

# Implementasi Arsitektur *Model View Controller* pada Website Toko *Online*

## Implementation of Model View Controller Architecture on Online Store Website

Alip<sup>1\*</sup>, Sandy Kosasi<sup>2</sup>, I Dewa Ayu Eka Yuliani<sup>3</sup>, Gusti Syarifudin<sup>4</sup>, David<sup>5</sup>

<sup>1,2,3,4,5</sup>STMIK Pontianak

alip55247@gmail.com<sup>1\*</sup>, sandykosasi@stmikpontianak.ac.id<sup>2</sup>,  
dewaayu.ekayuliani@stmikpontianak.ac.id<sup>3</sup>, guswet2907@stmikpontianak.ac.id<sup>4</sup>,  
davidLiauw@stmikpontianak.ac.id

Submitted: 18 November 2021, Revised: 23 November 2021, Accepted: 13 Desember 2021

**Abstrak** – Website merupakan media yang digunakan untuk penyampaian informasi, komunikasi, dan transaksi. Pengembangan website memiliki beberapa arsitektur diantaranya Model View Presenter, Model View ViewModel, Model View Controller dan lain-lain. Website memiliki standar pemrograman yang universal dengan arsitektur MVC dan RESTful API. REST berformat JSON dan menggunakan Json Web Token untuk hak akses, JSON memiliki ukuran kecil dibanding XML. Penelitian ini bertujuan untuk mengatasi kebutuhan yang ada pada toko Happy Belanja berupa website yang bisa digunakan sebagai media penjualan, pengelolaan barang dan cek stok produk tanpa harus berada di toko agar lebih terstruktur dan memperluas pangsa pasar dengan memanfaatkan arsitektur MVC dan RESTful API untuk menghubungkan dan mengintegrasikan framework laravel dan framework Vue.js. Framework laravel digunakan sebagai backend untuk manajemen produk yang berkaitan dengan data penjualan produk. Framework Vue.js sebagai view untuk menampilkan informasi produk dan transaksi penjualan. Hasil penelitian berupa website dengan menerapkan arsitektur MVC dan RESTful API yang memberikan kemudahan dalam pengembangan dimasa mendatang dan kemudahan dalam proses manajemen produk dan transaksi penjualan pihak Happy Belanja dapat lebih mudah dalam melakukan input data produk dan cek stok produk. Website yang telah dibuat setelah dilakukan pengujian dengan persentase keberhasilan 66,7% menunjukkan website toko online memiliki kemampuan dalam menangani data.

**Kata Kunci:** Toko Online, Model View Controller, Framework.

**Abstract** – Website is a medium used for the delivery of information, communication, and transactions. Website development has several architectures including Model View Presenter, Model View ViewModel, Model View Controller and others. The website has universal programming standards with MVC architecture and RESTful API. REST is in JSON format and uses Json Web Token for access rights, JSON is smaller than XML. This study aims to address the needs that exist in the Happy Shopping store in the form of a website that can be used as a medium of sales, management of goods and checking product stock without having to be in the store to be more structured and expand market share by utilizing the MVC architecture and RESTful API to connect and integrate laravel framework and Vue.js framework. Laravel framework is used as a backend for product management related to product sales data. Vue.js framework as a view to display product information and sales transactions. The results of the research are in the form of a website by applying the MVC architecture and RESTful API which provides convenience in future development and convenience in product management processes and sales transactions for Happy Shopping parties can be easier to input product data and check product stock. Websites that have been created after testing with a success percentage of 66,7% show that online store websites have the ability to handle data.

**Keywords:** Online Store, Model View Controller, Framework.

## 1. Pendahuluan

Tren perkembangan teknologi aplikasi berbasis *website* mengalami perubahan yang cukup signifikan dalam bentuk digitalisasi dimana pada proses pengembangan sistem website pada zaman sekarang mengikuti konsep arsitektur (*pattern*) dimana pengembangan yang dipisahkan dari proses bisnis maupun presentasi [1]. Perkembangan teknologi yang cepat pada zaman sekarang menuntut orang untuk beralih menggunakan media elektronik dengan alasan utama untuk meningkatkan kinerja bisnis dan memperluas pangsa pasar dalam meningkatkan penjualan [2].

*E-commerce* merupakan proses membeli, menjual, atau memperdagangkan data, barang, atau jasa melalui internet [3]. *E-commerce* didefinisikan sebagai transaksi komersial yang melibatkan pertukaran nilai yang dilakukan melalui atau menggunakan teknologi digital antara individu. Media *e-commerce* melibatkan penggunaan internet, *world wide web*, dan aplikasi atau browser pada perangkat mobile untuk bertransaksi. Pada awal berkembangnya *e-commerce*, satu-satunya media digital adalah web browser, namun saat ini media yang lebih banyak digunakan melalui aplikasi *mobile*, namun untuk mengatasi hal tersebut dibuatlah media digital berbasis website yang *responsive* saat digunakan pada perangkat *mobile*.

*Website* adalah lokasi di internet yang menyajikan kumpulan informasi yang berhubungan dengan profil pemilik situs. *Website* adalah suatu halaman yang memuat situs-situs *web page* yang berada di internet yang berfungsi sebagai media penyampaian informasi, komunikasi, atau transaksi [4]. MVC merupakan singkatan dari *Model View Controller*, merupakan sebuah metode untuk membuat sebuah aplikasi dengan memisahkan antara data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*). Secara umum konsep MVC merupakan suatu metode dalam pemrograman dengan memisahkan komponen utama yang membangun aplikasi, yaitu memanipulasi data, *user interface*, dan juga bagian yang mengontrol aplikasi [5].

Pengembangan sebuah *website* awalnya dibuat dengan bahasa pemrograman *Hypertext Preprocessor* (PHP) dimana arsitektur yang digunakan masih menggabungkan logika bisnis dengan presentasi (*view*) atau secara konvensional, pengembangan *website* dengan *pattern Model View Controller* (MVC) memisahkan komponen *website* menjadi beberapa bagian serta logika bisnis yang terpisah dari *view* [6]. Perancangan dan pengembangan sebuah *website* tentunya dihadapkan dengan beberapa permasalahan berupa logika bisnis yang bergabung menjadi satu dengan *view* dapat menambah biaya pengembangan saat melakukan penambahan fitur dan modifikasi dapat mempengaruhi bagian lainnya sehingga terjadi kesulitan dalam pengembangan dan penambahan fitur selanjutnya oleh *developer* yang berbeda dari sebelumnya dan memiliki ketergantungan dengan pengembang pertama.

Pemanfaatan MVC dalam membangun sebuah *website* sehingga memiliki *business logic* yang terpisah dari *model* saat melakukan modifikasi pada *coding* tidak mempengaruhi komponen lainnya yang tidak diubah dan proses pengembangan yang lebih cepat karena dapat melakukan *reuse of code* dimana fungsi ini berguna dalam pengembangan *website* tanpa harus melakukan *coding* dari awal [7]. Pihak *Happy Belanja* berkeinginan untuk memiliki sebuah *website* toko online yang dapat digunakan dalam mengelola dan memonitoring penjualan dan mengelola stok barang secara *realtime* sehingga dapat meminimalisir terjadinya selisih stok barang pada toko tanpa harus pemilik toko datang ke toko untuk melakukan pengecekan stok barang. Adanya suatu struktur organisasi tentu memiliki skala pekerjaan yang besar dengan dipisahkan bagian unit kerja menjadi kecil, sehingga pekerjaan pengolahan data barang, penjualan dan stok barang dapat saling bertukar informasi dan saling bekerja sama. Muncul suatu redundansi data pada pengolahan data barang yang dapat diselesaikan melalui integrasi data kedalam sebuah *database* serta dalam memisahkan unit kerja menjadi bagian yang kecil dan saling bertukar informasi dan data dapat saling bekerja sama dengan memanfaatkan arsitektur *Service-Oriented Architecture* [8].

*Framework* adalah kerangka kerja yang terdiri dari kumpulan *script* (terutama *class* dan *function*) yang dapat membantu *developer* dalam menangani berbagai masalah-masalah dalam

pemrograman seperti koneksi ke *database*, pemanggilan *variable*, file, dan lain-lain sehingga pekerjaan *developer* lebih fokus dan lebih cepat dalam membangun aplikasi [9]. Keuntungan menggunakan *framework* dapat mempersingkat waktu pembuatan aplikasi, kode aplikasi *website* menjadi lebih mudah dibaca, karena sedikit dan sifatnya pokok. *Website* menjadi lebih mudah diperbaiki, karena tidak perlu fokus ke semua komponen kode *website*, terutama kode *system framework*, mudah dikerjakan secara *teamwork* dengan keteraturan penulisan kode [10]. *Framework laravel* adalah sebuah *framework* PHP yang dirilis dibawah lisensi MIT dengan kode sumber yang sudah disediakan di github [11]. *Framework* Vue.Js adalah suatu *library javascript* yang digunakan untuk membangun *interface* sebuah *website* menjadi lebih interaktif, Vue.Js difokuskan hanya pada *view layer* dan mudah untuk di implementasi dan diintegrasikan dengan *library* lain ataupun *project* yang ada sebelumnya [12].

Pada arsitektur MVC memiliki *business logic* yang terpisah dari *model* dan *presentation*, saat melakukan modifikasi pada *coding* tidak mempengaruhi komponen lain yang tidak diubah, dan proses pengembangan yang lebih cepat, serta *reuse of code* dimana fungsi ini berguna dalam pengembangan *website* tanpa harus melakukan *coding* dari awal. Penelitian yang berjudul implementasi program PHP berbasis *Model View Controller* (MVC) pada *website e-commerce* studi kasus [www.indominiatur.com](http://www.indominiatur.com), penelitian yang dilakukan dengan menggunakan sistem yang memiliki arsitektur MVC (*Model View Controller*), pengembangan sistem yang lebih besar menjadi lebih mudah dilakukan karena pemisahan antara *model*, *view*, dan *controller*. Aplikasi *mobile* yang diimplementasikan dengan menggunakan *framework phonegap* pada sistem ini merupakan sebuah solusi yang tepat untuk mempermudah pengguna perangkat *mobile* untuk memperoleh informasi [13]. Penelitian yang berjudul perancangan aplikasi pelayanan *cleaning service* berbasis *web*, berupa pengembangan sistem aplikasi pelayanan *cleaning service* pada CV. Khalifah yang dibangun menggunakan bahasa pemrograman PHP dan *framework laravel*, menyediakan informasi dan pemesanan [14]. Penelitian yang berjudul perancangan aplikasi *cleaning service* berbasis *website*, implementasinya menggunakan konsep MVC dan *framework laravel*. Penelitian perancangan sistem *E-commerce* pada toko salsa *sport* berbasis *web*, membantu pemilik toko dalam mengatur transaksi pembelian, dan memberikan informasi mengenai barang pada toko salsa *sport* [15].

Penelitian ini memiliki objek yang berbeda dari penelitian sebelumnya dan pada penelitian ini memiliki perbedaan dari penelitian sebelumnya yaitu menggunakan *framework Laravel* versi 8.48.0 yang memiliki fitur-fitur terbaru seperti penambahan direktori *model*, *blade component* yang mirip dengan Vue.Js, *laravel airlock* digunakan untuk sistem otentikasi khusus *single page applications* (SPA), *mobile apps*, dan *simple token* agar lebih mudah digunakan, *HTTP Client*, *route caching*, *Blade X*, *Customizable Stubs*, *Custom Eloquent Casts*, dan *Mail Markdown Template* RESTful API dan *framework* Vue.Js. Penelitian ini bertujuan untuk menerapkan arsitektur *model view controller* pada *website* toko online dengan menggunakan *framework laravel* sebagai *server side* atau *backend* dan *framework* Vue.Js sebagai *client-side* atau *frontend* dengan menerapkan *service-oriented architecture* dan memanfaatkan RESTful API. Penelitian ini bertujuan untuk menghasilkan sebuah *website* dengan implementasi arsitektur *model view controller* pada toko *Happy Belanja* sehingga diharapkan dapat memberi kemudahan pekerjaan dalam melakukan pengelolaan barang pada toko serta memberikan kemudahan dalam cek stok barang yang ada pada toko secara *realtime*.

## 2. Metode Penelitian

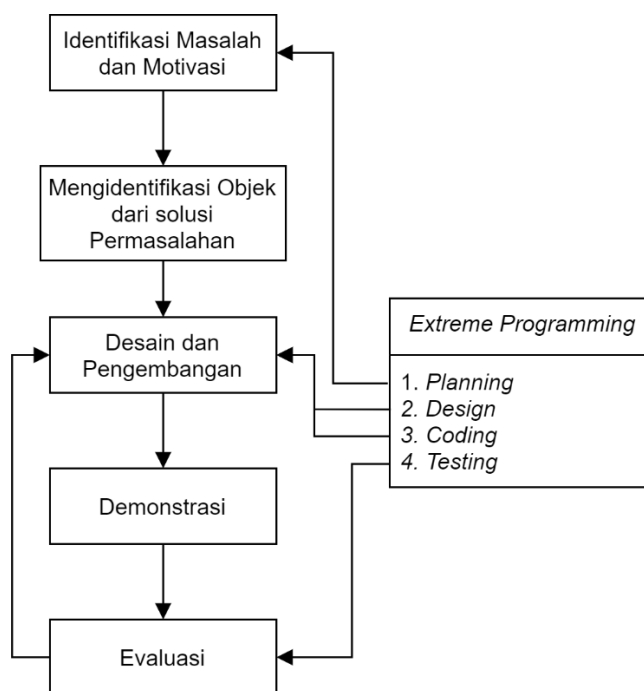
Bentuk dan metode penelitian yang penulis gunakan dalam penelitian adalah studi kasus pada toko *Happy Belanja* dan metode penelitian yang penulis gunakan adalah *Design Science Research Methodology* (DSRM) dengan pendekatan kualitatif. Tujuan menggunakan metode ini yaitu untuk melakukan implementasi arsitektur *Model View Controller* (MVC) pada *website* toko online *Happy Belanja* dengan menggabungkan prinsip, praktik, dan prosedur yang diperlukan agar penelitian yang dihasilkan memiliki tingkat keberhasilan dan kepuasan yang

baik bagi pengguna dengan tahapan yang digunakan pada penelitian ini yaitu *Identify problem and motivate*, *Define objectives of a solution*, *Design and development*, dan *Evaluation* [16].

Metode pengumpulan data yang digunakan dalam penelitian adalah data primer dan data sekunder. Data primer diperoleh secara langsung dari objek yang diteliti. Penulis melakukan observasi dan wawancara secara langsung dengan pemilik toko *Happy Belanja* untuk mendapatkan informasi dengan memberikan pertanyaan langsung secara lisan. Wawancara yang dilakukan dengan pemilik mengenai penjualan yang dilakukan secara *offline*, dan hambatan dalam proses persediaan barang yang masuk dan keluar pada *Happy Belanja*. Observasi dilakukan secara langsung untuk mengetahui bagaimana proses transaksi dan pengelolaan stok barang.

Variabel dalam penelitian ini adalah variabel tunggal yaitu implementasi arsitektur Model *View Controller (MVC)* pada *website* toko online *Happy Belanja*. Adapun aspek penelitian meliputi fitur-fitur *website* yang dibutuhkan pada toko *Happy Belanja* yaitu manajemen produk (kategori produk, produk, diskon produk, pengelolaan stok produk), manajemen pesanan, halaman utama penjualan, konten *website*, dan teknologi arsitektur *MVC* yang diimplementasikan.

Metode pengembangan perangkat lunak yang digunakan pada penelitian ini adalah *Extreme Programming* yang memiliki tahapan yaitu *planning* (perencanaan), *design* (perancangan), *coding* (pengkodean), dan *testing* (pengujian). Pengujian menggunakan metode *black box* dengan menggunakan teknik *boundary value analysis*.



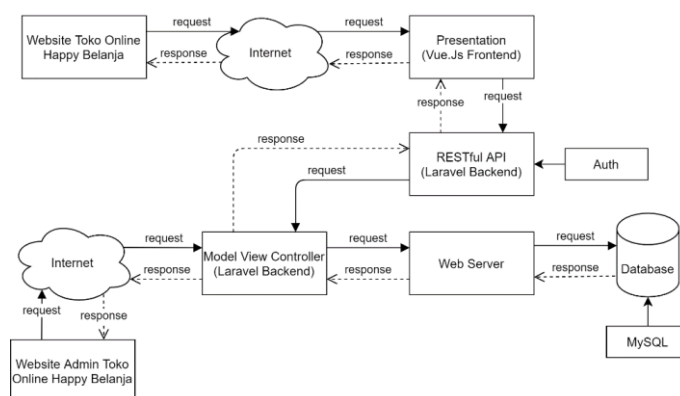
Gambar 1. Tahapan Penelitian

Tahapan penelitian yang ditunjukkan pada Gambar 1, ini menggunakan DSRM dengan tahapan awal penelitian melakukan identifikasi masalah yang terjadi pada toko *Happy Belanja* dengan melakukan wawancara dan survei langsung ke toko *Happy Belanja* serta menggabungkan *planning* dari *extreme programming* untuk menentukan perencanaan penentuan solusi yang akan diambil dalam mengatasi masalah yang terjadi pada *Happy Belanja*. Tahap Selanjutnya mengidentifikasi objek dari solusi permasalahan pada tahap ini ditentukan solusi yang dapat mengatasi permasalahan yang ditemukan sebelumnya, tahap ketiga desain dan pengembangan pada tahap ini dilakukan desain rancangan perangkat lunak menggunakan UML

berupa arsitektur *website*, *use case*, *sequence diagram*, *activity diagram*, dan *class diagram*, tahap ini melakukan desain rancangan perangkat lunak yang selanjutnya *coding* untuk mengimplementasikan desain yang telah dibuat. Pada tahap selanjutnya melakukan demonstrasi *website* toko online yang telah dbuat kepada pemilik toko dan pengguna bahwa fitur-fitur dan fungsi dari *website* telah sesuai dengan rencana awal pada tahap identifikasi masalah. Pada tahap evaluasi yaitu dilakukan pengujian *website* untuk mengetahui error dan bug yang masih terjadi pada *website* yang selanjutnya akan dilakukan kembali pada tahap desain dan pengembangan jika masih ditemukan *error* atau *bug* pada perangkat lunak.

### 2.1. Pemodelan Perangkat Lunak

Pemodelan perangkat lunak untuk melakukan Implementasi arsitektur *Model View Controller* dengan menggunakan *Unified Modelling Language (UML)* yang dimulai dari desain arsitektur *website*, desain arsitektur *model view controller*, *use case diagram*, *sequence diagram*, *activity diagram*, dan *class diagram*. Pemodelan perangkat lunak menggunakan UML. Pemodelan yang dibuat direncanakan terlebih dahulu dengan pemilik toko *Happy Belanja* untuk mempertimbangkan mengenai sistem yang akan dibangun agar selaras dengan bisnis yang sedang berjalan, sehingga memberikan kepuasan bagi pemilik dan pengguna yang dapat memberikan *feedback* positif terhadap sistem yang direncanakan sesuai kebutuhan pemilik toko dan sesuai dengan kebutuhan pengguna. Pemodelan perangkat lunak *website* toko online *Happy Belanja* dengan menerapkan arsitektur *model view controller* dari rencana awal pembuatan perangkat lunak agar nantinya akan sesuai dengan rencana tersebut maka diperlukan rancangan pemodelan yang dapat menjadi titik awal perancangan menyesuaikan dengan keinginan pemilik toko *Happy Belanja* dan *user* sebagai pengguna *website* tersebut, arsitektur *website* ditunjukkan pada Gambar

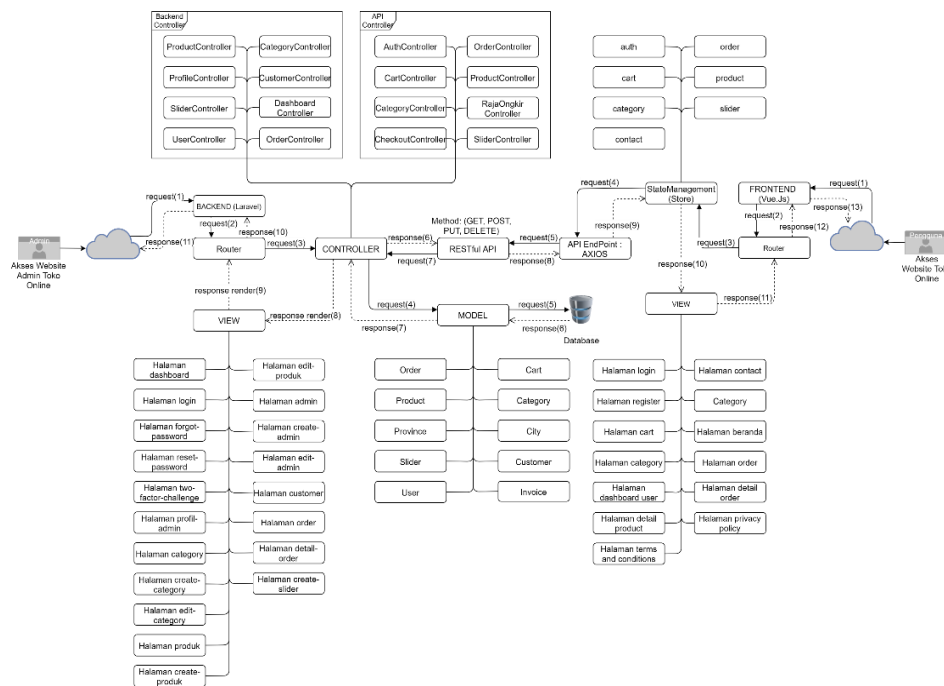


Gambar 2. Arsitektur Website

Arsitektur *website* yang ditunjukkan pada Gambar 2, yang digunakan pada toko online *Happy Belanja* pengguna dapat mengakses *website* melalui browser dengan adanya jaringan internet, browser melakukan *request* melalui alamat url yang diakses dan *framework* *Vue.js* sebagai *presentation* akan melakukan *response* dengan menampilkan halaman beranda *website* yang datanya hasil *response* dari *RESTful API* yang ada pada *backend framework laravel* dan *response* tersebut akan diteruskan ke bagian *model view controller (MVC)* untuk diproses lebih lanjut ke bagian *model* untuk melakukan *request* server dan *request* ke *database*, data hasil *request* tersebut akan dikembalikan dalam bentuk *response* oleh *controller* dan ditampilkan dalam bentuk *view* atau *presentation*.

Pengguna dapat mengakses toko secara online untuk melakukan pemesanan barang pada *website* toko online *Happy Belanja* pengguna diharuskan untuk membuat akun terlebih dahulu, jika sudah maka pengguna melakukan *login* dengan adanya fitur *Auth* maka akan ditangani oleh *Json Web Token* untuk menghasilkan kode token yang digunakan untuk hak akses keamanan yang dikelola oleh *controller* pada *framework laravel*. Implementasi arsitektur *MVC* pada

website toko online Happy Belanja untuk mengatasi kesulitan dalam maintenance dan penambahan fitur untuk kedepannya.



Gambar 3. Desain Arsitektur Model View Controller

Desain arsitektur *model view controller* ditunjukkan pada Gambar 3, pada bagian *router* berguna untuk memberi akses menuju halaman tertentu agar bisa diakses melalui web browser sehingga memudahkan saat berpindah halaman dari halaman satu ke halaman yang lainnya maupun ke halaman yang sebaliknya, pada saat *user* mengakses halaman *website* maka yang pertama melakukan *HTTP request* ke *controller* untuk mendapatkan *HTTP response* berupa *render view*.

*Controller* berguna untuk mengatur dan melakukan validasi data proses *request* dari pengguna baik itu antara *model* dan *view* yang saling berhubungan, penggunaan *controller* pada *website Happy Belanja* berupa *controller* untuk mengatur *backend* seperti *ProductController*, *CategoryController*, *CustomerController*, *ProfileController*, *SliderController*, *UserController*, *DashboardController*, *OrderController* bagian-bagian tersebut berguna untuk mengatur *request* dari pengguna sehingga pengguna dapat mengelola dan mengatur produk serta penjualan pada *website* di bagian *backend*, sedangkan bagian *API controller* merupakan *controller* yang mengatur proses *request* pengguna berupa *login*, *register*, transaksi dan *request* data produk melalui *framework* *Vue.js* yang menjadi *view* pada *website*.

*Happy Belanja* berupa *AuthController*, *CartController*, *CategoryController*, *CheckoutController*, *OrderController*, *ProductController*, *RajaOngkirController*, dan *SliderController* sehingga apa yang menjadi keinginan pengguna yang ditampilkan dengan *view* dapat terpenuhi dengan semestinya.

*Model* berguna untuk menerima *request* dari *controller* dan mengirimkan *response* sesuai permintaan dari *request* pengguna, berupa data yang berhubungan langsung dengan *database*, baik itu berupa data pelanggan, produk, kategori, admin, *banner slider*, dan data transaksi penjualan. *Model* juga berguna untuk melakukan relasi antar tabel dengan memanfaatkan fitur *eloquent relationships* seperti *one to one*, *one to many*, *many to many*, dan *one to many (inverse)*, *model* yang digunakan pada *Happy Belanja* berupa *Order*, *Cart*, *Product*, *Category*, *Province*, *City*, *Slider*, *Customer*, *User*, *Invoice*.

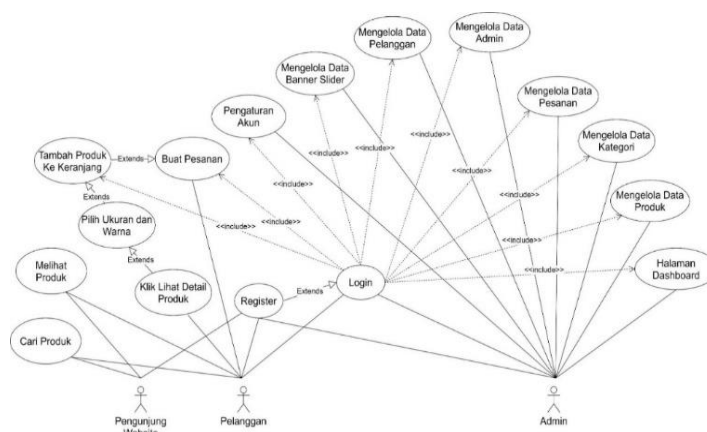
View berguna untuk menampilkan *render* halaman *website* sesuai proses *request* dari pengguna, sehingga bagian ini berhubungan langsung ke *user* melalui HTTP *request* yang diatur dan diarahkan oleh *router* baik berpindah halaman dari halaman yang satu ke halaman yang lainnya. Pada *backend Happy Belanja view* yang digunakan berupa halaman *dashboard*, halaman *login*, halaman *forgot password*, halaman reset *password*, halaman *two factor challenge*, halaman profil admin, halaman *category*, halaman tambah *category*, halaman edit *category*, halaman produk, halaman tambah produk, halaman edit produk, halaman admin, halaman edit admin, halaman *customer*, halaman *order*, halaman detail *order*, halaman tambah *banner*. RESTful API berguna untuk menghubungkan antara *framework laravel (backend)* dengan *framework Vue.js* yang dapat digunakan sebagai proses *GET, POST, PUT, DELETE* sehingga dapat melakukan pertukaran data melalui HTTP *request* dan HTTP *response* dengan adanya API *controller* yang selanjutnya dihubungkan dengan API *Endpoint*.

API *Endpoint* menggunakan *AXIOS* merupakan bagian dari *framework Vue.js* yang menerima API *Endpoint* dari *framework laravel* melalui RESTful API sehingga bisa terhubung dengan API *controller* yang ada pada *framework laravel*, jika telah terhubung maka *user* dapat melakukan proses *GET, POST, PUT, DELETE* pada *frontend framework Vue.js*.

*State Management (store)* merupakan bagian yang melakukan *request* data dari API *Endpoint* dan menyimpan data tersebut kedalam *state* dengan adanya *Vuex* sehingga data-data tersebut dapat ditampilkan pada *website* tanpa harus melakukan *refresh* halaman yang membuat *website* menjadi *single page application (SPA)*, pada *Happy Belanja* terdapat file yang mengatur dan menerima data berupa *response* seperti *auth, cart, order, category, product, slider, contact* dan akan mendapatkan *response* data ditampilkan pada *view*.

View berguna untuk menampilkan *render* halaman *website* sesuai proses *request* dari pengguna, sehingga bagian ini berhubungan langsung ke pengguna melalui HTTP *request* yang diatur dan diarahkan oleh *router* baik berpindah halaman dari halaman yang satu ke halaman yang lainnya. Pada *frontend Happy Belanja view* yang digunakan berupa halaman *login*, halaman *register*, halaman *cart*, halaman *category*, halaman *contact, category*, halaman beranda, halaman *order*, halaman *dashboard user*, halaman detail *order*, halaman detail *product*, halaman *privacy policy*, dan halaman *term and conditions*

*Router* berguna untuk mengatur, dan mengarahkan serta memberi akses menuju halaman tertentu agar bisa diakses melalui *web browser* sehingga memudahkan saat berpindah halaman dari halaman satu ke halaman yang lainnya maupun ke halaman yang sebaliknya, pada saat pengguna mengakses halaman *website* maka yang pertama melakukan HTTP *request* ke *State Management* dan menampilkan *render view* sebagai HTTP *response*.



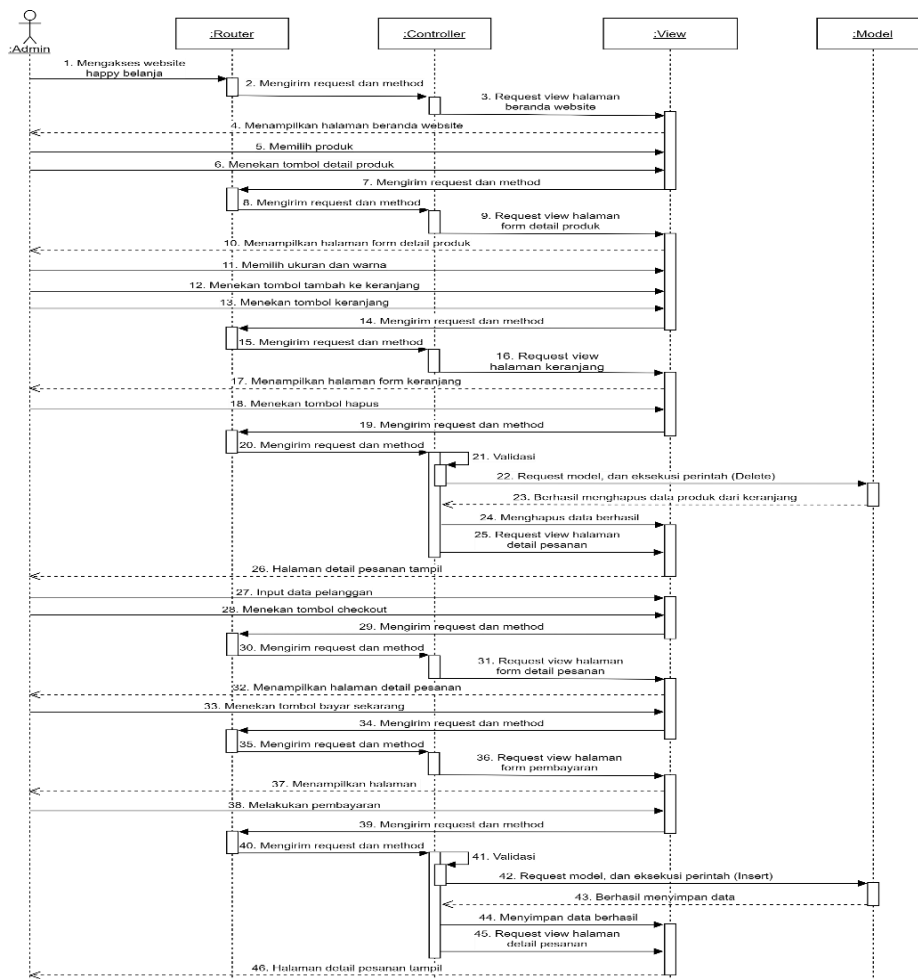
Gambar 4. Use Case Diagram

Use case diagram ditunjukkan pada Gambar 4, terdapat aktor admin, aktor *user* dan aktor pengunjung *website*. Admin mendapatkan hak akses dengan akun *default* yang diberikan saat

website selesai dibuat, namun admin dapat memperbarui info dan password untuk login untuk mengakses semua fitur maka admin harus login terlebih dahulu namun jika admin lupa password, bisa mengklik bagian lupa password, fitur yang dapat diakses oleh admin yaitu pengaturan akun, mengelola data banner, mengelola data pelanggan, mengelola data admin, mengelola data pesanan dari pelanggan, mengelola data kategori, dan mengelola data produk.

Aktor user dapat melihat produk terlebih dahulu tanpa harus melakukan register pada website namun jika user ingin melakukan penambahan barang ke keranjang dan pembelian barang harus melakukan register dan login terlebih dahulu atau jika belum memiliki akun bisa melakukan register terlebih dahulu, untuk menambahkan produk ke keranjang user dapat memilih produk mana yang akan dibeli selanjutnya dengan mengklik pada produk untuk melihat detail produk hal ini dilakukan agar user dapat memilih ukuran dan warna yang akan dibeli sesuai keinginan, setelah semua terpilih maka user mengklik tambah ke keranjang yang pada halaman keranjang user dapat menghapus produk yang tidak ingin dia beli dan bisa melanjutkan pembelian dengan mengisi form data pembelian berupa identitas dan alamat pembeli, setelah selesai maka user harus membayar sejumlah uang dari total belanjaan yang sebelumnya dilakukan dengan memilih metode pembayaran.

Aktor pengunjung website hanya bisa mengakses website untuk melihat produk dan mencari produk tanpa harus melakukan register dan login pada website, pengunjung website bisa melakukan register pada website toko online Happy Belanja dengan memasukkan nama, email, dan password setelah berhasil register dan memiliki akun maka user bisa melakukan login dan melakukan order produk pada website.

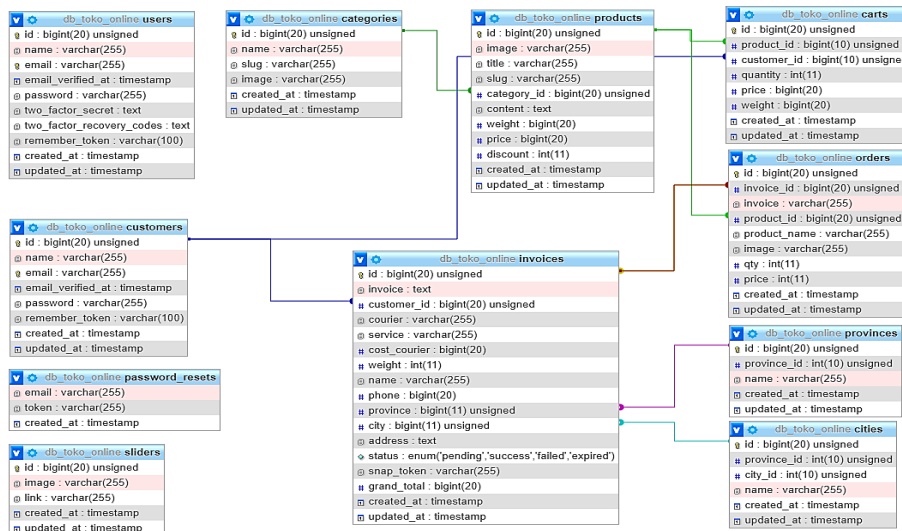


Gambar 5. Sequence Diagram Order



Sequence diagram order ditunjukkan pada Gambar 5, pelanggan dapat melakukan pembelian produk setelah dengan syarat harus memiliki akun, dan login ke website toko online Happy Belanja, pelanggan dapat memilih produk dan mengklik tombol detail produk kemudian sistem menampilkan halaman detail produk dan pelanggan menambahkan produk ke dalam keranjang setelah menambahkan produk ke keranjang pelanggan mengklik keranjang belanja user dapat memilih menghapus produk atau melanjutkan proses checkout dengan input data setelah melakukan input data dan memilih layanan pengiriman dan sistem akan menampilkan jumlah bayar beserta ongkos kirim. Selanjutnya sistem menampilkan halaman pembayaran dan pelanggan memilih metode pembayaran dan sistem menampilkan detail pembayaran. Kemudian pelanggan melakukan pembayaran melalui metode yang telah dipilih dan jika pembayaran berhasil maka sistem akan menampilkan halaman pesanan pelanggan.

Pelanggan yang berhasil melakukan pembelian dan pembayaran produk melalui yang dipilih tersebut akan menerima status *order success* yang artinya pelanggan tersebut telah berhasil melakukan pembayaran dan status no. resi pengiriman adalah pesanan diproses yang artinya pemilik toko sedang melakukan packing barang untuk dilakukan pengiriman berdasarkan jasa pengiriman yang dipilih oleh pelanggan dan jika telah selesai dilakukan pengiriman, maka admin melakukan *update* no. resi pengiriman agar pelanggan bisa mengecek secara langsung barang yang dipesan tersebut melalui jasa pengiriman yang dipilih. Status pesanan pelanggan akan berubah jika pesanan telah sampai dan diterima oleh pelanggan dan untuk mengetahui informasi pesanan dari pelanggan bisa dicek di *dashboard* pelanggan.



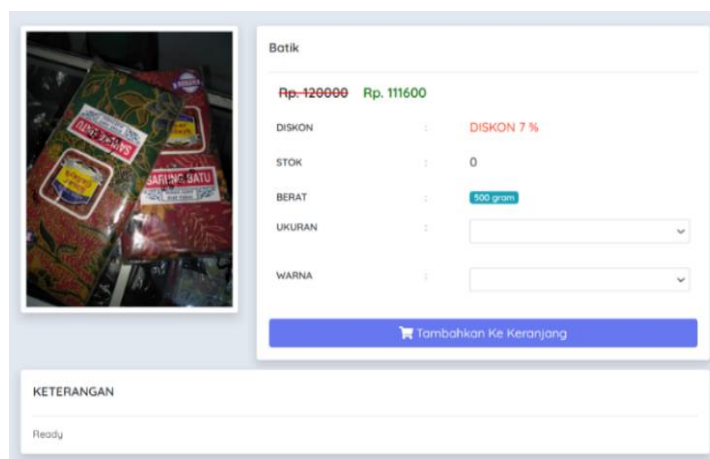
Gambar 6. Diagram Hubungan Entitas

Diagram hubungan entitas ditunjukkan pada Gambar 6, tabel-tabel pada diagram hubungan entitas diatas menunjukkan relasi antar penyimpanan dan data yang memberikan informasi mengenai tabel-tabel yang digunakan pada website toko online Happy Belanja. Hubungan yang terdapat pada tabel-tabel yang digunakan website toko online Happy Belanja dimana tabel categories memiliki relasi one to many dengan tabel products, tabel customers memiliki relasi many to many dengan tabel products, tabel customers memiliki relasi one to many dengan tabel invoices, tabel products memiliki relasi one to many dengan tabel carts dan tabel products juga memiliki relasi one to many dengan tabel orders, tabel invoices memiliki relasi one to one dengan tabel provinces dan tabel invoices juga memiliki relasi one to one dengan tabel cities. Relasi pada beberapa tabel tersebut menggambarkan relasi antar penyimpanan data dalam sebuah sistem website toko online Happy Belanja.

### 3. Hasil dan Pembahasan

Langkah awal penelitian dengan melakukan identifikasi masalah dan motivasi menggunakan tahapan pada metode penelitian DSRM yang diselaraskan dengan metode pengembangan perangkat lunak menggunakan *Extreme Programming* yaitu tahap *planning* (perencanaan) dimana tahap tersebut digunakan untuk mengidentifikasi masalah yang terjadi pada objek penelitian untuk mengetahui apa saja kebutuhan pengguna dan kebutuhan perangkat lunak yang akan digunakan dan untuk mendapatkan masukan dan saran dari pemilik toko serta data yang diperlukan untuk pengembangan perangkat lunak. Pengumpulan data dilakukan dengan observasi langsung ke objek penelitian, melakukan wawancara kepada pemilik toko, dan melakukan studi dokumentasi.

Adapun yang menjadi kebutuhan pengguna pada toko *Happy Belanja* berupa desain *web* harus bisa menampilkan informasi produk secara rinci dan mudah diakses oleh pengguna, Informasi yang ditampilkan dalam *website* harus lebih interaktif, pelanggan yang ingin melakukan pemesanan produk harus melakukan *login* terlebih dahulu agar dapat melanjutkan ke tahap *checkout* dan pembayaran, setiap pengunjung *website* dapat melakukan akses informasi dengan bebas kecuali ingin melakukan pemesanan barang, untuk mempermudah dalam rangka mengetahui bahwa pelanggan telah melakukan pembayaran, maka perlu adanya fitur yang dapat memberitahukan kepada admin jika pelanggan telah berhasil melakukan pembayaran, dan Autentikasi perlu dilakukan untuk tetap menjaga keamanan informasi dari *website* secara keseluruhan.



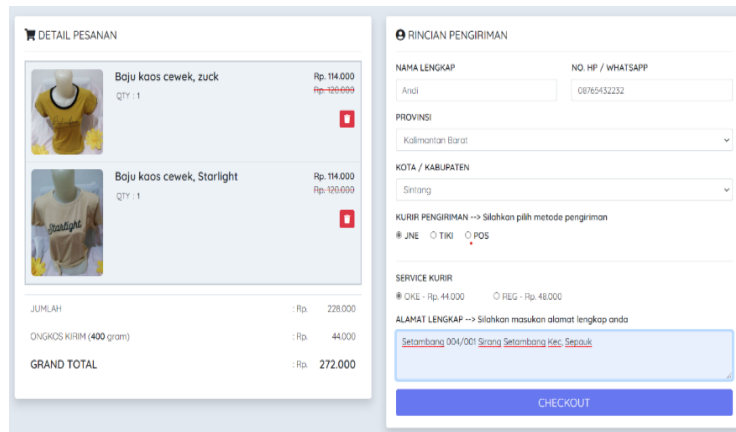
Gambar 7. Halaman Tambah Produk Ke Keranjang

Tambah produk ke keranjang ditunjukkan pada Gambar 7, yang digunakan untuk menampilkan halaman detail produk yang ditampilkan berdasarkan id produk, pengguna yang belum memiliki akun tidak bisa menambahkan produk ke keranjang dan diharuskan *register* terlebih dahulu setelah *user* memiliki akun maka bisa melakukan *login* untuk menambahkan produk ke keranjang dan melanjutkan ke tahap selanjutnya yaitu tahap *checkout*. Produk yang ditampilkan berdasarkan id dari produk yang diklik sebelumnya, *user* bisa menambahkan produk jika jumlah stok produk lebih dari nol atau tidak sama dengan nol.

*User* memilih ukuran dan warna produk yang ingin dibeli, ukuran dan warna produk yang tersedia pada halaman tersebut dapat dipilih sesuai keinginan *user*. Jika ukuran dan warna produk tidak tersedia maka produk tersebut tidak bisa ditambahkan ke dalam keranjang belanja, hal ini digunakan untuk melakukan validasi agar tidak terjadi kesalahan pemesanan produk setelah melakukan *checkout*, karena data yang diterima tidak sesuai dan dapat menurunkan minat pembeli saat pelayanan tidak sesuai dengan yang diharapkan.

*User* melakukan penambahan produk ke keranjang dengan mengklik tombol tambahkan ke keranjang, hal ini dapat dilakukan setelah melalui beberapa validasi. Selanjutnya produk akan

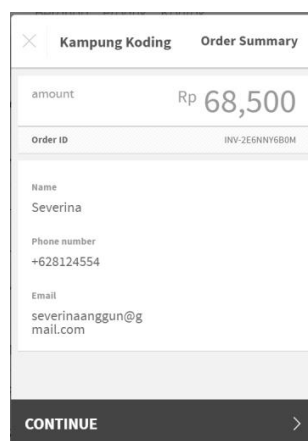
ditambahkan sesuai dengan pilihan ukuran dan warna produk yang ditangani oleh *vue* pada *vue.js* sehingga data produk yang ditambahkan secara *realtime* langsung masuk ke keranjang tanpa perlu *loading* terlebih dahulu, data tersebut selanjutnya ditangani oleh *controller* API untuk menyimpan data ke basis data keranjang, data tersebut akan terhapus setelah *user* melakukan *checkout* yang selanjutnya *user* dapat melakukan pembayaran dengan sejumlah uang yang harus dibayarkan.



Gambar 8. Halaman Order

Halaman *order* ditunjukkan pada Gambar 8, merupakan halaman yang berguna untuk *user* melakukan pemesanan barang yang telah dimasukkan ke dalam keranjang sebelumnya. Selanjutnya *user* melakukan input data pesanan sesuai dengan alamat pemesan, pilihan pengiriman dapat dilakukan setelah memilih provinsi dan kota, jika semua data telah dimasukkan dan pelanggan menekan tombol *checkout* dan akan ditangani oleh *Axios* melalui *RESTful* API, selanjutnya akan diterima oleh *API controller* sehingga data dapat dimasukkan ke dalam *database* dibagian *backend* dan *user* tinggal melakukan pembayaran.

Pelanggan dapat memilih jasa pengiriman yang didapat dari *API RajaOngkir* sehingga pelanggan dapat mengetahui jumlah biaya *service* kurir dan total semua yang harus dibayarkan untuk membeli produk. Pelanggan dapat menghapus data produk pada keranjang jika tidak jadi membeli produk tersebut dengan mengklik *icon* sampah pada *website* toko online *Happy Belanja*.



Gambar 9. Halaman Pembayaran

Halaman pembayaran ditunjukkan pada Gambar 9, digunakan ketika *user* ingin melanjutkan pembayaran setelah melakukan *checkout* pada halaman sebelumnya *user* telah mengisi data *order* yang tampil pada halaman pembayaran dengan no *invoice*, total harga *order* yang harus dibayarkan dan info pembeli akan muncul. Setelah mendapatkan *invoice* pembayaran maka *user* dapat melakukan pembayaran melalui transfer ke no rekening yang telah tersedia dan barang akan diproses oleh pemilik toko untuk dilakukan pengemasan dan pengiriman ke alamat yang tertera di *invoice*, pemilik toko dapat melakukan *update* no resi pengiriman agar *user* bisa melakukan *tracking* barang yang dipesan.

### 3.1 Pengujian

Tahap pengujian dilakukan untuk memastikan semua fungsi *website* dapat berjalan sesuai dengan kebutuhan. Metode *black-box* dengan metode *boundary value analysis* ini dipilih karena pengembangan *website* toko online *Happy Belanja* yang tidak mengetahui struktur internal *website*. *Boundary Value Analysis* merupakan salah satu teknik pengujian *black box* yang melakukan pengujian pada batas bawah dan batas atas dari suatu nilai yang di inputkan ke dalam aplikasi atau *website* dengan ketentuan memperhatikan nilai input dan nilai *output* dari pengujian yang dilakukan dan menguji input pada batas atas maupun batas bawah sebuah nilai yang valid.

Tabel 1. Pengujian *Form* Rincian Pengiriman.

Skenario Pengujian	Data Sampel Uji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
Jumlah karakter lebih banyak	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.	Data tidak dapat diinputkan	Data tidak dapat diinputkan	Berhasil
Jumlah karakter lebih sedikit	+6281	Data tidak dapat diinputkan	Data dapat diinputkan	Gagal
Jumlah karakter sesuai dengan ketentuan	081233455678	Data dapat diinputkan	Data dapat diinputkan	Berhasil
Kosong		Data tidak dapat diinputkan	Pesan kesalahan	Berhasil

Berdasarkan pengujian yang dilakukan pada Tabel 1, *form* rincian pengiriman dilakukan pengujian dengan menggunakan skenario jumlah karakter lebih banyak, jumlah karakter lebih sedikit, jumlah karakter sesuai dengan ketentuan, dan kosong. Pengujian yang dilakukan dengan menginputkan data sampel uji yang telah disediakan sebelumnya dengan tingkat kesuksesan pengujian 75%. Tabel 1 memiliki 3 data sampel berhasil yang bisa ditangani oleh *website* saat dilakukan pengujian.

Tabel 2. Pengujian *Form* Tambah Produk.

Skenario Pengujian	Data Sampel Uji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
Jumlah karakter lebih banyak	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.	Data dapat diinputkan	Data dapat diinputkan	Gagal
Jumlah karakter lebih sedikit	baju	Data tidak dapat diinputkan	Data dapat diinputkan	Gagal
Jumlah karakter sesuai dengan ketentuan	Baju Scarlet/outfit	Data dapat diinputkan	Data dapat diinputkan	Berhasil
Kosong		Data tidak dapat diinputkan	Pesan kesalahan	Berhasil

Berdasarkan pengujian yang dilakukan pada Tabel 2, *form* tambah produk dilakukan pengujian dengan menggunakan skenario jumlah karakter lebih banyak, jumlah karakter lebih sedikit, jumlah karakter sesuai dengan ketentuan, dan kosong. Pengujian yang dilakukan dengan menginputkan data sampel uji yang telah disediakan sebelumnya dengan tingkat kesuksesan pengujian 50%. Tabel 2 memiliki 2 data sampel berhasil yang bisa ditangani oleh *website* saat dilakukan pengujian.

Tabel 3. Pengujian *Form* Tambah Admin.

Skenario Pengujian	Data Sampel Uji	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
Jumlah karakter lebih banyak	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.	Data tidak dapat diinputkan	Data tidak dapat diinputkan	Berhasil
Jumlah karakter lebih	Budi	Data tidak dapat diinputkan	Data dapat diinputkan	Gagal

sedikit		diinputkan		
Jumlah	Budianto	Data dapat	Data dapat	Berhasil
karakter		diinputkan	diinputkan	
sesuai dengan				
ketentuan				
Kosong		Data tidak	Pesan	Berhasil
		dapat	kesalahan	
		diinputkan		

Berdasarkan pengujian yang dilakukan pada Tabel 3, *form* tambah produk dilakukan pengujian dengan menggunakan skenario jumlah karakter lebih banyak, jumlah karakter lebih sedikit, jumlah karakter sesuai dengan ketentuan, dan kosong. Pengujian yang dilakukan dengan menginputkan data sampel uji yang telah disediakan sebelumnya dengan tingkat kesuksesan pengujian 75%. Tabel 3 memiliki 3 data sampel berhasil yang bisa ditangani oleh *website* saat dilakukan pengujian.

#### 4. Kesimpulan

Berdasarkan hasil penelitian dengan menggunakan metode *Design Science Research Methodology* (DSRM) dalam melakukan implementasi arsitektur *Model View Controller* (MVC) menggunakan *framework laravel* dan *Vue.js* yang dihubungkan dengan memanfaatkan RESTful API pada *website* toko online *Happy Belanja*. Perancangan *website* toko online ini dimulai dari analisis kebutuhan pengguna, pengumpulan data dilakukan dengan data primer yang meliputi wawancara, dan observasi, sedangkan data sekunder mempelajari dokumentasi pendukung dari internet. Hasil akhir adalah *website* toko online *Happy Belanja* berbasis *framework laravel* sebagai *backend* yang dihubungkan dengan menggunakan RESTful API sehingga *framework Vue.js* sebagai *frontend*, selain itu yang paling utama dalam hasil pembuatan *website* toko online *Happy Belanja* adalah menerapkan arsitektur MVC. Hasil pengujian dengan menggunakan *black box* dari 3 *form* yang di uji menunjukkan persentase dari data sampel uji dengan persentase keberhasilan 66,7%. Penelitian yang telah dilakukan masih memiliki kekurangan baik berupa pengembangan perangkat lunak maupun analisis yang dilakukan, untuk penelitian selanjutnya masih perlu adanya penambahan fitur komentar, testimoni dan *rating* produk. Perlu adanya grafik penjualan agar penjual mengetahui seberapa besar minat pembeli dalam berbelanja pada *website* toko online, perlu dilakukan pengujian secara menyeluruh pada perangkat lunak yang telah dibuat seperti masih munculnya *error too long data* pada saat input data lebih dari 250 karakter.

#### Ucapan Terima Kasih

Saya mengucapkan terima kasih kepada STMIK Pontianak, para dosen pembimbing dan para mahasiswa yang telah memberikan dukungan dan bimbingan selama penelitian ini dilakukan sehingga saya dapat menyelesaikan penelitian ini, terima kasih juga kepada pemilik toko *Happy Belanja* yang telah bersedia untuk membantu dan memberi izin untuk saya melakukan penelitian ini dan terimakasih juga untuk BITE yang telah memberikan kesempatan kepada saya untuk publikasi.

#### Referensi

- [1] K. C. Laudon and C. Guercio. Traver, *E-commerce: business, technology, society*.
- [2] A. Afsar, Z. Nasiri, and M. O. Zadeh, "E-loyalty model in e-commerce," *Mediterranean Journal of Social Sciences*, vol. 4, no. 9, 2013, doi: 10.5901/mjss.2013.v4n9p547.
- [3] E. Turban, D. King, J. K. Lee, T.-P. Liang, and D. C. Turban, "Springer Texts in Business and Economics Electronic Commerce a Managerial and Social Networks Perspective Eighth Edition," 2015. [Online]. Available: <http://www.springer.com/series/10099>
- [4] P. S. Hasugian, "Perancangan Website Sebagai Media Promosi Dan Informasi," *Journal of Informatic Pelita Nusantara*, vol. 3, no. 1, 2018.

- [5] A. Subagia, *Membangun Aplikasi Web dengan Metode OOP*. 2018.
- [6] Arochman, "Penerapan Model Design-Pattern Mvc Untuk Efektifitas Dan Efisiensi Pengembangan Web Kantor Ristekin Kota Pekalongan," 2016. [Online]. Available: <http://jurnal.stmik-wp.ac.id>
- [7] B. Rizki, P. Surya, A. Putra Kharisma, and N. Yudistira, "Perbandingan Kinerja Pola Perancangan MVC, MVP, dan MVVM Pada Aplikasi Berbasis Android (Studi kasus : Aplikasi Laporan Hasil Belajar Siswa SMA BSS)," 2020. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [8] T. Wijaya, "Penerapan Service-Oriented Architecture Pada Sistem Informasi Eksepdisi," *Eksplora Informatika*, vol. 6, no. 2, 2017.
- [9] Y. Yudhanto and Prase, "Panduan Mudah Belajar Framework Laravel," *PT Alex Media Komputindo*, 2018.
- [10] Wardana, S.Hut, and M.Si, *Menjadi Master PHP dengan Framework Codeigniter*. 2010.
- [11] Aminudin, "Cara Efektif Belajar Framework Laravel," *Ilmu Teknologi Informasi*, vol. 1, no. 1, 2015.
- [12] Sigit, "Mengenal Vue.js," 2017. <https://medium.com/skyshidigital/mengenal-vue-js-ab8557042cd3> (accessed Jul. 01, 2021).
- [13] D. Prabowo, S. Kom, B. Wulan, S. S. Kom, P. Ventri, and H. S. Kom, "Implementasi Pemrograman PHP Berbasis Model View Controller (MVC) Pada Website E-Commerce ( Studi Kasus: [www.indominiatur.com](http://www.indominiatur.com) )," 2014, [Online]. Available: [www.indominiatur.com](http://www.indominiatur.com)
- [14] A. Armi and Y. Hendriyani, "Perancangan Aplikasi Pelayanan Cleaning Service Berbasis Web," *Voteteknika (Vocational Teknik Elektronika dan Informatika)*, vol. 7, no. 3, 2019, doi: 10.24036/voteteknika.v7i3.105687.
- [15] K. Kardinal and E. Tasrif, "Perancangan Sistem E-Commerce Pada Toko Salsa Sport Berbasis Web," *Voteteknika (Vocational Teknik Elektronika dan Informatika)*, vol. 6, no. 2, 2018, doi: 10.24036/voteteknika.v6i2.101985.
- [16] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, 2007, doi: 10.2753/MIS0742-1222240302.

