

Rancang Bangun Sistem Informasi Pembukuan Keuangan Menggunakan Arsitektur Hierarchical Model view controller Dengan Pendekatan PXP (Studi Kasus: CV. Anugerah Mandiri)

Ahmad Hidayat^{*1}, Aminudin², Ilyas Nuryasin³

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

hidayat121196@gmail.com^{*1}, aminudin2008@umm.ac.id², ilyas@umm.ac.id³

Abstrak

CV. Anugerah Mandiri merupakan salah satu perusahaan yang bergerak dalam bidang fotocopy, penjualan ATK, penjualan mesin fotocopy dan penjualan sparepart untuk mesin fotocopy. Masalah yang dialami perusahaan tersebut terjadi disebabkan pengelolaan pembukuan yang masih manual. Berdasarkan permasalahan yang dialami perusahaan, diperlukan sebuah sistem yang dapat mengelola serta menganalisa masalah pembukuan keuangan yang ada pada perusahaan yang diteliti, namun dengan metode ini peneliti harus siap akan kebutuhan client yang dapat berubah-ubah karena pada proses pengembangan client akan terlibat secara langsung sejak analisa kebutuhan dilakukan. Dengan arsitektur HMVC serta pendekatan akan dapat dengan mudah menyelesaikan sistem yang akan dibangun menimbang bahwa sistem yang digunakan ialah HMVC yang bersifat modularitas sehingga akan dengan mudah memenuhi kebutuhan tambahan dari client dengan cara menambahkan modul baru pada sistem, hal tersebut mudah dilakukan karena ketika ada penambahan fitur maka modul yang ditambahkan tidak mengganggu modul lain sebab modul satu dan lainnya memiliki model view controllernya masing-masing. Development, Design Pattern. Setelah penelitian berhasil dilakukan maka dapat ditarik kesimpulan bahwa sistem informasi pembukuan keuangan CV. Anugerah Mandiri dibangun menggunakan framework codeigniter dengan menggunakan pendekatan Personal Extreme Programming yang terdiri dari dua iterasi dan lima tahapan yaitu Requirement, Planning, Iteration initialization, System testing, dan Retrospective. Hierarchical Model view controller pada sistem ini mampu menangani perubahan yang dilakukan oleh user dibuktikan dengan tiga belas modul MVC yang terdiri dari admin, auth, barang, client, debit, kalender, keuangan, laporan, orderan, profile, sparepart, template dan user.

Kata Kunci: PXP, HMVC, Agile Software Development, Design Pattern

Abstract

CV. Anugerah Mandiri is one of the companies engaged in photocopying, ATK sales, copy machine sales and spare parts sales for photocopiers. The problem experienced by the company occurred due to the management of bookkeeping which is still manual. Based on the problems experienced by the company, we need a system that can manage and analyze financial accounting problems that exist in the company under study, but with this method researchers must be prepared for client needs that can change because the client development process will be directly involved since the analysis needs done. With HMVC architecture and approach will be able to easily complete the system to be built considering that the system used is HMVC which is modularity so that it will easily meet the additional needs of clients by adding new modules to the system, it is easy to do because when there are additional features then the added module does not interfere with other modules because one and the other modules have their own view controller models. Development, Design Pattern. After successful research, it can be concluded that the financial accounting information system CV. Anugerah Mandiri was built using a codeigniter framework using the Personal Extreme Programming approach which consists of two iterations and five stages, namely Requirements, Planning, Iteration initialization, System testing, and Retrospective. Hierarchical Model view controller on this system is able to handle changes made by the user as evidenced by thirteen MVC modules consisting of admin, auth, goods, client, debit, calendar, finance, reports, orders, profiles, spare parts, templates and users.

Keywords: PXP, HMVC, Agile Software Development, Design Pattern

1. Pendahuluan

Framework merupakan *software* yang memfasilitasi para *web developer* dalam membuat sebuah *website*, dengan bantuan *framework* proses pengkodean akan lebih singkat dikarenakan menggunakan fungsi bawaan yang telah disediakan oleh *software* tersebut [1]. *Framework* PHP yang beredar saat ini sudah cukup banyak jenisnya seperti *codeigniter*, *aura*, *cake PHP* dan *laravel*. Sama halnya dengan jenis-jenis *framework* PHP pada umumnya semua menggunakan konsep arsitektur *Model view controller* (MVC). MVC juga merupakan jenis *design pattern*.

Design pattern yang paling umum digunakan pada PHP ada empat jenis yaitu *singleton*, *strategy*, *factory*, dan MVC namun diantara *design pattern* tersebut MVC merupakan pilihan yang paling banyak digunakan, dikarenakan pola MVC memiliki pola yang cukup unik. Pola MVC dibagi menjadi tiga peran utama, sehingga lebih efisien ketika terjadinya kolaborasi [2]. Peran tersebut yaitu perkembangan, desain, dan integrasi, manfaat adanya tiga peran penting ini yaitu dapat mengisolasi sebuah unit fungsional semua modul sebanyak mungkin, sehingga sangat membantu dalam proses perancangan *website* dalam memahami setiap *unit* yang akan dibangun.

Pengembangan yang merujuk pada arsitektur MVC sudah sangat banyak akan tetapi pengembangan dalam *Hierarchical Model view controller* (HMVC) masih sangat jarang ditemukan dikarenakan arsitektur ini cukup baru dalam proses rancang bangun sebuah *website*. HMVC merupakan pengembangan dari MVC, HMVC adalah susunan MVC yang telah tersusun rapi sehingga menjadi sebuah kesatuan sistem yang mana strukturnya dibuat berlapis *hierarchy of parent-child MVC layer*. HMVC dibuat agar dapat meringankan beban beberapa *programmer* dalam proses pengkodean dan kompilasi kode dikarenakan sistem tersebut dibagi menjadi beberapa modul-modul yang sangat spesifik sehingga pada proses pengerjaan lebih fleksibel [3]. MVC mungkin menjadi salah satu *design pattern* yang banyak digunakan karena memiliki banyak fitur-fitur kelebihan dibanding *design pattern* yang lain akan tetapi HMVC juga tidak kalah menarik dengan arsitektur tersebut.

Arsitektur HMVC memiliki kelebihan dibanding MVC *standart* karena, dalam proses pengkodean ketika membuat sebuah *website* jelas HMVC lebih tertata rapi, modul-modul yang rapi tersebut juga sangat mudah dikompilasi ketika dipecah menjadi beberapa bagian. HMVC juga dapat di *reusable* dan *portable* tanpa melakukan banyak perubahan [4].

Pada penelitian sebelumnya mengimplementasikan salah satu metode pengembangan sistem informasi, metode yang digunakan adalah *Extreme Programming*, metode ini digunakan karena siap menangani perubahan-perubahan yang akan terjadi ketika proses pengembangan sistem sedang berlangsung [5]. Penelitian lainnya juga menggunakan arsitektur HMVC dalam membangun sistem informasi rumah sakit namun metode yang digunakan adalah *prototype* [6]. *Extreme Programming* merupakan salah satu dari metode pengembangan sistem informasi *agile software development* dimana metode ini merupakan metode pengembangan sistem cepat. Metode ini sangat mengedepankan komunikasi antara tim pengembang dengan *client* sehingga sangat fleksibel dengan *client* yang kebutuhannya dapat dengan mudah berubah-ubah [7]. Menggunakan metode *Extreme Programming* membutuhkan tim pengembang yang berjumlah lebih dari dua orang dikarenakan banyaknya peran yang akan dibagi menjadi beberapa bagian.

Pada penelitian kali ini yang menjadi titik pembeda adalah peneliti akan bekerja sendiri dalam proses pengembangan sistem yang nantinya peneliti akan menggunakan metode turunan dari *Extreme Programming*. *Personal Extreme Programming* (PXP) dapat diterapkan sebagai solusi penengah sebagai pengembangan sistem informasi dengan pemrograman tunggal secara cepat, fleksibel, dan efisien [8]. Berdasarkan permasalahan yang dialami peneliti saat ini, diperlukan sebuah sistem yang dapat mengelola serta menganalisa masalah pembukuan keuangan yang ada pada perusahaan yang diteliti, namun dengan metode ini peneliti harus siap akan kebutuhan *client* yang dapat berubah-ubah karena pada proses pengembangan *client* akan terlibat secara langsung sejak analisa kebutuhan dilakukan.

Dengan arsitektur HMVC serta pendekatan PXP peneliti berfikir akan dapat dengan mudah menyelesaikan sistem yang akan dibangun menimbang bahwa sistem yang digunakan ialah HMVC yang bersifat modularitas sehingga akan dengan mudah memenuhi kebutuhan tambahan dari *client* dengan cara menambahkan modul baru pada sistem, hal tersebut mudah dilakukan karena ketika ada penambahan fitur maka modul yang ditambahkan tidak mengganggu modul lain sebab modul satu dan lainnya memiliki *model view controller*nya masing-masing.

Oleh karena itu peneliti ingin mengimplementasikan arsitektur *Hierarchical Model view controller* dengan pendekatan *Personal Extreme Programming* pada CV. Anugerah Mandiri. Hal

ini dikarenakan pada penelitian sebelumnya arsitektur *Hierarchical Model view controller* dengan pendekatan *Personal Extreme Programming* belum pernah dilakukan dalam sebuah pengembangan sistem informasi pembukuan perusahaan sejenis.

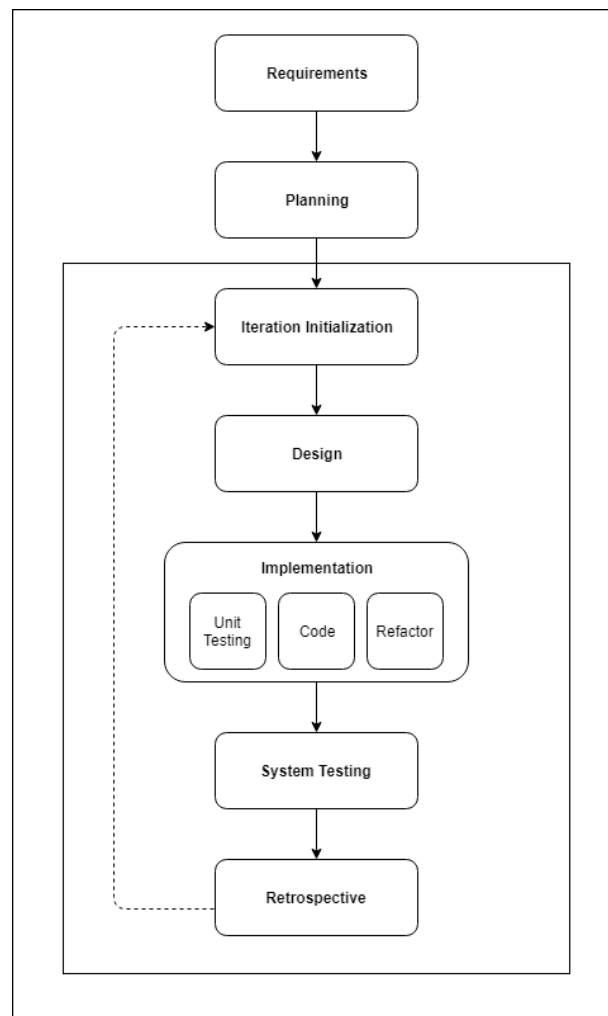
2. Metode Penelitian

2.1 Studi Literatur

Pada tahap studi literatur proses pengumpulan segala informasi untuk menunjang penelitian akan dilakukan. Segala informasi tersebut didapatkan dari beberapa sumber terpercaya seperti jurnal, buku, dan literatur penunjang lainnya.

2.2 Proses Pengembangan Sistem

Tahapan penelitian pada bab ini menjelaskan mengenai kegiatan yang akan dilakukan selama penelitian berlangsung, adapun skema tahapan penelitian yang akan dilakukan oleh peneliti dapat dilihat pada Gambar 1 berikut.



Gambar 1. Tahap Metode PXP [9]

1. Requirements

Pada tahap ini pengembang mengumpulkan kebutuhan dari manager CV. Anugerah Mandiri. Proses pengumpulan data dilakukan dengan cara wawancara langsung bersama pemilik perusahaan. Pada wawancara tersebut *client* menjabarkan mengenai segala aspek yang menjadi permasalahan perusahaan dari yang kecil hingga terbesar serta gambaran umum yang diharapkan pada sistem yang akan dibangun. Berdasarkan kebutuhan yang diinginkan *user* maka akan dibuat sebuah *user story* pada tiap masalah yang dialami *client*. Adapun format penulisan *user story* "Sebagai <pengguna>, saya ingin sistem <tindakan > sehingga <manfaat t>".

2. Planning

Pada fase *planning* serangkaian *task* yang akan di kerjakan berdasarkan *user story* yang telah didapatkan. Susunan *task* tersebut dibuat menggunakan *partice planning game*. Fase ini juga akan melakukan pemberian *value* pada tiap *story* pemberian *value* ini akan sangat berguna sebagai penentu prioritas pada setiap *user stories* yang ada dan pada fase ini juga akan dilakukan pemberian *CRC Card* pada tiap *class* yang akan digunakan [10].

3. Iteration Initialization

Pada iterasi pengembang telah mengerjakan keseluruhan *stories* yang telah ada pada bab sebelumnya. *Story* dikerjakan dimulai dari *risk* dan *value* yang paling tinggi hingga yang paling rendah

4. Design

Pada iterasi pertama desain dari sistem dibuat sesimpel mungkin, namun jika *client* ingin mengubah desain pada konten yang telah dibuat maka pengembang harus segera mengubah desain sesuai dengan keinginan *client*. Berdasarkan *CRC Card* yang ada pengembang membuat gambaran umum mengenai sistem yang akan dibangun sehingga pada tahap implementasi desain sementara dari program tidak akan berbeda jauh dibanding hasil dari program yang telah dibangun.

5. Implementation

Setelah desain dari mokup sederhana selesai maka pengkodean dari tiap *user story* akan dilakukan. Sistem dibuat menggunakan arsitektur HMVC codeigniter dimana setiap modul *class* yang dibangun memiliki *model view controllemya* masing-masing

6. System testing

Tahap *system testing* dilakukan dalam bentuk kuisisioner *User Acceptance Test* hal ini bertujuan agar menentukan apakah hasil pengembangan akan diterima atau ada sedikit perubahan.

7. Retrospective

Tahap ini menganalisis jalannya setiap fase pengembangan serta penyesuaian estimasi waktu yang telah dikerjakan pada tiap iterasi.

3. Hasil Penelitian dan Pembahasan

3.1 Requirements

Pada tahap ini pengembang mengumpulkan kebutuhan dari manager CV. Anugerah Mandiri. Proses pengumpulan data dilakukan dengan cara wawancara langsung bersama pemilik perusahaan. Pada wawancara tersebut *client* menjabarkan mengenai segala aspek yang menjadi permasalahan perusahaan dari yang kecil hingga terbesar serta gambaran umum yang diharapkan pada sistem yang akan dibangun. Hasil wawancara ini ialah didapatkannya tabel *story* yang diinginkan oleh *client*, isi *story client* dapat dilihat pada Tabel 1.

“ Saya menginginkan sebuah sistem pembukuan perusahaan berbasis website, yang dapat mengorganisir data piutang dan dapat menjumlahkan secara otomatis biar saya tidak perlu lagi menghitung pakai kalkulator, kalau bisa data-data tadi bisa di *print* sebagai bukti ambila barang mitra kami. Oh iya jangan lupa untuk tampilan home nya nanti buat iklan alat-alat, mesin, atau tinta fotocopy biar bisa sekaligus promosi. Tolong buat juga untuk perhitungan penghasilan kami, soalnya itu buat perhitungan untung, rugi perusahaan kalau bisa tabel sama grafik. Sistemnya juga bisa order dan liat orderan yang sudah dia ambil itu biar dia ndak nanya-nanya lagi sama kami berapa jumlah hutangnya”

Tabel 1. Daftar User Stories

Berdasarkan kebutuhan yang diinginkan *user* maka akan dibuat sebuah *user story* pada tiap masalah yang dialami *client*. Adapun format penulisan *user story* “Sebagai <pengguna>, saya ingin sistem <tindakan > sehingga <manfaat t>”. Setelah proses pengumpulan kebutuhan telah dilakukan maka dapat ditarik kesimpulan bahwa pengembang akan melakukan delapan penyelesaian masalah berdasarkan *user stroy* yang telah dibuat. *User story* yang didapatkan dapat dengan jelas dilihat pada Tabel 2.

Tabel 2. Daftar User Stories

Kode User Stories	User Stories
Story -01	Sebagai user, saya ingin sistem ini dapat mengorganisir data piutang, serta dapat menjumlahkan dan melakukan pengurangan total piutang secara otomatis ketika diinputkan sehingga dapat memudahkan saya dalam pengaturan pembukuan perusahaan
Story -02	Sebagai user, saya ingin sistem memiliki fitur <i>download</i> dan <i>print</i> , sehingga ketika saya dimintai list harga dan jumlah barang yang telah di ambil oleh <i>costumer</i> maka saya bisa menunjukannya.
Story -03	Sebagai user, saya ingin tampilan awal sistem sebelum login dapat menampilkan <i>sparepart</i> fotocopy dan harga mesin fotocopy, sehingga memudahkan saya dalam hal promosi.
Story -04	Sebagai user, saya ingin sistem memiliki pemberitahuan bulanan, harian, dan tahunan, sehingga memudahkan saya dalam menghitung keuntungan dan kerugian perusahaan
Story -05	Sebagai user, saya ingin sistem memiliki pemberitahuan dalam bentuk grafik, sehingga memudahkan saya dalam melihat untung rugi perusahaan.
Story -06	Sebagai user saya ingin agar <i>client</i> dapat melihat daftar ambilan barang yang telah dia ambil, serta dapat mengorder barang yang akan dia ambil.

3.2 Planning

Pada fase *planning* serangkaian *task* yang akan di kerjakan berdasarkan *user story* yang telah didapatkan. Susunan *task* tersebut dibuat menggunakan *partice planning game*. Fase ini juga akan melakukan pemberian *value* pada tiap *story* pemberian *value* ini akan sangat berguna sebagai penentu prioritas pada setiap *user stories* yang ada dan pada fase ini juga akan dilakukan pemberian *CRC Card* pada tiap *class* yang akan digunakan.

Sebelum pemberian *value stories* pada tahap ini menentukan estimasi waktu yang diperlukan pada tiap *user stories*. Estimasi waktu dibagi menjadi beberapa poin tergantung lama waktu pengerjaan semakin besar poinnya maka estimasi waktu yang diperlukan akan semakin besar. Setelah *story points* untuk masing-masing *user stories* yang telah ditentukan, pengembang dan klien akan berdiskusi untuk menentukan prioritas dari masing-masing *user stories* tersebut. Penentuan prioritas tersebut dilakukan dengan cara menyortir *user stories* tersebut berdasarkan *value* dan *risk*.

Value dari sebuah *user stories* di tentukan berdasarkan nilai bisnis dari *user stories* tersebut. Nilai *value* tersebut terbagi menjadi 3, yaitu *critical*, *significant business value*, dan *nice to have*. Hasil penentuan *value* akan dilanjutkan dengan penentuan *risk* berdasarkan *risk Index* yang berada pada faktor *completeness*, *volatility*, dan *complexcity*.

Adapun *completeness* dapat ditentukan dari pengetahuan pengembang berdasarkan detail dari *user stories* tersebut. *Volatility* ditentukan dari kemungkinan adanya perubahan kebutuhan dari *user stories*, dan *complexcity* dapat ditentukan dari sulitnya pengembangan sistem pada *user stories*. Semua penilaian dimulai dari 0 hingga 2 tergantung penilaian dari faktor sebelumnya Setelah berhasil ditentukan maka *Index* tersebut akan dijumlahkan sehingga total dari *Index* akan menentukan *risk* pada masing-masing *story*. *Index* tersebut terdiri dari *low* (0-1), *medium* (2-4) dan *high* (5-6), ditunjukkan pada Tabel 3.

Tabel 3. Hasil Perencanaan Iterasi

Iterasi	Kode User Stories	Value	Risk	Story Points
Iterasi 1	Story-01	Critical	(5) high	9
	Story-02	Critical	(1) low	3
	Story-03	Significant busniness vslue	(3) Medium	7
	Story-04	Critical	(3) Medium	8
	Story-05	Significant busniness vslue	(3) Medium	5
	Story-06	Critical	(5) high	10

Pada iterasi pertama pengembang telah mengerjakan keseluruhan *stories* yang telah ada pada bab sebelumnya. *Story* dikerjakan dimulai dari *risk* dan *value* yang paling tinggi hingga yang paling rendah. Pada iterasi sebelumnya *client* meminta agar adanya tambahan fitur pada sistem. Fitur yang diminta merupakan fitur yang simple sehingga tidak membutuhkan waktu yang lama bagi pengembang untuk menyelesaikan fitur tersebut, adapun fitur tambahan yang diinginkan ialah kalender, produk, dan *print* penghasilan seperti pada Tabel 4.

Tabel 4. Penambahan User Stories Iterasi 2

Kode User Stories	User Stories
Story -07	Tambah fitur kalender pada admin dan <i>client</i>
Story -08	Tambah fitur barang untuk cek harga dan lihat gambar barang
Story -04 & 05	Tambah fitur <i>print</i> tabel penghasilan

3.3 Design

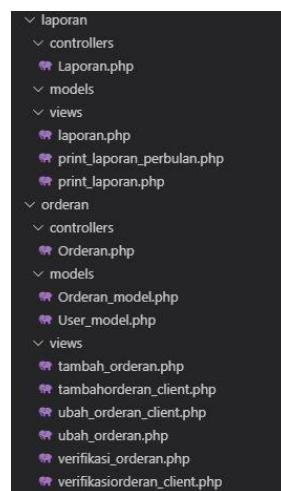
Class Responsibility Colaboration Card (CRC Card) bertugas sebagai penunjuk interaksi antar objek, serta aktifitas apa yang akan dilakukan objek dalam *card* yang telah dibuat, pembuatan CRC Card disini berdasarkan *story* yang telah dibuat. Berikut kumpulan daftar nama CRC Card yang telah dibuat dapat dilihat pada Tabel 5.

Tabel 5. Daftar CRC Cards

Iterasi	Kode User Stories	CRC Card
Iterasi 1	Story-01	Buku
	Story-02	Buku
	Story-03	Slider
	Story-04	Debit
	Story-05	Debit
	Story-06	Orderan
Iterasi 2	Story-07	Kalender
	Story-08	Barang
	Story-04&05	Debit

3.4 Pengembangan Sistem dan Pengujian

Setelah desain dari mokup sederhana selesai maka pengkodean dari tiap *user story* akan dilakukan. Sistem dibuat menggunakan arsitektur HMVC codeigniter dimana setiap modul *class* yang dibangun memiliki *model view controllemnya* masing-masing. Adapun bukti penerapan pada sistem yang di bangun dapat dilihat pada Gambar 1.



Gambar 2. Implementasi HMVC

Tahap *system testing* dilakukan dalam bentuk kuisisioner *User Acceptance Test* hal ini bertujuan agar menentukan apakah hasil pengembangan akan diterima atau ada sedikit perubahan. Pada iterasi pertama semua sistem telah diterima akan tetapi *user* meminta beberapa tambahan fitur pada sistem yang diinginkan. UAT pada sistem dapat dilihat pada Tabel 6.

Tabel 6. Hasil Pengujian Sistem

Iterasi	Kode User Stories	Hasil Uji
Iterasi 1	Story-01	Diterima
	Story-02	Diterima
	Story-03	Diterima
	Story-04	Diterima
	Story-05	Diterima
	Story-06	Diterima
Iterasi 2	Story-07	Diterima
	Story-08	Diterima
	Story-04&05	Diterima

3.4 Hasil Evaluasi

Pada tabel diatas dapat dilihat waktu pengerjaan semua *user stories* pada iterasi kedua telah melampaui estimasi yang telah ditentukan namun estimasi waktu yang diperkirakan tidak berbanding jauh dengan waktu estimasi pengerjaan. Setelah pengerjaan selesai *client* memutuskan untuk menyudahi pembangunan sistem dikarenakan semua fitur yang dibangun telah lengkap. Jika *Retrospective* ke dua dikatakan selesai maka tidak ada iterasi ke tiga dan seterusnya, oleh karena itu pembangunan sistem informasi pebukuan keuangan CV. Anugerah mandiri dikatakan selesai.

4. Kesimpulan

Setelah penelitan berhasil dilakukan maka dapat ditarik kesimpulan bahwa sistem informasi pembukuan keuangan CV. Anugerah Mandiri dibangun menggunakan *framework* codeigniter dengan menggunakan pendekatan *Personal Extreme Programming* yang terdiri dari dua iterasi dan lima tahapan yaitu *Requirment, Planning, Iteration initialization, System testing,* dan *Retrospective. Hierarchical Model view controller* pada sistem ini mampu menangani perubahan yang dilakukan oleh *user* dibuktikan dengan tiga belas modul MVC yang terdiri dari admin, auth, barang, *client*, debit, kalender, keuangan, laporan, orderan, profile, *sparepart*, template dan *user*.

5. Saran

Saran untuk penelitian kedepannya adalah sebagai berikut :

1. Pengujian pada sistem informasi selanjutnya disarankan menggunakan metode pengujian *testing* lainnya.
2. Pengembangan pada sistem informasi selanjutnya disarankan menggunakan metode agile lainnya seperti Scrum, *Rational Unified Process*. dan desain *sprint*.

Referensi

- [1] N. Prokofyeva and V. Boltunova, "Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems," *Procedia Comput. Sci.*, vol. 104, no. December 2016, pp. 51–56, 2016.
- [2] D. P. Pop and A. Altar, "Designing an MVC model for rapid web application development," *Procedia Eng.*, vol. 69, pp. 1172–1179, 2014.
- [3] Y. A. Susetyo, P. O. N. Saian, and R. Somya, "Pembangunan Sistem Informasi Zona Potensi Sumber Daya Kelautan Kabupaten Gunungkidul Berbasis HMVC Menggunakan Google Maps API dan JSON," *Indones. J. Model. Comput.* 2, 2018.
- [4] W. Eka, "Penerapan Pola Hierarchical Model-View-Controller Pada Rekayasa Sistem Berbasis Web Framework Eka," *ISSN 1979-8415*, vol. 3, no. 2, pp. 169–178, 2011.
- [5] R. Rahmi, R. P. Sari, and R. Suhatman, "Pendekatan Metodologi Extreme Programming pada Aplikasi E-Commerce (Studi kasus Sistem Informasi Penjualan Alat-alat Telekomunikasi)," *Pendekatan Metodol. Extrem. Program. pada Apl. E-Commerce (Studi kasus Sist. Inf. Penjualan Alat-alat Telekomun. Raisy*, vol. 2, no. 2, pp. 83–92, 2016.

-
- [6] P. Studi, T. Informatika, F. T. Informasi, U. Kristen, and S. Wacana, "Perancangan Dan Implementasi Sistem Informasi Rumah Sakit Menggunakan Arsitektur Hierarchical Model, View, Controller Berbasis Web (Studi Kasus: Rumah Sakit Puri Asih Salatiga)," *Artik. Ilm.*, no. April, 2017.
- [7] P. Patanakul and R. Rufo-mccarron, "Journal of High Technology Management Research Transitioning to agile software development: Lessons learned from a government-contracted program," *J. High Technol. Manag. Res.*, no. xxxx, pp. 0–1, 2018.
- [8] H. Rizal, S. Adhy, and P. W. Wirawan, "Perancangan Dan Pembuatan Mobile Learning Interaktif Berbasis Android Dengan Metode Personal Extreme Programming," vol. 2, no. 3, pp. 1–10, 2013.
- [9] Y. Dzhurov, I. Krasteva, and S. Ilieva, "Personal Extreme Programming—An Agile Process for Autonomous Developers," 2009.
- [10] M. Subekti, "Requirements Management pada Extreme Programming," no. August, 2015.