

Perancangan dan Implementasi Sistem Manajemen Peminjaman Mobil Dengan Metode Scrum di Universitas Internasional Batam

Fredian Simanjuntak, S. Kom., MMSI¹, Ghozi Bahri²

Information System Study Program, Universitas Internasional Batam Jl. Gajah

Mada, Baloi Sei Ladi Batam 29442

Universitas Internasional Batam

Email: fredian.pc@gmail.com, ghozi@uib.ac.id

Abstrak

Pada zaman yang modern ini sudah banyak perusahaan maupun industri serta lokasi-lokasi yang sudah mulai meninggalkan proses yang dilakukan secara manual. Pada zamannya cara manual sangat lah membantu karena manusia sudah terbiasa melakukan pekerjaannya seperti biasa, dengan perkembangan teknologi dibutuhkan sebuah sistem yang bisa membantu manusia dalam melakukan pekerjaannya. Dari mulai memasukkan data, menyimpan, mengorganisir serta berkomunikasi sesama pengguna. Maka dari itu dibuatlah sistem yang bisa membantu pekerjaan manusia dalam melakukan pekerjaannya sehari-hari. Sistem yang dibuat ini membantu dalam mengatur mobil dan mengatur peminjaman mobil yang biasa dilakukan dengan cara manual. Dengan menggunakan metode *scrum*, penulis merancang dan mengembangkan aplikasi secara rapih dan teratur. Hasil dari proyek ini yaitu dapat mempermudah dan menambah keamanan data yang dikelola karena data akan disimpan kedalam database dimana data tidak akan mengalami kerusakan secara fisik dan kecil kemungkinan terjadi kehilangan data. Dengan perancangan sistem ini dapat membantu proses manajemen mobil dan peminjaman mobil.

Kata kunci: *Scrum, aplikasi web, agile, manajemen mobil, peminjaman mobil, sistem informasi*

Abstract

In this modern era many companies and industries as well as locations have started to leave the manual process. In its era manual process was very helpful because humans were accustomed to doing their usual jobs, but in modern times like today with the development of technology needed a system that could help humans in doing their work. Start from entering data, store, organize and communicate with other users. Therefore, a system is created that can assist human work in doing their daily work. This system was made to help in managing car and managing car booking which are usually done manually. By using the scrum method, the authors design and develop applications neatly and regularly. The result of this project is that it can simplify process and increase the security of data because the data will be stored in a database where the data will not physically damaged and there is little possibility of data loss. With this system, we hope it can help the process in managing car and managing booking car.

Keyword: *Scrum, Web Application, Agile, Car Management, Car Booking Management, Information System*

Copyright © Journal of Information System and Technology. All rights reserved

I. PENDAHULUAN

Seiring berjalannya jam, bulan, tahun, hingga zaman, perkembangan teknologi informasi sudah banyak mengalami perubahan. Sistem yang dikembangkan memiliki berbagai fungsi

untuk meningkatkan kinerja sebuah perusahaan dan juga individunya. Salah satu pemanfaatan teknologi yaitu dapat membantu perusahaan untuk mengatur manajemen mobil serta membantu mengatur bagaimana proses peminjaman

mobil yang terjadi sehingga memudahkan perusahaan dan juga individu dalam melakukan tugasnya.

Dalam pengembangan perangkat lunak yang menggunakan pendekatan *agile*, pekerjaan dilakukan dalam tahap kecil, berdasarkan kolaborasi, perencanaan adaptif, pengiriman lebih awal, perbaikan secara bertahap, masukan pelanggan regular, seringnya pengulangan desain yang menghasilkan iterasi pengembangan perangkat lunak diberikan dalam iterasi yang sukses, dalam merespon perubahan kebutuhan pelanggan yang tiada henti (Matharu et al., 2015).

Dalam sebuah proyek, *Requirement Engineering* memiliki peran yang penting dalam lingkaran kehidupan pengembangan perangkat lunak. Karena *Requirement Engineering* adalah titik mulai lingkaran kehidupan dalam pengembangan perangkat lunak. Apabila terdapat perubahan pada kebutuhan yang di telah di tetapkan, akan menyebabkan biaya dan waktu menjadi berlebih. *Scrum* sebagai salah satu *agile development* mendapat banyak perhatian karena kemampuannya dalam menghadapi lingkungan yang berubah-ubah (Darwish & Megahed, 2016).

Sehingga seorang proyek manager harus bisa mengelola waktu dan juga sumber daya untuk bisa menyelesaikan dan bisa mengembangkan sebuah sistem perangkat lunak yang dimana sistem tersebut dapat membantu kelancaran sebuah perusahaan atau individu tersebut. Dengan menggunakan pendekatan *agile* proyek manager akan lebih mudah mengatur serta memberikan tugas kepada para developer agar bisa melakukan tugas dengan lebih teratur dan lebih cepat dan juga lebih bisa untuk memperjelas kapan proyek tersebut harus selesai. Ada beberapa metodologi kebutuhan prioritas yang dapat membantu dalam menentukan pilihan akan tetapi yang paling penting banyak yang melupakan faktor penting yang memiliki pengaruh besar dalam memprioritaskan kebutuhan. (Asghar et al., 2016)

Dengan latar belakang tersebut penulis menerapkan metode *scrum* pada sistem manajemen mobil dalam menentukan pekerjaan yang lebih dahulu dikerjakan. Judul dari penelitian ini adalah **“Perancangan dan Implementasi Sistem Manajemen Peminjaman Mobil Dengan Menggunakan Metode Scrum di Universitas Internasional Batam”**.

Adapun rumusan masalah dalam penelitian ini sebagai berikut:

1. Bagaimana Sistem Manajemen Mobil dan Manajemen Peminjaman Mobil dapat mempermudah proses yang saat ini telah dilakukan?
2. Sistem Manajemen Mobil dan Manajemen Peminjaman Mobil dengan Metode Scrum bisa diterapkan?

Dengan sumber hasil penelitian dari beberapa peneliti sebelumnya yang menggunakan sistem untuk mempermudah sebuah pekerjaan dengan tujuan agar meningkatkan perkembangan teknologi yang terjadi hingga saat ini.

II. LANDASAN TEORI

2.1 TINJAUAN PUSTAKA

Telah banyak penelitian mengenai *scrum* itu sendiri, banyak yang telah meneliti seberapa bagusnya dan bergunanya metode *scrum* ini. Penelitian yang dilakukan mengenai *scrum* salah satunya yaitu yang dilakukan oleh (Permana, 2015) dengan judul *Scrum Method Implementation in a Software Development Project Management* yang menjelaskan bahwa dengan metode *scrum* ini bisa sangat membantu dalam pembuatan perangkat lunak.

Penelitian dengan judul *Development of Reusable Hybrid Test Automation Framework for Web Based Scrum Projects* yang dilakukan oleh (Tyagi et al., 2018) yang dimana menjelaskan bahwa *web base application* sudah mulai mendapatkan popularitas yang luas dan hamper semua aplikasi di kembangkan menggunakan metode pengembangan *agile* terutama *scrum*.

Salah satu dari penelitian *scrum* yang telah ada yaitu yang dilakukan oleh (Matharu et al., 2015) tentang Empirical Study of Agile Software Development Methodologies: A Comparative Analysis bahwa metodologi *agile* yang paling populer antara lain adalah *scrum*, *extreme programming*, dan *kanban*. Dari survei yang telah mereka lakukan di dapatkan kesimpulan bahwa pada saat ini pengembangan Berbasis *scrum* lebih tinggi penggunaannya dibanding *extreme programming* dan *kanban methodologies*.

Sistem Informasi Mahasiswa Asing adalah judul dari penelitian tentang bagaimana cara untuk mempermudah mahasiswa asing yang sedang ingin mengajukan perihal izin tinggal yang di mohon kepada Pejabat Imigrasi. Salah satu hal yang dilakukan dari (Robiansyah & Angreani, 2017), adalah dengan memakai *scrum* yang merupakan salah satu *agile methodology*.

2.2 LANDASAN TEORI

2.2.1 Web Application

Menurut (Rizkita et al., 2018) *web application* atau aplikasi web wadah sebuah pekerjaan dengan menggunakan internet.

2.2.2 Scrum

Menurut (Firdaus, 2017), *scrum* memiliki perbedaan Segitiga manajemen proyek yang biasanya diisi oleh kualitas, biaya dan waktu. Pada bagian kualitas diganti oleh fungsionalitas. Hal ini disebabkan bahwa kualitas bukan menjadi salah satu *variable* yang sangat menentukan pada sebuah proyek yang menggunakan *scrum*.

2.2.3 PHP

Menurut (Das & Saikia, 2016) *PHP : Hypertext Preprocessor* merupakan Bahasa *scripting* yang sangat diketahui sering dipakai dalam pengembangan web walaupun sebenarnya tidak hanya itu kegunaannya. Berdasarkan w3techs.com *PHP* merupakan bahasa *scripting* yang sering dipakai di internet sebesar 82%.

2.2.4 Codeigniter

Menurut (Arrhioui et al., 2017), *Codeigniter* adalah kerangka kerja pengembangan aplikasi *PHP* berdasarkan arsitektur yang terstruktur. *Codeigniter* memiliki tujuan untuk memberikan alat bantu yang dibutuhkan seperti *helpers and libraries* untuk mengimplementasi tugas yang biasa dilakukan. Dengan demikian, pengembangan proyek menjadi lebih mudah dan cepat. Dan pengembang tidak perlu menulis lagi dari awal.

2.2.5 MVC (Model, View, Controller)

Menurut (Endra & Aprilita, 2018), *MVC* merupakan suatu metode yang digunakan dalam pengembangan system yang menggunakan prinsip memisahkan setiap kerangka kerjanya, yaitu *logic*, *presentation*, dan *process* atau biasa yang kita tahu *model*, *view*, *controller*.

2.2.6 Definisi Website

Menurut (Endra & Aprilita, 2018), *website* terhubung dengan suatu jaringan internet yang akan membawa pengguna kesuatu tujuan yang diinginkan oleh pengguna dengan cara mengklik link yang berupa teks, gambar.

2.2.7 HTML

Menurut (Endra & Aprilita, 2018), *HTML* atau *Hypertext Markup Language* merupakan salah satu bahasa yang biasa digunakan oleh pengguna dalam membuat tampilan yang digunakan oleh *web application*.

2.2.8 jQuery

Menurut (Wahyudi, 2017), *jquery* merupakan sekumpulan kode *Javascript* yang dibuat dalam berbagai modul dan digunakan sesederhana mungkin.

2.2.9 CSS

Menurut (Wahyudi, 2017), *CSS* adalah suatu Bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web akan lebih rapi, terstruktur, dan seragam.

2.2.10 Bootstrap

Menurut (Nugroho & Setiyawati, 2019), *bootstrap* adalah *framework css* untuk membuat tampilan *web*. *Bootstrap*

menyediakan *class* dan komponen yang sudah siap dipakai.

2.2.11 Database

Menurut (Hesananda et al., 2017), *Database* ialah suatu wadah untuk menampung sebuah data yang ada pada sebuah sistem. Database juga bias diartikan sebagai kumpulan data. Database juga biasa dikenal formal dan tegas. Database juga bias diartikan dengan kumpulan data yang terintegrasi yang dapat dimanipulasi, diambil dan dicari secara cepat.

2.2.12 DBMS (Database Management System)

Menurut (Hesananda et al., 2017), *DBMS* adalah kumpulan program yang digunakan untuk mendefinisikan, mengatur dan memproses *database*. *DBMS* juga sering disebut sebagai *server database*.

2.2.13 MySQL

MySQL bekerja menggunakan *SQL Language (Structure Query Language)*, yang dapat diartikan bahwa *MySQL* merupakan standar penggunaan *database* di dunia untuk pengolahan data. Kelebihan yang dimiliki *MySQL* yaitu bersifat *open source*, yang memiliki kemampuan untuk dikembangkan lagi. (Wahyudi, 2017).

2.2.14 Entity Relationship Diagram (ERD)

Biasanya *Entity Relationship Diagram* dikenal dengan diagram *ER*. Perancangan *web* memerlukan. *ERD* sangat berbeda dengan *DFD*, karena *DFD* merupakan model jaringan fungsi yang dilaksanakan oleh sistem, sedangkan *ERD* merupakan model jaringan data yang menekankan pada struktur dan *relationship* data. (Wahyudi, 2017).

2.2.15 (UML) Inified Modelling Language

Dari penelitian (Rizkita et al., 2018), bahwa *UML* atau biasa diblilang *Unified Modeling Language* merupakan teknik yang dapat mengembangkan sistem dengan menggunakan salah satu bahasa yaitu bahasa grafis sebagai alat pendokumentasian dan juga dalam melakukan spesifikasi sistem. *UML*

memiliki banyak diagram, dan diagram itu digunakan untuk melakukan pemodelan data maupun sistem.

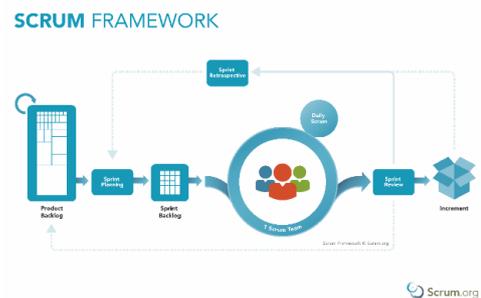
2.2.16 Flowchart

Flowchart merupakan representasi secara grafik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah. (Lasminiasih et al., 2016).

III. METODOLOGI PENELITIAN

3.1 Alur Penelitian

Salah satu peran pengembangan yang berguna sebagai acuan ataupun proses yang harus dilakukan untuk mencapai sebuah tujuan, yang dimana tujuan tersebut akan menghasilkan hasil yang maksimal juga. Berikut adalah ringkasan alur yang akan dibuat dalam proses *scrum* (Gambar 3. 1).



Gambar 3. 1 Alur Penelitian

Hal yang harus dilakukan pertama kali dengan metode *scrum* adalah menentukan *product backlog*. *Product backlog* ialah proses pengumpulan data-data maupun fitur yang dibutuhkan oleh sebuah sistem. Disini kita menentukan fitur apa saja yang akan menjadi *product backlog*. Setelah tau apa saja yang dibutuhkan dan telah di kelompokkan di dalam *product backlog*, barulah kita akan menentukan planning untuk merencanakan apa saja yang harus dilakukan dalam kurun waktu yang telah di tentukan, proses ini dinamakan *sprint planning*.

Setelah *sprint planning* selesai barulah kita memulai membagikan *sprint backlog*. *Sprint backlog* ini ialah daftar dari seluruh *product backlog* yang telah di rencanakan untuk selesai pada waktu yang telah ditentukan. Ini dapat

memudahkan dalam proses pengerjaan produk karna sudah di rencanakan dengan matang. Setelah semua proses diatas selesai saat itu juga kita akan menentukan tim yang akan mengerjakan *backlog* tersebut. *Scrum team* akan di *brief* setiap hari untuk mengetahui sejauh apa proses dari *backlog* yang telah di assign terhadap *scrum team* tersebut.

Setelah selesai akan ada yang namanya *sprint review*. *Sprint review* ialah proses mereka ulang apakah semua *product backlog* telah selesai semua dikerjakan atau masih ada pekerjaan yang tertinggal. Proses ini sangat penting karena disinilah kita bisa tau seberapa bagusnya pekerjaan yang telah dilakukan. Setelah semua pekerjaan telah selesai, lalu kembali merencanakan pada *sprint planning* diawal, proses ini dinamakan *sprint retrospective*. Apabila semua proses sudah success maka baru bisa kita bungkus dan kita *deliver kepada client* hasil pekerjaan yang telah dikerjakan. Semua *product backlog* serta *sprint planning* telah berjalan sukses lalu masuk ke proses *increment*. Maka alur proses sudah selesai. Dilakukan dalam kurun waktu yang telah di tentukan, proses ini dinamakan *sprint planning*. Setelah *sprint planning* selesai barulah kita memulai membagikan *sprint backlog*. *Sprint backlog* ini ialah daftar dari seluruh *product backlog* yang telah di rencanakan untuk selesai pada waktu yang telah ditentukan. Ini dapat memudahkan dalam proses pengerjaan produk karna sudah di rencanakan dengan matang.

Setelah semua proses diatas selesai saat itu juga kita akan menentukan tim yang akan mengerjakan *backlog* tersebut. *Scrum team* akan di *brief* setiap hari untuk mengetahui sejauh apa proses dari *backlog* yang telah di assign terhadap *scrum team* tersebut. Setelah selesai akan ada yang namanya *sprint review*. *Sprint review* ialah proses mereka ulang apakah semua *product backlog* telah selesai semua dikerjakan atau masih ada pekerjaan yang tertinggal. Proses ini sangat penting karena disinilah kita bisa tau seberapa bagusnya pekerjaan yang

telah dilakukan. Setelah semua pekerjaan telah selesai, lalu kembali merencanakan pada *sprint planning* diawal, proses ini dinamakan *sprint retrospective*. Apabila semua proses sudah success maka baru bisa kita bungkus dan kita *deliver kepada client* hasil pekerjaan yang telah dikerjakan. Semua *product backlog* serta *sprint planning* telah berjalan sukses lalu masuk ke proses *increment*. Maka alur proses sudah selesai.

3.2 Analisis Permasalahan

Salah satu inti dari penelitian ini yaitu agar bisa memudahkan staff dalam proses manajemen mobil dan proses peminjaman mobil. Mengurangi penggunaan kertas sehingga bisa mengurangi keuangan yang dikeluarkan untuk kebutuhan saat menggunakan proses manual.

3.2.1 Analisis Sistem yang Sedang Berjalan

Pada saat ini belum ada sistem yang berjalan untuk membantu dalam proses pekerjaan. Sejauh ini proses peminjaman biasanya dilakukan secara manual dengan mengajukan permintaan ke bank untuk peminjaman mobil, lalu dicatat sehingga ada kemungkinan apabila kertas tersebut hilang akan menyebabkan kehilangan data yang telah di masukkan pada saat itu.

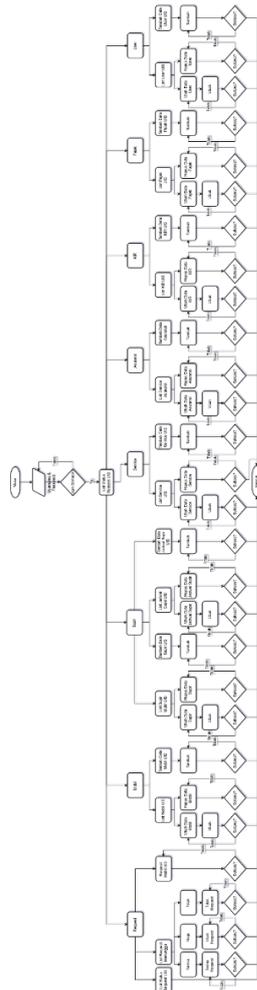
3.2.2 Analisis Strategi Pemecahan Masalah

Penulis akan membuat sistem untuk memudahkan user dan juga admin untuk bisa saling berkomunikasi tanpa harus datang ke tempat ketika ingin melakukan pengajuan. Dengan kata lain user bisa merequest mobil kapan pun dan dimanapun dengan menggunakan sistem. Dan juga admin bisa kapanpun mengolah data mobil seperti pajak, asuransi, dll. Dan data yang tersimpan akan lebih teratur dan memudahkan dalam pencarian data kedepannya.

3.3 Perancangan Sistem

3.3.1 Flowchart

Gambar 3. 2 berikut adalah alur dari *flowchart* yang dirancang berdasarkan aplikasi yang di kembangkan oleh penulis.

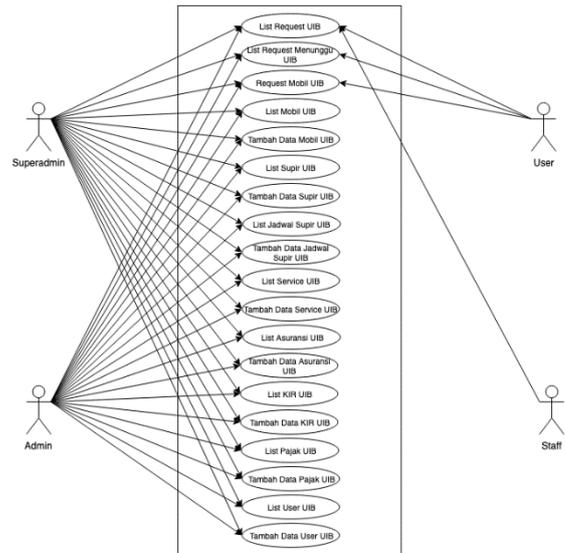


Gambar 3. 2 Flowchart Sistem

Pada saat pertama kali pengguna menggunakan aplikasi ini, pengguna harus melakukan proses *login* terlebih dahulu, pengguna akan memasukkan username dan juga password. Setelah itu user akan diverifikasi apakah data yang dimasukkan benar apa salah, apabila data yang dimasukkan benar maka pengguna akan masuk ke halaman *list request UIB* dimana halaman ini berisi data pengguna yang telah mengajukan permohonan ke UIB.

Dari halaman ini apabila pengguna *login* sebagai *superadmin* pengguna dapat berpindah ke halaman yang telah diberikan akses. Halaman yang dapat diakses oleh pengguna *superadmin* meliputi semua halaman yaitu halaman *Request, Mobil, Supir, Service, Asuransi, KIR, Pajak* dan *User*.

3.3.2 Use Case Diagram



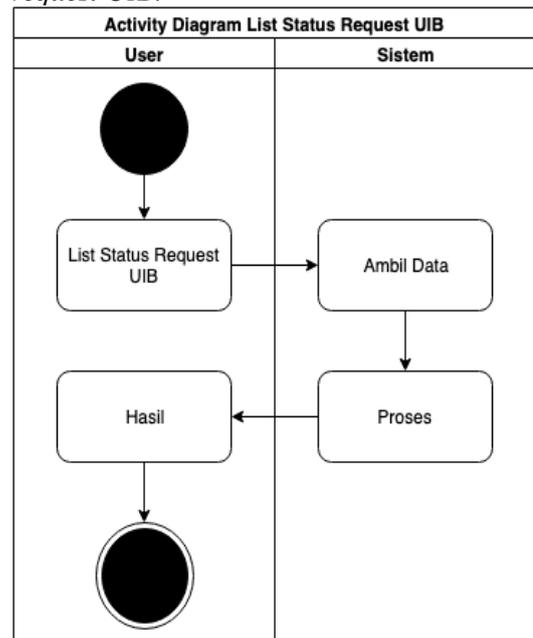
Gambar 3. 3 Use Case Diagram Sistem

Pada *use case diagram* diatas bisa dilihat bahwa pada aplikasi ini memiliki 4 aktor yaitu *superadmin, admin, user* dan *staff*. Dan setiap perannya memiliki akses yang berbeda beda seperti yang terlihat pada gambar diatas. Peran *superadmin* merupakan peran tertinggi yang ada dalam aplikasi tersebut.

3.3.3 Activity Diagram

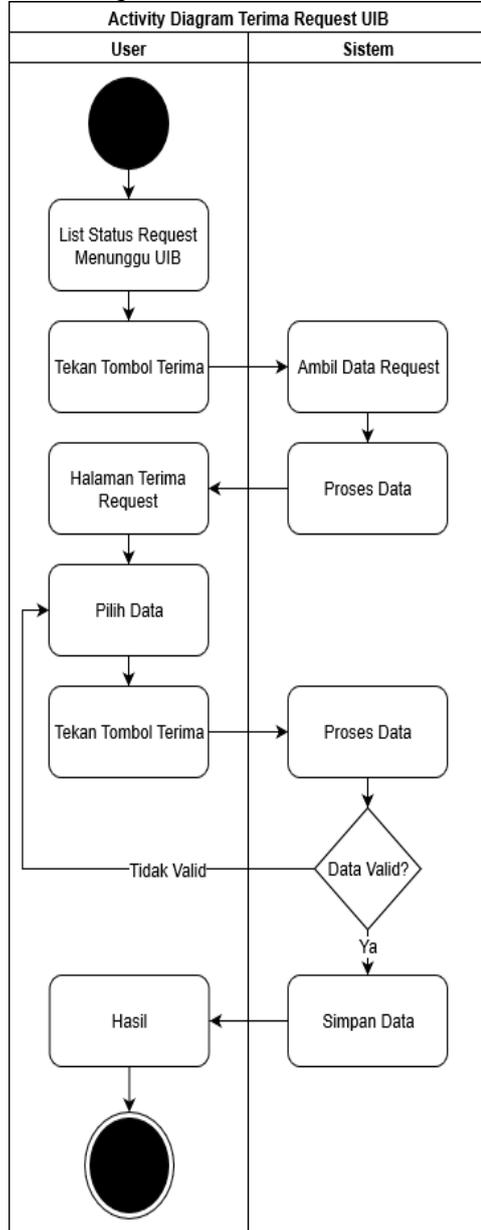
Activity diagram berfungsi untuk menggambarkan alur-alur aktivitas umum sebuah sistem.

Activity diagram berikut menjelaskan proses halaman *list status request UIB*.



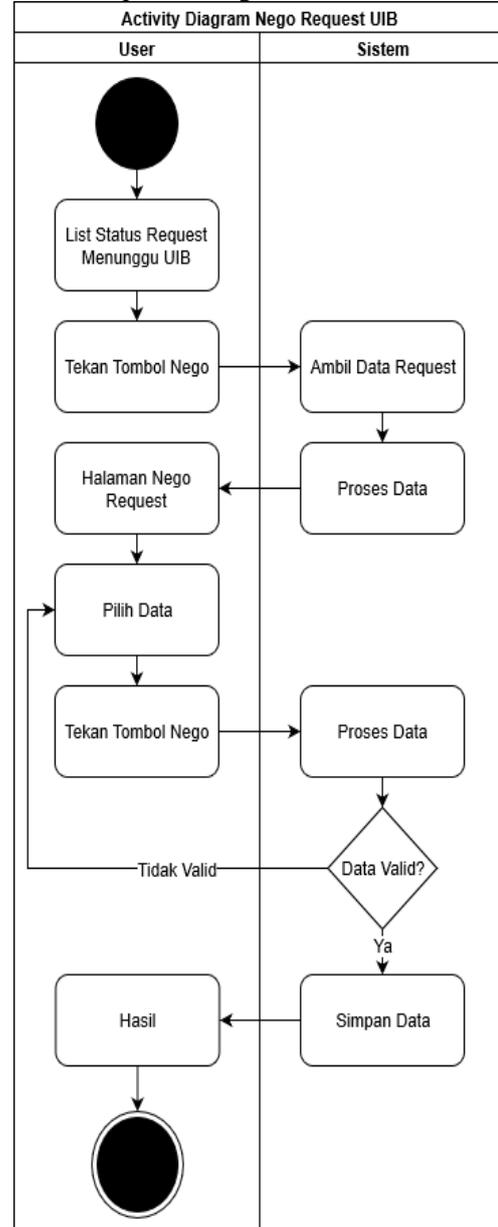
Gambar 3. 2 Activity Diagram List Status Request UIB

Berikut adalah *activity diagram* yang menjelaskan bagaimana proses ketika *request* diterima.



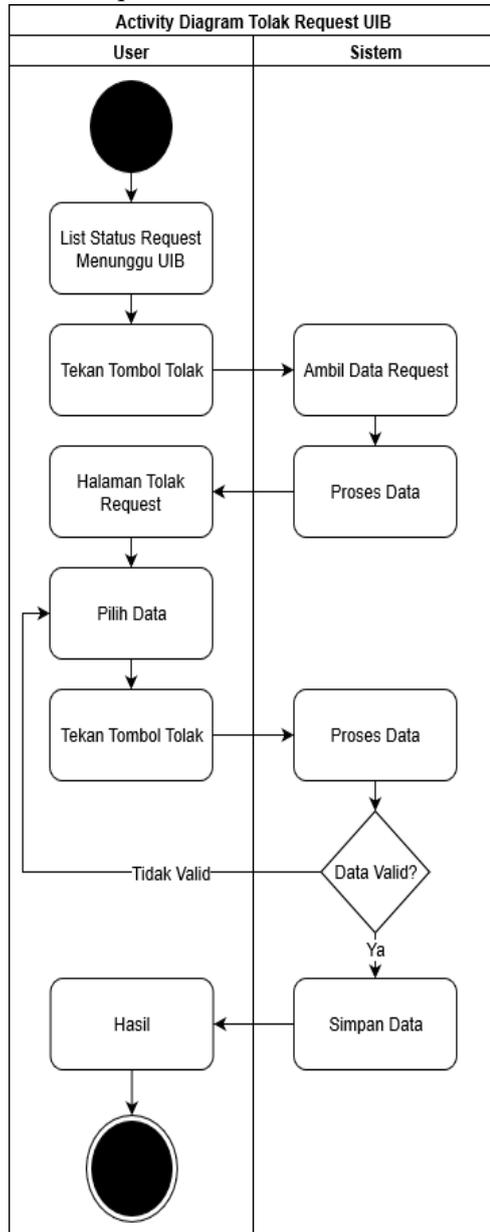
Gambar 3. 3 Activity Diagram Terima List Request Menunggu UIB

Berikut adalah *activity diagram* yang menjelaskan bagaimana proses ketika *request* dinego.



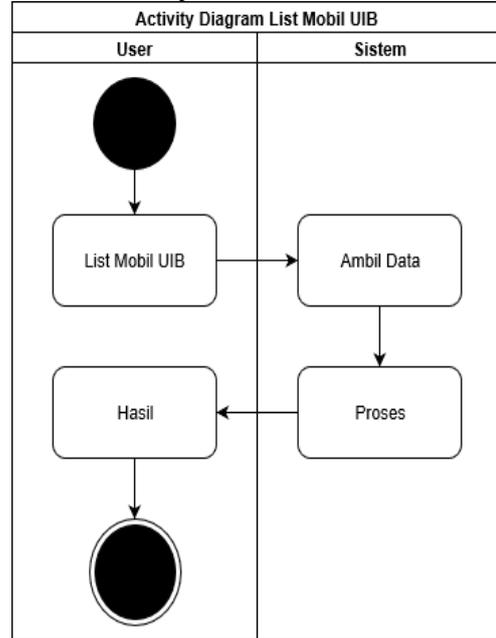
Gambar 3. 4 Activity Diagram Nego List Request Menunggu UIB

Berikut adalah *activity diagram* yang menjelaskan bagaimana proses ketika *request* ditolak.



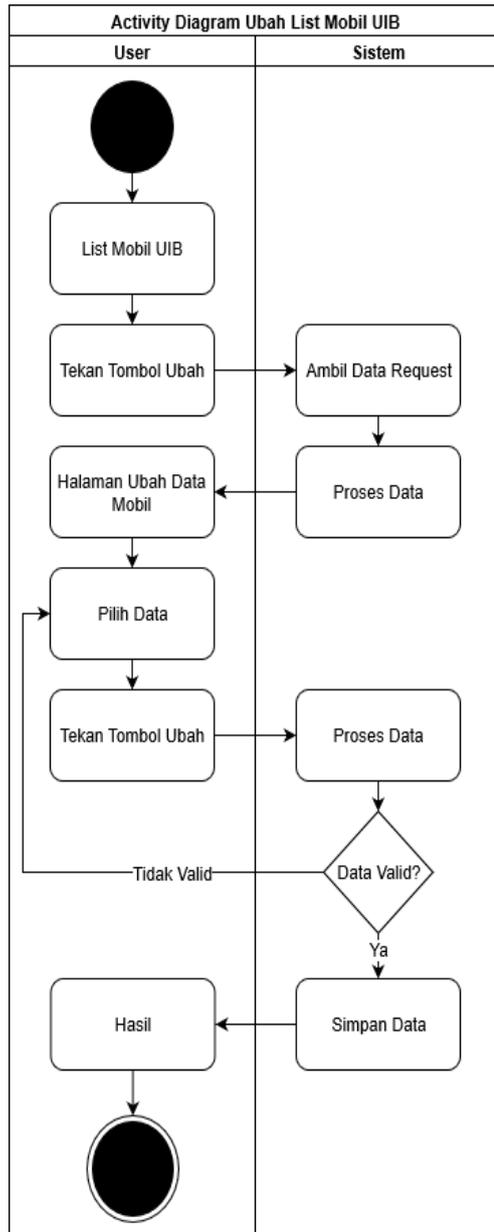
Gambar 3. 5 Activity Diagram Tolak List Request Menunggu UIB

Berikut adalah *activity diagram* yang menjelaskan bagaimana proses ketika menampilkan *list mobil UIB*.



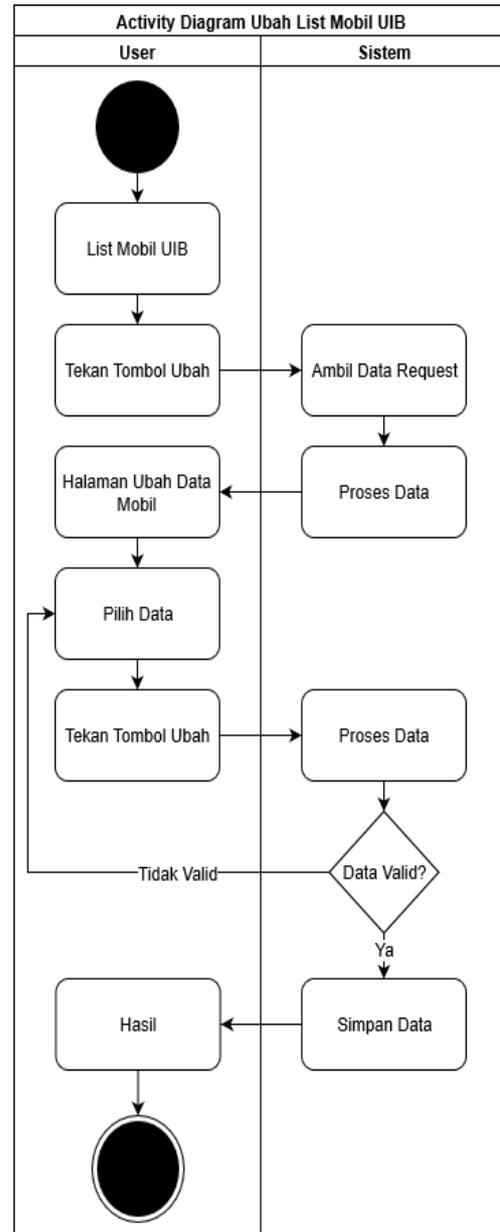
Gambar 3. 8 Activity Diagram List Mobil UIB

Berikut adalah *activity diagram* yang menjelaskan bagaimana proses ketika mengubah *list mobil UIB*.



Gambar 3. 6 Activity Diagram Ubah List Mobil UIB

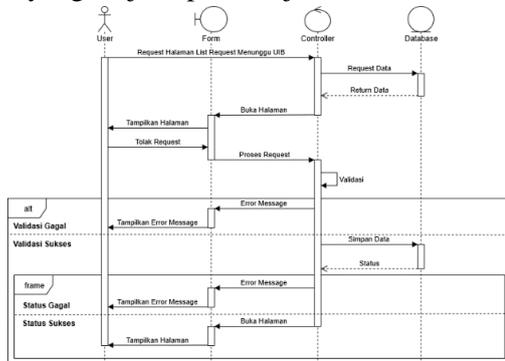
Berikut adalah *activity diagram* yang menjelaskan bagaimana proses ketika menghapus *list mobil UIB*.



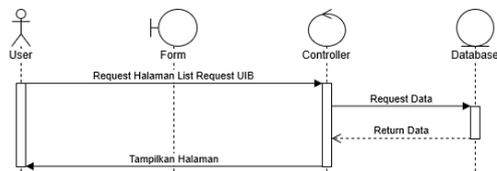
Gambar 3. 7 Activity Diagram Hapus List Mobil UIB

3.3.4 Sequence Diagram

Fungsi dari *sequence diagram* ialah untuk memperlihatkan interaksi yang terjadi pada objek secara berurut.

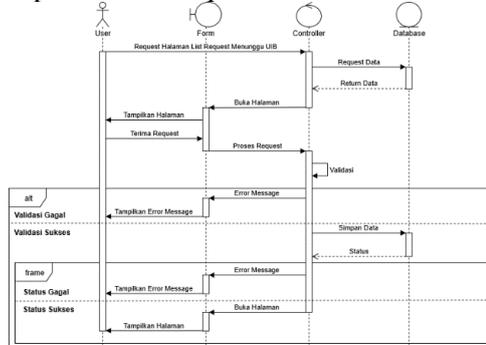


Berikut adalah proses pengambilan data untuk halaman *list request UIB*.



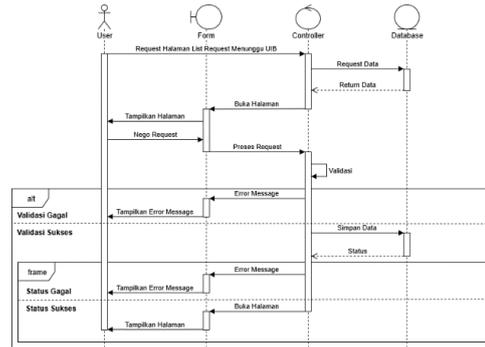
Gambar 3. 8 Sequence Diagram List Request UIB

Sequence diagram berikut adalah proses manipulasi data yang dilakukan oleh *user* untuk proses manipulasi data seperti *terima request*.



Gambar 3. 9 Sequence Diagram Terima List Request Menunggu UIB

Sequence diagram berikut adalah proses manipulasi data yang dilakukan oleh *user* untuk proses manipulasi data seperti *nego request*.



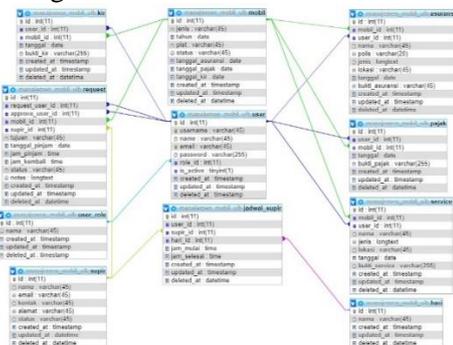
Gambar 3. 10 Sequence Diagram Nego List Request Menunggu UIB

Sequence diagram berikut adalah proses manipulasi data yang dilakukan oleh *user* untuk proses manipulasi data seperti *tolak request*.

Gambar 3. 11 Sequence Diagram Tolak List Request Menunggu UIB

3.3.5 Entity Relationship Diagram

Bagian ini merupakan salah satu bagian yang penting dimana serangkaian data yang telah di buat oleh penulis dalam proses pengembangan aplikasi yang sedang dibuat. Berikut adalah relasi yang terjadi pada aplikasi yang sedang dibangun.



Gambar 3. 12 Entity Relationship Diagram Sistem

3.3.6 Rancangan Antarmuka

Halaman *login* berikut adalah tampilan dimana halaman yang pertama kali pengguna akses ketika membuka halaman web aplikasi tersebut. Pada halaman ini pengguna harus memasukkan username dan password agar bisa masuk ke halaman utama sekaligus sesuai peran apa yang mereka telah di buat.

Sistem akan menampilkan halaman *request* setelah pengguna berhasil masuk kedalam aplikasi. Pada halaman ini user dapat melihat list

request yang telah dia lakukan selama dia melakukan *request* pada aplikasi tersebut.

Pengguna juga dapat mengakses halaman mobil. Pada halaman ini user dapat melihat list mobil yang tersedia ataupun yang tidak tersedia pada saat itu pada aplikasi tersebut. Dan juga pengguna juga bisa merubah serta menghapus data.

Pengguna juga dapat mengakses halaman supir. Pada halaman ini user dapat melihat list supir yang tersedia ataupun yang tidak tersedia pada saat itu pada aplikasi tersebut. Dan juga pengguna juga bisa merubah serta menghapus data.

Pengguna juga dapat mengakses halaman service. Pada halaman ini user dapat melihat list service yang tersedia ataupun yang tidak tersedia pada saat itu pada aplikasi tersebut. Dan juga pengguna juga bisa merubah serta menghapus data.

Pengguna juga dapat mengakses halaman asuransi. Pada halaman ini user dapat melihat list asuransi yang tersedia ataupun yang tidak tersedia pada saat itu pada aplikasi tersebut. Dan juga pengguna juga bisa merubah serta menghapus data.

Pengguna juga dapat mengakses halaman KIR. Pada halaman ini user dapat melihat list KIR yang tersedia ataupun yang tidak tersedia pada saat itu pada aplikasi tersebut. Dan juga pengguna juga bisa merubah serta menghapus data.

Pengguna juga dapat mengakses halaman pajak. Pada halaman ini user dapat melihat list pajak yang tersedia ataupun yang tidak tersedia pada saat itu pada aplikasi tersebut. Dan juga pengguna juga bisa merubah serta menghapus data.

Pengguna juga dapat mengakses halaman user. Pada halaman ini user dapat melihat list user yang tersedia ataupun yang tidak tersedia pada saat itu pada aplikasi tersebut. Dan juga pengguna juga bisa merubah serta menghapus data.

IV. IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi

Implementasi merupakan salah satu tahapan dalam perancangan dan pengembangan aplikasi. Dimulai dari pembuatan database, desain, hingga tampilan antar muka yang memudahkan *user* untuk bisa menggunakan aplikasi tersebut. Serta dengan melakukan pengujian terhadap aplikasi tersebut.

4.2 Implementasi Antarmuka

Dibawah ini adalah tampilan tatap muka yang telah dibangun :

4.2.1 Halaman Login

Pertama kali *user* menggunakan aplikasi, *user* harus melakukan *login* terlebih dahulu.

4.2.2 Halaman List Request

Setelah *login*, *user* akan masuk ke halaman *request* terlebih dahulu.

4.2.3 Halaman View Project

Setelah itu *user* bisa menuju halaman *list mobil* apabila masuk sebagai seorang admin dan juga superadmin.

4.2.4 Implementasi List Supir

Setelah itu *user* bisa menuju halaman *list supir* apabila masuk sebagai seorang admin dan juga superadmin.

4.2.5 Halaman List Service

Setelah itu *user* bisa menuju halaman *list service* apabila masuk sebagai seorang admin dan juga superadmin.

4.2.6 Halaman List Asuransi

Setelah itu *user* bisa menuju halaman *list asuransi* apabila masuk sebagai seorang admin dan juga superadmin.

4.2.7 Halaman List KIR

Setelah itu *user* bisa menuju halaman *list KIR* apabila masuk sebagai seorang admin dan juga superadmin.

4.2.8 Halaman List Pajak

Setelah itu *user* bisa menuju halaman *list pajak* apabila masuk sebagai seorang admin dan juga superadmin.

4.2.9 Halaman List User

Setelah itu *user* bisa menuju halaman *list user* apabila masuk sebagai seorang admin dan juga superadmin.

4.3 Pengujian Aplikasi

Pada setiap sprint, penulis akan melakukan pengujian pada sistem dan menyimpulkan pada bentuk table sebagai berikut.

4.3.1 Pengujian Fungsi Login

Tabel 4. 1 Pengujian Fungsi

Login

| No | Pengujian yang dilakukan | Hasil yang Diharapkan | Hasil Pengujian |
|----|---|---|-----------------|
| 1 | <i>Login Bypass</i> | Melindungi sistem dari <i>login bypass</i> | Ok |
| 2 | Melakukan <i>Login</i> dengan data <i>user</i> yang tidak terdaftar | <i>Error Message</i> yang menunjukkan <i>login</i> gagal akan ditampilkan | Ok |
| 3 | Melakukan <i>Login</i> dengan data <i>user</i> yang terdaftar | Akan muncul <i>home page</i> sesuai masing-masing <i>role user</i> | Ok |

4.3.2 Pengujian Fungsi List Request

Tabel 4. 2 Pengujian Fungsi List

Request

| No | Pengujian yang dilakukan | Hasil yang Diharapkan | Hasil Pengujian |
|----|---|-----------------------|-----------------|
| 1 | Melakukan panggilan halaman dengan user yang tidak memiliki hak | Halaman tidak terbuka | Ok |
| 2 | Melakukan | Halaman terbuka | Ok |

| | | | |
|--|---|---|--|
| | panggilan halaman dengan user yang memiliki hak | dengan data <i>list request</i> yang tercatat | |
|--|---|---|--|

4.3.3 Pengujian Fungsi List Mobil

Tabel 4. 3 Pengujian Fungsi List

Mobil

| No | Pengujian yang dilakukan | Hasil yang Diharapkan | Hasil Pengujian |
|----|---|---|-----------------|
| 1 | Melakukan panggilan halaman dengan user yang tidak memiliki hak | Halaman tidak terbuka | Ok |
| 2 | Melakukan panggilan halaman dengan user yang memiliki hak | Halaman terbuka dengan data <i>list</i> mobil yang tercatat | Ok |

4.3.4 Pengujian Fungsi List Supir

Tabel 4. 4 Pengujian Fungsi List

Supir

| No | Pengujian yang dilakukan | Hasil yang Diharapkan | Hasil Pengujian |
|----|---|---|-----------------|
| 1 | Melakukan panggilan halaman dengan user yang tidak memiliki hak | Halaman tidak terbuka | Ok |
| 2 | Melakukan panggilan halaman dengan user yang memiliki hak | Halaman terbuka dengan data <i>list</i> supir yang tercatat | Ok |

4.3.5 Pengujian Fungsi List Service

Tabel 4. 5 Pengujian Fungsi List

Service

| No | Pengujian yang dilakukan | Hasil yang Diharapkan | Hasil Pengujian |
|----|---|---|-----------------|
| 1 | Melakukan pemanggilan halaman dengan user yang tidak memiliki hak | Halaman tidak terbuka | Ok |
| 2 | Melakukan pemanggilan halaman dengan user yang memiliki hak | Halaman terbuka dengan data <i>list service</i> yang tercatat | Ok |

4.3.6 Pengujian Fungsi List

Asuransi

Tabel 4. 6 Pengujian Fungsi List

Asuransi

| No | Pengujian yang dilakukan | Hasil yang Diharapkan | Hasil Pengujian |
|----|---|--|-----------------|
| 1 | Melakukan pemanggilan halaman dengan user yang tidak memiliki hak | Halaman tidak terbuka | Ok |
| 2 | Melakukan pemanggilan halaman dengan user yang memiliki hak | Halaman terbuka dengan data <i>list asuransi</i> yang tercatat | Ok |

4.3.7 Pengujian Fungsi List KIR

Tabel 4. 7 Pengujian Fungsi List

KIR

| No | Pengujian yang dilakukan | Hasil yang Diharapkan | Hasil Pengujian |
|----|---|---|-----------------|
| 1 | Melakukan pemanggilan halaman dengan user yang tidak memiliki hak | Halaman tidak terbuka | Ok |
| 2 | Melakukan pemanggilan halaman dengan user yang memiliki hak | Halaman terbuka dengan data <i>list KIR</i> yang tercatat | Ok |

4.3.8 Pengujian Fungsi List Pajak

Tabel 4. 8 Pengujian Fungsi List

Pajak

| No | Pengujian yang dilakukan | Hasil yang Diharapkan | Hasil Pengujian |
|----|---|---|-----------------|
| 1 | Melakukan pemanggilan halaman dengan user yang tidak memiliki hak | Halaman tidak terbuka | Ok |
| 2 | Melakukan pemanggilan halaman dengan user yang memiliki hak | Halaman terbuka dengan data <i>list pajak</i> yang tercatat | Ok |

4.4 Implikasi

Dari hasil pengembangan yang telah dilakukan oleh penulis, menghasilkan sebuah *web application* yang digunakan untuk mengatur mobil, KIR, Pajak, Asuransi, etc, dan bisa juga digunakan untuk me-request mobil. *Web application* ini telah di coba oleh 20 *user* dengan *role* yang berbeda-beda yaitu super admin, admin, user, staff. Dari hasil pengumpulan data dapat disimpulkan bahwa sistem ini terbukti mempermudah

kinerja yang biasa dilakukan secara manual, dan data lebih terjaga, tetapi ada beberapa kekurangan dari sistem ini yaitu sistem ini masih kekurangan beberapa fitur yang dikemudian hari ada kemungkinan diperlukan oleh *user*.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penulis mengambil kesimpulan dari mulai pengembangan, implementasi hingga pengujian dapat disimpulkan hal-hal yang bisa menjadi landasan yaitu:

1. Manajemen mobil dan juga manajemen peminjaman mobil adalah hal yang sangat dibutuhkan pada saat sekarang ini, banyak yang sudah pandai dalam mengatur sesuatu dengan mengandalkan manajemen, maka dari itu agar bisa mengembangkan manajemen mobil dan manajemen peminjaman mobil yang sudah ada dibuatlah sistem yang dapat menjaga data serta mengatur mobil.
2. Pada Perancangan sistem ini menggunakan metode *scrum* untuk mengetahui apa saja kebutuhan dari sistem yang ingin dibangun ini, dengan mengetahui hal tersebut maka akan mudah dalam merancang waktu dan *resource* yang dibutuhkan untuk pengembangan sistem ini.
3. Dengan menggunakan CodeIgniter penulis mengembangkan sistem yang nantinya akan memudahkan dalam pengembangan.
4. Hasil dari penelitian ini adalah sebuah *web application* yang nantinya dapat digunakan dengan mudah oleh *user*.
5. *Outcome* dari sistem yang dibuat oleh penulis memudahkan *admin* dalam mengelola mobil dan juga mengelola *request* peminjaman mobil yang dilakukan oleh *user* yang ingin meminjam mobil.

5.2 Saran

Dalam pembuatan sistem, adapun saran yang dapat diterapkan pada sistem yaitu:

1. Pengembangan *web application* perlu dilakukan seiring zaman untuk menyesuaikan dengan teknologi yang akan berkembang.
2. Pengembangan *web application* dapat memberikan kemudahan dan keamanan sehingga mengurangi rasa khawatir terhadap *user* yang menggunakan.
3. Evaluasi sistem secara berkala sangat dibutuhkan untuk menjaga sistem tetap bekerja dengan baik.

DAFTAR PUSTAKA

- Arrhioui, K., Mbarki, S., Betari, O., Roubi, S., & Erramdani, M. (2017). A Model Driven Approach for Modeling and Generating PHP CodeIgniter based Applications. *Transactions on Machine Learning and Artificial Intelligence*, 5(4), 259–266.
- Asghar, A. R., Tabassum, A., Bhatti, S. N., Sultan, Z., & Abbas, R. (2016). Role of Requirements Elicitation & Prioritization to Optimize Quality in Scrum Agile Development. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 7(12), 300–306.
- Darwish, N. R., & Megahed, S. (2016). Requirements Engineering in Scrum Framework. *International Journal of Computer Applications*, 149(8), 24–29.
- Das, R., & Saikia, L. P. (2016). Comparison of Procedural PHP with Codeigniter and Laravel Framework. *International Journal of Current Trends in Engineering & Research (IJCTER)*, 2(6), 42–48.
- Endra, R. Y., & Aprilita, D. S. (2018). E-Report Berbasis Web Menggunakan Metode Model View Controller Untuk Mengetahui Peningkatan Perkembangan Prestasi Anak Didik. *Explore – Jurnal Sistem Informasi Dan*

- Telematika*, 9(3), 15–22.
- Firdaus, M. A. (2017). Implementasi Kerangka Kerja Scrum Pada Manajemen Pengembangan Sistem Informasi. *Seminar Nasional Teknologi Informasi Dan Multimedia 2017*, 1(2), 283–288.
- Hesananda, R., Warnars, H. L. H. S., & Sianipar, N. F. (2017). Supervised Classification Karakter Morfologi Tanaman Keladi Tikus (*Typhonium Flagelliforme*) Menggunakan Database Management System. *Jurnal Sistem Komputer*, 7(2), 50–58.
- Lasminiasih, P. S., Akbar, A., Andriansyah, M., & Utomo, R. B. (2016). Perancangan Sistem Informasi Kredit Mikro Mahasiswa Berbasis Web. *Jurnal Sistem Informasi (JSI)*, 8(1), 883–893.
- Matharu, G. S., Mishra, A., Singh, H., & Upadhyay, P. (2015). Empirical Study of Agile Software Development Methodologies: A Comparative Analysis. *ACM SIGSOFT Software Engineering Notes*, 40(1), 1–6.
- Nugroho, A. A., & Setiyawati, N. (2019). Perancangan Dan Implementasi Aplikasi It Investment Log Berbasis Web (Studi Kasus : PT. XYZ). *Jurnal of Business and Audit Information Systems*, 2(1), 38–47.
- Permana, P. A. (2015). Scrum Method Implementation in a Software Development Project Management. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 6(9), 198–204.
- Rizkita, N., Rosely, E., & Nugroho, H. (2018). Aplikasi Pendaftaran dan Transaksi Pasien Klinik Hewan di Bandung Berbasis Web. *EProceedings of Applied Science*, 4(3), 1512–1520.
- Robiansyah, H., & Angreani, L. S. (2017). Sistem Informasi Mahasiswa Asing. *MATICS : Jurnal Ilmu Komputer Dan Teknologi Informasi*, 9(1), 23–26.
- Tyagi, S., Sibal, R., Suri, B., Wadhwa, B., & Shekhar, S. (2018). Development of Reusable Hybrid Test Automation Framework for Web Based Scrum Projects. *Journal Of Applied Science and Engineering*, 21(3), 455–462.
- Wahyudi, T. (2017). Rancang Bangun Sistem Informasi Pondok Pesantren (Studi Kasus Darul Abror Watumas). *Simposium Nasional Ilmu Pengetahuan Dan Teknologi (SIMNASIPTEK) 2017*, 1(1), 23–30.