Strategis)*. Bandung: Salemba Empat.
7. Brainmatics. 2017. *Business Process Modelling and Notation (BPMN)*. (Online). (<http://brainmatics.com/business-process-model-and-notation-bpmn/>, diakses 10 Juli 2017).
8. Sutabri, Tata. 2012. *Analisis Sistem Informasi*. Yogyakarta: Penerbit ANDI.
9. Cloud Hosting Indonesia. 2016. *Pengertian dan Keunggulan Framework Laravel*. (Online). (<https://idcloudhost.com/pengertian-dan-keunggulan-framework-laravel/>, diakses 27 Maret 2017).
10. Pondok Programmer. 2014. *Tutorial Dasar Laravel*. (Online). (http://pondokprogrammer.com/wp-content/uploads/2014/12/tutorial_dasar_Laravel.pdf , diakses 27 Maret 2017).
11. The Hospitality and Tourism College. 2015. *Food and Beverage Services*. (Online). (<http://www.thtc.co.za/wp-content/uploads/2015/02/food-and-beverage-service-9-edition.pdf>, diakses 30 Juli 2017).

12. Siregar, Ivan Michael. 2012. *Membongkar Teknologi Pemrograman Web Service*. Yogyakarta: Gava Media.