

PENGEMBANGAN APLIKASI PENJUALAN DENGAN METODE *EXTREME PROGRAMMING* DAN PENERAPAN MODEL *MULTI- TENANCY*

Ahmadi¹, Arie Surachman²

Program Studi Teknik Informatika, Universitas Indraprasta PGRI Jakarta^{1,2}
anggiahmadi@gmail.com¹, ariesurachmanmkom@gmail.com²

Submitted September 4, 2023; Revised March 26, 2024; Accepted March 26, 2024

Abstrak

Penelitian ini membahas pengembangan aplikasi penjualan dengan metode *extreme programming* dan penerapan model *multi-tenancy* untuk mendukung bisnis *software as a service* (SaaS) pada perusahaan perangkat lunak berskala kecil. Hasil observasi mengungkapkan aplikasi berbasis web menggunakan bahasa pemrograman PHP dengan framework Laravel dan *database* MariaDB. Aplikasi memiliki fasilitas untuk mengelola penjualan, kontrak kerjasama, dan berbagai fitur lainnya dalam bisnis *serviced office*, *virtual office*, dan MICE. Pengembangan perangkat lunak dilakukan oleh dua pengembang dalam dua iterasi yang terfokus dan terbatas. Hasil pengembangan mencakup fitur-fitur kunci seperti autentikasi pengguna, pengelolaan hak akses, pengelolaan data utama penjualan, pengelolaan pelanggan, transaksi penjualan, dan dashboard. Aplikasi yang dihasilkan menerapkan model *multi-tenancy*, penting dalam mendukung bisnis SaaS. Kesimpulan dari penelitian ini adalah bahwa model *multi-tenancy* dan metode *extreme programming* cocok untuk pengembangan aplikasi SaaS, bahkan dengan tim pengembangan kecil. Keunggulan teknis dalam penulisan kode perangkat lunak merupakan kunci utama dalam metode ini.

Kata Kunci : Pengembangan aplikasi, *Extreme Programming*, *Multi-Tenancy*, *Software as a Service* (SaaS), Perangkat lunak berskala kecil

Abstract

This research discusses the development of sales applications using extreme programming methods and implementing a multi-tenancy model to support software as a service (SaaS) businesses in small-scale software companies. Observation results reveal that the web-based application uses the PHP programming language with the Laravel framework and MariaDB database. The application has facilities for managing sales, cooperation contracts, and other features in serviced offices, virtual offices, and MICE businesses. Software development using only two developers in two focused and limited iterations. The development results include key features such as user authentication, access rights management, sales data management, customer management, sales transactions, and dashboards. The resulting application implements a multi-tenancy model essential in supporting SaaS businesses. This research concludes that the multi-tenancy model and extreme programming methods suit SaaS application development, even with small development teams. Technical excellence in writing software code is the primary key to this method.

Keywords : *Application development, Extreme Programming, Multi-Tenancy, Software as a Service (SaaS), Small-scale software.*

1. PENDAHULUAN

Dalam era teknologi saat ini persaingan pada perusahaan jasa pembuatan perangkat lunak sangat kompetitif tidak hanya dari harga yang ditawarkan, melainkan fasilitas

produk yang diberikan. Kondisi tersebut ini mendorong penyedia layanan untuk terus efektif dan efisien dalam melayani pengguna. Pada perusahaan skala kecil sangatlah membutuhkan beberapa strategi

untuk dapat bersaing dengan perusahaan yang lebih besar, salah satu strateginya yaitu dengan menerapkan metodologi *agile development* dalam pengembangan perangkat lunak. Metodologi *agile development* memberikan fleksibilitas dan kemampuan adaptif sehingga dapat meningkatkan efisiensi dan efektivitas dalam proses pengembangan perangkat lunak [1]. Terdapat beberapa model *agile development* yang ada, seperti *scrum*, *kanban*, *extreme programming*, dan lain-lain. Namun dalam penelitian ini merujuk kepada model *agile development* yang paling sesuai dengan tim perusahaan skala kecil yaitu *extreme programming*, metode ini dapat digunakan untuk tim pengembang yang memiliki kapasitas dua orang [2].

Software as services (SaaS) adalah layanan aplikasi yang menggunakan akses internet oleh para pengguna, serta dapat dikonfigurasi dan dikustomisasi sesuai dengan kebutuhan dan batasan yang dimiliki aplikasi [3]. Untuk mencapai nilai ekonomis, SaaS membutuhkan model *multi-tenancy*. *Multi-tenancy* adalah model aplikasi yang dapat melayani beberapa pengguna (*Multi-user*) dengan menggunakan *single source code*. Model *multi-tenancy* ini digunakan untuk dapat menurunkan biaya pemeliharaan dan infrastruktur lainnya, sehingga sangat sesuai dengan kondisi perusahaan perangkat lunak (*software house*) skala kecil bahkan bagi perusahaan yang baru berkembang. Efektivitas SaaS ini sangatlah bergantung pada konfigurasi dan kustomisasi di dalam penerapan pembuatan aplikasinya. Hal ini disebabkan adanya keberagaman penyewa jasa software atau aplikasi dalam menerapkan sesuai kebutuhannya masing-masing, sehingga hal tersebut dapat memberikan masalah pada saat *programming* melakukan kustomisasi dan konfigurasi. Maka bagi perusahaan penyedia layanan aplikasi menerapkan *Software as services* (SaaS) ketika sedang melakukan rancangan aplikasi dengan

terstruktur, diharapkan bahwa pada saat melakukan kustomisasi dan konfigurasi bagi setiap pengguna aplikasi ini tidak memberikan dampak perubahan yang signifikan atau meminimalisir adanya perubahan pada *source codenya*.

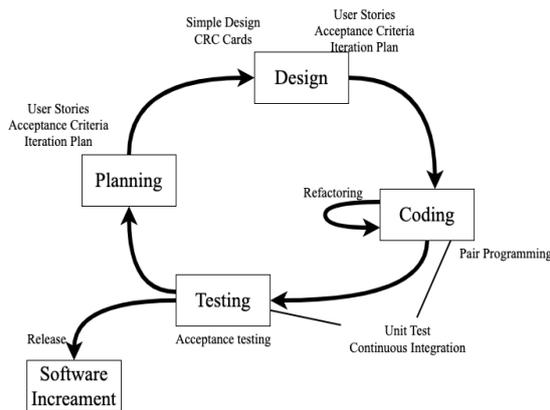
Pada studi kasus di dalam penulisan artikel ilmiah ini yaitu dimana adanya sebuah perusahaan perangkat lunak skala kecil dengan kapasitas empat orang, mereka memiliki aplikasi pengelolaan penjualan untuk para pelanggannya, namun aplikasi yang dimiliki mereka tersebut belum menerapkan model *multi-tenancy* [4], sehingga di dalam proses kustomisasi, konfigurasi, dan instalasi pada setiap pelanggan baru masih membutuhkan adanya perubahan yang signifikan berpengaruh terhadap *source code* aplikasinya. Upaya memaksimalkan proses bisnis perusahaan tersebut, maka perusahaan memutuskan dan menetapkan untuk membangun setiap aplikasi yang dibuatnya dalam versi terbaru, menggunakan pendekatan *multi-tenancy* sehingga dapat menerapkan bisnis *Software as a services*. Dalam penerapan proses pengembangan aplikasi versi terbaru maka perusahaan harus memastikan menggunakan model desain *multi-tenancy* yang terstruktur dengan baik.

Penelitian ini melakukan pengembangan aplikasi pengelolaan penjualan milik tim skala kecil yang belum mendukung *multi-tenancy* menjadi aplikasi yang mendukung *multi-tenancy*, sehingga dapat menjalankan model bisnis *software as services* (SaaS). Pengembangan aplikasi dengan menerapkan metode *extreme programming* (XP) yang sesuai dengan skala tim pengembang kecil yang terdiri dari dua orang pengembang saja. Keterbaruan yang ingin dicapai pada penelitian ini yaitu, fokus penerapan praktik-praktik pada *extreme programming* yang menghasilkan perangkat lunak yang berkualitas dan

penerapan model *multi tenancy* dalam proses pengembangan.

2. METODE PENELITIAN

Metode penelitian ini menggunakan metode pengembangan *extreme programming* pada gambar 1. *Extreme programming* terdiri dari empat proses utama yang dimulai dengan *planning*, *design*, *coding*, dan *testing*.



Sumber: [1], [5]

Gambar 1. Metode Extreme Programming

Berikut penjelasan secara ringkas terkait langkah – langkah dari metode *extreme programming*.

1. *Planning* : tahap ini merupakan perencanaan untuk satu iterasi dalam *extreme programming*. Untuk kasus penelitian ini pada iterasi pertama yaitu dengan melakukan observasi pada aplikasi lama untuk mendapatkan gambaran umum aplikasi yang sudah berjalan. Sedangkan secara umum tahapan ini berfungsi untuk merumuskan kebutuhan aplikasi yang akan dikembangkan dalam satu iterasi kedepan [5].
2. *Design* : tahapan kedua yaitu perancangan terhadap aplikasi yang akan dibuat. Perancangan aplikasi menggunakan alat bantu CRC (*Class Responsibility Collaborator*) yang

akan dipetakan menjadi UML (*Unified Modified Language*) diagram [1].

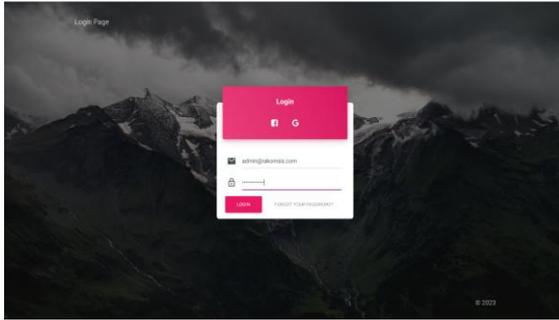
3. *Coding* : Pada fase ini yaitu melakukan pengembangan aplikasi dan pada fase ini melakukan *coding* dengan menerapkan beberapa praktik dari *extreme programming* seperti : *standard coding*, *pair programming*, dan *test-driven development* [6]. Sehingga secara praktik fase *coding* dilakukan bersamaan dengan fase *testing*. Khusus pada penelitian ini akan melakukan *coding* dengan menggunakan bahasa pemrograman PHP dengan *framework Laravel*.
4. *Testing* : seperti yang dijelaskan pada fase *coding*, fase ini dilakukan bersama – sama khususnya pada praktik seperti : *extreme programming* seperti : *standard coding*, *pair programming*, dan *test-driven development* [6]. Sedangkan terdapat aktivitas pengujian langsung oleh user atau disebut dengan *user acceptance test (UAT)*. Setelah dilakukan UAT, maka fitur aplikasi yang sudah sesuai dengan kebutuhan pengguna dirilis.

3. HASIL DAN PEMBAHASAN

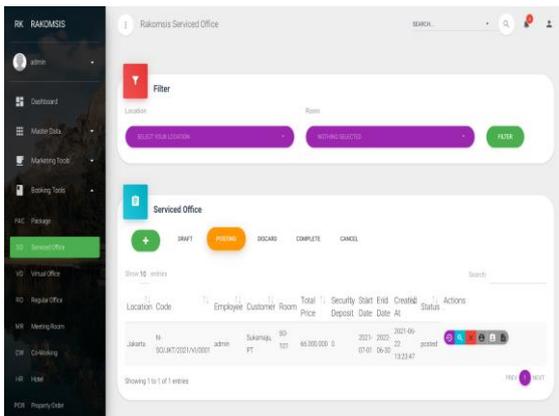
Hasil penelitian ini terbagi menjadi dua bagian, yaitu hasil observasi dan hasil pengembangan perangkat lunak dengan menerapkan empat proses *extreme programming*.

Hasil Observasi

Peneliti mendapatkan hasil arahan penggunaan aplikasi dari tim pengembang. Gambar 2 dan Gambar 3 berikut contoh dari tangkap layar aplikasi versi lama.



Gambar 2. Tampilan Login Aplikasi Lama



Gambar 3. Tampilan Menu Penjualan Aplikasi Lama

Aplikasi yang dituangkan kedalam *initial requirement*. Hasil observasi aplikasi yaitu sebagai berikut :

- Teknologi: Aplikasi berbasis web yang dibangun menggunakan bahasa pemrograman PHP dengan menggunakan *framework Laravel*. Database yang digunakan adalah *MariaDB*.
- Gambaran Umum : Aplikasi dibuat untuk menangani aktivitas *sales* untuk melakukan pencatatan penjualan dan pembuatan kontrak kerjasama, yang ditujukan untuk bisnis *serviced office*, *virtual office*, dan *MICE (Meeting Room, Incentive, Convention, and Exhibition)*
- Fitur - fitur aplikasi : *login, logout, update profile, update password, master data - access group, master data - location, master data - room, master data - product, master data – employee,*

master data – customer, manage sales order, dan dashboard.

- Kelompok Pengguna : *administrator, manager* atau *supervisor, dan salesman.*

Planning

Berdasarkan hasil *initial requirement*, langkah selanjutnya adalah perencanaan pengembangan aplikasi. Pada fase ini merumuskan rencana pengembangan aplikasi yang dituliskan pada Tabel 1.

Tabel 1. Rencana Pengembangan

| Fase | Penjelasan |
|-----------------------------|--|
| <i>Planning</i> | <ul style="list-style-type: none"> ▪ Pada fase ini menentukan fitur fitur yang dikembangkan ▪ Pengembangan fitur dalam 1x iterasi harus dapat dikerjakan oleh dua orang pengembang selama 2 – 3 pekan (10 - 15 hari kerja) waktu pengerjaan. |
| <i>Design</i> | <ul style="list-style-type: none"> ▪ Pada fase ini menterjemahkan fitur – fitur yang dikembangkan ke dalam CRC. ▪ Perancangan CRC harus memenuhi kaedah <i>multi tenancy</i>, sehingga satu aplikasi dapat melayani beragam kelompok pengguna. |
| <i>Coding & Testing</i> | <ul style="list-style-type: none"> ▪ Fase berikutnya adalah mengembangkan perangkat lunak berdasarkan CRC yang sudah dibuat. ▪ Fase pengembangan menggabungkan dua fase XP, yaitu <i>coding</i> dan <i>testing</i> sebagai bentuk implementasi praktik <i>test-driven development</i>. |
| <i>Software Increment</i> | <ul style="list-style-type: none"> ▪ Fase ini yaitu menerbitkan aplikasi secara bertahap. ▪ Ketentuan penerbitan aplikasi adalah <i>user stories</i> yang sudah dikembangkan dan memenuhi skenario pengujian |

Berdasarkan ketentuan pengembangan, maka perangkat lunak yang baru dikembangkan selama dua kali iterasi *extreme programming*. Tabel 2 berikut merupakan rincian rencana pengembangan perangkat lunak berdasarkan urutan iterasi, waktu pengerjaan per-iterasi, dan hasil konversi *initial requirement* menjadi *user stories*.

Tabel 2. Rincian Rencana Pengembangan

| Iterasi | Initial Requirement | CRC |
|--|----------------------------|----------------------------------|
| Pertama 10 hari kerja (2 pekan) | Login | User Autentikasi |
| | Logout | |
| | Update Profile | |
| | Update Password | Pengelolaan Hak Akses |
| | Master Data - Access Group | |
| | Master Data - Employee | |
| Kedua 15 hari kerja (3 pekan) | Master Data - Product | Pengelolaan Tim Internal |
| | Master Data - Location | Pengelolaan Data Utama Penjualan |
| | Master Data - Room | |
| | Master Data - Customer | Pengelolaan Pelanggan |
| | Manage Sales Order | Transaksi Penjualan |
| | Dashboard | |

Design

Pada fase perencanaan ini yaitu merumuskan kebutuhan perangkat lunak yang dituangkan menjadi CRC. Penekanan rancangan yaitu untuk bisa memfasilitasi model *multi-tenancy* yaitu dengan menghadirkan "kode user" sebagai pembeda antara satu tenant dengan tenant yang lain. Tabel 3 merupakan contoh CRC untuk user autentikasi.

Tabel 3. CRC User Autentikasi

| Nama Dokumen | Rancangan CRC |
|---------------------------------|---|
| Nama Aplikasi | User Autentikasi |
| Aktifitas | Fungsi Aplikasi |
| Pelaksanaan | 31 Juli 2023 s/d 1 Agustus 2023 |
| Penjelasan aplikasi yang dibuat | <ul style="list-style-type: none"> • User dapat login ke-dalam aplikasi dengan menggunakan kode user, email, dan password. • Jika salah satu dari autentikasi salah, maka aplikasi akan menolak dan menampilkan pesan gagal masuk (login). • Setelah berhasil masuk (login), user dibawa kehalaman profil dengan menyajikan informasi terkait data tenant dan data profil dari user tersebut. • Selanjutnya user dapat merubah data profil dan password. • User dapat keluar (logout) dari aplikasi dan masuk (login) kembali menggunakan password terbaru |

Coding

Pada fase ini yaitu melakukan *coding* aplikasi terhadap rancangan yang dibangun. *Coding* dalam *extreme programming* yaitu perlu memperhatikan penerapan praktik – praktik yang sudah disebutkan seperti *standard coding*, *pair programming*, dan *test-driven development*.

Penerapan praktik yang pertama adalah *pair programming* dengan total 2 (dua) orang *programmer*. Proses *coding* dengan cara saling bergantian antara kedua *programmer* dengan melakukan *coding* selama 25 menit dan istirahat selama 5 menit atau yang dikenal dengan *pomodoro technique* [7]. Tabel 4 merupakan ilustrasi pekerjaan *programmer* dalam satu hari kerja.

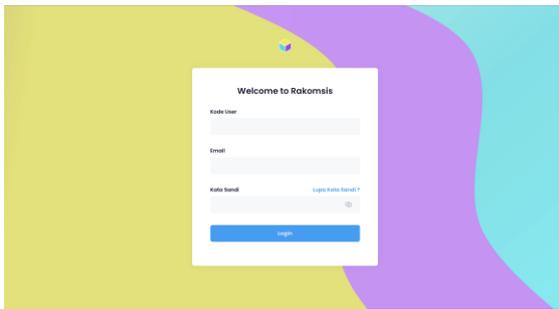
Tabel 4. Ilustrasi Pair Programming

| Waktu | Programmer 1 | Programmer 2 |
|---------------|--|--|
| 09:00 – 09:25 | Menulis program (<i>coding</i>) | Memantau tulisan program dari programmer 1 |
| 09:30 – 09:55 | Memantau tulisan program dari programmer 2 | Menulis program (<i>coding</i>) |
| 10:00 – 10:25 | Menulis program (<i>coding</i>) | Memantau tulisan program dari programmer 1 |
| 10:30 – 11:55 | Memantau tulisan program dari programmer 2 | Menulis program (<i>coding</i>) |

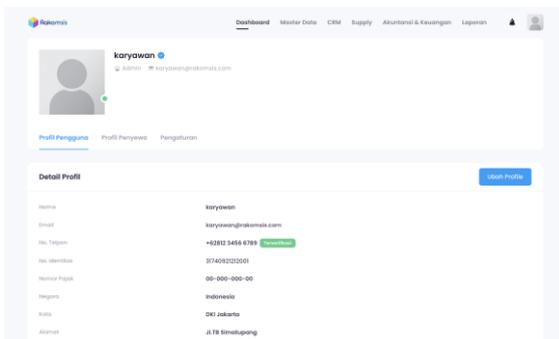
Penerapan praktik yang kedua adalah *standard coding* dengan mengikuti kaidah pada Bahasa pemrograman PHP dengan *framework Laravel* [8]. Serta penerapan praktik yang ketiga adalah *test-driven development* dengan menuliskan skenario pengujian terlebih dahulu sebelum menulis program utama. Skenario pengujian ditulis sesuai dengan CRC pada poin "penjelasan aplikasi yang dibuat".

Dalam Bahasa pemrograman PHP dengan *framework Laravel*, maka penerapan praktik *test-driven development* yaitu dengan menggunakan PHP *unit test*. Gambar 4 merupakan hasil pengerjaan aplikasi untuk CRC user autentikasi dengan menampilkan halaman login. Gambar 5 merupakan hasil pengerjaan aplikasi untuk

CRC *user* autentikasi dengan menampilkan halaman profil *user*.



Gambar 4. Tampilan Login Aplikasi Baru



Gambar 5. Tampilan Profil Aplikasi Baru

Testing

Pada fase ini yaitu melakukan *testing* aplikasi terhadap dibangun. Namun dalam penerapan praktik *test-driven development*, maka sebagian fase testing sudah dikerjakan bersamaan dengan fase *coding*. Tetapi ada praktik yang secara khusus dilakukan pada fase *testing* dan tidak dikerjakan bersamaan dengan fase *coding*, yaitu *user acceptance testing*.

User acceptance testing dilakukan terhadap setiap CRC yang sudah dikerjakan sampai fase *testing*. Contoh yang dilakukan yaitu melakukan pengujian terhadap CRC *user* autentikasi, dengan menjalankan skenario pada poin “penjelasan aplikasi yang dibuat”.

4. SIMPULAN

Kesimpulan yang diperoleh dari penelitian ini adalah metode *extreme programming* dapat dikerjakan dengan tim pengembang yang minim yang terdiri dari dua orang saja.

Selain itu metode *extreme programming* mempercepat proses penyajian aplikasi dengan iterasi yang singkat.

Penerapan praktik – praktik *extreme programming* berperan besar dalam menghasilkan perangkat lunak yang berkualitas. Praktik penulisan CRC pada fase perencanaan (*design*) memudahkan pemahaman bagi tim pengembang terhadap bentuk aplikasi yang ingin dibangun. Contoh pada penelitian ini adalah penerapan model *multi-tenancy*. Selain itu, praktik pada *extreme programming* fokus pada kualitas perangkat lunak seperti *standard coding* dan *test-driven development*. *Extreme programming* dapat meningkatkan kerjasama antara tim pengembang dengan praktik *pair programming*. Praktik yang menuntut dua orang menulis kode (*coding*) secara bersamaan dengan satu orang menulis dan satu orang mengawasi, sehingga pengawasan dilakukan bersamaan dengan penulisan program.

Saran untuk penelitian kedepannya, yaitu penggunaan metode *extreme programming* untuk tim pengembang yang lebih besar dari dua orang tim pengembang. Serta penerapan praktik – praktik *extreme programming* yang lebih banyak dalam proses pengembangan perangkat lunak.

DAFTAR PUSTAKA

- [1] R. I. Borman, A. T. Priandika, and A. R. Edison, “Implementasi Metode Pengembangan Sistem Extreme Programming (XP) pada Aplikasi Investasi Peternakan,” *Jurnal Sistem dan Teknologi Informasi (Justin)*, vol. 8, no. 3, p. 272, Jul. 2020, doi: 10.26418/justin.v8i3.40273.
- [2] E. S. Dasawati and B. S. P. Halim, “Pengembangan Aplikasi Odo dengan Command Line Interface Terhadap Warehouse Management System Menggunakan Model Extreme Programming Studi Kasus: Pt Indoguna Utama,” *Jurnal*

- Informatika Dan Bisnis*, vol. 8, no. 2, Dec. 2019.
- [3] S. Raghavan R., J. K.R., and R. V. Nargundkar, "Impact of software as a service (SaaS) on software acquisition process," *Journal of Business & Industrial Marketing*, vol. 35, no. 4, pp. 757–770, Apr. 2020, doi: 10.1108/JBIM-12-2018-0382.
- [4] W. Zheng *et al.*, "Target-based Resource Allocation for Deep Learning Applications in a Multi-tenancy System," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, IEEE, Sep. 2019, pp. 1–7. doi: 10.1109/HPEC.2019.8916403.
- [5] B. G. Sudarsono, S. P. Lestari, A. U. Bani, J. Chandra, and J. F. Andry, "Using an extreme programming method for hotel reservation system development," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 6, pp. 2223–2228, 2020, doi: 10.30534/ijeter/2020/01862020.
- [6] K. Haryono, "The extreme programming approach for financial management system on local government," in *Proceedings 2015 International Conference on Science and Technology, TICST 2015*, Institute of Electrical and Electronics Engineers Inc., 2015, pp. 29–34. doi: 10.1109/TICST.2015.7369335.
- [7] S. Mancuso, *The Software Craftsman: Professionalism, Pragmatism, Pride*. in Robert C. Martin series. Prentice Hall, 2014. [Online]. Available: <https://books.google.co.id/books?id=qqLfBQAAQBAJ>
- [8] Alexeymezenin, "Laravel best practices."

Click or tap here to enter text.