

## Implementasi Model PXP Pada Sistem Informasi Laboratorium Infokom Universitas Muhammadiyah Malang

Benny Indriawan Priatmaja\*<sup>1</sup>, Ilyas Nuryasin<sup>2</sup>, Wildan Suharso<sup>3</sup>

<sup>1,2,3</sup>Universitas Muhammadiyah Malang

benny.indriawanp@gmail.com\*<sup>1</sup>, ilyas@umm.ac.id<sup>2</sup>, wsuharso.umm.ac.id<sup>3</sup>

### Abstrak

Laboratorium Infokom merupakan Laboratorium yang secara langsung dikelola oleh Lembaga Infokom. Laboratorium Infokom juga memiliki fungsi untuk melaksanakan kegiatan pendidikan, pelatihan, dan praktikum teknologi informasi dan komunikasi serta mengelola data – data tersebut. Beberapa masalah yang terjadi dalam pelaksanaan kegiatan disebabkan oleh pengelolaan data kegiatan masih dijalankan secara manual. Oleh karena itu, diperlukan sebuah sistem informasi yang dapat dibangun secara tepat dan cepat. Metode pengembangan perangkat lunak yang dapat memenuhi hal tersebut termasuk kedalam agile software development. Personal Extreme Programming atau PXP merupakan salah satu metode yang ada didalam agile software development dan juga merupakan metode pengembangan dari Extreme Programming yang disesuaikan agar dapat dikerjakan oleh pengembang tunggal. Sistem informasi yang akan dibangun dalam penelitian ini ialah sistem informasi laboratorium. Tahapan awal dari PXP ialah pencarian kebutuhan setelah itu dilanjutkan dengan tahap perencanaan iterasi dengan menentukan prioritas kebutuhan dan diskusi bersama. Pengembangan sistem akan dilakukan secara iteratif berdasarkan perencanaan yang telah dilakukan sebelumnya. Apabila terdapat perubahan atau penambahan kebutuhan, maka akan dilakukan diskusi dan perencanaan ulang. Berdasarkan hasil penelitian ini, PXP dapat diimplementasikan dalam Studi Kasus ini. Kemampuan, pengalaman serta pemahaman dari seorang pengembang dalam mengerjakan suatu sistem dapat menentukan kesesuaian estimasi pengerjaan sistem dalam metode PXP. Interaksi dengan klien juga berperan penting dalam proses pengembangan sistem agar proses tersebut tetap terarah dan sesuai dengan kebutuhan klien.

**Kata Kunci:** Agile Software Development, Extreme Programming, Personal Extreme Programming, Sistem Informasi Laboratorium

### Abstract

Infokom Laboratory is a laboratory directly managed by the Infokom Institute. The Infokom Laboratory also has a function to carry out educational activities, training, and information and communication technology practicum as well as managing these data. Several problems that occurred in the implementation of activities were caused by manual activity data management. Therefore, we need an information system that can be built appropriately and quickly. Software development methods that can fulfill this are included in agile software development. Personal Extreme Programming or PXP is one of the methods that exist in agile software development and is also a development method of Extreme Programming that is tailored so that it can be done by a single developer. The information system to be built in this research is a laboratory information system. The initial stage of PXP is the search for needs, after which it is continued with the iteration planning stage by determining priority needs and joint discussion. The system development will be carried out iteratively based on the previous planning. If there are changes or additional needs, there will be discussion and re-planning. Based on the results of this study, PXP can be implemented in this case study. The ability, experience and understanding of a developer in working on a system can determine the suitability of the estimated system performance in the PXP method. Interaction with clients also plays an important role in the system development process so that the process remains focused and in accordance with client needs.

**Keywords:** Agile Software Development, Extreme Programming, Personal Extreme Programming, Information Sistem of Laboratory

---

## 1. Pendahuluan

Di era perkembangan teknologi informasi saat ini, kecepatan dan ketepatan informasi merupakan aspek penting dalam suatu organisasi agar mampu menyediakan pelayanan optimal bagi para pihak yang berhubungan. Dengan dikembangkannya bahasa pemrograman dan framework merupakan salah satu faktor pendukung untuk memudahkan orang membuat sistem informasi [1].

Sistem informasi merupakan teknologi informasi pada suatu organisasi yang memiliki program pengolahan transaksi harian sehingga dapat mendukung fungsi operasi organisasi yang bersifat strategis dan manajerial. Dengan sistem informasi suatu organisasi dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu [2].

Laboratorium Infokom merupakan Laboratorium yang secara langsung dikelola oleh Lembaga Infokom, Laboratorium Infokom juga memiliki fungsi untuk melaksanakan pendidikan, pelatihan, dan praktikum teknologi informasi dan komunikasi bagi internal kampus Universitas Muhammadiyah Malang. Laboratorium Infokom juga mengelola serta mengatur data – data tersebut sehingga data tersebut tersedia apabila dibutuhkan oleh pihak atasan atau pihak kampus [3]. Akan tetapi pada pelaksanaan pengelolaan data dan pelaporan data dari aktivitas pendidikan, pelatihan dan praktikum tersebut masih secara terpisah dan tidak dilakukan dalam satu sistem informasi. Keterbatasan fasilitas tersebut seringkali menjadi kendala dalam aktivitas yang dilakukan pada laboratorium infokom. Selain permasalahan diatas, tidak adanya sistem tersebut juga menyulitkan untuk menyediakan informasi atau laporan dari data tersebut agar dapat diakses secara real time seperti dalam hal jadwal dan peminjaman ruangan laboratorium yang masih manual. Dimana jadwal dan peminjaman ruangan laboratorium dilakukan dengan cara menginput menggunakan Ms. Office, perubahan pada jadwal hanya dapat dilihat dengan melapor ke kantor laboratorium dan papan pengumuman yang ada dilaboratorium infokom.

Dari permasalahan diatas peneliti ingin membuat sistem informasi laboratorium pada laboratorium infokom Universitas Muhammadiyah Malang dengan mengimplementasikan salah satu metode dari *agile software development* yaitu *extreme programming*. Pada penelitian yang dilakukan sebelumnya menjelaskan bahwa pada proses pengembangannya metode *extreme programming* merupakan konsep metode yang adaptif atau dapat menangani perubahan-perubahan [4]. *Extreme Programming* pada penerapannya biasa diimplementasikan oleh tim pengembang yang kecil, dan pengerjaan sistemnya dilakukan secara perpasangan [5]. Karena hal tersebut, peneliti tidak dapat mengimplementasikan metode *Extreme Programming*. Akan tetapi, dalam perkembangannya metode tersebut dikembangkan lagi menjadi *Personal Extreme Programming* sehingga dapat dilakukan oleh pengembang [5], [6].

## 2. Metodologi

Ini merupakan tahap awal dimana berfungsi sebagai landasan sebelum melanjutkan ke tahap-tahap berikutnya. Pada tahap studi literatur peneliti mengumpulkan dan mempelajari dari buku, jurnal, skripsi serta artikel yang berhubungan dengan tema yang peneliti ambil sebagai landasan dan diharapkan nantinya dapat membantu peneliti tentang teknik dan strategi dalam melaksanakan penelitian.

### 2.1 Penelitian Terdahulu

Pada bagian ini peneliti akan memaparkan beberapa penelitian yang sudah dilakukan sebelumnya. Penelitian-penelitian tersebut berhubungan dengan *Extreme Programming* dan *Personal Extreme Programming* dimana *Personal Extreme Programming* sendiri merupakan pengembangan dari metode *Extreme Programming*.

### 2.2 Pengembangan Sistem

Ini merupakan bagian yang akan menjelaskan prosedur – prosedur yang dilakukan oleh peneliti pada proses pengembangan. Metode pengembangan perangkat lunak yang akan digunakan adalah metode *Personal Extreme Programming (PXP)*. Pada bagian pengembangan sistem model pengembangan *PXP* memiliki beberapa langkah-langkah pengembangannya, *Requirements, Planning, Iteration Inialization, Design, Implementation, System Testing,* dan *Retrospective*. Pengembangan sistem informasi laboratorium ini akan dibangun dengan menggunakan bahasa pemrograman PHP yang dibantu dengan *framework* CodeIgniter dan sistem basis data MySQL. Berikut merupakan rincian dari penjelasan prosedur – prosedur yang ada pada metode pengembangan PXP.

### 2.2.1 Requirement

Tahap awal pengembangan ialah *requirement* atau pengumpulan kebutuhan – kebutuhan pada sistem yang akan dibangun. Pengumpulan kebutuhan tersebut dilakukan dengan cara wawancara atau diskusi langsung bersama klien. Kebutuhan–kebutuhan yang sudah dikumpulkan tersebut akan ditulis ulang dalam bentuk *user stories*.

### 2.2.2 Planning

Selanjutnya masuk ke tahap *planning*, pada tahap ini, pengembang melakukan perencanaan proses pengembangan sistem berdasarkan *requirements* yang sudah didapatkan. Pada proses *Planning Game*, pengembang akan menentukan prioritas, memberikan estimasi waktu pengerjaan dari masing - masing *user stories* dan menentukan jumlah iterasi yang akan dibutuhkan selama proses pengembangan terjadi.

### 2.2.3 Iteration Initialization

Pada *iteration development* tahap awal yang dilakukan ialah *iteration inialization*. Pada tahap ini akan ditentukan iterasi - iterasi yang akan di kerjakan. Iterasi tersebut memiliki fokus utama yaitu, memilih *user story* berdasarkan prioritas. Prioritas tersebut didapatkan dari hasil perencanaan pada tahap *planning game*.

### 2.2.4 Design

Pada tahap *Design*, pengembang akan membuat model dari modul sistem yang akan di implementasikan selama proses iterasi. Desain modul tersebut hanya memenuhi kebutuhan dari klien, akan tetapi desain tersebut bisa saja sewaktu–waktu berubah mengikuti hasil diskusi dari klien dan pengembang. Desain yang di buat oleh pengembang merupakan *Mock up* antarmuka dari sistem yang juga merupakan model desain *prototype*, desain tersebut berdasarkan *CRC Cards* dari iterasi yang sedang dijalankan.

### 2.2.5 Implementation

Pada tahap *Implementation* akan dibuat kode pemrograman berdasarkan desain yang sudah dibuat sebelumnya. Objek-objek yang terdapat pada tahap design akan diimplementasikan oleh pengembang. Tahap ini dibagi lagi menjadi tiga fase, yaitu *Unit Testing*, *Code Generation*, dan *Code Refactoring*.

#### a. Unit Testing

*Unit testing* adalah pengujian pada fitur dari iterasi yang diimplementasikan. Karena pengembang membuat sistem menggunakan bahasa pemrograman PHP, maka *unit testing* yang akan dilakukan oleh pengembang ialah menggunakan *library* PHP unit.

#### b. Code Generation

Pada tahap *code generation* pengembang akan membuat kode program pada setiap modul dari *user stories* pada iterasi yang diimplementasikan. Kode program akan dibuat menggunakan bahasa pemrograman php dan dibantu dengan *framework* CodeIgniter.

#### c. Code Refactoring

*Code refactoring* adalah kode program yang sudah dibuat akan di optimalisasi dan kode program tersebut akan diintegrasikan dengan kode program yang sudah dibuat pada iterasi sebelumnya.

### 2.2.6 System Testing

Pada tahap *system testing* semua modul hasil dari implementasi pada setiap iterasi akan dilakukan pengujian fungsionalitas. Pengujian yang akan dijalankan yaitu *User Acceptance Test* dimana klien sendiri yang akan melakukan pengujian sistem.

Pada proses pengujian klien akan didampingi oleh pengembang agar klien dapat lebih mudah mengetahuinya fitur - fitur sistem yang dikerjakan dalam proses iterasi. Hal ini juga bertujuan untuk mengetahui kecocokan antara fitur sistem yang sudah dihasilkan dengan kebutuhan awal pada tahap *requirement* dan *planning*. Kemudian klien memberikan verifikasi terhadap fitur - fitur tersebut apakah sudah memenuhi kriteria yang diinginkan oleh klien.

### 2.2.7 Retrospective

*Retrospective* merupakan tahap terakhir dari proses iterasi. Pada tahap *retrospective* dilakukan verifikasi terhadap semua tahapan pada setiap iterasi. Verifikasi tersebut ialah melakukan analisis terhadap jalannya setiap tahap pengembangan sistem, penyesuaian estimasi waktu pengerjaan, dan faktor penyebab terjadinya keterlambatan ketika proses pengembangan. Verifikasi dilakukan agar pada iterasi selanjutnya hal tersebut tidak terulang kembali.

### 2.3 Penyusunan Laporan

Penyusunan laporan adalah tahap akhir dari penelitian. Pada tahap ini, aktivitas yang terjadi pada proses pengembangan sistem, kemudian hasil yang didapatkan, dan kesimpulan dari penelitian akan peneliti tulis kedalam laporan. Sistematika dari penulisan laporan ini terdiri dari pendahuluan, tinjauan pustaka, metodologi penelitian, hasil dan pembahasan, serta penutup.

## 3. Hasil Penelitian dan Pembahasan

Pada tahap ini, langkah-langkah dalam pengembangan sistem menggunakan metode *Personal Extreme Programming* dan hasil evaluasi dari pengembangan tersebut, berikut penjelasannya.

### 3.1 Requirement

Pada tahap ini pengembang akan mengumpulkan kebutuhan fungsional dan kebutuhan non-fungsional pada sistem. Ada banyak metode yang bisa digunakan salah satunya ialah dengan cara melakukan wawancara dan diskusi bersama klien yang terlibat. Pada wawancara tersebut klien akan menguraikan kendala yang terjadi. Kendalanya adalah masih adanya keterbatasan dalam penyampaian informasi dari setiap kegiatan yang ada dilaboratorium tersebut. Karena masih belum adanya sistem yang mengelola klien juga kesulitan dalam melakukan pelaporan dari setiap kegiatan yang ada dilaboratorium. Berdasarkan permasalahan diatas, kemudian klien akan menyebutkan kebutuhan - kebutuhan dari sistem yang akan dibangun nantinya.

Setelah melakukan wawancara kebutuhan-kebutuhan tersebut nantinya akan ditulis dalam bentuk user story dengan format "*Sebagai <jenis pengguna>, saya ingin <melakukan tindakan tertentu> sehingga <mendapatkan manfaat dari tindakan tersebut>*". Setelah dilakukan proses requirements, pengembang mendapatkan 11 *user stories*. *User stories* yang telah didapatkan dapat dilihat pada Tabel 1.

Tabel 1. User Stories

Kode	User Stories
Stories-01	Sebagai Admin, saya ingin menambahkan data peserta pelatihan dan mengorganisir data tersebut, sehingga data peserta dapat tersimpan didalam sistem dan mempermudah saya memeriksa apabila ada terjadi kesalahan pada pengimputan data tersebut.
Stories-02	Sebagai Admin, saya ingin menambahkan nilai evaluasi pelatihan peserta dan mengorganisir data tersebut, sehingga dapat mengetahui peserta mana yang lulus dan tidak lulus pelatihan.
Stories-03	Sebagai Admin, saya ingin menambahkan data peserta dan mengorganisir data tersebut, sehingga data peserta dapat tersimpan didalam sistem dan mempermudah saya memeriksa apabila ada terjadi kesalahan pada pengimputan data tersebut.
Stories-04	Sebagai Admin, saya ingin menambahkan nilai evaluasi praktikum peserta dan mengorganisir data tersebut, sehingga dapat mengetahui peserta mana yang lulus dan tidak lulus praktikum.
Stories-05	Sebagai Admin, saya ingin menambahkan data peserta recruitment dan mengorganisir data tersebut, sehingga data peserta dapat tersimpan didalam sistem dan mempermudah saya memeriksa peserta yang sudah memenuhi kriteria recruitment.

Stories-06	Sebagai Admin, saya ingin menambahkan tes evaluasi recruitment peserta dan mengorganisir data tersebut, sehingga dapat mengetahui peserta mana yang lulus dan tidak lulus recruitment.
Stories-07	Sebagai Admin, saya ingin membuat jadwal pemakaian ruangan laboratorium sehingga saya dapat mengorganisir jadwal ruangan agar tidak terjadi bentrokan antar jadwal kegiatan.
Stories-08	Sebagai Peserta, saya ingin melihat profil data dan hasil evaluasi milik saya sehingga saya dapat mengetahui apabila ada kesalahan dan pengumuman hasil dari kegiatan saya ikuti.
Stories-09	Sebagai Peserta, saya ingin melihat jadwal kegiatan laboratorium pada sistem sehingga saya dapat mengetahui apabila ada perubahan jadwal pada laboratorium.
Stories-10	Sebagai Ketua Lab, saya ingin melihat laporan jumlah peserta, proses dari setiap pertemuan dan hasil akhir sehingga saya dapat memonitoring kegiatan yang ada pada Laboratorium.
Stories-11	Sebagai Ketua Lab, saya ingin melihat jadwal kegiatan laboratorium pada sistem sehingga saya dapat mengkonfirmasi apabila ada perubahan jadwal pada laboratorium.

### 3.2 Planning

Pada tahap ini pengembang akan menyusun beberapa *task* yang akan dikerjakan pada tiap-tiap iterasi berdasarkan dari *User Stories* yang sudah didapatkan. *Practice planning game* akan digunakan pengembang dalam penyusunan *task*. Pada proses *Planning Game*, pengembang akan memberikan estimasi waktu pengerjaan dari masing-masing *user stories*. Kemudian, pengembang akan melakukan analisa terhadap prioritas yang akan diberikan ke masing-masing *user stories*. Penentuan prioritas tersebut dilakukan menggunakan penyortiran *user stories* berdasarkan *value* dan *risk*.

Nilai bisnis dari *user stories* akan menentukan *value* dari *user stories* tersebut. Sebuah *user stories* dapat dibagi menjadi tiga *value* yaitu, *critical*, *significant business value*, dan *nice to have*. Kemudian ada tiga kategori *risk* dari sebuah *user stories* yaitu, *low*, *medium*, dan *high*. *Risk index* akan menentukan seberapa besar *risk* pada *user stories*. Kategori - kategori tersebut ialah *completeness*, *volatility*, dan *complexity*. Setelah estimasi waktu pengerjaan dan prioritas dari *user stories* tersebut sudah diketahui, pengembang akan menentukan *user stories* apa saja yang akan dikerjakan tiap iterasinya.

Setelah dilakukan tahap estimasi waktu pengerjaan dan penentuan prioritas dari masing-masing *user stories*, tahap *planning* dilanjutkan dengan perencanaan iterasi. Dalam proses pengembangannya, PXP dilakukan secara pengulangan atau iterasi. Dalam proses tersebut penentuan *user stories* yang dikerjakan dalam iterasi ditentukan dengan berdiskusi dengan klien dan berdasarkan prioritas dari *user stories* tersebut akan ditemukan nilai *velocity*.

Nilai *velocity* merupakan nilai yang ditentukan oleh pengembang untuk menentukan lamanya proses iterasi. *Story points* dari masing-masing *user stories* akan dijumlahkan hingga memiliki nilai yang sama dengan nilai *velocity* dan akan membentuk iterasi pertama. *User story* yang tidak termasuk kedalam iterasi pertama akan dijumlahkan lagi hingga membentuk iterasi kedua. Hal ini dilakukan sampai tidak ada *user stories* yang tersisa. Hasil perencanaan iterasi dapat dilihat pada Tabel 2.

Tabel 2. Hasil Perencanaan Iterasi

Iterasi	Kode User Stories	Value	Risk	Story Points
Iterasi ke-1	Stories-01	Critical	(4) Medium	4
	Stories-02	Critical	(2) Medium	2
Iterasi ke-2	Stories-03	Critical	(4) Medium	4
	Stories-04	Critical	(2) Medium	2
Iterasi ke-3	Stories-05	Critical	(4) Medium	4
	Stories-06	Critical	(2) Medium	2
Iterasi ke-4	Stories-10	Critical	(3) Medium	3
	Stories-11	Critical	(3) Medium	3
Iterasi ke-5	Stories-07	Critical	(4) Medium	3

Stories-08	Critical	(3) Medium	2
Stories-09	Critical	(3) Medium	2

Pada tabel diatas nilai velocity pada tiap iterasi ialah bernilai 6 hanya iterasi terakhir yang bernilai 7, sehingga tiap iterasi akan dikerjakan selama 12 hari dan iterasi terakhir selama 14 hari. Keseluruhan *user stories* berjumlah 11, yang rencananya akan dikerjakan selama 62 hari. Selama proses pengembangan sistem, klien dapat mengusulkan perubahan ataupun penambahan *user stories* baru. Pengembang kemudian akan mengestimasi dan menentukan prioritas dari *user stories* tersebut, lalu menambahkannya kedalam proses iterasi yang sedang berjalan ataupun iterasi berikutnya.

### 3.3 Iteration Initialization

Pada tahap ini pengembang akan mengerjakan iterasi ke- 1, pada iterasi ke-1 berisi Stories-01 dan Stories-02. Stories-01 memiliki fungsi untuk mengelola data peserta dari pelatihan dari penambahan peserta baru hingga menghapus data peserta. Sedangkan untuk Stories-02 memiliki fungsi mengelola data nilai peserta pelatihan hingga dijadikan dalam bentuk diagram laporan.

### 3.4 Design

Pada tahap ini, pengembang akan membuat desain berdasarkan iterasi yang sedang berjalan. Pengembang akan membuat desain *CRC Card* dan *spike solution prototype*. *CRC Card* akan menunjukkan tugas dari masing-masing *class* serta *class* lain yang berinteraksi dengan *class* tersebut. *Spike solution prototype* merupakan skema desain interface berdasarkan *CRC Card* dari *user stories* pada iterasi ke- 1.

Berdasarkan hasil analisa Stories-01, diketahui terdapat tiga objek didalam *user stories* ini yaitu mahasiswa, dosen, dan staff. *CRC Card* dari *user stories* ini dapat dilihat pada gambar 1 berikut.

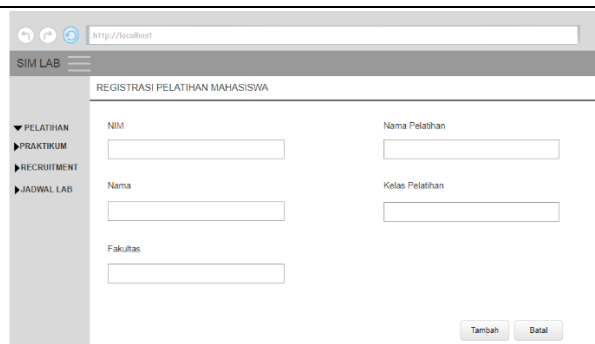
<<CRC>> Pelatihan Mahasiswa	<<CRC>> Pelatihan Dosen	<<CRC>> Pelatihan Staff
<b>RESPONSIBILITIES</b> <ul style="list-style-type: none"> <li>- Menambahkan peserta mahasiswa</li> <li>- Melihat daftar peserta pelatihan mahasiswa</li> <li>- Mengedit data peserta mahasiswa</li> <li>- Menghapus data peserta mahasiswa</li> </ul>	<b>RESPONSIBILITIES</b> <ul style="list-style-type: none"> <li>- Menambahkan peserta dosen</li> <li>- Melihat daftar peserta pelatihan dosen</li> <li>- Mengedit data peserta dosen</li> <li>- Menghapus data peserta dosen</li> </ul>	<b>RESPONSIBILITIES</b> <ul style="list-style-type: none"> <li>- Menambahkan peserta staff</li> <li>- Melihat daftar peserta pelatihan staff</li> <li>- Mengedit data peserta staff</li> <li>- Menghapus data peserta staff</li> </ul>
<b>COLLABORATIONS</b> <ul style="list-style-type: none"> <li>- Mahasiswa DB</li> <li>- Jurusan DB</li> <li>- Pelatihan DB</li> </ul>	<b>COLLABORATIONS</b> <ul style="list-style-type: none"> <li>- Dosen DB</li> <li>- Jurusan DB</li> <li>- Pelatihan DB</li> </ul>	<b>COLLABORATIONS</b> <ul style="list-style-type: none"> <li>- Staff DB</li> <li>- Jurusan DB</li> <li>- Pelatihan DB</li> </ul>

Gambar 1. *CRC Card* dari *user stories-01*

*Card* tersebut menunjukkan responsibility serta collaborator dari class pelatifa mahasiswa, pelatihan dosen, dan pelatihan staff. Responsibility dari class pelatihan mahasiswa terdiri dari menambahkan, melihat, merubah, menghapus data mahasiswa yang ada pada sistem. Sedangkan untuk collaborator pada class tersebut yaitu database mahasiswa, jurusan dan pelatihan. Pada class pelatihan dosen terdiri dari menambahkan, melihat, merubah, dan menghapus data dosen yang pada sistem. Sedangkan untuk collaborator pada class tersebut yaitu database dosen, jurusan dan pelatihan. Class pelatihan staff juga terdiri dari menambahkan, melihat, merubah, dan menghapus data staff yang ada pada sistem. Sedangkan untuk collaborator pada class tersebut yaitu database staff, jurusan dan pelatihan.

Berdasarkan *CRC Cards* tersebut, pada halaman pengelolaan mahasiswa terdapat data – data mahasiswa yang sudah diinputkan ke dalam sistem. Untuk penambahan data mahasiswa dapat dilakukan melalui tombol tambah data baru, lalu sistem akan menampilkan form yang akan diisi oleh user. Sedangkan untuk mengubah dan menghapus data dapat dilakukan melalui tombol aksi yang tersedia.

Begitu juga fungsi yang berlaku pada halaman data dosen dan staff. Desain *prototype* dari *CRC Cards* yang telah dibuat dapat dilihat pada Gambar 2 dibawah.



Gambar 2. Prototype Stories-01 Form Registrasi Pelatihan Mahasiswa

Pada Stories-02 juga dapat dilakukan analisa sehingga objek pada user stories dapat diubah menjadi *CRC Cards* kemudian dari cards tersebut ditemukan *responsibility* serta *collaborator*. Setelah *responsibility* dan *collaborator* diketahui tahap selanjutnya ialah membuat desain *prototype* dari *CRC Card* yang telah dibuat seperti yang dilakukan pada Stories-01.

### 3.5 Implementation

Setelah menyelesaikan desain *prototype* maka akan dilanjutkan dengan tahapan selanjutnya yaitu pembuatan kode program. Pada fase ini akan dibagi menjadi tiga tahap, yaitu *Unit Testing*, *Code Generation*, dan *Code Refactoring*.

#### a. Unit Testing

Pada proses pembuatan kode program untuk unit testing, mula-mula pengembang membuat unit testing berdasarkan berdasarkan *CRC Card* yang telah dibuat. Pembuatan unit testing menggunakan library *PHPUnit*. Unit testing dijalankan dengan menggunakan fitur *Test Suite* agar hanya unit test tertentu saja yang dijalankan.

Dalam iterasi ini, terdapat beberapa test yang dibuat kedalam 9 buah file, yakni unit test untuk pelatihan mahasiswa, pelatihan dosen, pelatihan staff, evaluasi nilai sementara dan nilai akhir untuk mahasiswa, dosen, dan staff.

Unit testing tersebut terdiri dari 29 skenario pengujian database pada sistem yang akan dibuat. Pengujian itu terdiri dari menambahkan, mengubah, dan menghapus data mahasiswa, dosen, dan staff serta pengelolaan data evaluasi nilai mereka kedalam kegiatan pelatihan.

#### b. Code Generation

Setelah semua unit testing telah lulus pengujian, proses pengembangan sistem selanjutnya ialah dengan melengkapi kode program untuk setiap user stories dari iterasi ke-1. Kode program tersebut tersebut terdiri dari fitur-fitur untuk memasukkan, mengubah, dan menghapus data mahasiswa, dosen dan staff serta menambahkan, mengubah dan menghapus data mereka kedalam kegiatan pelatihan.

#### c. Code Refactoring

Di tahap ini setelah kode program selesai dibuat, pengembang melakukan optimalisasi terhadap kode program dan mengintegrasikan kode program tersebut dengan kode program yang telah dibuat pada iterasi sebelumnya. Berdasarkan hasil implementasi yang telah dilakukan, pengembang mengintegrasikan kode program untuk pengelolaan mahasiswa, dosen, staff peserta pelatihan dan evaluasi pelatihan dengan sistem utama seperti mengatur route untuk fitur-fitur pada modul tersebut.

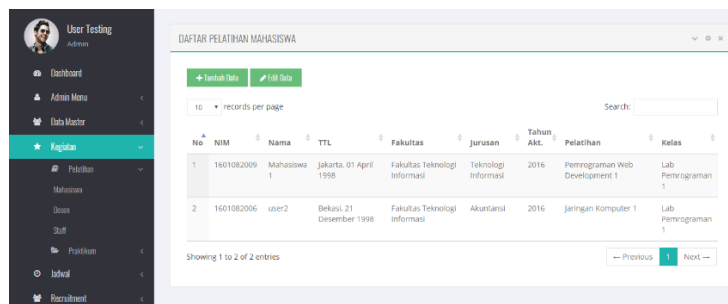
### 3.6 System Testing

Tahap selanjutnya merupakan tahap system testing atau pengujian terhadap keseluruhan hasil implementasi user stories pada iterasi ke-1. Pengujian akan dilakukan oleh klien dan setelah pengujian klien akan memberikan *feedback* atau masukan mengenai sistem. Sistem dihosting agar dapat diakses secara online. Selanjutnya pengembang akan mengarahkan user agar dapat mengoperasikan fungsi – fungsi sistem yang di kerjakan dalam proses iterasi. Hasil pengujian dapat dilihat pada Tabel 3.

Tabel 3. Hasil System Testing

User Story	Unit Modul	Hasil Uji
Stories-01	Menambahkan peserta baru	Berhasil
	Melihat daftar peserta pelatihan	Berhasil
	Mengedit data mahasiswa	Berhasil
	Menghapus data mahasiswa	Berhasil
	Mengedit data dosen	Berhasil
	Menghapus data dosen	Berhasil
	Mengedit data staff	Berhasil
	Menghapus data staff	Berhasil
	Stories-02	Menambahkan nilai pelatihan peserta
Menampilkan daftar nilai peserta		Berhasil
Mengedit nilai pelatihan peserta		Berhasil
Menghapus nilai pelatihan peserta		Berhasil

Berdasarkan hasil yang ditunjukkan pada tabel 3, disimpulkan bahwa hasil implementasi telah diterima oleh klien dan ada sedikit masukan dari klien pada bagian tampilan menu pada sistem. Berikut tampilan dari hasil implementasi dapat dilihat pada Gambar 3.



Gambar 3. Tampilan Daftar Pelatihan Mahasiswa

Tampilan pada gambar 3 merupakan tampilan dari daftar pelatihan mahasiswa yang ada didalam sistem. User dapat menambahkan peserta pelatihan yang baru dengan menekan tombol "Tambah Data" dan sistem akan menampilkan form untuk diisi dengan data peserta pelatihan. Untuk melakukan perubahan atau menghapus data dapat dilakukan dengan menekan tombol "Edit Data".

### 3.7 Retrospective

Setelah tahap pengujian telah selesai dan diterima oleh user, akan dilanjutkan dengan tahap restrospective. Pada tahap ini pengembang menganalisis jalannya setiap fase pengembangan modul serta kesesuaian estimasi waktu pengerjaan pada iterasi ke-1. Hasil verifikasi estimasi waktu pengerjaan iterasi ke-1 dapat dilihat pada Tabel 4.

Tabel 4. Verifikasi Estimasi Waktu Pengerjaan Iterasi ke-1

Iterasi ke-1			
User Story	Story Points	Waktu Estimasi	Waktu Pengerjaan
Stories-01	4	8	11
Stories-02	2	4	4
Total	6	12	15

Pada Tabel 4, dapat dilihat bahwa waktu pengerjaan keseluruhan user stories pada iterasi ke-1 adalah 15 hari. Hal ini menandakan bahwa waktu pengerjaan tidak sesuai dengan waktu estimasi yang telah ditentukan pada tahap *planning*. Hal ini disebabkan karena kurangnya kemampuan pengembang dalam pembuatan unit testing dan penggunaan PHPUnit. Selain itu, pada awalnya pengembang cukup kesulitan dalam pembuatan fungsi menambahkan dan mengubah peserta pelatihan. Pada bagian tersebut user meminta fitur agar sistem dapat



melakukan autoloading dimana pada saat penambahan data sistem akan secara otomatis mencari data peserta dengan cara memasukkan nomor induk milik peserta sehingga menyebabkan waktu pengerjaan bertambah 3 hari ketika pengerjaan Stories-01. Ketika pengerjaan Stories-02, pengembang tidak mengalami masalah tersebut karena pengembang telah mempelajari hal tersebut ketika melakukan pengerjaan user stories yang sebelumnya. Selama proses iterasi ke-1, user hanya meminta sedikit perubahan pada tampilan menu. Proses iterasi ini terus dilakukan hingga proses pengembangan sistem telah selesai.

### 3.8 Hasil Evaluasi

Pada bagian ini akan menjelaskan hasil evaluasi dari proses pengembangan sistem selama 5 iterasi yang telah dilakukan. Evaluasi tersebut ialah pengerjaan user stories yang melebihi estimasi pengerjaan dan timeline selama proses pengembangan sistem informasi laboratorium infokom Universitas Muhammadiyah Malang.

Selama proses pengembangan sistem, terdapat pengerjaan user stories yang melebihi estimasi dikarenakan kurangnya kemampuan dan pengalaman pengembang dalam pengelolaan waktu dan pengembangan suatu sistem berbasis web. Daftar user stories yang pengerjaannya tidak sesuai dengan estimasi dapat dilihat pada Tabel 5 berikut.

*Tabel 5. Ketidaksesuaian Estimasi Pengerjaan User Stories*

User Story	Prioritas		Waktu	Waktu
	Value	Risk	Estimasi	Pengerjaan
Stories-01	Critical	Medium (4)	8	11
Total			8	11

Tabel di atas menunjukkan user stories yang pengerjaannya melebihi waktu estimasi yang ditentukan. Bertambahnya lama waktu pengerjaan disebabkan karena alur program pada modul yang rumit sehingga waktu pengerjaan bertambah 3 hari. Kurangnya pengalaman pengembang dalam mengerjakan modul tersebut membuat pengembang tidak dapat menyelesaikan iterasi tersebut secara akurat. Dalam proses pengembangan sistem, pengembang juga terbantu dengan adanya package pada teknologi framework yang memudahkan pengembang dalam mengerjakan modul-modul lain yang memiliki fungsi yang hampir sama sehingga dapat dikerjakan sesuai dengan estimasi waktu yang diperkirakan. Kemampuan pemahaman dan pengalaman dari seorang pengembang dalam mengerjakan suatu sistem menjadi penentu untuk menyesuaikan estimasi pengerjaan sistem dalam metode PXP.

Ketidaksesuaian waktu pengerjaan user stories pada iterasi ke-1 menyebabkan estimasi awal pengembangan sistem yang digunakan yaitu 62 hari menjadi bertambah 3 hari dan total keseluruhan 65 hari.

## 4. Kesimpulan dan Saran

### 4.1 Kesimpulan

Berdasarkan hasil penelitian selama pengembangan sistem informasi laboratorium Infokom Universitas Muhammadiyah Malang didapatkan kesimpulan:

1. Bahwa pada studi kasus ini metode pengembangan Personal Extreme Programming dapat diimplementasikan.
2. Pada proses pengembangan pengumpulan kebutuhan dilakukan melalui diskusi bersama klien. Kemudian hasil dari diskusi dengan klien akan diproses dan direpresentasikan menjadi user stories. Ada 11 user stories yang sudah didapatkan berdasarkan kebutuhan – kebutuhan dari klien.
3. Penentuan estimasi waktu dari pengembangan sistem adalah 62 hari. Namun dalam implementasinya sendiri, waktu pengembangan sistem yang digunakan bertambah 3 hari dan menjadi 65 hari dikarenakan pengembang cukup kesulitan dalam pembuatan fungsi menambahkan dan mengubah data peserta pelatihan. Pada bagian tersebut user meminta fitur agar sistem dapat melakukan autoloading dimana pada saat penambahan data, sistem akan secara otomatis mencari data peserta dengan cara memasukkan nomor induk milik peserta sehingga menyebabkan waktu pengerjaan bertambah 3 hari.

**Referensi**

- [1] D. Riana, R. Sanjaya, O. Kalsoem, "Sistem Informasi Manajemen Laboratorium Patologi Anatomi Menggunakan Model MVC Berbasis Laravel Framework," *Konferensi Nasional Sistem Informasi (KNSI)*, pp. 237-242, Maret, 2018.
- [2] V. R. Handayani, R. Wijianto, A. Anggoro, "Sistem Informasi Pendaftaran Seleksi Kerja Berbasis Web pada BKK (Bursa Kerja Khusus) Tunas Insan Karya SMK Negeri 2 Banyumas," *Jurnal Evolusi*, Vol. 6., No. 1., pp. 76-84, 2018.
- [3] Lembaga Infokom, "Tugas Pokok dan Fungsi (Tupoksi) Infokom Universitas Muhammadiyah Malang" [online]. Available: <http://infokom.umm.ac.id/id/pages/tupoksi.html>
- [4] R. Rahmi, R. Perdana Sari, and R. Suhatman, "Pendekatan Metodologi Extreme Programming pada Aplikasi E-Commerce (Studi kasus Sistem Informasi Penjualan Alat-alat Telekomunikasi)," *J. Komput. Terap.*, vol. 2, pp. 83-92, 2016.
- [5] R. Agarwal and D. Umphress, "Extreme Programming for a Single Person Team," in *Proceedings of the 46<sup>th</sup> Annual Southeast Regional Conference*, pp. 82-87, 2008.
- [6] Y. Dzhuurov, I. Krasteva, and S. Ilieva, "Personal Extreme Programming - An Agile Process for Autonomous Developers," 2009.
- [7] I. G. Stamelos and P. Sfetsos, *Agile Software Development Quality Assurance*. Information Science Reference, 2007.
- [8] R. F. Malik, M. Fachrurrozi, R. Prabowo, "Sistem Informasi Manajemen Laboratorium Menggunakan Metode Agile dengan Konsep Model-View-Controller Data Access Object," *KNTIA*, Vo. 4., pp. 65-69, Januari, 2017.
- [9] S. Suehring and J. Valade, *PHP, MySQL, JavaScript & HTML All-in-One For Dummies*. Hoboken, New Jersey: John Wiley & Sons, Inc, 2003.
- [10] R. Fahmillah, *Pembangunan Website Pondok Pesantren Darul Iman Karawang Menggunakan Framework CodeIgniter*. Bandung: Fakultas Teknik UNPAS, 2018.
- [11] G. I. Marthasari, W. Suharso, and F. Ardiansyah, "Personal Extreme Programming with MoSCoW Prioritization for Developing Library Information System," *Proceeding of EECSI*, pp. 537-541, October, 2018.
- [12] A. F. Septiyanto, W. Suharso, and I. Nuryasin, "Sistem Informasi Program Keluarga Harapan (PKH) Menggunakan Metode Personal Extreme Programming dengan Metode Prioritas Ranking," *Jurnal Repositor*, Vol. 2., No. 12, 2020.
- [13] H. Rizal, S. Adhy, P. W. Wirawan, "Perancangan dan Pembuatan Mobile Learning Interaktif Berbasis Android Dengan Metode Personal Extreme Programming," *Journal of Informatics and Technology*, Vol. 2., No. 3., pp. 103-112, 2013.
- [14] S. F. Taslim, G. I. Marthasari, and W. Suharso, "Rancang Bangun Sistem Manajemen Pergudangan Berbasis Website Pada PT. Astragraphia (Cabang Depo Jayapura)," *Jurnal Repositor*, Vol. 2., No. 6, 2020.
- [15] R. G. Hidayat, I. Nuryasin, and W. Suharso, "Implementasi Sistem Informasi Penjualan dan Persediaan Menggunakan webERP Pada Cribro Inc Malang," *Jurnal Repositor*, Vol. 2., No. 8, 2020.
- [16] S. Kosasi, "Pembuatan Sistem E-Commerce Produk Meubel Berbasis Komponen," *IC ITECS*, November, 2014.
- [17] A. Hidayat, Aminudin, I. Nuryasin, "Rancang Bangun Sistem Informasi Pembukuan Keuangan Menggunakan Arsitektur Hierarchical Model view controller Dengan Pendekatan PXP (Studi Kasus: CV. Anugrah Mandiri)," *Jurnal Repositor*, Vol. 2., No.7, 2020.