

Rancang Bangun API untuk Odoo ERP pada Modul *Sales*

Salma Nurkhafidoh, Nurul Fajrin Ariyani, dan Abdul Munif
Departemen Informatika, Fakultas Teknologi Informasi dan Komunikasi,
Institut Teknologi Sepuluh Nopember (ITS)
e-mail: snurhafidoh33@gmail.com¹⁾, nurulfajrin@if.its.ac.id²⁾, munif@if.its.ac.id³⁾

Abstrak— Sebagai salah satu aplikasi ERP terbaik di dunia, Odoo terdiri dari banyak modul, mulai dari manajemen proyek, CRM, *Sales*, hingga penagihan pembayaran. Salah satu modul paling penting dalam Odoo yang dibahas dalam tugas akhir ini adalah modul *Sales*. Modul *Sales* merupakan modul yang berisi semua proses transaksi mulai dari pembuatan penawaran, pengiriman, penawaran kepada pelanggan, hingga pesanan yang siap difakturkan. Namun saat ini aplikasi Odoo hanya dapat menjalankan fungsionalitasnya ketika perangkat dalam keadaan *online*. Jika perangkat sedang tidak dapat mengakses atau mengalami gangguan koneksi internet, semua proses transaksi pada modul *Sales* tidak dapat dijalankan. Sehingga pengguna tidak dapat mencatat, mengubah, ataupun mengakses data-data dalam modul *Sales*. Keterbatasan tersebut dapat menghambat pekerjaan jika pengguna ingin melakukan transaksi data namun perangkat sedang tidak terhubung dengan internet. Untuk mengatasi batasan tersebut, diperlukan pengembangan aplikasi lebih lanjut pada modul *Sales*. Oleh karena itu, dalam tugas akhir ini dibuatlah API Odoo pada modul *Sales* agar aplikasi dapat dikembangkan sesuai keinginan pengembang. Selain itu, tugas akhir ini juga mengimplementasikan Firebase pada Odoo untuk melakukan pertukaran data pada modul *Sales* secara lokal ketika sedang tidak dapat mengakses internet, kemudian melakukan sinkronisasi data segera setelah mendapatkan akses internet. Dari pengujian yang telah dilakukan menunjukkan bahwa implementasi API dan *offline storage* menggunakan Firebase terhadap fitur-fitur pada modul *Sales* telah berhasil diimplementasikan.

Kata Kunci— API, ERP, Firebase, Odoo, Offline Storage

I. PENDAHULUAN

DENGAN adanya perkembangan teknologi informasi seperti saat ini, para pebisnis tidak akan kesulitan lagi untuk mengontrol semua aktivitas bisnis (mulai dari produksi, customer relation, hingga penjualan) yang dijalankan karena semua aktivitas dapat terintegrasi pada satu sistem. Dari sistem tersebut, semua pekerja dari berbagai macam divisi dapat melaporkan aktivitas dari divisi masing-masing.

Sebagai salah satu aplikasi ERP terbaik di dunia, Odoo terdiri dari banyak modul, mulai dari manajemen proyek, CRM, *Sales*, hingga penagihan pembayaran. Salah satu modul paling penting dalam Odoo yang dibahas dalam tugas akhir ini adalah modul *Sales*. Modul *Sales* merupakan modul yang berisi semua proses transaksi mulai dari pembuatan penawaran, pengiriman, penawaran kepada pelanggan, hingga pesanan yang siap difakturkan.

Dalam pengembangan modul *Sales* yang telah diterapkan sebelumnya, telah dilakukan pengembangan menggunakan metode *sure step* yang digunakan untuk mendapatkan jumlah

pasti produk yang tersedia pada waktu yang diinginkan serta penerapan metode RAD (*Rapid Application Development*) untuk menganalisa proses bisnis dan kebutuhan perusahaan untuk menyesuaikan dengan kebutuhan masing-masing perusahaan. Sedangkan saat ini aplikasi pelaporan bisnis berbasis *mobile* sangatlah diperlukan untuk mengoptimalkan pelaporan dan pengaksesan transaksi pada modul *Sales*. Untuk mengatasi hal itu, dalam tugas akhir ini dibuat API untuk mengintegrasikan aplikasi Android yang dibuat dengan memanfaatkan aplikasi Odoo ERP.

Selain itu, untuk mengintegrasikan semua data dari berbagai divisi harus dilakukan secara *online* agar semua data dapat ter-*update*. Hal ini dapat menghambat pekerjaan jika pengguna ingin memasukkan data namun sedang tidak dapat mengakses internet. Oleh karena itu, dalam artikel ini mengimplementasikan Firebase untuk menyimpan data secara lokal ketika perangkat sedang tidak dapat mengakses internet, kemudian melakukan sinkronisasi data segera setelah mendapatkan akses internet.

II. DASAR TEORI

A. Penelitian Terkait

Artikel ini membahas tentang rancang bangun API untuk Odoo ERP pada modul *Sales* dan mengembangkannya menjadi aplikasi berbasis Android yang memiliki kemampuan *offline storage* dengan memanfaatkan Firebase. Tabel 1 merupakan perbandingan beberapa pengembangan modul *Sales* yang telah dikembangkan sebelumnya dengan pengembangan modul *Sales* pada tugas akhir ini. Penelitian [1] berjudul Pengembangan Sistem Informasi Berbasis *Enterprise Resource Planning* Modul *Sales* pada Odoo dengan Metode *Rapid Application Development* di PT. Brodo Ganesha Indonesia dan penelitian [2] berjudul Penerapan

Tabel 1.
Penelitian Terkait

Penelitian	Pengembangan yang Dilakukan	ERP
Penelitian [1]	Menggunakan metode RAD (<i>Rapid Application Development</i>) untuk menganalisa proses bisnis dan kebutuhan perusahaan untuk menyesuaikan dengan kebutuhan masing-masing perusahaan.	Odoo ERP
Penelitian [2]	Menggunakan metode <i>sure step</i> untuk mendapatkan jumlah pasti produk yang tersedia di waktu yang diinginkan.	Open ERP
Rancang Bangun API untuk Odoo ERP pada Modul <i>Sales</i>	Menggunakan Firebase sebagai <i>tools</i> untuk penyimpanan data secara <i>offline</i> .	Odoo ERP

Sistem *Sales Management* dan *Warehouse Management* pada UMKM Konveksi Rajutan Berbasis OpenERP dengan Metode *Sure Step*.

B. Application Programming Interface (API)

Application Programming Interfaces (API) adalah alat yang membuat data situs web dapat dicerna untuk komputer. Karakteristik yang membuat situs web optimal bagi manusia, namun membuat sulit digunakan oleh komputer sehingga dibutuhkanlah sebuah API. Melalui API, komputer dapat melihat dan mengedit data, sama seperti seseorang yang dapat memuat halaman dan submit formulir. Apa yang mungkin membutuhkan waktu berjam-jam dengan manusia untuk menyelesaikannya, namun dapat memakan waktu beberapa detik dengan komputer melalui API.

Ketika dua sistem terhubung melalui API, dapat dikatakan bahwa dua sistem tersebut terintegrasi. Satu sisi yaitu server dan sisi lain yaitu klien. Dalam sisi server sebenarnya yang menyediakan API. Jika dalam sisi klien, dapat mengetahui data apa yang tersedia melalui API dan dapat memanipulasinya, biasanya atas permintaan dari pengguna. *Request method* nantinya akan memberitahu server tindakan apa yang diinginkan pengguna untuk diambil oleh server[3]. Terdapat empat *method* yang sering digunakan pada API, yaitu:

1. GET : Meminta server untuk mengambil data
2. POST : Meminta server untuk menambah data
3. PUT : Meminta server untuk mengedit atau memperbarui data
4. DELETE : Meminta server untuk menghapus data

C. Enterprise Resource Planning (ERP)

Enterprise Resource Planning (ERP) adalah sebuah sistem yang diperuntukkan bagi perusahaan manufaktur maupun jasa yang berperan mengintegrasikan dan mengotomasikan proses bisnis yang berhubungan dengan aspek operasi, produksi, maupun distribusi di perusahaan yang bersangkutan. ERP sendiri bermula pada kebutuhan bisnis sebuah perusahaan skala *enterprise* dalam menjalankan semua proses bisnisnya.

Beberapa modul *software* ERP digunakan untuk mengelola proses bisnis, menggunakan *database* yang tersentralisasi. Proses bisnis adalah sekumpulan aktivitas yang menerima *input* dan menghasilkan *output*. Umumnya dimulai dari memproses data yang masuk, melacak status dari penjualan, inventory, pengiriman barang, dan memperkirakan bahan baku serta kebutuhan sumber daya manusia[4].

Jadi pada dasarnya, ERP (*Enterprise Resource Planning*) menggabungkan beberapa fungsi manajemen ke dalam sistem yang terintegrasi dan memfasilitasi semua arus informasi pada fungsi manajemen tersebut. ERP dirancang untuk mengotomasikan proses-proses dasar pada seluruh proses bisnis melalui database terpusat dan menghilangkan kebutuhan sistem yang berbeda yang dikelola oleh berbagai unit kerja dalam suatu perusahaan.

D. Odoo

Odoo merupakan sebuah *platform open source* yang digunakan untuk keperluan bisnis. Aplikasi atau modul-modul yang terintegrasi dibangun di atas *platform* tersebut, meliputi semua bisnis dari CRM, *Sales*, stok, dan lain

sebagainya. Odoo dibangun menggunakan bahasa pemrograman Phyton, XML, dan JavaScript.

Odoo termasuk sebagai software *Enterprise Resource Planning* (ERP). Odoo dulunya dikenal sebagai OpenERP. Odoo dibangun secara *open source*, sehingga Odoo mendukung pemanfaatan kembali *library* yang telah ada dan setiap orang dapat terlibat dalam pengembangannya. *Platform* Odoo terdiri dari tiga komponen utama, yaitu *database* PostgreSQL sebagai *database* bawaannya, *application server* Odoo, dan *web server*. Database PostgreSQL menampung semua data yang berhubungan dengan data dan konfigurasi Odoo. Selain sebagai aplikasi, Odoo juga dapat berfungsi sebagai *framework* atau kerangka kerja bagi para Software Developer[5]. Odoo menyediakan modul-modul dasar yang mendukung fungsi bisnis, sehingga setiap modul dapat dikostumisasi sesuai dengan kebutuhan. Modul-modul tersebut terdiri dari 8 kelompok aplikasi, antara lain:

1. Aplikasi website : website builder, blog, e-commerce, forum, slides, live chat, appointments
2. Aplikasi *sales management* : Customer Relationship Management (CRM), point of sale, Sales, subscriptions
3. Aplikasi keuangan : accounting, invoicing, expenses
4. Aplikasi operasi : inventory, timesheets, project, purchase, helpdesk, documents
5. Aplikasi human resources : recruitment, employees, fleet, leaves, appraisal
6. Aplikasi marketing : marketing automation, mass mailing, events, survey
7. Aplikasi manufaktur : MRP, PLM, equipment, quality
8. Aplikasi komunikasi : discuss, eSignature.

E. Modul Sales pada Odoo

Modul *Sales* merupakan salah satu modul utama pada Odoo karena fungsi utama modul ini berfungsi untuk menangani transaksi penjualan, mulai dari membuat penawaran, melihat pesanan yang siap difakturkan, hingga melihat pesanan yang telah terkirim[6].

Tabel 2.
Menu pada modul *Sales*

No	Menu	Fungsi
1	Quotation	Membuat dan mengelola surat penawaran kepada pelanggan setelah adanya permintaan penawaran.
2	Sale order	Mencatat dan mengelola data transaksi penerimaan pesanan dari pelanggan
3	Sales team	Mengelola data tim penjualan.
4	Customer	Mengelola data pelanggan.
5	Orders to invoice	Melihat data pesanan yang siap difakturkan.
6	Product	Mengelola data produk yang dijual
7	Product Variant	Mengelola data varian produk yang dijual
8	Sales report	Melihat laporan penjualan dalam rentang waktu tertentu.
9	Quotation template	Mengelola data template penawaran.
10	Product Attribute	Mengelola data atribut produk.
11	Product Attribute Value	Mengelola data nilai atribut produk.

Tabel 3.
Perancangan API endpoint

No	Endpoint URI	Method	Deskripsi	No	Endpoint URI	Method	Deskripsi
1	/quotation	GET	Menampilkan semua data penawaran	19	/product-variant	POST	Menambah data varian produk
2	/quotation/{id}	GET	Mengambil data penawaran berdasarkan Id	20	/product-variant-edit	POST	Mengubah data varian produk
3	/quotation	POST	Menambah data penawaran	21	/product-variant/{id}	DELETE	Menghapus data varian produk
4	/quotation/{id}	DELETE	Menghapus data penawaran	22	/quotation-template	GET	Mengambil semua data quotation template
5	/quotation-edit	POST	Mengubah data penawaran	23	/quotation-template/{id}	GET	Mengambil data quotation template sesuai Id
6	/quotation-sent/{id}	POST	Mengubah status menjadi sent	24	/quotation-template	POST	Menambah data quotation template
7	/quotation-confirm/{id}	POST	Mengkonfirmasi data penawaran	25	/quotation-template-edit	POST	Mengubah data quotation template
8	/quotation-cancel/{id}	POST	Membatalkan penawaran	26	/quotation-template/{id}	DELETE	Menghapus data quotation template
9	/quotation-draft/{id}	POST	Menetapkan status sebagai penawaran	27	/product-attribute	GET	Mengambil semua data atribut produk
10	/sale-order	GET	Mengambil semua data order penjualan	28	/product-attribute/{id}	GET	Mengambil data atribut produk sesuai Id
11	/invoice-order	GET	Mengambil semua data order siap difakturkan	39	/product-attribute	POST	Menambah data atribut produk
12	/product	GET	Mengambil semua data produk	30	/product-attribute-edit	POST	Mengubah data atribut produk
13	/product/{id}	GET	Mengambil data Produk sesuai Id	31	/product-attribute/{id}	DELETE	Menghapus data atribut produk
14	/product	POST	Menambah data produk	32	/product-attribute-value	GET	Mengambil semua data nilai atribut produk
15	/product-edit	POST	Mengubah data produk	33	/product-attribute-value	POST	Menambah data nilai atribut produk
16	/product/{id}	DELETE	Menghapus data produk	34	/product-attribute-value/{id}	POST	Mengubah data nilai atribut produk
17	/product-variant	GET	Mengambil semua data varian produk	35	/product-attribute-value/{id}	DELETE	Menghapus data nilai atribut produk
18	/product-variant/{id}	GET	Mengambil data varian produk sesuai Ide				

Tabel 2. merupakan beberapa menu yang terdapat dalam modul Sales:

F. Laravel

Laravel adalah sebuah framework PHP yang dirilis dibawah lisensi MIT dengan kode sumber yang sudah disediakan oleh Github. Sama seperti framework yang lain, Laravel dibangun dengan konsep MVC yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, serta untuk meningkatkan pengalaman bekerja dengan menyediakan sintaks yang ekspresif, jelas, dan menghemat waktu. Selain itu, Laravel juga dilengkapi dengan *command line tool* yang bernama Artisan yang bisa digunakan untuk packaging bundle dan instalasi bundle melalui *command prompt*[7].

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti manipulasi data, controller, dan user interface. Berikut merupakan penjelasan dari masing-masing komponen dari MVC:

1. Model : Mewakili suatu struktur data. Biasanya berisi fungsi-fungsi yang membantu dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data, dan lain sebagainya.
2. View : Merupakan bagian yang mengatur tampilan untuk pengguna.

3. Controller: Merupakan bagian yang menjembatani model dan view.

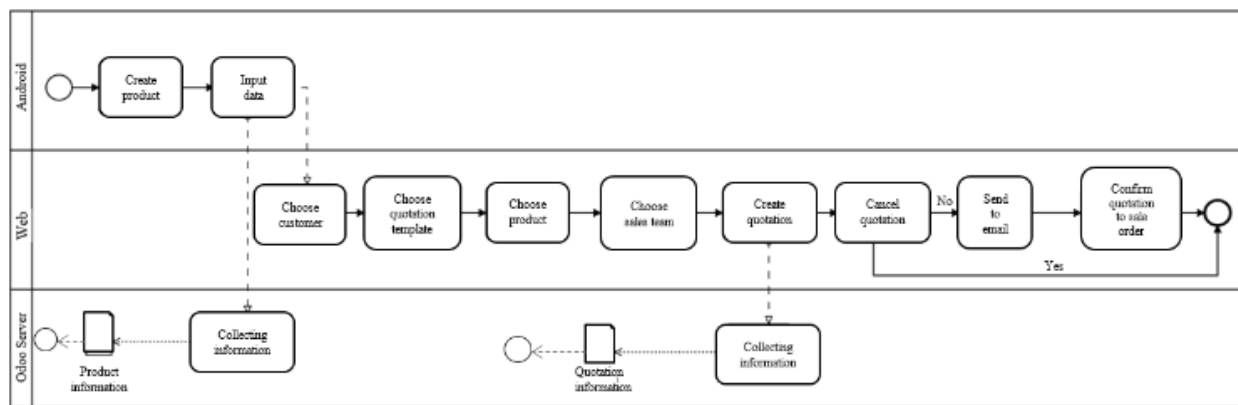
G. Laradoo

Laradoo adalah sebuah *open source package* dan merupakan implementasi dari API Odoo ERP untuk Laravel yang dikembangkan oleh Eduardo Marcos. Dengan menggunakan Laradoo, memudahkan para pengguna untuk membuat API dan mengakses langsung Odoo dengan akun miliknya sendiri. Untuk langsung dapat mengakses akun miliknya sendiri, Laradoo memiliki konfigurasi sendiri pada Laravel yang bernama laradoo.php. Pada konfigurasi ini nantinya dapat langsung menambahkan nama *database* yang sudah dibuat saat instalasi Odoo, dan *username* serta *password* akun Odoo yang sudah dibuat sebelumnya.

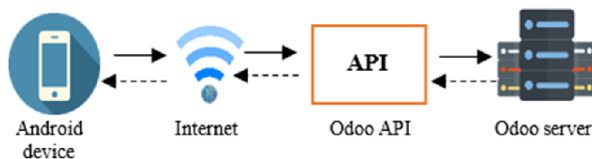
H. JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan oleh komputer. Format ini merupakan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 –Desember 1999. JSON merupakan format teks yang menggunakan gaya bahasa yang umum digunakan oleh *programmer*, seperti bahasa C, C++, C#, Java, JavaScript, Python, dll. JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama atau nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai *object*, *record*, *struct*, *dictionary*, *hash table*, *keyed list*, atau *associative array*.



Gambar 1. Proses bisnis pada modul Sales Odoo ERP



Gambar 2. Alur aplikasi dalam keadaan online



Gambar 3. Alur aplikasi dalam keadaan offline

2. Daftar nilai terurutkan (*an ordered list of values*). Pada beberapa bahasa, hal ini dinyatakan sebagai array, vector, list, atau sequence.

Struktur data ini disebut sebagai struktur data universal, karena format data mudah dipertukarkan dengan semua bahasa pemrograman modern yang juga berdasarkan pada struktur data ini[8].

I. *Firestore*

Firestore merupakan layanan *Database as a Service* dengan konsep *realtime* yang berbasis di San Fransisco, California[9]. Produk utama dari Firestore yaitu menyediakan *realtime database* dan *backend* sebagai layanan (*Backend as a Service*). Selain itu, Firestore merupakan suatu aplikasi yang memungkinkan pengembang membuat API untuk disinkronisasikan untuk *client* yang berbeda-beda dan disimpan pada *cloud* Firestore.

Dengan menggunakan produk utama Firestore, yaitu Firestore Realtime Database, aplikasi dapat disinkronkan pada semua klien secara *realtime* dan tetap tersedia meski aplikasi sedang dalam keadaan *offline*. Semua data pada Firestore Realtime Database disimpan dalam bentuk JSON dan di-host di *cloud*. Selain itu, data juga disinkronkan secara *realtime* ke setiap klien yang terhubung[10].

III. ANALISIS DAN PERANCANGAN

A. *Analisis Permasalahan*

Sebagai salah satu aplikasi ERP terbaik di dunia, tentunya Odoo memiliki banyak fitur yang menjadi kelebihanannya dibanding aplikasi ERP yang serupa. Hal ini dikarenakan Odoo mencakup semua jenis operasi bisnis yang

dibutuhkan, mulai dari manajemen proyek, CRM, hingga penjualan. Dengan banyaknya kelebihan yang dimilikinya, tentunya Odoo memiliki banyak pengguna yang mengandalkan aplikasi Odoo untuk mengintegrasikan semua data perusahaan dimanapun dan kapanpun.

Namun saat ini aplikasi Odoo hanya dapat menjalankan fungsionalitasnya ketika perangkat dalam keadaan *online*. Hal ini tentunya dapat menghambat pekerjaan jika pengguna ingin memasukkan data namun sedang tidak dapat mengakses atau mengalami gangguan koneksi internet.

Untuk mengatasi batasan tersebut, diperlukan pengembangan aplikasi lebih lanjut. Untuk itu, dalam tugas akhir ini dibuatlah API Odoo agar aplikasi dapat dikembangkan sesuai keinginan pengembang.

Selain itu, dalam artikel ini juga membahas bagaimana cara mengimplementasikan Firestore pada Odoo untuk menyimpan data secara lokal ketika sedang tidak dapat mengakses internet, kemudian melakukan sinkronisasi data segera setelah mendapatkan akses internet.

B. *Deskripsi Umum Sistem*

Pada artikel ini akan dibuat sebuah rancang bangun API untuk Odoo ERP pada modul *Sales* agar aplikasi dapat dikembangkan sebagai aplikasi berbasis Android yang dapat menerapkan *offline storage*. Pembuatan API Odoo ini akan diimplementasikan dengan memanfaatkan *Laradot* yang merupakan API Odoo ERP untuk *Laravel*.

Secara garis besar proses bisnis yang ada pada modul *Sales* ditunjukkan pada Gambar 1. Proses tersebut dimulai dengan penambahan data produk baru yang dapat dilakukan melalui perangkat Android. Setelah penambahan data, data yang berupa informasi akan tersimpan pada server Odoo. Produk yang telah dibuat dapat diteruskan untuk dimasukkan dalam form quotation. Selain data produk, data lain yang dibutuhkan untuk form quotation yaitu data customer, quotation template, dan *Sales team*. Setelah semua *field* pada form quotation terisi, data quotation baru akan tersimpan pada server. Data quotation yang telah tersimpan dapat dibatalkan jika pengguna menginginkannya. Jika quotation dibatalkan, maka proses akan otomatis berakhir. Jika data tidak dihapus, pengguna dapat meneruskan penawaran melalui email ke pelanggan. Jika pelanggan telah mengkonfirmasi penawaran tersebut, maka administrator akan mengkonfirmasi status sehingga menjadi *sale order*.

Dengan kemampuan *offline storage* yang memanfaatkan Firestore Realtime Database, aplikasi ini akan tetap dapat

Tabel 4.
Hasil pengujian fungsionalitas API

Kode Pengujian Fungsionalitas	Kasus Penggunaan	Hasil
PF – 01	Mengambil semua data penawaran	Berhasil
PF – 02	Mengambil semua data order penjualan	Berhasil
PF – 03	Mengambil semua data order siap difakturkan	Berhasil
PF – 04	Mengambil semua data Produk	Berhasil
PF – 05	Mengambil semua data <i>Quotation Template</i>	Berhasil
PF – 06	Mengambil semua data Atribut Produk	Berhasil
PF – 07	Mengambil semua data Nilai Atribut Produk	Berhasil

Tabel 5.
Skenario pengujian API

Kode Skenario	Nama Skenario	Deskripsi
A	Parameter Sesuai	Pengujian dilakukan dengan mengirim seluruh parameter dengan benar yang dibutuhkan oleh <i>endpoint</i> . Pengujian berhasil jika <i>status code</i> bernilai 200.
B	Parameter Tidak Sesuai	Pengujian dilakukan dengan mengirim sebagian parameter atau parameter yang salah yang dibutuhkan oleh <i>endpoint</i> . Pengujian berhasil jika <i>status code</i> bernilai 500.

menjalankan fungsionalitasnya walaupun dalam keadaan *offline* (tanpa internet yang terhubung). Setelah itu, sistem akan otomatis melakukan sinkronisasi ketika perangkat terhubung dengan internet.

Gambar 2 merupakan alur aplikasi jika perangkat dalam keadaan online. Untuk mendapatkan atau mengolah data, dapat diperoleh dari Odoo API yang langsung terhubung dengan Odoo server. Untuk mempermudah dalam memahami, tanda panah lurus merupakan proses request dari device ke Odoo server, sedangkan untuk tanda panah dengan garis putus-putus merupakan proses *response* dari Odoo server ke *device*.

Sedangkan pada Gambar 3 menjelaskan alur aplikasi ketika perangkat dalam keadaan *offline*. Semua data yang diakses terdapat pada Firebase Realtime Database. Untuk mempermudah dalam memahami, tanda panah lurus merupakan proses *request* dari *device* ke Firebase Realtime Database, sedangkan untuk tanda panah dengan garis putus-putus merupakan proses *response* dari Firebase Realtime Database ke *device*.

C. Perancangan API Endpoint

Pembuatan API diimplementasikan dengan memanfaatkan Laradoo. Perancangan API *endpoint* untuk modul Sales pada Odoo ERP dijelaskan pada Tabel 3.

D. Perancangan Offline Storage dan Sinkronisasi pada Android

Implementasi *offline storage* pada artikel ini menggunakan produk utama dari Firebase, yaitu Firebase Realtime Database. Firebase akan bekerja sebagai penyimpanan sementara ketika perangkat sedang dalam keadaan *offline* atau tidak ada koneksi internet dan akan melakukan sinkronisasi dengan mendorong ke server begitu perangkat kembali online. Hal ini akan terlihat pada arsitektur yang dijelaskan pada Gambar 4. Ketika perangkat dalam keadaan *offline*, pertukaran data dilakukan dengan Firebase.

Tabel 6.
Hasil pengujian API menggunakan skenario

Kode Pengujian	Nama <i>Endpoint</i>	Hasil Pengujian Skenario	
		A	B
PS – 01	Read quotation, sales order, orders to invoice	Berhasil	Berhasil
PS – 02	Store quotation	Berhasil	Berhasil
PS – 03	Update quotation	Berhasil	Berhasil
PS – 04	Delete quotation	Berhasil	Berhasil
PS – 05	Send quotation	Berhasil	Berhasil
PS – 06	Confirm quotation	Berhasil	Berhasil
PS – 07	Cancel quotation	Berhasil	Berhasil
PS – 08	Change to draft quotation	Berhasil	Berhasil
PS – 09	Read product	Berhasil	Berhasil
PS – 10	Store product	Berhasil	Berhasil
PS – 11	Update product	Berhasil	Berhasil
PS – 12	Delete product	Berhasil	Berhasil
PS – 13	Read product variant	Berhasil	Berhasil
PS – 14	Store product variant	Berhasil	Berhasil
PS – 15	Update product variant	Berhasil	Berhasil
PS – 16	Delete product variant	Berhasil	Berhasil
PS – 17	Read quotation template	Berhasil	Berhasil
PS – 18	Store quotation template	Berhasil	Berhasil
PS – 19	Update quotation template	Berhasil	Berhasil
PS – 20	Delete quotation template	Berhasil	Berhasil
PS – 21	Read product attribute	Berhasil	Berhasil
PS – 22	Store product attribute	Berhasil	Berhasil
PS – 23	Update product attribute	Berhasil	Berhasil
PS – 24	Delete product attribute	Berhasil	Berhasil
PS – 25	Store product attribute value	Berhasil	Berhasil
PS – 26	Update product attribute value	Berhasil	Berhasil
PS – 27	Delete product attribute value	Berhasil	Berhasil

Tabel 7.
Skenario Pengujian Offline Storage

Kode Skenario	Nama Skenario	Deskripsi
A	Tidak Terhubung Koneksi Internet	Pengujian dilakukan dengan melakukan semua fitur terhadap data customer dengan <i>device</i> tidak terhubung koneksi internet.
B	Sinkronisasi	Pengujian dilakukan dengan melakukan sinkronisasi terhadap perubahan data yang terjadi di antara <i>device</i> dan Odoo server.

Sinkronisasi data pada Firebase dengan server Odoo dapat dilakukan ketika perangkat kembali mendapatkan koneksi internet kembali. Saat sinkronisasi dilakukan, data dari server akan ditarik terlebih dahulu untuk mengecek apakah ada perubahan pada data yang telah tersimpan, kemudian semua data yang berada pada Firebase akan di dorong ke server Odoo.

IV. UJI COBA DAN EVALUASI

A. Hasil Pengujian Fungsionalitas API

Pada subbab ini merupakan hasil pengujian terhadap pengambilan semua data yang dikirimkan tanpa memerlukan parameter.

B. Hasil Pengujian API Menggunakan Skenario

Terdapat dua skenario pengujian yang diterapkan pada *endpoint* yang memerlukan pengiriman parameter, yaitu skenario A dan skenario B. Deskripsi dari setiap skenario dapat dilihat pada Tabel 5.

Hasil pengujian menggunakan skenario secara keseluruhan dapat dilihat pada Tabel 6.

Tabel 8.
Hasil Pengujian *Offline Storage*

Kode Pengujian	Kasus Penggunaan	Hasil Pengujian Skenario	
		A	B
PO – 01	Pengambilan data saat <i>offline</i>	Berhasil	Berhasil
PO – 02	Penambahan data saat <i>offline</i>	Berhasil	Berhasil
PO – 03	Pengubahan data saat <i>offline</i>	Berhasil	Berhasil
PO – 04	Penghapusan data saat <i>offline</i>	Berhasil	Berhasil

C. Hasil Pengujian *Offline Storage*

Pengujian dilakukan untuk mengetahui keberhasilan *offline storage* yang diimplementasikan menggunakan Firebase. Terdapat dua skenario pengujian yang diterapkan. Deskripsi dari setiap skenario dapat dilihat pada Tabel 7.

Hasil pengujian *offline storage* secara keseluruhan dapat dilihat pada Tabel 8.

D. Hasil Pengujian Sinkronisasi Data

Pengujian dilakukan terhadap sinkronisasi data pada Firebase dengan data yang ada pada server Odoo. Pengujian bertujuan untuk mengetahui apa yang terjadi apabila di antara dua perangkat melakukan penghapusan dan perubahan pada data yang sama secara *offline* dengan melakukan sinkronisasi dengan waktu yang bersamaan. Deskripsi dari setiap skenario dapat dilihat pada Tabel 9.

Hasil pengujian sinkronisasi data terhadap dua skenario dapat tersebut menunjukkan bahwa fitur yang dijalankan terlebih dahulu secara *offline* akan terlebih dahulu dieksekusi saat kedua perangkat disinkronkan secara bersamaan.

Jika dilakukan penghapusan pada perangkat A terlebih dahulu dibanding pengubahan data yang sama pada perangkat B ketika kedua perangkat dalam keadaan *offline*, kemudian kedua perangkat disinkronkan secara bersamaan, maka yang akan dieksekusi terlebih dahulu merupakan fitur penghapusan data dan pada perangkat yang melakukan pengubahan akan memunculkan pesan bahwa data terhapus oleh perangkat lain.

Sedangkan jika dilakukan pengubahan pada perangkat A terlebih dahulu dibanding penghapusan data yang sama pada perangkat B ketika kedua perangkat dalam keadaan *offline*, kemudian kedua perangkat disinkronkan secara bersamaan, maka yang akan dieksekusi terlebih dahulu merupakan fitur pengubahan lalu dilanjutkan dengan penghapusan data.

Sehingga dapat ditarik kesimpulan bahwa aplikasi mengeksekusi perintah sesuai urutan waktu penyimpanan saat *offline*. Hal ini dikarenakan penyimpanan dalam Firebase merupakan satu penyimpanan untuk semua perangkat.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Pembuatan API Odoo pada modul *Sales* untuk mengintegrasikan aplikasi *mobile* dan aplikasi web telah berhasil diimplementasikan dengan menggunakan bahasa pemrograman PHP dan memanfaatkan Laradoo. Hal ini dibuktikan dengan keberhasilan seluruh uji coba API yang ditunjukkan pada Tabel 4 dan Tabel 6.

Pada pengimplementasian *offline storage* dengan menggunakan Firebase berhasil diterapkan pada aplikasi

Tabel 9.
Skenario Pengujian Sinkronisasi Data

Kode Skenario	Nama Skenario	Deskripsi
A	Penghapusan data dilakukan terlebih dahulu secara <i>offline</i> .	Pengujian dilakukan dengan melakukan penghapusan pada perangkat A terlebih dahulu dibanding pengubahan data yang sama pada perangkat B ketika kedua perangkat dalam keadaan <i>offline</i> . Kemudian kedua perangkat disinkronkan secara bersamaan.
B	Pengubahan data dilakukan terlebih dahulu secara <i>offline</i> .	Pengujian dilakukan dengan melakukan pengubahan pada perangkat A terlebih dahulu dibanding penghapusan data yang sama pada perangkat B ketika kedua perangkat dalam keadaan <i>offline</i> . Kemudian kedua perangkat disinkronkan secara bersamaan.

berbasis Android. Pengimplementasian Firebase dapat diterapkan pada semua fitur dengan penambahan *field x_softdelete* pada model Odoo dengan tujuan agar penghapusan data pada saat *offline* tetap dapat dilakukan. Hal ini tidak mengurangi satupun fungsionalitas yang ada pada aplikasi ini. Hal ini dibuktikan dengan keberhasilan seluruh uji coba fungsionalitas yang ditunjukkan pada Tabel 8.

Dengan mengeksplorasi *offline storage* menggunakan Firebase, didapatkan beberapa hal penting, yaitu: a) Data *offline* yang tersimpan merupakan data yang tersimpan pada server Firebase, (b) Semua data pada Firebase disimpan sebagai objek JSON, (c) Dalam Firebase yang tidak berbayar, data yang dapat disimpan ke dalam cache adalah sebesar 10 MB.

B. Saran

Berdasarkan API Odoo pada modul *Sales* yang telah dibangun dalam tugas akhir ini, pengembangan modul *Sales* pada aplikasi Android dapat dilakukan dengan mengimplementasikan Firebase sebagai penyimpanan *offline*.

Untuk implementasi *offline storage* pada modul *Sales*, disarankan untuk menggunakan Firebase versi yang berbayar agar semua data dapat tersimpan dalam *cache*.

Untuk pengembangan aplikasi selanjutnya, disarankan agar aplikasi dapat digunakan pada semua zona waktu karena aplikasi yang telah dibangun dalam tugas akhir ini hanya dapat digunakan pada zona waktu GMT+7.

DAFTAR PUSTAKA

- [1] R. R. Juniar *et al.*, "Pengembangan Sistem Informasi Berbasis Enterprise Resource Planning Modul Warehouse Management Pada Odoo Dengan Metode Rapid Application Development Di Pt. Brodo Ganesha Indonesia," *eProceedings Eng.*, vol. 3, no. 2, pp. 3551–3556, 2016.
- [2] P. Setiadi, I. Darmawan, and R. W. Witjaksono, "Penerapan Sistem Sales Management Dan Warehouse Management Pada Umkm Konveksi Rajutan Berbasis Openerp Dengan Metode Sure Step," *eProceedings Eng.*, vol. 2, no. 2, 2015.
- [3] G. M. Rama and A. Kak, "Some structural measures of API usability," *Softw. Pract. Exp.*, vol. 45, no. 1, pp. 75–110, 2015.
- [4] J. A. Marakas and G. M. O'Brien, "Management information systems New York." McGraw-Hill/Irwin, 2010.
- [5] D. Reis, *Odoo development essentials*. Packt Publishing Ltd, 2015.
- [6] "An Overview of Sales Management Module with Odoo - Globalteckz." [Online]. Available: <https://www.globalteckz.com/solutions/business-solutions/openerp/openerp-modules/Sales-management-apps/>.

- [Accessed: 21-Jan-2020].
- [7] A. Aminudin, "Cara Efektif Belajar Framework Laravel." Lokomedia, 2015.
- [8] "JSON." [Online]. Available: <https://www.json.org/json-en.html>. [Accessed: 21-Jan-2020].
- [9] D. Wahyujati and others, "Implementasi Teknologi Firebase Pada Aplikasi Pencarian Lokasi Service Kamera Berdasarkan Rating Berbasis Android," STM IK AKAKOM Yogyakarta, 2017.
- [10] "Firebase Realtime Database." [Online]. Available: <https://firebase.google.com/docs/database>. [Accessed: 21-Jan-2020].