

Implementasi Model *Personal Extreme Programming* Pada Perancangan Aplikasi *Project Report*

Risah Subariah*

Sistem Informasi, Universitas Pamulang, Tangerang Selatan, Indonesia

*e-mail *Corresponding Author*: risah.subariah@gmail.com

Abstract

Project reports are needed to ensure that a project is completed on schedule and produces an appropriate product. Dero Jasa Web Apps is a software company that focuses on creating software applications. This company has many projects, so it requires project activity reports from programmers to managers. The Personal Extreme Programming (XP) model applied in design only focuses on specifications and design. This stage includes Unified Modeling Language, database design, and interface design. The design of features and menus on the application display is adjusted to user access rights, thereby maintaining data privacy, and improving the performance of programmers and managers in creating project reports and monitoring project progress.

Keywords: *Project; Report; Monitoring; Management*

Abstrak

Project report diperlukan untuk memastikan bahwa suatu proyek selesai sesuai jadwal dan menghasilkan produk yang sesuai. Dero Jasa Web Apps merupakan perusahaan perangkat lunak yang berfokus pada pembuatan aplikasi perangkat lunak. Perusahaan ini memiliki banyak project sehingga memerlukan laporan kegiatan project dari programmer kepada manager. Model Personal Extreme Programming (XP) yang diterapkan dalam perancangan hanya berfokus pada persyaratan dan desain. Tahapan ini meliputi Unified Modeling Language, perancangan database, dan desain antarmuka. Rancangan fitur dan menu pada tampilan aplikasi disesuaikan dengan hak akses pengguna, sehingga menjaga kerahasiaan data dan meningkatkan kinerja programmer dan manager dalam membuat laporan project serta memantau kemajuan project.

Kata kunci: *Project; Report; Pemantauan; Manajemen*

1. Pendahuluan

Project adalah aktivitas sementara yang memiliki durasi terbatas, memerlukan sumber daya tertentu, dan bertujuan untuk menghasilkan produk atau hasil dengan kriteria kualitas yang jelas [1]. Keberhasilan suatu *project* tidak semata-mata ditentukan oleh waktu penyelesaian dan hasil akhir, namun juga oleh pemantauan berkala terhadap laporan kemajuan *project* agar tetap mendapat informasi mengenai perkembangan *project*. Laporan atau *report* merupakan suatu dokumen yang memberikan penjelasan mengenai peristiwa atau kegiatan yang telah dilakukan oleh seseorang atau suatu kelompok.

Project report diperlukan untuk memastikan suatu *project* selesai tepat waktu dan menghasilkan produk yang sesuai dengan kebutuhan. Dero Jasa Web Apps merupakan *software house* yang bergerak dalam bidang pembuatan aplikasi perangkat lunak. *Software house* ini memiliki banyak *project* perangkat lunak sehingga dalam operasionalnya membutuhkan laporan *project* yang dilakukan oleh *programmer* kepada *project manager*. Aktivitas *project report* yang telah berjalan pada Dero Jasa Web Apps memanfaatkan *Google Sheet* dan *electronic mail* sebagai media pelaporan *project* antar *programmer* dengan *manager*. Namun, hal ini dianggap kurang efektif karena terdapat ratusan *project* yang akan diperiksa oleh *manager* setiap harinya. Akibatnya, ketika data *project* diperlukan untuk laporan, pengumpulannya memerlukan waktu karena diperlukannya *email programmer* sebagai bukti laporan *project*.

Berdasarkan penelitian terdahulu terkait dengan pelaporan kegiatan harian proyek, menunjukkan hasil bahwa dengan adanya aplikasi *monitoring* kegiatan proyek dapat membantu perusahaan dalam memantau kemajuan proyek [2]. Penelitian serupa juga pernah dilakukan oleh beberapa peneliti yang membahas pelaporan aktivitas proyek yang sebelumnya dilakukan secara manual, penelitian tersebut membuktikan bahwa dengan adanya aplikasi *project monitoring* dapat memudahkan dalam pengumpulan informasi terkait aktivitas harian analisis dan kemajuan *project* [3].

Dari pembahasan yang telah dijelaskan sebelumnya, maka diperlukan perancangan aplikasi *project report* yang diharapkan dapat memantau dan mendokumentasikan detail penting perkembangan *project* yang ada pada Dero Jasa Web Apps. Tahapan perancangan aplikasi *project report* akan menggunakan model *Personal Extreme Programming* (PXP). Model PXP merupakan proses pengembangan perangkat lunak yang dirancang khusus untuk insinyur perangkat dalam penelitian ini meliputi tahapan *requirements, planning, iteration initialization* dan *design* [4]. Sehingga proses perancangan aplikasi yang dikembangkan akan menyesuaikan dengan proses pengembangan perangkat yang ada pada model PXP. Dengan adanya rancangan aplikasi *project report* diharapkan dapat menjadi solusi bagi Dero Jasa Web Apps dalam menangani permasalahan pelaporan *project*.

2. Tinjauan Pustaka

Pada penelitian yang dilakukan oleh Andi Megawati dan Dian Gustina yang membahas mengenai pembuatan aplikasi yang dapat membantu manajer proyek dalam pekerjaannya. Aplikasi tersebut dapat digunakan untuk membuat, mengatur, dan memantau informasi proyek berbasis web dengan metode *Web Development Life Cycle* (WDLC) [2].

Abert Tandilintin dkk. Dengan judul "Perancangan Aplikasi *Project Monitoring* Pada PT Cyber Solution Berbasis Web" yang membahas mengenai *project monitoring* yang dilakukan secara manual. Penelitian tersebut membuat aplikasi yang dapat membantu berbagai langkah seperti mendaftarkan proyek, pembentukan tim, dan mengirimkan aplikasi proyek sehingga memudahkan pengumpulan informasi dari klien [3].

Penelitian ini menyajikan beberapa penelitian terdahulu yang sesuai dengan topik *project report*. Handrie Noprisson yang membahas mengenai perancangan aplikasi monitoring magang dilakukan oleh mahasiswa dengan menerapkan *Personal Extreme Programming* (PXP). Penelitian ini menghasilkan aplikasi yang dapat memantau aktivitas magang yang dilakukan oleh mahasiswa [5].

Agung Sirajuddin Vidianto dan Wachyu Hari Haji yang membahas mengenai pembuatan aplikasi khusus untuk membantu memonitor proyek-proyek. Dalam sistem tersebut, dapat melihat perkembangan proyek, memberikan tugas dan sumber daya, serta membuat laporan tentang apa yang telah dilakukan [6].

Bagus Budi Pangestu dan Niki Ratama membahas mengenai pembuatan sistem informasi manajemen proyek dengan pendekatan *Rapid Application Development* (RAD) yang dapat mengatur sistem penjadwalan kerja sehingga menunjang kemajuan pekerjaan proyek [7].

Penelitian ini mengeksplorasi topik yang sering dibicarakan, khususnya pemantauan proyek dalam suatu perusahaan. Namun ada beberapa aspek yang membedakannya dengan penelitian-penelitian sebelumnya. Misalnya, peneliti menggunakan metode *Personal Extreme Programming* (PXP) dalam aplikasi *project report*, kemudian rancangan desain dan fitur aplikasi yang dibangun. Selain itu, penelitian ini memiliki fokus objek kajian yang berbeda dibandingkan dengan penelitian sebelumnya.

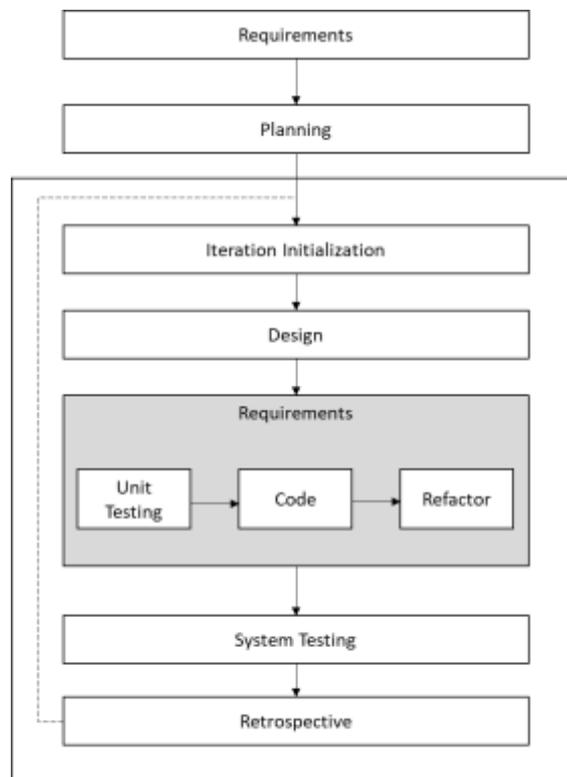
3. Metodologi

3.1. Metode Pengumpulan Data

Metode penelitian berfungsi sebagai prosedur, menguraikan berbagai langkah dan yang digunakan untuk mengatasi dan memecahkan masalah secara efektif, memastikan tingkat pemahaman dan kejelasan yang lebih tinggi. Dalam penelitian ini menggunakan metode kualitatif, yaitu pengumpulan data didapat dari observasi, studi pustaka dan wawancara. Data yang telah dikumpulkan akan dijadikan bahan untuk merancang aplikasi *project report* yang ada pada *software house* Dero Jasa Web Apps.

3.2. Metode Pengembangan Sistem

Personal Extreme Programming (XP) menjadi model pengembangan sistem yang digunakan dalam penelitian ini. *Personal Extreme Programming* (XP) adalah pendekatan pengembangan perangkat lunak yang beroperasi dengan keyakinan bahwa proses pembuatan sistem informasi sering mengalami modifikasi [8]. Dalam kerangka XP, ada tujuh fase berbeda yang diikuti selama proses pengembangan seperti pada Gambar 1 di bawah ini.



Gambar 1. Fase *Personal Extreme Programming*

Penelitian ini akan berfokus pada tahap perancangan aplikasi, sehingga pembahasan yang akan dijelaskan yaitu *requirements*, *planning*, *iteration initialization*, dan *design*.

1) *Requirements*

Langkah pertama dalam proses ini adalah mengumpulkan dan mengidentifikasi persyaratan yang diperlukan untuk hasil yang diinginkan [8]. Tahap ini melibatkan pengumpulan dan analisis data untuk menentukan persyaratan aplikasi, serta persyaratan perangkat keras dan perangkat lunak. Setelah persyaratan ini diidentifikasi, langkah selanjutnya adalah menganalisisnya dan membentuk arsitektur program yang sesuai. Peneliti akan mengumpulkan informasi dari *user* yang akan menggunakan sistem dengan berbicara kepada mereka atau melihat dokumen tentang apa yang dapat sistem lakukan.

2) *Planning*

Selama fase perencanaan, fokusnya adalah menentukan skala prioritas, dan fitur aplikasi, serta mengidentifikasi kebutuhan fungsional dari pengembangan proses bisnis atau aplikasi. Fase ini melibatkan pemeriksaan persyaratan untuk pengembangan sistem yang akan dibangun dan penyusunan strategi untuk mendapatkan pemahaman komprehensif tentang karakteristik dan pengoperasian perangkat lunak yang akan dikembangkan (*user stories*) [9].

3) *Iteration Initialization*

Tahap inisialisasi iterasi meliputi transformasi hasil perencanaan dari tahap perencanaan ke dalam format *Unified Modeling Language* (UML) untuk memberikan penjelasan

rinci mengenai fungsionalitas aplikasi yang akan dikembangkan. Fase inialisasi iterasi melibatkan penentuan fungsionalitas dan alur bisnis yang ditetapkan, yang ditentukan selama tahap perencanaan dan kemudian dijelaskan lebih lanjut melalui penggunaan diagram [8].

4) *Design*

Pada tahap desain, pengembang akan membuat model sistem yang akan dibangun selama proses iterasi. Pada tahap khusus ini, setiap model sistem akan didokumentasikan dalam bentuk *Entity-Relationship Diagram* (ERD) untuk tujuan perancangan *database* serta merancang tampilan antarmuka pengguna [10].

3.3. Identifikasi Kebutuhan

Tahapan ini termasuk kedalam bagian *requirements* dalam model *Personal Extreme Programming* (PXP). Berikut ini adalah identifikasi kebutuhan pada sistem:

1) *Data User*

- a) Sistem dapat mengelola data user.
- b) Sistem memiliki level untuk membedakan user dengan level *programmer* dan *manager*.
- c) Sistem dapat melakukan pencarian data *user*.
- d) Sistem dapat mencetak data *user* yang telah dimiliki oleh Dero Jasa Web Apps.

2) *Data Project*

- a) Sistem dapat mengelola data *project*.
- b) Sistem memiliki periode *project* baik tanggal mulai dan tanggal selesai.
- c) Sistem dapat melakukan pencarian data *project*.
- d) Sistem dapat mencetak data *project* yang telah dimiliki oleh Dero Jasa Web Apps.

3) *Project Activity*

- a) Sistem dapat mengelola data *project activity*.
- b) *Project activity* memiliki kolom yang berisi kegiatan yang telah dilakukan oleh *programmer*.
- c) Terdapat filter tanggal *project activity* untuk menampilkan *project activity status* yang telah dicek dan yang menunggu untuk dicek.

4) *Project Activity Checking*

- a) Sistem dapat mengelola *project activity checking* yang telah diisi oleh semua level *programmer*.
- b) Terdapat filter tanggal *project activity checking* untuk menampilkan *project activity checking* yang telah dicek dan yang menunggu untuk dicek.

5) *Report*

- a) Sistem dapat mengelola data *project activity report* secara keseluruhan.
- b) Terdapat filter tanggal untuk menampilkan periode *project* dan *project activity* yang telah dilakukan.
- c) Sistem dapat mencetak *report* yang telah dipilih.

3.4. Analisis Kebutuhan Sistem

Pada model *Personal Extreme Programming* (PXP) bagian ini merupakan tahapan yang ada dalam *planning*. Analisis kebutuhan sistem mencakup analisis kebutuhan fungsional dan non-fungsional. Berikut adalah analisis kebutuhan sistem dalam aplikasi *project report*.

3.4.1. Analisis Kebutuhan Fungsional

Persyaratan fungsional mengacu pada data spesifik yang perlu dimasukkan ke dalam sistem agar dapat memenuhi fungsi yang dimaksudkan [11]. Persyaratan ini berfungsi sebagai landasan bagi pengoperasian sistem dan memainkan peran penting dalam menentukan fungsionalitas dan efektivitas keseluruhan. Berikut ini adalah analisis kebutuhan fungsional dalam aplikasi *project report*:

1) *Manager*

- a) *Manager* dapat mengelola (menambah, mengubah, menghapus, dan mencetak) data *user*.
- b) *Manager* dapat mengelola (menambah, mengubah, menghapus, dan mencetak) data *project*.

- c) *Manager* dapat mengelola (mengubah, menghapus, dan mencetak) data *project activity checking*.
- d) *Manager* dapat mencetak *project report* yang telah berjalan.
- 2) *Programmer*
 - a) *Programmer* dapat menambah data *project activity*.
 - b) *Programmer* dapat mengubah dan menghapus data *project activity* yang belum dicek oleh *manager*.

3.4.2. Analisis Kebutuhan Non Fungsional

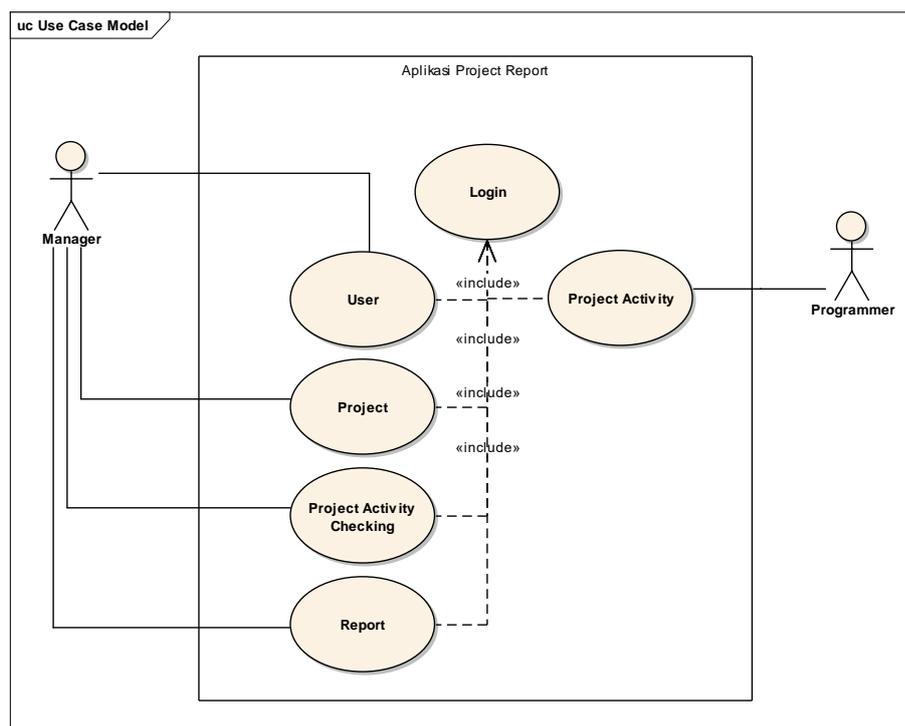
Proses analisis kebutuhan non-fungsional yang ada dalam penelitian ini adalah sebagai berikut:

- 1) Tampilan
 - Tampilan aplikasi yang dirancang mudah digunakan (*user friendly*).
- 2) Akses
 - a) Permintaan data pada bagian laporan dapat diakses tidak boleh lebih dari 15 menit.
 - b) Aplikasi membutuhkan akses internet agar dapat terhubung dengan *database*.

3.5. Implementasi Unified Modeling Language (UML)

Tahap *Iteration Initialization* dalam model PXP melibatkan konversi dari tahap perencanaan ke dalam format *Unified Modeling Language* (UML) untuk memberikan penjelasan komprehensif tentang fungsionalitas aplikasi *project report*.

3.5.1. Use Case Diagram



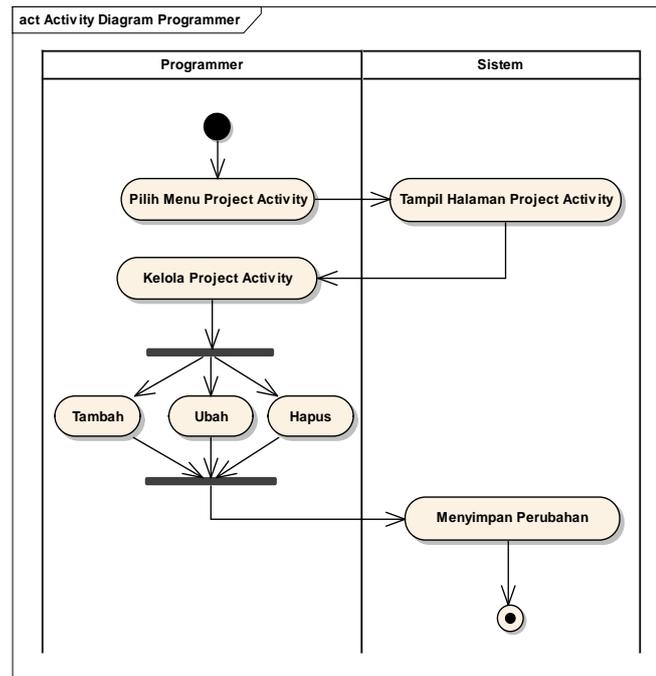
Gambar 2. Use Case Diagram Aplikasi Project Report

Use case diagram adalah alat yang berguna untuk memodelkan suatu sistem dan secara efektif menggambarkan interaksi antara aktor dan sistem [11]. Selain itu, diagram *use case* memfasilitasi pengembangan aplikasi dan memberikan pemahaman yang jelas tentang bagaimana setiap aktor berinteraksi dan fungsionalitas yang tersedia dalam sistem. Pada *use case project report* terdapat dua aktor yaitu *manager* dan *programmer*. Aktor *programmer* wajib *login* terlebih dahulu untuk dapat mengisi *project activity* dan hanya dapat mengakses *project*

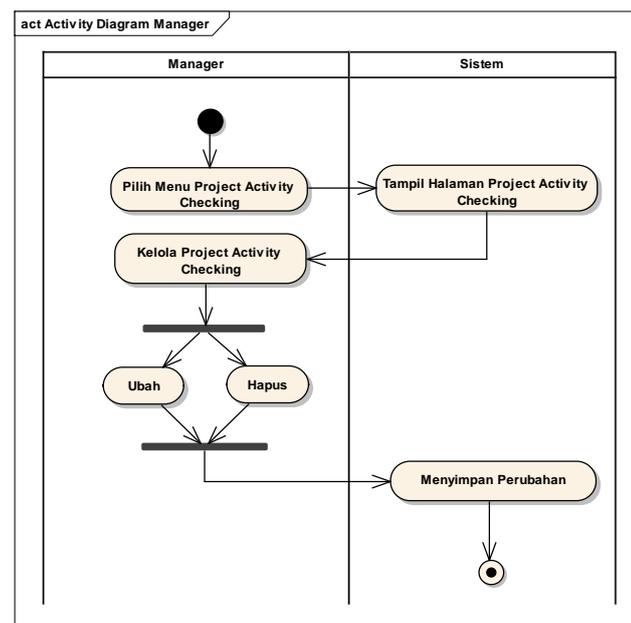
activity. Untuk aktor *manager* dapat mengelola *user*, *project*, *project activity checking*, dan *report* yang ada dalam aplikasi *project report*

3.5.2. Activity Diagram

Activity Diagram adalah representasi visual yang menguraikan urutan tugas atau tindakan yang terlibat dalam sistem atau proses bisnis [12]. Dengan memberikan gambaran secara *detail* tentang bagaimana berbagai aktivitas berinteraksi dan mengalir dalam aplikasi *project report*, memungkinkan pemahaman yang lebih baik tentang fungsionalitas dan efisiensinya. Pada Gambar 3 merupakan *activity diagram programmer* dalam aplikasi *project report*. *Programmer* dapat memilih menu *project activity* kemudian melakukan penambahan, perubahan, dan juga penghapusan selama data *project activity* belum divalidasi oleh *manager*.



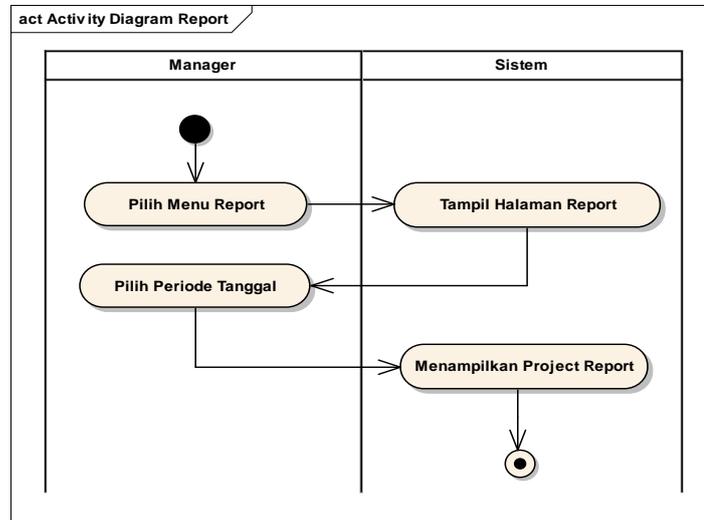
Gambar 3. Activity Diagram Programmer



Gambar 4. Activity Diagram Manager

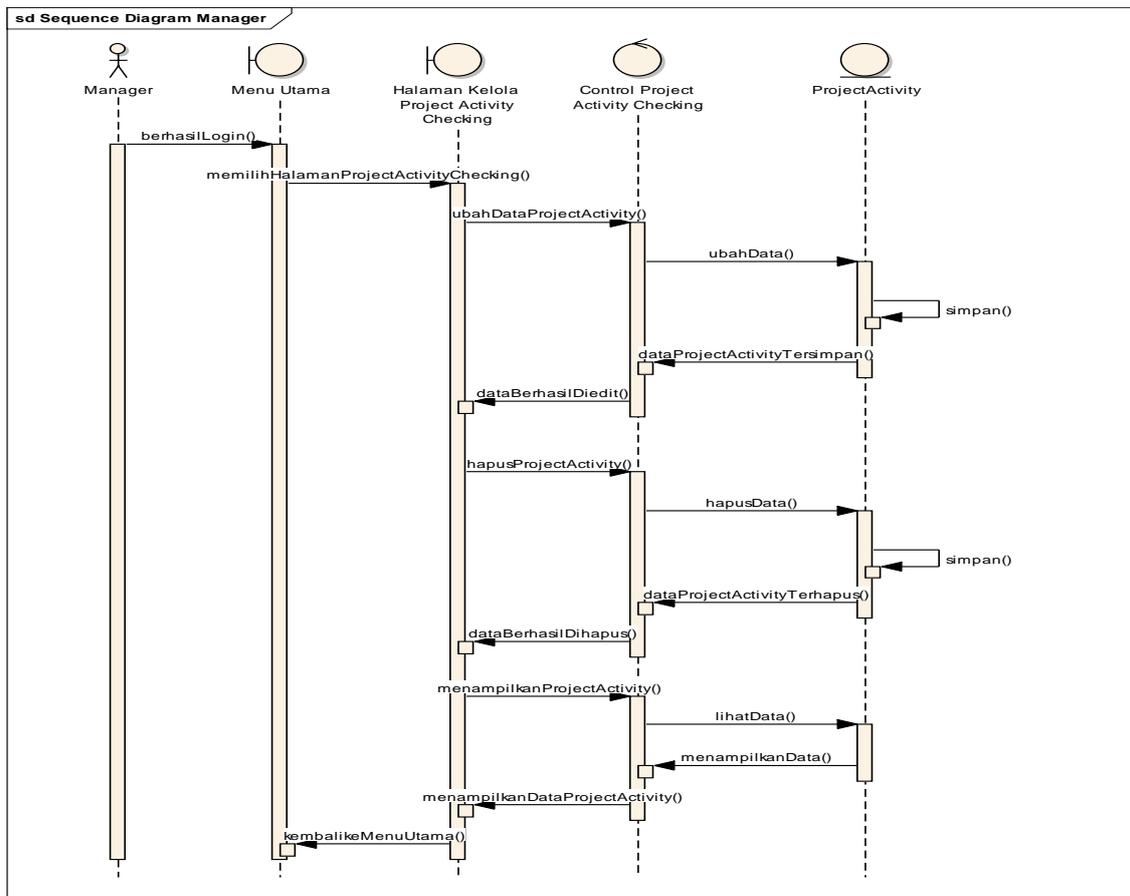
Pada Gambar 4 merupakan *activity diagram manager*, yang menggambarkan aktivitas *manager* pada *project activity checking*. *Manager* dapat mengubah status *project activity* dan memberikan catatan apabila diperlukan pada *project activity checking*.

Activity diagram report memberikan gambaran aktivitas *manager* dalam menampilkan data *project* yang ada dalam aplikasi *project report*.



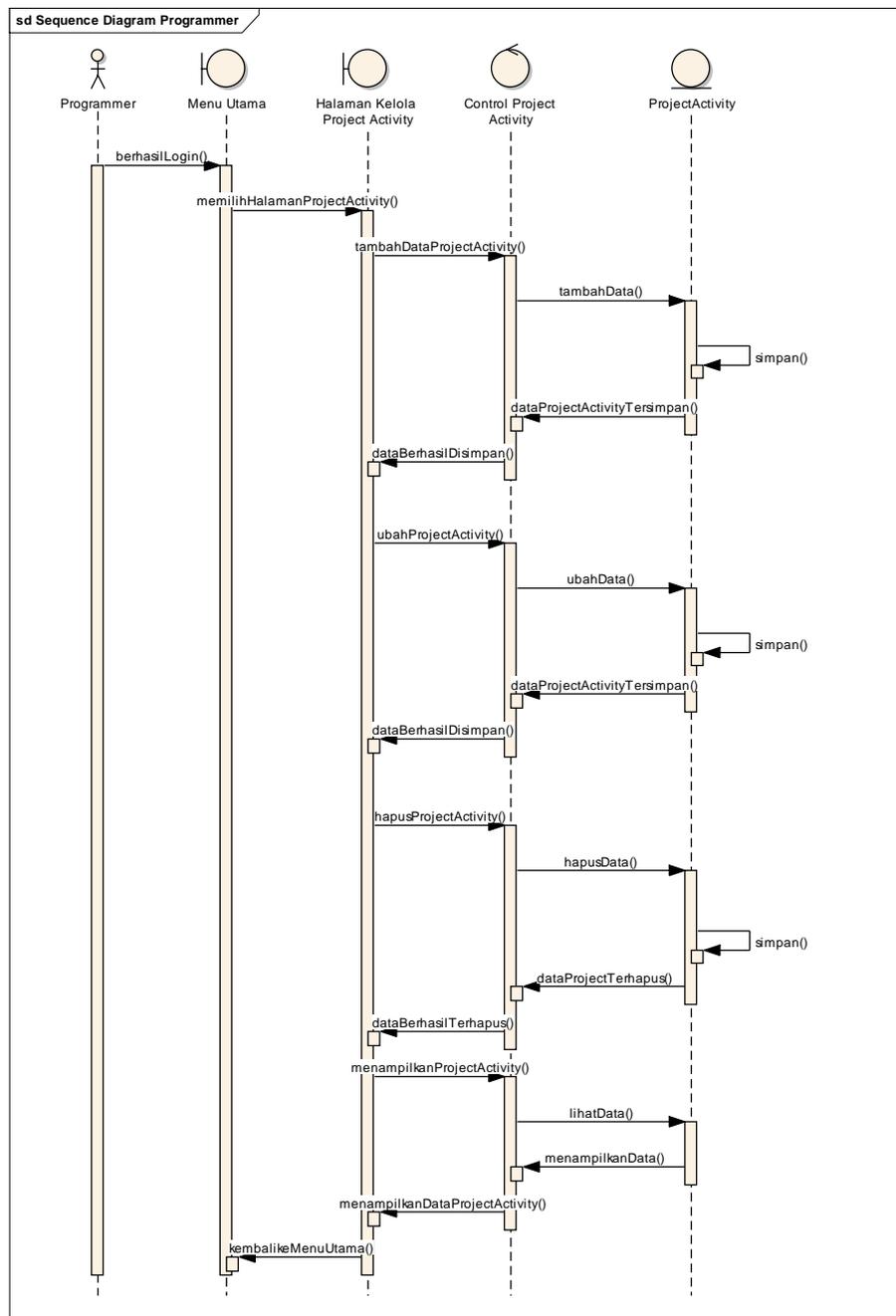
Gambar 5. Activity Diagram Report

3.5.3. Sequence Diagram



Gambar 6. Sequence Diagram Manager

Sequence diagram merupakan representasi visual yang menggambarkan aliran interaksi dan komunikasi antara berbagai objek dalam suatu sistem [12]. Pada *sequence diagram project activity checking manager* dapat melakukan interaksi dalam halaman *project activity checking*. *Manager* dapat melakukan perubahan setelah mengecek *project activity* yang diisi oleh *programmer* dengan memberikan keputusan apakah *project activity* yang telah diisi oleh *programmer* telah sesuai dengan *progress* yang ada.

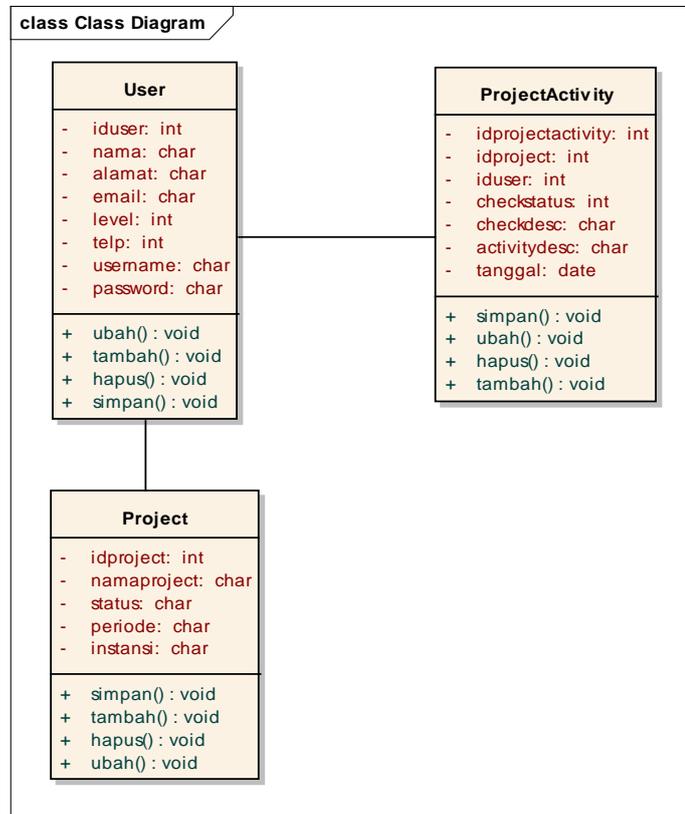


Gambar 7. *Sequence Diagram Programmer*

Gambar 7 menggambarkan *sequence diagram* yang dirancang khusus untuk *programmer*, merepresentasikan berbagai interaksi yang terjadi selama pengelolaan *project activity*. Pada diagram ini memungkinkan *programmer* untuk memasukkan rincian aktivitas *project* yang telah dikerjakan, serta memodifikasi atau menghapus data *project activity* yang ada jika diperlukan.

3.5.4. Class Diagram

Class diagram memberikan representasi ilustratif arsitektur sistem dengan menguraikan berbagai kelas yang akan dikembangkan untuk membangun sistem [13]. *Class diagram* aplikasi *project report* ini berfungsi sebagai alat visual untuk menjelaskan struktur sistem melalui penggambaran hubungan dan interaksi antar kelas diantaranya *User*, *Project* dan *Project Activity*.

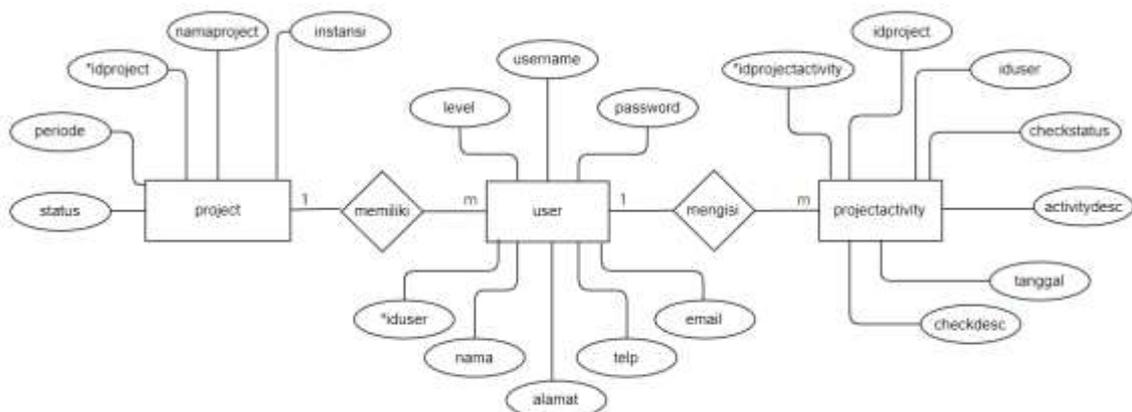


Gambar 8. Class Diagram

3.6. Pemodelan Sistem

Pemodelan sistem termasuk dalam tahap design yang ada pada PXP. Tahap ini menggambarkan ERD dan Diagram Konteks aplikasi *project report* yang akan dikembangkan.

3.6.1. Entity Relational Diagram (ERD)



Gambar 9. ERD Aplikasi Project Report

Entity Relational Diagram (ERD) adalah model konseptual yang menggambarkan hubungan antar penyimpanan [14]. ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Berikut ini adalah ERD pada perancangan aplikasi *project report*, terdapat 3 entitas yang saling terhubung pada Gambar 9

3.6.2. Diagram Konteks

Diagram konteks berfungsi sebagai desain awal sistem, memungkinkan pemahaman yang lebih mendalam tentang cara kerja sebuah sistem dan disebut sebagai level 0 (No1) [15]. Pada aplikasi *project report* terdapat dua pengguna yaitu *manager* dan *programmer* yang saling terhubung dengan aplikasi *project report* yang digambarkan dalam diagram konteks pada Gambar 10.



Gambar 10. Diagram Konteks Aplikasi *Project Report*

4. Pembahasan

Penelitian ini mengambil landasan dari temuan penelitian sebelumnya [5], yang juga mengeksplorasi dan meneliti penelitian serupa.

4.1 Rancangan Desain Antarmuka Pengguna

Tahapan design dalam PXP yang terakhir adalah merancang tampilan antarmuka pengguna pada aplikasi *project report*. Terdapat tiga tampilan rancangan yang nantinya akan diimplementasikan ke dalam aplikasi *project report*.

4.1.1. Halaman *Login*

Pada Gambar 11 merupakan halaman *login* yang berfungsi sebagai halaman awal yang menyambut pengguna saat membuka aplikasi melalui *browser*. Pengguna dipersilakan untuk mengisi *username* dan *password*, setelah itu dapat melanjutkan dengan memilih tombol *login*. Jika pengguna lupa *password*, maka dapat memilih "lupa *password*" yang akan mengarahkan pada halaman lupa *password*. Hal ini memudahkan pengguna, karena sebelumnya proses pelaporan *programmer* diwajibkan untuk masuk ke akun Google masing-masing.

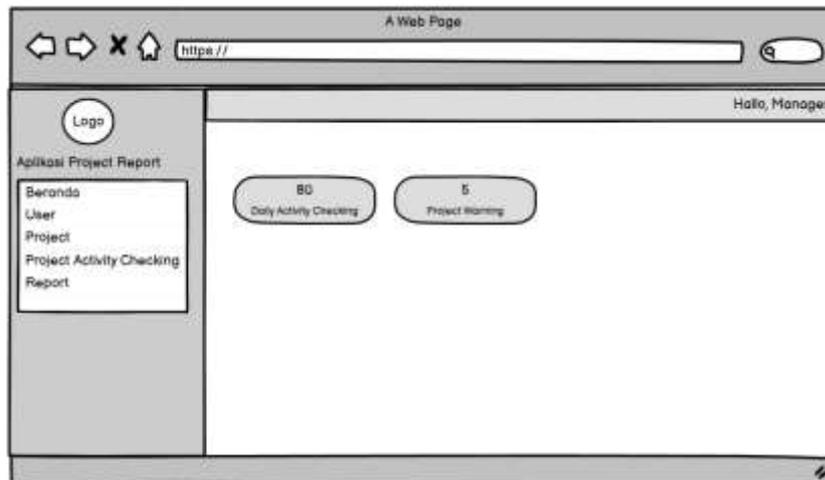
The screenshot shows a login form titled 'Login' for 'Aplikasi Project Report'. It features a circular logo placeholder. Below the logo are two input fields: 'Username' and 'Password'. At the bottom left is a 'Login' button, and at the bottom right is a link labeled 'Lupa Password'.

Gambar 11. Halaman *Login*

Perancangan sistem didasarkan pada dua hak akses yaitu *manager* dan *programmer*. *Manager* mempunyai akses penuh terhadap seluruh fitur yang ada pada sistem, sedangkan *programmer* hanya terbatas pada menu *project activity*. Selain itu, hanya *manager* yang diperbolehkan mendaftarkan akun baru di sistem.

Desain antarmuka pengguna disesuaikan dengan kedua hak akses tersebut. Antarmuka *manager* dirancang khusus untuk memenuhi kebutuhan, dengan fitur seperti menambahkan pengguna, *project*, *project activity checking*, dan *report*. Sedangkan *programmer* hanya memiliki akses untuk mengisi *project activity*. Tampilan hak akses yang disesuaikan dengan pengguna dapat memastikan setiap hak akses pengguna, sehingga menjaga kerahasiaan data pada aplikasi *project report*.

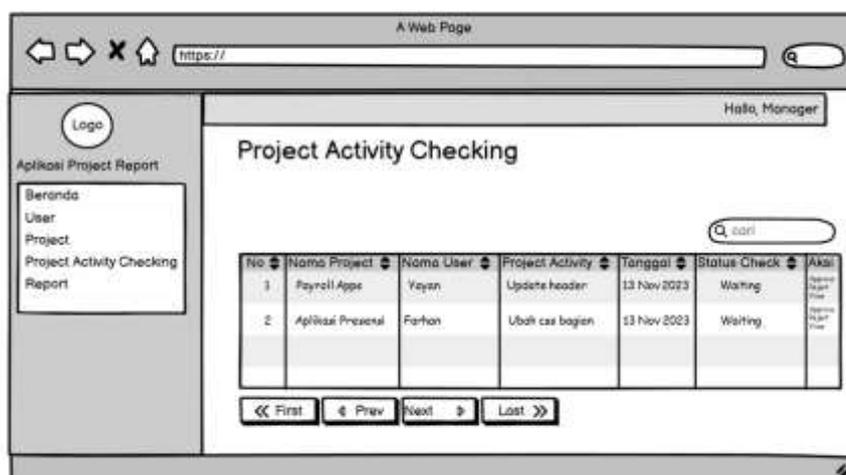
4.1.2. Halaman Beranda *Manager*



Gambar 12. Halaman Beranda *Manager*

Gambar di atas menampilkan *landing page* utama *manager*, yang berfungsi sebagai pusat berbagai aktivitas. Secara khusus, serangkaian laporan *project activity programmer* menunggu *approval manager* dalam antrian yang sistematis. Selain itu, dengan lokasi strategis di beranda ini, *manager* diberikan akses mudah ke proyek-proyek yang mendekati tenggat waktu yang ditentukan, memungkinkan *manager* mengawasi dan melacak kemajuan *project* secara efektif. Hal ini membuat tampilan aplikasi *project report* berbeda dengan sistem terdahulu. Sistem sebelumnya mewajibkan *manager* mengakses *electronic mail* untuk melihat pesan masuk terkait laporan aktivitas *programmer*.

4.1.3. Halaman *Project Activity Checking*



Gambar 13. Halaman *Project Activity Checking*

Halaman *project activity* checking akan menampilkan kumpulan data *project activity* yang telah diselesaikan oleh *programmer* dan menunggu evaluasi dari *manager* hal ini ditampilkan pada Gambar 13. *Manager* memegang wewenang untuk menyetujui atau menolak *project activity* jika ditemukan ketidakkonsistenan pada *project*. Pada Gambar 13 *manager* dapat dengan mudah mencari data *project activity* tertentu yang diperlukan. Selain itu *manager* dapat melihat detail penting seperti keterangan aktivitas yang telah dikerjakan oleh *programmer* dengan memilih aksi *view* pada tabel, kemudian akan tampil halaman *project activity checking detail* seperti pada Gambar 14.

4.1.4. Halaman *Project Activity Checking Detail*

Gambar 14. Halaman *Project Activity Checking Detail*

Halaman *project activity checking detail* akan menampilkan detail penting aktivitas yang telah dilakukan oleh *programmer*. Pada halaman ini *manager* dapat memilih status *approve*, *waiting*, atau *reject*. Terdapat catatan sebagai tambahan apabila terdapat masukan yang ingin disampaikan oleh *manager* kepada *programmer*. Sebelumnya interaksi yang dilakukan untuk *feedback* setiap kegiatan dilakukan melalui *electronic mail*.

4.1.5. Halaman *Report Manager*

No	Nama Project	Instansi	Periode	Status
1	Payroll Apps	PT ABCD	14 Februari 2024	On Progress
2	Aplikasi Presensi	PT EFGH	01 Maret 2024	On Progress

Gambar 15. Halaman *Project Report*

Halaman *report* menampilkan data *project* dengan pilihan status *On Progress* yang berarti *project* yang sedang dikerjakan, sedangkan status *Done* memiliki arti *project* telah selesai dikerjakan. Terakhir, *Warning* yang memiliki arti bahwa *project* telah melewati batas waktu yang ditentukan. Halaman ini memudahkan proses pemantauan *project* yang ada pada Dero Jasa Web Apps karena dilengkapi dengan periode tanggal dan status yang dapat disesuaikan dengan kebutuhan *manager*.

4.1.6. Halaman *Project Activity Programmer*



Gambar 16. Halaman *Project Activity Programmer*

Halaman *project activity* merupakan halaman yang akan tampil ketika *programmer* telah masuk ke dalam aplikasi *project report*. Halaman ini menyajikan detail penting aktivitas yang wajib diisi oleh *programmer*. Rancangan halaman *project activity* memudahkan *programmer* dalam mengisi laporan aktivitas. Jika dibandingkan pada sistem sebelumnya *programmer* wajib mengisi *Google Sheet* yang berisi aktivitas yang telah dikerjakan. Setelah itu *programmer* diharuskan mengirim file *Google Sheet* tersebut melalui *electronic mail*. Dengan adanya rancangan yang ditawarkan *programmer* hanya perlu mengisi halaman *project activity* tanpa harus mengirim email kepada *manager*.

Perbandingan tahapan proses pelaporan *project activity* antara sistem berjalan dengan aplikasi yang dirancang ditunjukkan pada Tabel 1:

Tabel 1. Perbandingan Sistem

No	Keterangan	Tahapan Sistem Berjalan	Tahapan Sistem Usulan
1	Pengisian aktivitas laporan <i>project activity</i> oleh <i>programmer</i>	<ol style="list-style-type: none"> 1. <i>Programmer</i> mengisi <i>Google Sheet</i> 2. <i>Programmer</i> mengirim <i>Google Sheet</i> melalui <i>electronic mail</i> 	<i>Programmer</i> mengisi halaman <i>project activity</i>
2	<i>Manager</i> memeriksa laporan <i>project activity</i> yang telah diisi oleh <i>programmer</i>	<ol style="list-style-type: none"> 1. <i>Manager</i> mengecek <i>Email</i> 2. <i>Manager</i> mengunduh <i>sheet</i> 3. <i>Manager</i> membuka <i>sheet</i> 4. <i>Manager</i> membalas email untuk <i>approval</i> dan <i>feedback</i> 	<ol style="list-style-type: none"> 1. <i>Manager</i> dapat mengecek pada halaman <i>project activity checking</i> 2. <i>Manager</i> dapat memberikan <i>approval</i> 3. <i>Manager</i> dapat melihat masing-masing laporan <i>project activity detail</i> dan dapat memberikan <i>feedback</i>
3	<i>Manager</i> mengumpulkan	<ol style="list-style-type: none"> 1. <i>Manager</i> membuka 	<ol style="list-style-type: none"> 1. <i>Manager</i> mengecek halaman <i>report</i>

No	Keterangan	Tahapan Sistem Berjalan	Tahapan Sistem Usulan
	laporan <i>project</i>	email 2. <i>Manager</i> mencari <i>email</i> yang berkaitan dengan <i>project</i> yang sedang dikerjakan 3. <i>Manager</i> membuka <i>email</i> dan mengunduh <i>sheet</i> 4. <i>Manager</i> mengumpulkan <i>email</i> yang berkaitan dengan <i>project</i> dan mendata kembali melalui <i>Google Sheet</i>	2. <i>Manager</i> menampilkan <i>project</i> sesuai dengan kebutuhan 3. <i>Manager</i> dapat mencetak laporan

Sistem yang dirancang memungkinkan *manager* memantau dan mendokumentasikan detail penting yang berkaitan dengan perkembangan *project* pada *software house* Dero Jasa Web Apps. Selain itu, semua aktivitas laporan *project* pada sistem yang dirancang, dilakukan dalam aplikasi yang sama yaitu aplikasi *project report*. Rancangan aplikasi *project report* ini diharapkan dapat membantu *manager* dalam mengukur kemajuan *project* dengan mudah merujuk hasil perbandingan pada Tabel 1. Penjelasan ini didukung oleh temuan penelitian sebelumnya yang dilakukan oleh Agung Sirajuddin Vidiyanto dan Wachyu Hari Haji yang mengeksplorasi pengembangan aplikasi khusus untuk meningkatkan pemantauan proyek. Dalam aplikasi yang dikembangkan, pengguna dapat mengamati kemajuan proyek, mengalokasikan tugas dan sumber daya, serta menghasilkan laporan yang mencakup aktivitas yang telah diselesaikan [6]. Penelitian lainnya dilakukan oleh Bagus Budi Pangestu dan Niki Ratama yang membahas mengenai pembuatan sistem informasi manajemen proyek untuk mengatur tugas dan proyek yang ada dalam perusahaan. Penelitian ini membuktikan dengan adanya sistem yang dibuat dapat mempermudah perusahaan dan karyawan dalam melakukan *monitoring* pekerjaan maupun pembuatan laporan [7]. Perancangan aplikasi *project report* yang menerapkan model *Personal Extreme Programming* (XP) dalam penelitian ini diharapkan dapat memberikan solusi terhadap permasalahan manajemen laporan *project* yang dihadapi Dero Jasa Web Apps.

5. Simpulan

Desain sistem yang dikembangkan dengan model *Personal Extreme Programming* (XP) memberikan solusi dalam menyederhanakan proses pembuatan laporan *project*. *Programmer* dapat dengan mudah menyelesaikan tiap laporan kegiatan *project*, sementara *manager* dapat dengan mudah melacak kemajuan anggota tim dan *project*. Sebelumnya, proses pelaporan bergantung pada *Google Sheet* dan *electronic mail*, yang terbukti sangat memakan waktu bagi para *manager*, karena mengalokasikan banyak waktu untuk meninjau setiap laporan satu per satu.

Berdasarkan penelitian yang ada, disarankan agar penelitian selanjutnya berfokus pada implementasi rancangan sistem dan antarmuka yang diusulkan, dilanjutkan dengan melakukan uji coba yang menyelaraskan dengan berbagai tahapan model XP. Hal ini memungkinkan dilakukannya pengujian sistem terhadap fungsionalitas dan efektivitas sistem.

Daftar Referensi

- [1] Wirapraja *et al.*, *Manajemen Proyek Perangkat Lunak*. Yayasan Kita Menulis, 2021.
- [2] A. Megawati and D. Gustina, "Membangun Sistem Informasi Monitoring Kegiatan Proyek Pemancar Sinyal BTS Berbasis Web Pada PT. Swatama Mega Teknik," *Jurnal Ilmiah FIFO*, vol. 10, no. 1, p. 22, Jun. 2018, doi: <https://doi.org/10.22441/fifo.v10i1.2937>.
- [3] A. Tandilintin, A. P. Candra, and G. S. Adji, "Perancangan Aplikasi Project Monitoring Pada Pt Cyber Solution Berbasis Web," *ICIT Journal*, vol. 5, no. 1, pp. 68–76, Feb. 2019, doi: <https://doi.org/10.33050/icit.v5i1.104>.
- [4] A. M. Fikri and I. P. D. Arthawan Sugih Prabowo, "Rancang Bangun Sistem Informasi Buku Tamu Pada Dinas Pemuda, Olahraga Dan Pariwisata Kota Balikpapan Dengan Metode

- Personal Extreme Programming,” *MULTITEK INDONESIA*, vol. 14, no. 2, pp. 101–110, Jan. 2021, doi: <https://doi.org/10.24269/mtkind.v14i2.2568>.
- [5] H. Noprisson, “Perancangan Aplikasi Monitoring Magang Untuk Mendukung Peningkatan Kualitas Pengalaman Bekerja Bagi Mahasiswa,” *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 5, no. 2, pp. 72–90, Dec. 2022, doi: <https://doi.org/10.31539/intecom.v5i2.4925>.
- [6] A. S. Vidiyanto and W. H. Haji, “Sistem Informasi Manajemen Proyek Berbasis Kanban (Studi Kasus: PT. XYZ),” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 2, p. 283, Feb. 2020, doi: <https://doi.org/10.25126/jtiik.2020701676>.
- [7] B. Budi Pangestu and N. Ratama, “Perancangan Sistem Informasi Project Management Dengan Metode Rapid Application Development Berbasis Web (Studi Kasus : Kelaspejwan.com),” *OKTAL : Jurnal Ilmu Komputer Dan Science*, vol. 02, no. 02, pp. 388–397, Feb. 2023.
- [8] A. Suharto and M. Mardiana, “Aplikasi Eresha Mobile Berbasis Android dengan Metode Personal Extreme Programming di Era Industri 4.0,” *Jurnal Teknologi Informasi*, vol. 4, no. 2, pp. 335–344, Dec. 2020, doi: <https://doi.org/10.36294/jurti.v4i2.1706>.
- [9] R. Satria, Royana Afwani, and Sri Endang Anjarwani, “Rancang Bangun Sistem Informasi Geografis Untuk Pemetaan Lokasi Tempat Ibadah Di Kota Mataram Menggunakan Metode Extreme Programming,” *Journal of Computer Science and Informatics Engineering (J-Cosine)*, vol. 5, no. 2, pp. 111–119, Dec. 2021, doi: <https://doi.org/10.29303/jcosine.v5i2.395>.
- [10] D. H. Ardianzah, I. Nuryasin, and B. S. Wiyono, “Pengembangan Sistem Pengelolaan Peminjaman Auditorium Universitas Muhammadiyah Malang Berbasis Web Menggunakan Metode Personal Extreme Programming,” *Jurnal Repositor*, vol. 4, no. 2, pp. 137–146, May 2022, doi: <https://doi.org/10.22219/repositor.v4i2.1342>.
- [11] Ratna Candra Sari, Fikri Hamidy, and Suaidah Suaidah, “SISTEM INFORMASI AKUNTANSI PERHITUNGAN HARGA POKOK PRODUKSI PADA KONVEKSI SJM BANDAR LAMPUNG,” *JURNAL Teknologi dan Sistem Informasi (JTSI)*, vol. 2, no. 1, pp. 65–73, Mar. 2021, doi: <https://doi.org/10.33365/jtsi.v2i1.740>.
- [12] Dimas, Puja Irawan, and Petrus Sokibi, “Rancang Bangun Sistem Pengarsipan Surat Kedinasan Berbasis Web Menggunakan Framework Codeigniter,” *MISI (Jurnal Manajemen Informatika & Sistem Informasi)*, vol. 3, no. 2, pp. 157–165, Jul. 2020, doi: <https://doi.org/10.36595/misi.v3i2.146>.
- [13] Ruly Sutrisno Sinukun, Roys Pakaya, and S. Suleman, “Perancangan Sistem Informasi Perjalanan Dinas (SIMPERNAS) Menggunakan Metode UML,” *Energy - Jurnal Ilmiah Ilmu-Ilmu Teknik*, vol. 12, no. 1, pp. 18–24, May 2022, doi: <https://doi.org/10.51747/energy.v12i1.1040>.
- [14] F. F. Adiwijaya, D. S. Amaruloh, and A. R. Mulya, “Sistem Registrasi Surat Perintah Tugas (Spt) Di Dinas Pekerjaan Umum, Penataan Ruang Dan Pertanahan Provinsi Kepulauan Riau,” *Komputa: Jurnal Ilmiah Komputer dan Informatika*, vol. 10, no. 2, pp. 70–77, Oct. 2021, doi: <https://doi.org/10.34010/komputa.v10i2.6806>.
- [15] A. Fatoni, R. Effendi, Fikri Hadiyansyah, and Masum Masum, “Rancang Bangun Sistem Absensi Pegawai Menggunakan Qr Code Pada Kantor Desa Sidamukti,” *Jurnal Ilmiah Sains dan Teknologi*, vol. 6, no. 2, pp. 146–158, Oct. 2022, doi: <https://doi.org/10.47080/saintek.v6i2.2273>.