

Implementasi REST dan Sistem Manajemen Pada Virtual Shop Berbasis 3D

Gusti Alfian M. P^{*1}, Maskur², Lailatul Husniah³

^{1,2,3}Teknik Informatika/Universitas Muhammadiyah Malang

gustialfianmp@gmail.com^{*1}, maskur.informatika@gmail.com², husniah@umm.ac.id³

Abstrak

Virtual reality (VR) merupakan teknologi yang memungkinkan pengguna untuk merasakan sensasi di suatu tempat walaupun tidak di tempat tersebut, dengan membuat pengguna bisa berinteraksi dengan objek atau ruangan yang ditampilkan. Saat ini VR berkembang dengan pesat berkat perusahaan seperti Google dan Facebook yang melakukan investasi besar terhadap teknologi tersebut. Perkembangan VR membuka peluang bagi e-commerce untuk melakukan inovasi dengan menambahkan VR sebagai salah satu fitur. Bagi e-commerce yang menggunakan web service untuk menyalurkan data ke klien baik itu web atau mobile. Permasalahan yang didapatkan adalah bagaimana melakukan integrasi VR dengan web service yang dimiliki e-commerce. Pada penelitian ini membahas pembuatan web service berbasis Representational state transfer (REST) dan integrasi Virtual Reality dengan web service berbasis REST. Penelitian ini menghasilkan tahapan pembuatan web service dan integrasinya dengan virtual reality.

Kata Kunci: VR, REST, e-commerce

Abstract

Virtual reality (VR) is a technology that allows users to feel the sensation in place that is not in place, the user can interact with objects or the room. Currently, VR is growing rapidly thanks to companies like Google and Facebook that make substantial investments in the technology. The development of VR opens opportunities for e-commerce to innovate by adding VR as one of the features. For e-commerce that uses web services to distribute data to clients either web or mobile. The problem is how to integrate VR with web services owned by e-commerce. In this study discusses the creation of web service based on representation of state transfer (REST) and Virtual Reality integration with REST based web service. This research resulted in the creation of web services and its integration with virtual reality.

Keywords: VR, REST, e-commerce

1. Pendahuluan

Virtual reality merupakan teknologi yang memungkinkan pengguna untuk merasakan sensasi di suatu tempat walaupun tidak di tempat tersebut, dengan membuat pengguna bisa berinteraksi dengan objek atau ruangan yang ditampilkan [1]. Saat ini perusahaan besar seperti Google dan Facebook sedang berlomba untuk mengembangkan virtual reality, Google sudah memiliki platform untuk Virtual reality, yaitu Daydream dan Facebook dengan Oculusnya. Pengembangan aplikasi virtual reality sekarang sudah sampai pada ranah e-commerce yaitu virtual shop.

Virtual Shop adalah aplikasi virtual reality dimana penggunanya bisa berbelanja layaknya di toko biasa. Pada virtual shop, data yang ada akan sering berubah dan akan muncul masalah konsistensi data serta cara mengatur data yang mengalir di dalam virtual shop. Karena kebanyakan virtual reality saat ini difokuskan untuk menyajikan gambar 360 derajat dan game. Sedangkan untuk proses bisnis seperti transaksi pembelian masih sedikit.

Pada penelitian sebelumnya Wilma Waterlander meneliti penggunaan virtual reality pada minimarket yang hanya menyediakan pangan yaitu sayuran, buah, dll. Hasil pada penelitian sebelumnya cara penggunaan virtual reality mudah dipahami pengguna. Tetapi memiliki kendala untuk memvalidasi pembelian di virtual reality dibanding di dunia nyata [2].

Penelitian ini merupakan bagian dari penelitian lebih besar, mencakup beberapa komponen yang nantinya dapat diintegrasikan menjadi satu kesatuan. Peneliti berkontribusi

pada web service dan Sistem Manajemen Virtual Shop. Ada beberapa permasalahan yang ada yaitu mengatur data pada virtual reality dan memvalidasi pembelian yang didapatkan dari penelitian sebelumnya [2].

Peneliti mengajukan solusi untuk menangani permasalahan tersebut dengan membuat Sistem Manajemen Virtual Shop yang berfungsi mengatur data yang ada pada Virtual Shop, serta web service berbasis REST untuk mengintegrasikan komponen – komponen yang memerlukan data. Harapan peneliti bahwa dengan metode ini bisa menyelesaikan permasalahan sulitnya mengatur data yang ada di dalam Virtual Shop dan sulitnya memvalidasi transaksi pembelian.

2. Tinjauan Pustaka

2.1 Web Service

Web Service adalah sistem dari perangkat lunak yang dibuat untuk mempermudah menukar informasi antar perangkat lunak. Web service menggunakan teknologi web yaitu HTTP untuk menjadi perantara pengiriman atau penerimaan data. Format data yang digunakan web service adalah XML atau JSON karena web service berfokus kepada komunikasi antara perangkat lunak. Web service memiliki antarmuka yang digunakan perangkat lunak lain untuk berkomunikasi dengannya [3].

2.3 REST

REST (Representational state transfer) adalah salah satu arsitektur web service yang berfungsi untuk menyediakan cara menukar informasi antara perangkat lunak melalui HTTP. REST atau REST memungkinkan sistem untuk melakukan proses manipulasi representasi textual dari data yang sebenarnya menggunakan URI yang ditentukan secara stateless. REST bisa menggunakan JSON sebagai format data yang menjadi representasi [4].

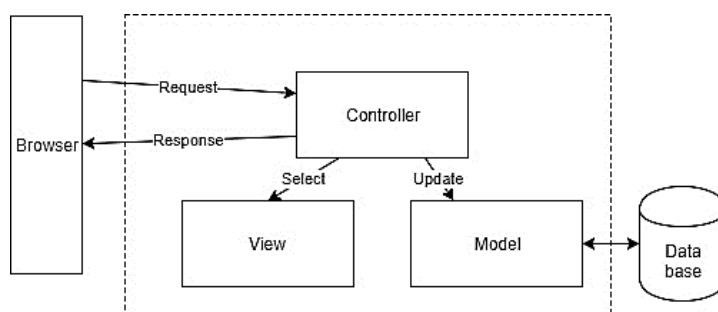
2.4 Virtual Reality

Virtual reality adalah sebuah istilah yang digunakan untuk lingkungan 3D dihasilkan oleh komputer, yang memungkinkan penggunaanya untuk masuk dan berinteraksi dengan realitas alternatif [5]. Virtual reality mulai digunakan oleh beberapa perusahaan untuk melakukan pemasaran produk mereka contoh Coca-cola, McDonald, IKEA. Selain itu perusahaan-perusahaan teknologi besar seperti Google dan Facebook sedang berlomba untuk mengembangkan platform VR merka masing-masing Google dengan Daydream dan Facebook dengan Oculus Rift.

2.5 MVC

MVC (Model View Controller) adalah salah satu jenis arsitektur perangkat lunak. MVC membagi perangkat lunak menjadi tiga bagian yang terhubung yaitu model, view, dan controller. Setiap bagian pada MVC berdiri sendiri dan memiliki fungsi yang jelas. Oleh karena itu MVC berguna dalam pengembangan modular serta meningkatkan efisiensi, perawatan sistem dan penggunaan ulang code [6].

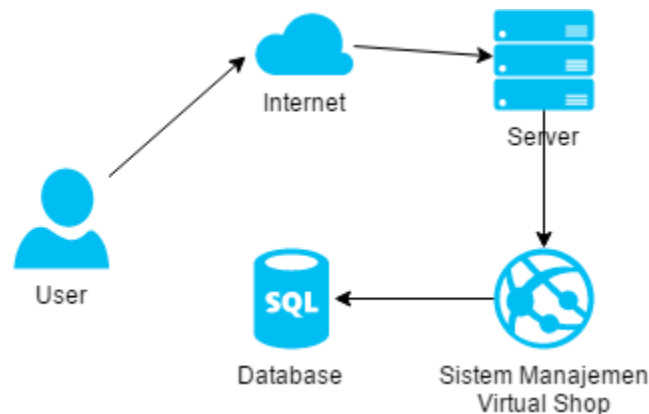
Berikut pada Gambar 1 digambarkan relasi antara bagian pada MVC. Pada saat pertama kali browser melakukan request, request tersebut diterima oleh controller. Tergantung dari route yang ada pada request controller akan mengakses model. Controller akan membaca, mengubah, atau menghapus data pada database melalui perantara model. Setelah itu controller akan memilih view yang sesuai dengan request yang dikirim browser dan mengolah template pada view dan data dari model untuk dikirimkan lagi ke browser berupa responses [7].



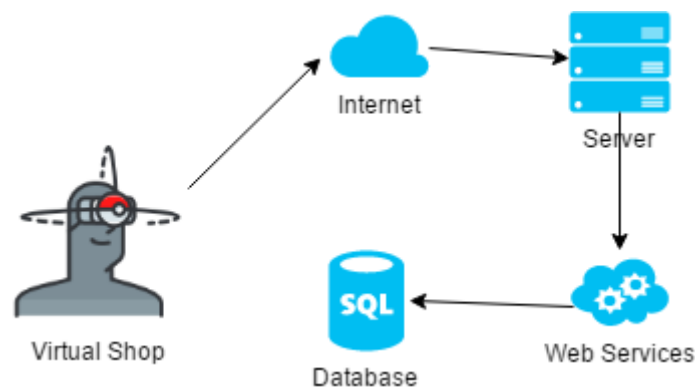
Gambar 1. Gambaran Umum MVC

3. Perancangan Sistem

3.1 Analisa Sistem



Gambar 2. Gambaran Umum Sistem Manajemen Virtual Shop



Gambar 3. Gambaran Umum REST pada VS

Pada Gambar 2 no 1 merupakan proses user melakukan request ke internet. No 2 internet sebagai perantara menghubungkan ke server. No 3 server menjalankan sistem manajemen virtual shop. No 4 sistem manajemen virtual shop melakukan query pada database. Pada Gambar 3 no 1 merupakan proses client melakukan request ke internet. No 2 internet sebagai perantara menghubungkan ke server. No 3 melakukan response pada client. No 4 pada saat sebelum melakukan responses server mengambil data yang di perlukan untuk inisiasi VR.

Sistem Manajemen Virtual Shop merupakan aplikasi berbasis web yang merupakan klien REST untuk mengatur data yang nantinya akan digunakan pada Virtual Shop. Pengguna melakukan perubahan data pada Virtual Shop menggunakan Sistem Manajemen Virtual Shop, Sistem Manajemen Virtual Shop melakukan request ke server, setelah itu server akan meneruskan ke web service, web service akan melakukan routing dari request pengguna, melakukan manipulasi data pada database dan memberikan respond untuk ditampilkan di Sistem Manajemen Virtual Shop.

Peran Virtual Shop disini sebagai klien REST berbeda dengan Sistem Manajemen Virtual Shop. Virtual Shop merupakan aplikasi VR yang menerima data dari REST dan melakukan proses bisnis berupa login, promosi, menampilkan produk, melakukan transaksi melalui beberapa endpoint pada REST.

3.2 Analisa Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang menggambarkan fungsionalitas sistem atau layanan-layanan sistem. Kebutuhan fungsional dapat berupa pernyataan-pernyataan yang menggambarkan apa yang sistem harus lakukan dan layanan-layanan apa yang dapat diberikan oleh sistem kepada pengguna secara mendetail:

- a. Melalui web service pengguna dapat menyediakan katalog barang yang dijual.
- b. Melalui web service pengguna dapat mengatur produk yang ada pada sistem.

- c. Melalui web service pengguna dapat mengatur pembelian.
- d. Melalui web service pengguna dapat mengatur pengguna dalam sistem.
- e. Melalui web service pengguna dapat mengatur tata letak barang saat dijual.
- f. Melalui web service pengguna dapat mengatur diskon yang dimiliki produk.

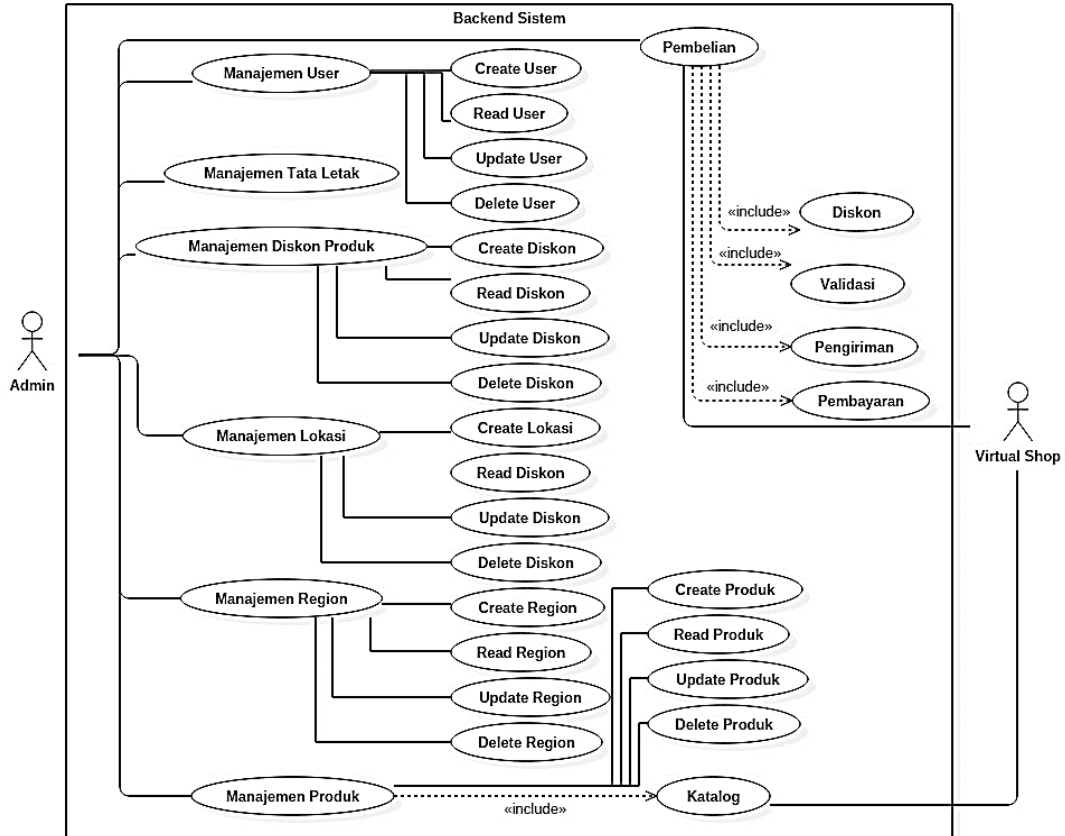
3.3 Analisa Kebutuhan Non Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang menentukan atribut atau kualitas secara keseluruhan dari suatu sistem. Kebutuhan non-fungsional menempatkan batasan pada produk yang sedang dikembangkan, proses pengembangannya, dan menentukan batasan-batasan eksternal yang harus dipenuhi oleh produk tersebut.

- a. *Usability*
Antar muka di desain untuk mempermudah penggunaan *web service* untuk *admin*.
- b. *Security*
 - 1. Hanya admin yang mempunyai wewenang untuk membuat, mengubah serta menghapus data pada *web service*.
 - 2. *Web service* menggunakan *JWT* untuk melakukan autentikasi pengguna.
- c. *Reliability*
Data tersimpan pada *web service*, dan hanya admin yang dapat melakukan pengolahan data.
- d. *Performance*
REST client minimal memiliki koneksi 3G, untuk memastikan request dan response antara *REST client* dan *web service* bisa dilakukan secara cepat.

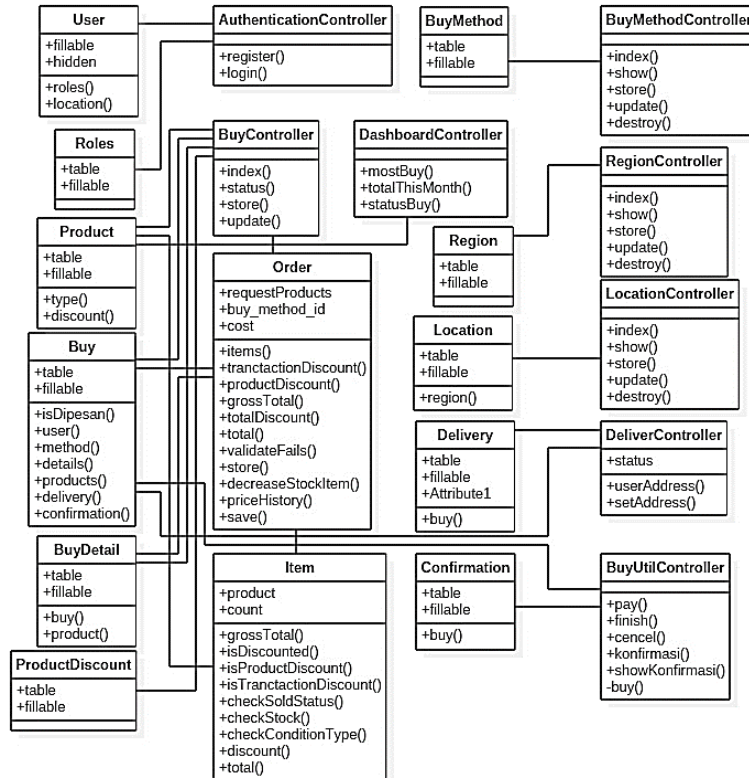
3.4 Desain Sistem

Use case dibuat berdasarkan dari kebutuhan fungsional dan non fungsional. Use case berfungsi menjelaskan kebutuhan – kebutuhan aktor dalam sistem. Pada Gambar 4 merupakan use case diagram yang dibuat.

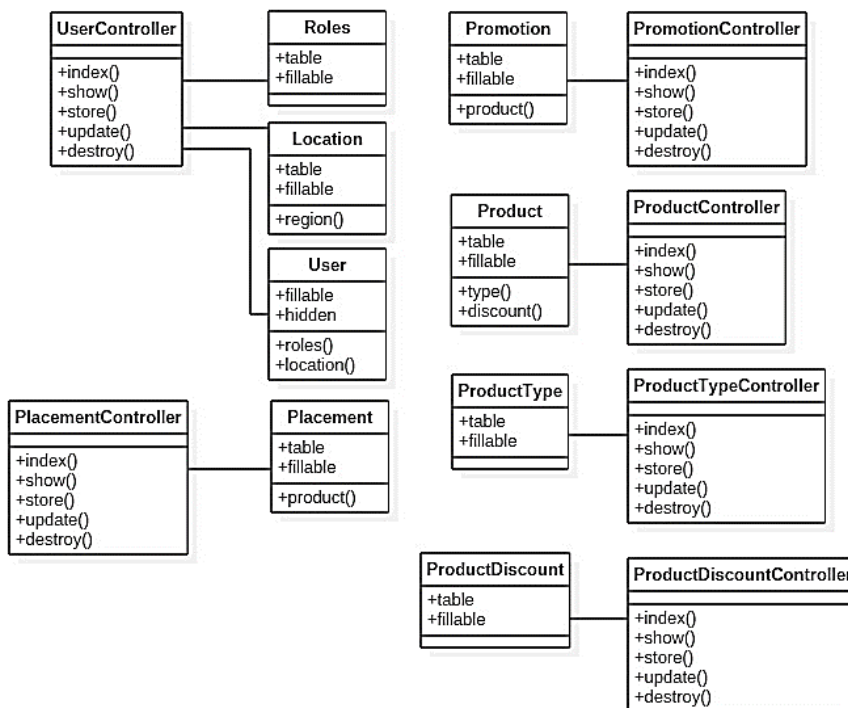


Gambar 4. Use Case Diagram

Dari use case dibuat definisi use case, scenario use case, activity diagram, sequence diagram, dan class diagram. Class diagram merupakan spesifikasi dari pengembangan dan desain berorientasi objek yang menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Pada Gambar 5 dan Gambar 6 merupakan class diagram dari sistem ini.



Gambar 5. Class Diagram 1



Gambar 6. Class Diagram 2

Selain desain sistem pada tahap ini dilakukan juga disain tampilan. Desain tampilan dari aplikasi dirancang semenarik mungkin agar pengguna dapat dengan mudah dalam mengoperasikan aplikasi ini. Pada platform WEB untuk membangun layout menggunakan HTML, CSS, dan JS. Seperti yang ditunjukkan Gambar 7 dan Gambar 8.



Gambar 7. Desain Halaman Dashboard

▼ Head 1	▼ Head 2	▼ Head 3	
Cell 1	Cell 2	Cell 3	<input type="checkbox"/>
Cell 4	Cell 5	Cell 6	<input checked="" type="checkbox"/>
Cell 7	Cell 8	Cell 9	<input type="radio"/>
Cell 10	Cell 11	Cell 12	<input type="radio"/>

Gambar 8. Desain Halaman Manajemen

4. Implementasi

4.1 Implementasi Web Service Berbasis REST

Langkah pertama dalam pembuatan sistem ini adalah pembuatan *web service* berbasis REST. Peran *web service* pada sistem ini adalah sebagai pusat pertukaran data dari semua client yang lain sistem manajemen dan VS oleh karena alasan tersebut. *Web service* yang nanti akan bangun menggunakan *framework lumen* untuk mempermudah pengerjaan [8] [9].

Hasil dari implementasi ini adalah berupa *URI* yang nantinya akan digunakan oleh VS dan Sistem Manajemen VS. Routing merupakan pemetaan antara *URI* dan *controller* yang nanti bertugas untuk mengatur logika *web service* dan menerima *request HTTP* yang nanti akan diolah dan memberikan *response*. Pada tabel 4.1 merupakan daftar route yang akan dibuat.

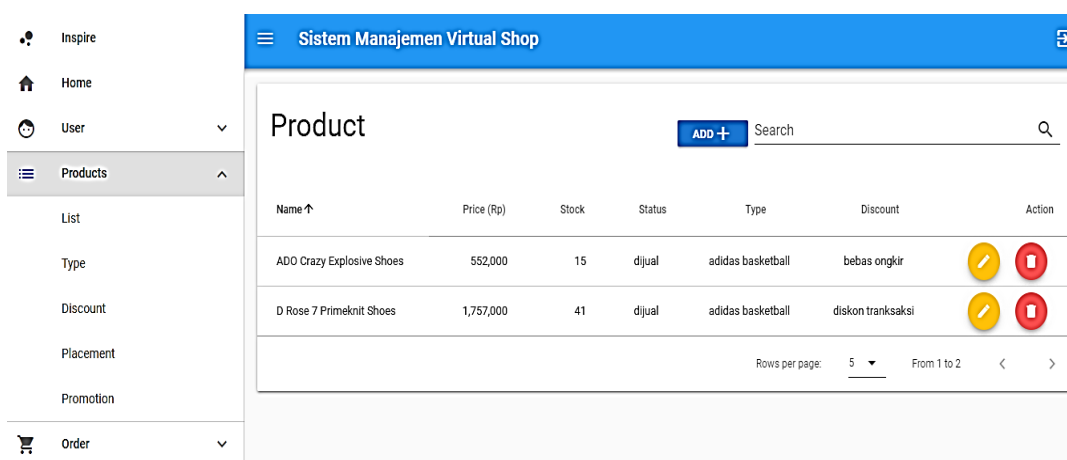
Implementasi *controller* pada *web service* ini adalah sebagai tempat bisnis proses dari sistem secara keseluruhan. Pada *controller* akan memanggil *model* untuk melakukan manipulasi data pada *database* baik itu memasukan data, membaca data, mengubah data, menghapus data semua dilakukan di *controller*. *controller* dibuat dengan cara mengetikkan *command line php artisan make:controller <nama_controller>* pada *terminal*, setelah itu akan ada file baru pada *path app/Http/Controller* difile tersebut akan dilakukan *coding*. Berikut daftar *controller* yang akan diimplementasi.

Tabel 1. Daftar Route

No	Method	URI	Controller
1	POST	/auth/register	AuthenticationController
2	POST	/auth/login	AuthenticationController
3	GET	/catalog	SellProductController
4	GET	/most-buy	DashboardController
5	GET	/total-this-month	DashboardController
6	GET	/status-buy	DashboardController
7	GET	/pay/{value}	BuyUtilController
8	GET	/cancel/{value}	BuyUtilController
9	GET	/finish/{value}	BuyUtilController
10	GET, POST	/confirmation/{value}	BuyUtilController
11	GET	/deliver/{value}/user	DeliverController
12	GET	/deliver/{value}	DeliverController
13	GET, POST	/buy	BuyController
14	GET	/buy/{value}/status	BuyController
15	GET, POST, PUT, DELETE	/buy-method	BuyMethodController
16	GET, POST, PUT, DELETE	/location	LocationController
17	GET, POST, PUT, DELETE	/product	ProductController
18	GET, POST, PUT, DELETE	/product-discount	ProductDiscountController
19	GET, POST, PUT, DELETE	/product-type	ProductTypeController
20	GET, POST, PUT, DELETE	/promotion	PromotionController
21	GET, POST, PUT, DELETE	/region	RegionController
22	GET, POST, PUT, DELETE	/users	UserController
23	GET, POST, PUT, DELETE	/product-placement	PlacementController

4.2 Implementasi Sistem Manajemen Virtual Shop

Sistem Manajemen *Virtual Shop* adalah sebuah web yang berfungsi untuk mengatur data yang pada VS. Sistem Manajemen *Virtual Shop* diimplementasi menggunakan *vuejs*. *Vuejs* adalah javascript library yang digunakan untuk membangun tampilan dari sebuah web. Sebagai contoh pada Gambar 9 merupakan hasil implementasi.



Gambar 9. Halaman Product

5. Pengujian

5.1 Pengujian web service

Pengujian *webservice REST* menggunakan *cyclometric complexity* untuk mengetahui tingkat kompleksitas dari *webservice REST*. Tahapan pengujian *cyclometric complexity* adalah pertama membuat *flow graph* dari *source code*. Setelah *flow graph* dari program tergambar, hitung jumlah *node* dan *edge flow graph*, dan hitung nilai *cyclometric complexity* menggunakan Persamaan 1 [10].

$$M = E - N + 2 \quad (1)$$

Pengujian Webservice REST ini menghitung nilai *cyclometric complexity* pada *sourcecode controller*, karena pada *controller* terletak bisnis *logic* dari sistem. Agar hasil pengujian lebih akurat maka digunakan alat bantu yaitu *phpMetrics* untuk menganalisa hasil *cyclometric complexity* dari setiap *controller*. Pada Tabel 2 merupakan hasil dari pengujian:

Tabel 2. Hasil Pengujian Webservice REST

No	Controller	Cyclometric Complexity
1	LocationController	7
2	DeliverController	8
3	ProductDiscountController	5
4	PlacementController	7
5	DashboardController	1
6	SellProductController	2
7	PromotionController	5
8	BuyMethodController	5
9	ProductTypeController	5
10	AuthenticationController	2
11	RegionController	5
12	UserController	11
13	BuyController	8
14	VshopController	2
15	ProductController	7
16	BuyUtilController	5

Pengujian performa *web servis* merupakan pengujian non fungsionalitas. Alat yang akan digunakan untuk pengujian adalah *Postman* nilai yang akan diamati dalam pengujian performa adalah waktu *response* dari *web service*. waktu *response* adalah selisih waktu antara *response web service* dengan *request web service*. *Web service* yang akan diuji menggunakan *hosting* di Domainsia. Prosedur pengujian yang akan dijalankan adalah menguji satu persatu seluruh *endpoint* yang dimiliki oleh *web service*, pada Tabel 3 merupakan hasil dari pengujian.

Tabel 3. Hasil Pengujian Performa

No	Method	URI	Response Time (ms)
1	POST	/auth/register	217
2	POST	/auth/login	207
3	GET	/catalog	122
4	GET	/most-buy	178
5	GET	/total-this-month	69
6	GET	/status-buy	67
7	GET	/pay/{value}	255
8	GET	/cancel/{value}	209
9	GET	/finish/{value}	167
10	GET	/confirmation/{value}	199
11	POST	/confirmation/{value}	152
12	GET	/deliver/{value}/user	133
13	GET	/deliver/{value}	137
14	GET	/buy	174

15	POST	/buy	231
18	GET	/buy/{value}/status	140
22	GET	/buy-method	215
23	POST	/buy-method	167
24	PUT	/buy-method	154
25	DELETE	/buy-method	142
26	GET	/location	159
27	POST	/location	184
28	PUT	/location	156
29	DELETE	/location	172
30	GET	/product	133
31	POST	/product	171
32	PUT	/product	171
33	DELETE	/product	149
34	GET	/product-discount	149
35	POST	/product-discount	154
36	PUT	/product-discount	198
37	DELETE	/product-discount	157
38	GET	/product-type	166
39	POST	/product-type	129
40	PUT	/product-type	153
41	DELETE	/product-type	246
42	GET	/promotion	74
43	POST	/promotion	243
44	PUT	/promotion	191
45	DELETE	/promotion	168
46	GET	/region	189
47	POST	/region	168
48	PUT	/region	158
49	DELETE	/region	218
50	GET	/users	207
51	POST	/users	168
52	PUT	/users	191
53	DELETE	/users	157
54	GET	/product-placement	185
55	POST	/product-placement	163
56	PUT	/product-placement	166
57	DELETE	/product-placement	192

5.2 Pengujian Sistem Manajemen *Virtual Shop*

Pengujian Sistem Manajemen VS menggunakan pengujian fungsionalitas. Apakah fungsionalitas tersebut sesuai dengan yang diharapkan atau tidak, Pada Tabel 4 merupakan adalah sebagian hasil pengujian:

Tabel 4. Hasil Pengujian Sistem Manajemen *Virtual Shop*

No	Case	Pengujian
		Deskripsi
		Pengujian untuk melakukan manajemen produk seperti melihat data, menambahkan data baru, mengubah data yang ada, menghapus data.
		Prosedur Pengujian
1	Manajemen Produk	a. Membuka halaman produk b. Menambahkan data baru c. Mengubah data yang sudah ditambahkan d. Menghapus data yang sudah ditambahkan
		Keluaran yang diharapkan
		a. Halaman produk tampil dengan daftar produk b. Data yang dimasukkan terdapat pada halaman produk

		c. Data yang diubah terdapat pada halaman produk d. Data yang dihapus tidak terdapat pada halaman produk
		Kriteria Evaluasi Hasil
		Sesuai
		Deskripsi
		Pengujian untuk memasukan data letak dari sebuah produk pada virtual shop
		Prosedur Pengujian
2	Manajemen Tata letak	a. Membuka halaman tata letak b. Memetakan produk dengan kode tempat
		Keluaran yang diharapkan
		a. Halaman tata letak terbuka dengan data yang sudah dimasukan sebelumnya b. Data yang di petakan dapat terlihat pada halaman tata letak
		Kriteria Evaluasi Hasil
		Sesuai
		Deskripsi
		Pengujian untuk melakukan tindakan – tindakan pada saat pembelian seperti pembayaran, pengiriman.
		Prosedur Pengujian
3	Pembelian	a. Mebuka halaman <i>order</i> b. Mebuka detil order c. Melakukan pembayaran d. Melakukan pengiriman
		Keluaran yang diharapkan
		a. Halaman <i>order</i> tampil dengan daftar <i>order</i> b. Halaman detil menampilkan detil <i>order</i> gambar 4.12 c. Saat melakukan pembayaran status order menjadi lunas d. Saat melakukan pengiriman status order menjadi pengiriman
		Kriteria Evaluasi Hasil
		Sesuai

5.3 Pengujian REST pada Virtual Shop

Pengujian REST pada VS menggunakan pengujian fungsionalitas. Apakah fungsionalitas tersebut sesuai dengan yang diharapkan atau tidak. Pada Tabel 5 adalah hasil pengujian:

Tabel 5. Hasil Pengujian REST Pada VS

No	Case	Pengujian
		Deskripsi
		Pengujian untuk mengambil data produk yang dijual dan ditampilkan pada VS
		Prosedur Pengujian
1	Katalog	a. Membukan VS b. Melakukan <i>request</i> ke <i>web service</i> dengan <i>URL /catalog</i>
		Keluaran yang diharapkan
		a. Data produk berhasil sampai
		Kriteria Evaluasi Hasil
		Sesuai
		Deskripsi
		Pengujian untuk mengambil data produk yang dijual dan ditampilkan pada VS
		Prosedur Pengujian
2	Pembelian	a. Membukan VS b. Melakukan <i>request</i> ke <i>web service</i> dengan <i>URL /buy</i>
		Keluaran yang diharapkan
		a. Terjadi pembelian berdasarkan produk yang dipilih

		Kriteria Evaluasi Hasil
		Sesuai
		Deskripsi
		Pengujian untuk mengambil data letak produk yang dijual dan ditampilkan pada VS
		Prosedur Pengujian
3	Manajemen Tata Letak	a. Membuka VS b. Melakukan <i>request</i> ke <i>web service</i> dengan URL <i>/product-placement</i>
		Keluaran yang diharapkan
		a. Data produk dengan kode tempat berhasil sampai
		Kriteria Evaluasi Hasil
		Sesuai

6. Kesimpulan

Dari hasil penelitian "Implementasi REST dan Sistem Manajemen pada Virtual Shop Berbasis 3D" ini, dapat disimpulkan sebagai berikut:

- 1 Hasil pengujian cyclometric complexity untuk Web service berbasis REST didapatkan rata – rata nilai cyclometric complexity untuk setiap controller adalah 5.13, nilai cyclometric complexity yang tertinggi ditemukan pada controller usercontroller yaitu 11, sedangkan nilai cyclometric complexity yang terendah ditemukan pada controller dashboardcontroller yaitu 1.
- 2 Rata – rata reponse time pada pengujian performa web service adalah 169ms, dimana endpoint yang paling lama adalah HTTP GET *"/pay/{value}"* dengan nilai response time 255ms, dan yang paling cepat adalah HTTP GET *"/status-buy"* dengan nilai response time 67ms.
- 3 Pada pengujian Sistem Manajemen Virtual Shop dimana sistem mengkonsumsi data dari web service, semua fungsionalitas pada sistem bisa berjalan sesuai harapan.
- 4 Pada pengujian RESTFUL pada Virtual Shop dimana virtual reality mengkonsumsi data dari web service, semua fungsionalitas pada sistem bisa berjalan sesuai harapan.

Referensi

- [1] R.Silva, J.C. Oliveira G.A.Giraldi, "Introduction to Virtual Reality," *LNCC Research Report, Vol. 6*, 2003.
- [2] Wilma Waterlander, "The Use of Virtual Reality in Study Complex Interventions, in Our Every-Day Food Environment".
- [3] Patrick H. Smith, Ralph E. Johnson, and Munawar Hafiz Paul Adamczyk, "REST and Web Services In Theory," 2012.
- [4] Roy Thomas Fielding, "Architectural Styles and the Design of Network-based Software Architectures," 2000.
- [5] Yudi Nugraha Bahar, "APLIKASI TEKNOLOGI VIRTUAL REALTY BAGI," 2014.
- [6] Sarker and K. Apu Iqbal H, "MVC Architecture Design and Implementation of Java Framework for Developing Desktop Application," 2014.
- [7] Zhiqiang Wei, Yongquan Yang Dandan Zhang, "Research on Lightweight MVC Framework Based on Spring MVC and Mybatis," 2013.
- [8] Lin Huang, LiJing Liang, Jing Li Wei Cui, "The Research of PHP Development Framework Based on MVC Pattern," 2009.
- [9] Njeru Mwendu Edwin, "Software Frameworks, Architectural and Design Patterns," 2014.
- [10] Geoffrey K Gill, "Cyclometric Complexity Density and Software maintenance productivity," 1997.

