

Rancang Bangun Web Service API Aplikasi Apotek Online Menggunakan Metode REST

I Putu Fajar Tapa Mahendra^{a1}, I Wayan Santiyasa^{a2}

^aProgram Studi Informatika, Universitas Udayana
Jimbaran, Badung, Bali, Indonesia 80361

¹ftapamahendra@gmail.com

²santiyasa@unud.ac.id (Corresponding author)

Abstract

The development of information technology is very rapid, making information technology an important part of our lives that cannot be separated. Due to the effectiveness and efficiency of information technology, information technology is also used in the business world to assist in developing existing businesses, one of which is e-business. E-Business is a business that integrates information systems into a business, so that it can be accessed anywhere and anytime. To support transaction processing in e-business, we need a web service that can be accessed anywhere. In implementing the web service, the researcher uses the REST architecture. The reason for using this architecture is because this architecture is commonly used and easier to learn, as well as lightweight so it is suitable for use for information systems that require large scale. In the test, the Blackbox method is used. The blackbox method is a software test that is carried out by entering an input into the software and then checking whether the output produced by the software is in accordance with the expected results. from the test results, the results are 100% successful.

Keywords: Web Service, E-Business, REST API, Blackbox, Technology

1. Introduction

Dalam beberapa dekade terakhir ini perkembangan teknologi terutama teknologi informasi mengalami perkembangan yang pesat. Perkembangan ini membuat kita tidak bisa lepas dari dari teknologi informasi karena kita sangat bergantung pada teknologi ini. Hampir seluruh kegiatan yang kita lakukan sehari-hari tidak lepas dari dukungan teknologi informasi, baik untuk hal-hal pribadi maupun bisnis, semua bergantung pada teknologi informasi. Baik besar maupun kecil, entah apapun bidangnya, hampir semua bisnis berlomba-lomba memanfaatkan teknologi ini untuk mengembangkan usaha mereka. Penerapan teknologi informasi yang efektif sehingga mengurangi pengeluaran yang tidak diperlukan serta kinerjanya yang efisien dan cepat menjadi alasan utama. [1]

Banyaknya bisnis-bisnis yang menerapkan teknologi ini untuk mengembangkan usaha mereka menyebabkan munculnya berbagai jenis aplikasi dan layanan, mulai dari layanan *e-commerce*, *e-business*, *e-banking* dan lainnya [1]. *E-business* adalah kegiatan bisnis yang menggunakan teknologi informasi untuk mengintegrasikan sistem informasi ke bisnis. Dengan begitu segala transaksi yang dilakukan, baik itu pemesanan ataupun pengadaan barang dapat dilakukan dengan meminimalisir kontak langsung dengan pihak terkait [2]. Salah satu bisnis yang dapat melakukan *e-business* menurut penulis adalah bisnis apotek. Dengan menggunakan suatu sistem informasi, kontak antara pihak apotek dengan konsumen dapat diminimalisir.

Agar dapat menjalankan E-Business yang baik, E-business itu harus dapat dijangkau oleh orang banyak dan dimana saja. Agar dapat menjalankan e-business sehingga dapat diakses dimana saja dan di berbagai macam perangkat, diperlukan layanan *Web Service*. *Web Service* adalah suatu mekanisme komunikasi antara dua aplikasi atau mesin terlepas dari arsitektur atau teknologi yang digunakan. Pada antarmukanya, detail dari implementasi disembunyikan sehingga dapat digunakan secara terpisah dari *hardware*, *software* dan bahasa pemrograman yang digunakan [3].

Pada penelitian ini, sistem akan dibangun menggunakan arsitektur *REST API*. *Representational State Transfer* (REST) adalah salah satu arsitektur dari *web service*, yang mana arsitektur ini

bersifat *client server* dimana *client* atau dari sisi *user* mengirim *request* ke *server* dan kemudian *server* memproses *request* tersebut sebelum akhirnya mengembalikan sebuah *response* [4]. Alasan kenapa REST API digunakan, adalah karena arsitektur ini umum digunakan dan juga lebih mudah untuk dipelajari. Pada penggunaannya, data yang dikirim dapat direpresentasikan dalam berbagai format, mulai dari JSON, HTML, XML, Python, PHP, atau *plain text*. Pada penelitian ini format JSON digunakan karena lebih mudah untuk dibaca baik bagi manusia maupun mesin [5].

Berdasarkan uraian diatas, maka pada penelitian ini, peneliti akan mengangkat judul “Rancang Bangun Web Service API Aplikasi Apotek Online Menggunakan Metode REST” sebagai topik penelitian. Pada penelitian ini akan dibangun *web services* menggunakan metode *REST*. Tujuan kenapa bisnis Apotek dipilih adalah, dengan adanya sistem ini diharapkan dapat mengurangi interaksi antara pihak Apotek dengan konsumen sehingga mengurangi kemungkinan penyebaran penyakit, serta dapat mempermudah pemesanan suatu obat jikalau konsumen tidak dapat pergi ke apotek secara langsung. Penelitian ini juga diharapkan dapat menjadi tambahan wawasan bagi para pembaca.

2. Metode Penelitian

Pada penelitian ini, digunakan model pengembangan *waterfall*. Model *waterfall* adalah salah satu jenis *System Development Life Cycle* (SDLC) yang sudah banyak dikenal serta digunakan untuk membantu dalam pengembangan perangkat lunak agar menjadi lebih sistematis. Model *waterfall* menggunakan pendekatan yang berurutan dan sistematis, sehingga cocok untuk pengembangan perangkat lunak yang dimana kontrol kualitas menjadi perhatian utama. Pada model ini, pengembangan dibagi menjadi beberapa tahap yang biasanya selalu dimulai dengan fase perencanaan, analisis, perancangan dan implementasi. [6]

2.1. Perencanaan

Pada tahap ini dilakukan identifikasi masalah serta perencanaan bagaimana nantinya perangkat lunak akan dikembangkan. Pada tahap ini lebih fokus untuk menentukan gambaran kasar dari sistem yang akan dibuat.

2.2. Analisis Kebutuhan

Pada tahap ini, berdasarkan dari perencanaan tadi, akan dianalisis apa saja yang akan dibutuhkan dalam pengembangan sistem nantinya. Analisis yang dilakukan meliputi analisis data, informasi serta masalah yang dihadapi. Hal ini dilakukan agar penulis dapat menentukan solusi alternatif dari identifikasi masalah.

2.2.1. Identifikasi Masalah

Agar dapat mempermudah konsumen dalam memesan suatu obat melalui web service yang digunakan dilakukan analisis, apa saja yang kira-kira akan diperlukan untuk implementasikan ke dalam web service. Dari analisis yang dilakukan ditemukan beberapa masalah, yaitu:

- Pencarian Produk: User harus dapat dengan mudah menemukan obat yang ingin mereka pesan, baik itu pencarian berdasarkan nama obat, tipe obat, atau kategori obat.
- Resep Dokter: Untuk beberapa obat tertentu, diperlukan resep dokter untuk melakukan pemesanan terhadap obat tersebut, oleh karena itu pada web service perlu dilakukan pengecekan terhadap pesanan obat.
- Dapat digunakan oleh berbagai platform : Agar mempermudah user dalam mengakses web services, web service harus dapat diakses dimana saja dan kapan saja.

2.2.2. Analisis Data serta Informasi

Setelah identifikasi masalah, dilakukan pengumpulan informasi serta data untuk mendukung dalam pengembangan aplikasi. Dalam pengumpulan data serta informasi digunakan teknik studi literatur

2.2.3. Alternatif Solusi

Berdasarkan identifikasi masalah serta studi literatur, maka ditetapkanlah solusi berikut sebagai solusi terhadap masalah diatas.

- Diperlukan sistem yang dapat melakukan pencarian terhadap produk yang tersedia menggunakan beberapa kriteria tertentu
- Diperlukan sistem yang dapat menentukan apakah obat yang dipesan oleh customer memerlukan resep dari dokter atau tidak
- Digunakan arsitektur *REST API* sebagai arsitektur dalam pembangunan *web service*

2.3. Desain Sistem

Pada tahap ini, desain dari sistem dibuat terlebih dahulu agar lebih mudah dalam memvisualisasikan bagaimana sistem bekerja dan akan dibuat. Desain dari sistem dibuat dengan menerapkan *Unified Modelling Language* (UML). UML adalah salah satu metode yang sering digunakan untuk memodelkan sistem yang akan dibuat. Dengan menggunakan UML kita juga dapat dengan mudah melakukan pemetaan secara langsung dari model yang ada pada UML ke bahasa pemrograman berorientasi objek [7]. Pada penelitian ini digunakan diagram *use case* untuk membantu memvisualisasikan cara kerja sistem serta ERD untuk membantu memahami relasi antar entitas yang ada pada sistem.

2.4. Implementasi Sistem

Pada tahap ini, sudah dimulai pembangunan sistem menggunakan bahasa pemrograman. Pada penelitian ini digunakan bahasa pemrograman PHP dengan bantuan *framework* Laravel untuk membangun *Web Service API*. Untuk database dari sistem digunakan MySQL.

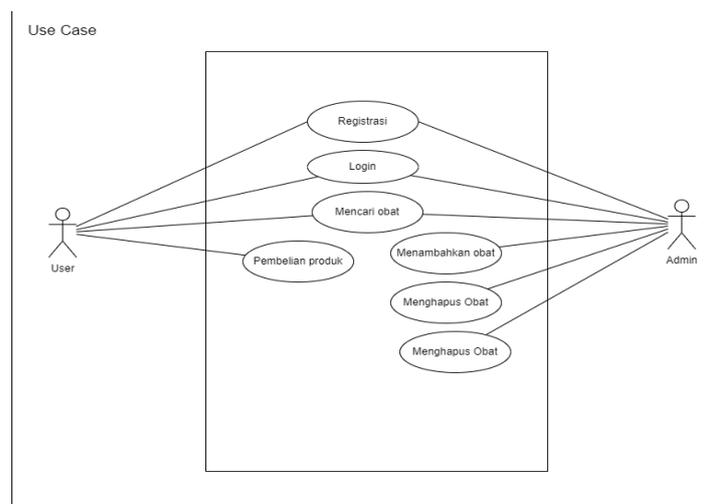
2.5. Uji Coba

Pada tahap ini dilakukan uji coba terhadap *Web Service API* yang telah dibuat. Pada penelitian ini pengujian dilakukan dengan metode *Blackbox*. *Blackbox testing* adalah pengujian perangkat lunak yang dilakukan dengan cara memasukkan suatu inputan ke dalam perangkat lunak lalu mengecek apakah keluaran yang dihasilkan oleh perangkat lunak tersebut sudah sesuai dengan hasil yang diharapkan. Pada *Blackbox testing* ini detail tentang bagaimana sistem bekerja tidak perlu diperhatikan. [8]

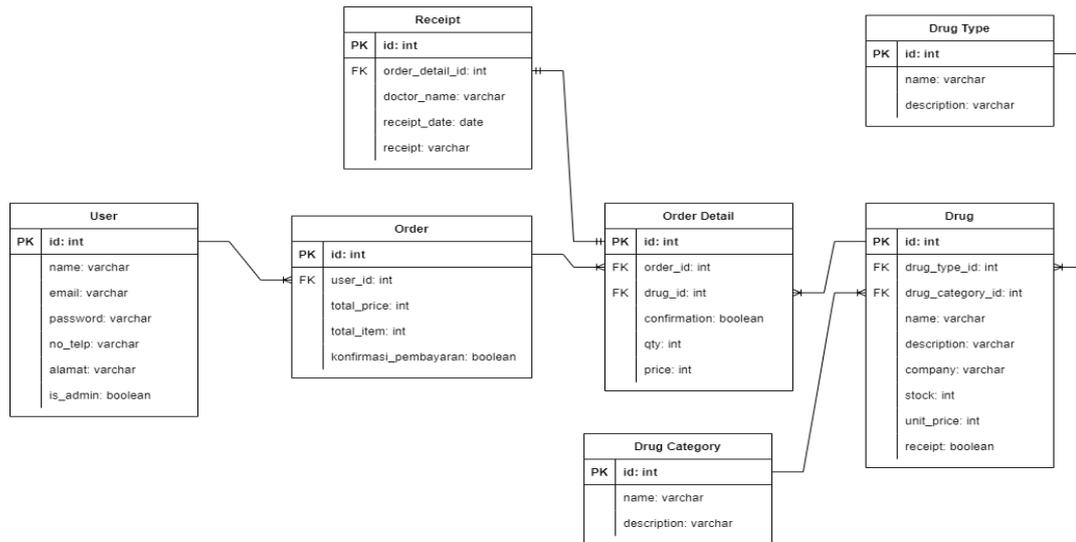
3. Hasil dan Pembahasan

3.1. Perancangan Sistem

Agar dapat membangun sistem yang baik, rancangan pemodelan dari sistem yang dibuat sangat dibutuhkan agar pada saat mengerjakan kita dapat fokus pada rancangan yang telah dibuat. Pada gambar 1 diperlihatkan rancangan diagram *use case* diagram. *Use case* diagram tersebut memperlihatkan interaksi antara user biasa dan admin dalam berinteraksi dengan sistem yang dibuat. Pada gambar 2 diperlihatkan *Entity Relationship Diagram* (ERD) untuk menggambarkan hubungan antar entitas di dalam database.



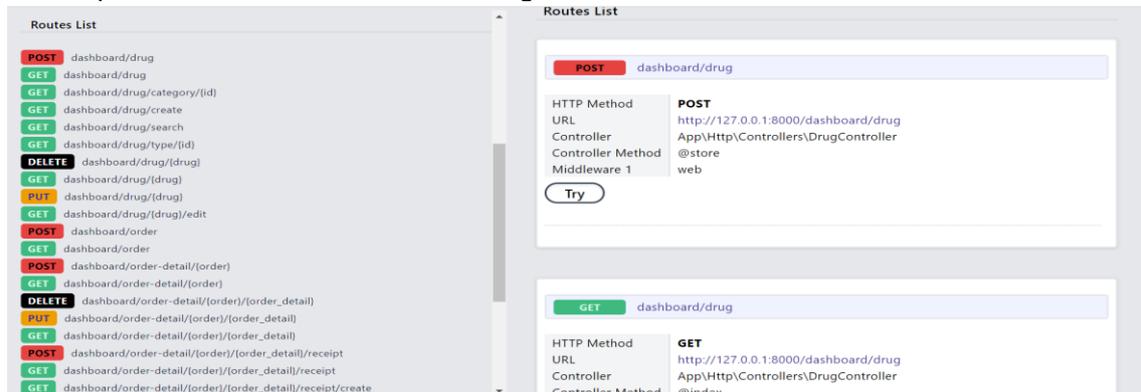
Gambar 1. Use case diagram



Gambar 2. ERD

3.2. Implementasi Sistem

Pada penelitian ini implementasi dilakukan dengan menggunakan bahasa pemrograman PHP dan juga menggunakan *framework* Laravel. Implementasi dibuat dengan arsitektur *REST API*. Untuk database dari sistem, digunakan MySQL dengan menggunakan fitur *migration* pada Laravel untuk membuatnya. Pada gambar 3 diperlihatkan hasil dari implementasi menggunakan dokumentasi dari API yang dibuat. Pada dokumentasi juga terlihat penerapan dari *REST API* karena pada dokumentasi *HTTP method* digunakan.



Gambar 3. Dokumentasi API

3.3. Pengujian

Pada bagian ini pengujian dilakukan menggunakan metode *Blackbox*. Input serta hasil dari pengujian ditampilkan pada Tabel – Tabel.

a. Registrasi

Tabel 1. Registrasi *Blackbox*

No	Request	Hasil yang Diharapkan	Hasil	Kesimpulan
1	Post Json { 'name' => 'Jack', 'email' => 'jackalive@gmail.com', 'password' => \$password, 'no_telp' => '081234567890', 'alamat' => 'kapal bajak laut', 'is_admin' => false	Dapat melakukan registrasi	Registrasi dapat dilakukan	Sesuai

	} Post Json { 'name' => 'Jack', 'email' => 'randomuser', 'password' => \$password, 'no_telp' => '081234567890' }	Tidak dapat melakukan registrasi karena email tidak valid	Registrasi tidak dapat dilakukan	Sesuai
--	---	---	----------------------------------	--------

b. Login

Tabel 2. Login *Blackbox*

No	Request	Hasil yang Diharapkan	Hasil	Kesimpulan
1	Post Json Body{ 'email' => 'randomuser@gmail.com', 'password' => 'password' }	Dapat melakukan login	Login berhasil dilakukan	Sesuai
2	Post Json Body { 'email' => 'randomuser@gmail.com', 'password' => 'pass000000' }	Tidak dapat melakukan Login karena password salah	Login gagal	Sesuai
3	Json Body { 'email' => 'randomuser', 'password' => 'pass000000' }	Tidak dapat melakukan Login karena tidak valid	Login gagal	Sesuai

c. Drug

Tabel 3. Drug *Blackbox*

No	Request	Hasil yang Diharapkan	Hasil	Kesimpulan
1	Get Json	Menampilkan seluruh senarai obat	Senarai obat ditampilkan	Sesuai
2	Get Json Param {'drug_type_id' => 1}	Menampilkan senarai obat berdasarkan tipe	Senarai obat dengan tipe yang sama ditampilkan	Sesuai
3	Get Json Param {'drug_category_id' => 22}	Menampilkan senarai obat berdasarkan kategori	Senarai obat dengan kategori yang sama ditampilkan	Sesuai
4	Get Json Body{ 'search' => 'Obat 2' }	Menampilkan senarai obat yang namanya mengandung 'Obat 2'	Senarai obat dengan nama yang mengandung 'Obat 2' ditampilkan	Sesuai
5	Get Json Param{'drug' => 3}	Menampilkan obat dengan id yang sesuai	Obat dengan id yang sesuai ditampilkan	Sesuai

6	<pre>Post Json Body{ 'name' => 'obat 4', 'description' => 'obat baru', 'company' => 'suka maju', 'stock' => 5, 'unit_price' => 45000, 'drug_type_id' => 1, 'drug_category_id' => 13, 'receipt' => true }</pre>	Menambahkan data obat baru	Data obat baru berhasil dtambahkan	Sesuai
7	<pre>Put Json Param{'drug' => 1} Body{ 'name' => 'obat nya sudah diganti', 'description' => 'obat untuk sakit kepala', 'company' => 'farmasi', 'stock' => 10, 'unit_price' => 457000, 'drug_type_id' => 1, 'drug_category_id' => 12, 'receipt' => false }</pre>	Memperbarui salah satu data obat	Data obat diperbarui	Sesuai
8	<pre>Delete Json Param{'drug' => 1}</pre>	Menghapus salah satu data obat	Data obat berhasil dihapus	Sesuai

d. Order

Tabel 4. Order *Blackbox*

No	Request	Hasil yang Diharapkan	Hasil	Kesimpulan
1	Get Json	Menampilkan senarai order yang telah dibuat user	Senarai data order ditampilkan	Sesuai
2	Post Json	Membuat order baru	Data order baru berhasil dibuat	Sesuai
3	Get Json Param{'order' => 1}	Menampilkan salah satu data order	Data order berhasil ditampilkan	Sesuai
4	Get Json Param{'order' => 2}	Tidak dapat melihat data order karena bukan user yang bersangkutan	Data order tidak ditampilkan	Sesuai
5	Delete Json Param{'order' => 2}	Tidak dapat menghapus data order karena bukan user yang bersangkutan	Data order tidak dapat dihapus	Sesuai
6	Delete Json Param{'order' => 1}	Menghapus salah satu data order	Data order berhasil dihapus	Sesuai

e. Order Detail

Tabel 5. Order Detail *Blackbox*

No	Request	Hasil yang Diharapkan	Hasil	Kesimpulan
1	Get Json Param{'order' => 1}	Menampilkan seluruh order detail yang berkaitan dengan order	Seluruh data order detail yang bersangkutan ditampilkan	Sesuai
2	Get Json Param{'order' => 1, 'order_detail' => 1}	Menampilkan salah satu order detail	Order detail ditampilkan	Sesuai
3	Post Json Param{'order' => 1} Body{'drug_id' => 3, 'qty' => 1,}	Menambahkan satu order detail	Order detail berhasil ditambahkan	Sesuai
4	Put Json Param{'order' => 1, 'order_detail' => 1} Body{'qty' => 3, 'confirmation' => false}	Mengubah data salah satu order detail	Data order detail berhasil diubah	Sesuai
5	Delete Json Param{'order'=>1, 'order_detail'=> 1}	Menghapus salah satu data order detail	Data order detail berhasil dihapus	Sesuai

f. Receipt

Tabel 6. Receipt *Blackbox*

No	Request	Hasil yang Diharapkan	Hasil	Kesimpulan
1	Get Json Param{'order' => 3, 'order_detail' => 3}	Menampilkan receipt yang bersangkutan dengan order detail	Data receipt berhasil ditampilkan	Sesuai
2	Post Json Param{'order' => 2, 'order_detail' => 2} Body{'doctor_name'=>'fajar', 'receipt_date'=>'2022-10-15', 'receipt'=> \$receipt}	Memambahkan data receipt ke order detail yang bersangkutan	Data receipt berhasil ditambahkan	Sesuai
Note: \$receipt adaha variabel untuk melakukan input file gambar dummy				
3	Post Json Param{'order' => 1, 'order_detail' => 1} Body{	Tidak dapat menambah data receipt karena obat yang diperlukan tidak	Data receipt tidak ditambahkan	Sesuai

	<pre>'doctor_name'=>'fajar', 'receipt_date'=>'2022-10-15', 'receipt'=> \$receipt}</pre>	memerlukan resep dari dokter		
	Note: \$receipt adaha variabel untuk melakukan input file gambar dummy			
4	<pre>Put Json Param{ 'order' => 3, 'order_detail' => 3, 'receipt' => 1} Body{ 'doctor_name'=>damar, 'receipt_date'=>'2022-11-15}</pre>	Mengubah data receipt	Data receipt berhasil diubah	Sesuai
5	<pre>Delete Json Param{ 'order' => 3, 'order_detail' => 3, 'receipt' => 1}</pre>	Menghapus data receipt	Data receipt berhasil dihapus	Sesuai

4. Kesimpulan

Berdasarkan hasil uji yang dilakukan, dari 29 pengujian yang dilakukan 29 berhasil lolos, sehingga dapat dikatakan bahwa hasil persentasi dari pengujian adalah 100%, sehingga dapat dikatakan *Web Service API* yang dibuat telah berjalan dengan baik. Sistem yang dibuat pada penelitian ini juga sudah menerapkan arsitektur REST sehingga API nya dapat digunakan di platform lain. Selama user memasukan URL dengan benar dan mengirimkan request yang sesuai, data yang ada pada *web service* akan dikirimkan tidak peduli *browser* yang digunakan.

References

- [1] Business Management Laboratory, "TEKNOLOGI INFORMASI DAN PERKEMBANGAN BISNIS," 2019. <https://bbs.binus.ac.id/bbslab/2019/11/teknologi-informasi-dan-perkembangan-bisnis/> (accessed Oct. 1, 2022).
- [2] Midtrans, "E-Business Adalah: Definisi, Jenis, dan Perbedaannya dengan E-Commerce". <https://midtrans.com/id/blog/e-business-adalah> (accessed Oct. 1, 2022).
- [3] Binus University, "Pengenalan Web Services", 2019. <https://socs.binus.ac.id/2019/12/26/pengenalan-web-services/> (accessed Oct. 1, 2022).
- [4] G. W. Wiku, A. Issa, and R. Bayu " Implementasi Teknologi Restful Web Service Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website " *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 2, pp. 680-689. 2020.
- [5] Red Hat, " What is a REST API?". 2020. <https://www.redhat.com/en/topics/api/what-is-a-rest-api/> (accessed Oct. 27, 2022).
- [6] A. W. Aceng, " Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi" *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*, 2020
- [7] N. P. Hendra, " mplementasi Diagram UML (Unified Modelling Language) dalam Perancangan Aplikasi Data Pasien Rawat Inap pada Puskesmas Lubuk Buaya" *Sinkron : Jurnal Dan Penelitian Teknik Informatika*, vol. 2, no. 2, p. 67-77, 2018
- [8] C. N. Fadhila et al. " Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions". *Jurnal Informatika Universitas Pamulang*, vol. 4, no. 4, p. 125-130, 2019