

**IMPLEMENTASI METODE SCRUM DALAM
PENGEMBANGAN SISTEM TRANSPORTATION
MANAGEMENT SYSTEM PADA PERUSAHAAN
PADMATIRTA GROUP**

PROPOSAL SKRIPSI



Oleh :

PRASETYO WICAKSONO

06.2022.1.90521

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI ADHI TAMA SURABAYA**

2024

**IMPLEMENTASI METODE SCRUM DALAM PENGEMBANGAN SISTEM
TRANSPORTATION MANAGEMENT SYSTEM PADA PERUSAHAAN
PADMATIRTA GROUP**

PROPOSAL SKRIPSI

Diajukan Guna Memenuhi Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer
Pada

Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Adhi Tama Surabaya

Surabaya, 22 Januari 2024

Mengetahui / Menyetujui,

Dosen Pembimbing I

Dr. Rinci Kembang Hapsari, S.Si.,M.Kom

Nip. 133011

**Jurusan Teknik Informatika
Ketua,**

Dr. Rinci Kembang Hapsari, S.Si.,M.Kom

Nip. 133011

**LEMBAR PENGESAHAN
SKRIPSI BERJUDUL**

**IMPLEMENTASI METODE SCRUM DALAM
PENGEMBANGAN SISTEM *TRANSPORTATION
MANAGEMENT SYSTEM* PADA PERUSAHAAN
PADMATIRTA GROUP**

Yang dipersiapkan dan disusun oleh

PRASETYO WICAKSONO

06.2022.1.90521

Dipertahankan di depan penguji skripsi

Jurusan Teknik Informatika

Institut Teknologi Adhi Tama Surabaya

Tanggal: 22 Januari 2023

Dinyatakan telah memenuhi syarat untuk diterima

PENGUJI

1. Nama : Muchamad Kurniawan, S.Kom.,M.Kom.

NIP : 153045

Ketua Sidang/Penguji I

2. Nama : Hendro Nugroho, S.T.,M.Kom..

NIP : 153047

Penguji II

3. Nama : Dr. Rinci Kembang Hapsari, S.Si.,M.Kom

NIP : 133011

Pembimbing Utama/Penguji III

ABSTRAK

Secara umum, *Transportation Management System* bertujuan untuk mengatur kegiatan logistik pengiriman barang dan memantau keadaan kendaraan yang akan digunakan. Fitur pengiriman barang merupakan salah satu wujud pengembangan dari *Transportation Management System* yang bertujuan untuk memonitoring proses pengiriman barang hingga sampai ke pelanggan. Fitur ini memantau kondisi barang secara detail sehingga memungkinkan pengguna dapat melacak keadaan barang selama proses permintaan pengiriman hingga sampai ke pelanggan. Selain fitur pengiriman barang juga terdapat fitur perbaikan kendaraan, penerimaan suku cadang, pencatatan biaya operasional dan *monitoring tracking* kendaraan. Metode *Scrum* merupakan metodologi yang dinilai dapat menghasilkan perangkat lunak yang sesuai dengan keinginan pengguna. Metode ini dapat diandalkan dalam pembuatan sistem dengan skala proyek yang besar, serta lebih mudah beradaptasi dengan banyak tambahan dan perubahan. Proses pengerjaan *scrum* dibagi menjadi beberapa proses kecil, dimana proses-proses tersebut dilakukan pengujian. Sehingga hasil akhir proyek sudah teruji satu per satu. Proses pengembangan yang cepat. Dikarenakan *scrum* memprioritaskan fitur utama, maka proyek dapat di presentasikan tanpa menunggu keseluruhan proyek terselesaikan. Mengurangi resiko proyek gagal, alasannya antara lain karena dalam prosesnya telah mengalami perbaikan sedini mungkin. Penelitian ini dilakukan pada Proyek Warehouse Manajemen sistem untuk mengelola proses pengiriman barang pada Padmatirta Group. Hasil penelitian yang telah dilakukan menghasilkan menu pengiriman yang membantu memantau pergerakan barang sesuai alur pengiriman.

ABSTRACT

In general, the Transportation Management System aims to organize logistics activities for shipping goods and monitor the condition of the vehicles that will be used. The goods delivery feature is a form of development of the Transportation Management System which aims to monitor the goods delivery process until it reaches the customer. This feature monitors the condition of goods in detail, allowing users to track the condition of goods during the delivery request process until it reaches the customer. Apart from the goods delivery feature, there are also vehicle repair features, receiving spare parts, recording operational costs and monitoring vehicle tracking. The Scrum method is a methodology that is considered capable of producing software that meets the wishes of users. This method can be relied on in creating systems with large project scales, and is easier to adapt to many additions and changes. The Scrum work process is divided into several small processes, where these processes are tested. So that the final project results have been tested one by one. Fast development process. Because Scrum prioritizes main features, projects can be presented without waiting for the entire project to be completed. Reducing the risk of a project failing, the reason for this is because the process has undergone improvements as early as possible. This research was conducted on the Warehouse Management System Project for managing the goods delivery process at the Padmatirta Group. The results of the research that has been carried out produce a delivery menu that helps monitor the movement of goods according to the delivery flow.

KATA PENGANTAR

Dengan mengucapkan puji syukur kehadiran Tuhan Yang Maha Esa yang telah melimpahkan rahmat dan hidayah-Nya. Sehingga dapat terselesaikannya proposal skripsi ini dengan baik dan tepat waktu. Maksud dari penyusunan proposal skripsi ini adalah dalam rangka memenuhi syarat mengikuti siding skripsi. Keberhasilan penyusunan proposal skripsi ini mulai dari tahap studi kasus, pengumpulan data, pengerjaan aplikasi, hingga penyusunan laporan ini tidak terlepas dari bimbingan dan bantuan dari berbagai pihak. Untuk itu tidak lupa ucapan terima kasih ditujukan kepada:

1. Ibu Dr. Rinci Kembang Hapsari, S.Si., M.Kom. selaku Ketua Jurusan Teknik Informatika.
2. Ibu Dr. Rinci Kembang Hapsari, S.Si., M.Kom. selaku dosen pembimbing proposal skripsi.
3. Bapak Wahono Wisnu Prabowo selaku manager sistem analis padmakriya digital.
4. Seluruh karyawan Padmatirta Group.
5. Orang tua dan keluarga besar yang senantiasa memberikan dukungan dan doanya.
6. Semua pihak yang telah banyak membantu sampai dengan terwujudnya laporan ini.

Penulis menyadari sepenuhnya bahwa laporan kerja praktek ini jauh dari sempurna karena keterbatasan dari penulis. Untuk itu kritik dan saran yang membangun dari pembaca sangat penulis harapkan dengan adanya perbaikan kearah yang lebih sempurna. Semoga laporan ini dapat bermanfaat bagi pembaca dan semua pihak.

DAFTAR ISI

| | |
|---|------|
| HALAMAN JUDUL | i |
| LEMBAR PENGESAHAN | ii |
| ABSTRAK | iii |
| ABSTRACT | iv |
| KATA PENGANTAR..... | v |
| DAFTAR ISI | vi |
| DAFTAR GAMBAR..... | viii |
| DAFTAR TABEL | x |
| BAB I..... | 1 |
| PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 4 |
| 1.3 Tujuan Penelitian | 4 |
| 1.4 Batasan Masalah | 4 |
| 1.5 Manfaat Penelitian | 5 |
| 1.6 Sistematika Penulisan | 5 |
| BAB II | 6 |
| LANDASAN TEORI | 6 |
| 2.1 Sistem Informasi | 6 |
| 2.2 Transportation Management System | 7 |
| 2.3 System Development Life Cycle | 8 |
| 2.4 Scrum | 9 |
| 2.5 UML..... | 13 |
| 2.6 ISO 9126 | 20 |

| | | |
|----------------------------------|-------------------------------------|-----------|
| 2.7 | Skala Likert..... | 25 |
| BAB III PERANCANGAN | | 24 |
| 3.1 | Membuat tim <i>scrum</i> | 24 |
| 3.2 | Menunjuk <i>Product Owner</i> | 25 |
| 3.3 | Pembuatan <i>Backlog</i> | 25 |
| 3.4 | Fase <i>Sprint</i> | 27 |
| 3.5 | <i>Sprint Review</i> | 69 |
| DAFTAR PUSTAKA..... | | 71 |
| LAMPIRAN | | |

DAFTAR GAMBAR

| | |
|--|----|
| <i>Gambar 3. 1 Tahap-Tahap Scrum</i> | 24 |
| <i>Gambar 3. 2 Tim Scrum</i> | 25 |
| <i>Gambar 3. 3 Sprint Timeline</i> | 27 |
| <i>Gambar 3. 4 Usecase Diagram</i> | 33 |
| <i>Gambar 3. 5 Class Diagram TMS</i> | 35 |
| <i>Gambar 3. 6 Activity Melihat Laporan Keuangan Operasional Kendaraan</i> | 36 |
| <i>Gambar 3. 7 Activity Melihat Laporan Stok Suku Cadang</i> | 37 |
| <i>Gambar 3. 8 Activity Memasukkan Biaya Operasional</i> | 38 |
| <i>Gambar 3. 9 Activity Monitoring Traking Kendaraan</i> | 39 |
| <i>Gambar 3. 10 Activity Penerimaan Suku Cadang</i> | 40 |
| <i>Gambar 3. 11 Activity Pengiriman Barang</i> | 41 |
| <i>Gambar 3. 12 Activity Perbaikan Kendaraan</i> | 43 |
| <i>Gambar 3. 13 Sequence Cek Status Pengiriman Barang</i> | 44 |
| <i>Gambar 3. 14 Sequence Cek Status Perbaikan Kendaraan</i> | 45 |
| <i>Gambar 3. 15 Sequence Keuangan Operasional</i> | 46 |
| <i>Gambar 3. 16 Sequence Konfirmasi Pengiriman Barang</i> | 47 |
| <i>Gambar 3. 17 Sequence Konfirmasi Perbaikan Kendaraan</i> | 48 |
| <i>Gambar 3. 18 Sequence Melihat Laporan Keuangan</i> | 49 |
| <i>Gambar 3. 19 Sequence Melihat Laporan Suku Cadang</i> | 50 |
| <i>Gambar 3. 20 Sequence Monitoring Traking Kendaraan</i> | 51 |
| <i>Gambar 3. 21 Sequence Penerimaan Suku Cadang</i> | 52 |
| <i>Gambar 3. 22 Sequence Pengiriman Barang</i> | 53 |
| <i>Gambar 3. 23 Sequence Perbaikan Kendaraan</i> | 54 |
| <i>Gambar 3. 24 collaboration diagram pengiriman barang</i> | 55 |
| <i>Gambar 3. 25 collaboration diagram cek status pengiriman barang</i> | 56 |

| | |
|--|----|
| <i>Gambar 3. 26 collaboration diagram konfirmasi pengiriman barang</i> | 57 |
| <i>Gambar 3. 27 collaboration diagram monitoring tracking kendaraan</i> | 58 |
| <i>Gambar 3. 28 collaboration diagram perbaikan kendaraan</i> | 59 |
| <i>Gambar 3. 29 collaboration diagram cek status perbaikan kendaraan</i> | 60 |
| <i>Gambar 3. 30 collaboration diagram konfirmasi perbaikan kendaraan</i> | 61 |
| <i>Gambar 3. 31 collaboration diagram penerimaan suku cadang</i> | 62 |
| <i>Gambar 3. 32 collaboration diagram keuangan operasional</i> | 63 |
| <i>Gambar 3. 33 collaboration diagram melihat laporan keuangan</i> | 64 |
| <i>Gambar 3. 34 collaboration diagram melihat laporan suku cadang</i> | 65 |

DAFTAR TABEL

| | |
|--|-----------|
| <i>Tabel 2. 1 Simbol usecase diagra.....</i> | <i>15</i> |
| <i>Tabel 2. 2 Simbol sequence diagram.....</i> | <i>17</i> |
| <i>Tabel 2. 3 Simbol activity diagram</i> | <i>19</i> |
| <i>Tabel 2. 4 Simbol class diagram</i> | <i>20</i> |
| <i>Tabel 2. 5 Karakteristik dan sub-karakteristik ISO 9126.....</i> | <i>23</i> |
| <i>Tabel 2. 6 Skala Likert</i> | <i>26</i> |
| <i>Tabel 3. 1 Backlog.....</i> | <i>26</i> |
| <i>Tabel 3. 2 Tabel sprint ke 1.....</i> | <i>28</i> |
| <i>Tabel 3. 3 Tabel sprint ke 2.....</i> | <i>29</i> |
| <i>Tabel 3. 4 Tabel sprint ke 3.....</i> | <i>29</i> |
| <i>Tabel 3. 5 Tabel sprint ke 4.....</i> | <i>30</i> |
| <i>Tabel 3. 6 Tabel sprint ke 5.....</i> | <i>30</i> |
| <i>Tabel 3. 7 Tabel sprint ke 6.....</i> | <i>30</i> |
| <i>Tabel 3. 8 Tabel sprint ke 7.....</i> | <i>31</i> |
| <i>Tabel 3. 9 Tabel sprint ke 8.....</i> | <i>31</i> |
| <i>Tabel 3. 10 Tabel sprint ke 9.....</i> | <i>31</i> |
| <i>Tabel 3. 11 Tabel sprint ke 10.....</i> | <i>32</i> |
| <i>Tabel 3. 12 Tabel pengujian menu biaya operasional</i> | <i>66</i> |
| <i>Tabel 3. 13 Tabel pengujian menu perbaikan kendaraan</i> | <i>66</i> |
| <i>Tabel 3. 14 Tabel pengujian menu penerimaan suku cadang</i> | <i>67</i> |
| <i>Tabel 3. 15 Tabel pengjian menu pengiriman barang</i> | <i>67</i> |
| <i>Tabel 3. 16 Tabel menu tracking kendaraan.....</i> | <i>68</i> |
| <i>Tabel 3. 17 Tabel laporan biaya operasional</i> | <i>68</i> |
| <i>Tabel 3. 18 Tabel menu laporan perbaikan kendaraan</i> | <i>69</i> |
| <i>Tabel 3. 19 Tabel menu laporan stok suku cadang</i> | <i>69</i> |

| | |
|--|-----------|
| <i>Tabel 3. 20 Menu biaya operasional.....</i> | <i>71</i> |
| <i>Tabel 3. 21 Menu perbaikan kendaraan</i> | <i>74</i> |
| <i>Tabel 3. 22 Menu penerimaan suku cadang.....</i> | <i>77</i> |
| <i>Tabel 3. 23 Menu Pengiriman Barang</i> | <i>80</i> |
| <i>Tabel 3. 24 Menu Monitoring Kendaraan.....</i> | <i>83</i> |
| <i>Tabel 3. 25 Menu laporan biaya operasional</i> | <i>86</i> |
| <i>Tabel 3. 26 Menu laporan perbaikan kendaraan.....</i> | <i>89</i> |
| <i>Tabel 3. 27 Menu laporan stok suku cadang.....</i> | <i>92</i> |

BAB I

PENDAHULUAN

1.1 Latar Belakang

Memiliki sarana transportasi yang memadai dalam logistik merupakan salah satu komponen utama dalam perusahaan yang efektif dan hampir di setiap sektor ekonomi. Transportasi adalah kegiatan yang berkaitan dengan pergerakan orang dan barang material dengan cara yang tepat. Transportasi sangat penting dalam logistik, karena dalam perekonomian nasional memungkinkan terjadinya pertukaran barang dan jasa, mengangkut bahan mentah dan produk setengah jadi untuk produksi dan produk jadi untuk konsumsi pribadi [1]. Transportasi dalam perusahaan mencakup seluruh kegiatan dalam periode dimana muatan sedang di bongkar dan muat, di angkut, di tengah perjalanan, dan selama parkir. Transportasi seiring pergerakan juga mencakup layanan tambahan seperti logistik dan pengiriman barang. Layanan ini terikat dengan perusahaan dan pengelolaan pengiriman barang, begitu juga layanan jasa yang terkait dengan transportasi.

Biaya yang terkait dengan transportasi seringkali memakan jumlah yang tidak sedikit. Penanggung jawab operasional di perusahaan bertanggung jawab atas keputusan, apakah akan menggunakan jasa transportasi sendiri atau eksternal, khususnya bertanggung jawab atas pilihan jasa pengiriman tertentu dan rute khusus pengangkutan kargo [1], Karyawan perlu mengetahui biaya transportasi, aturan hukum yang berlaku dalam hal peraturan dan ketentuan tentang transportasi, serta memiliki kemampuan untuk mengelola sumber daya

manusia dan keuangan. Oleh karena itu transportasi perlu di kelola secara baik karena dapat meningkatkan produksi dan penjualan. Penanggung jawab perlu menentukan pemilihan kapasitas angkut kendaraan, jenis dan metode transportasi, pemilihan rute, biaya operasional dan kecepatan pengiriman [2]. Di masa sekarang banyak perusahaan besar mulai menggunakan jasa transportasi milik sendiri agar mudah dalam menemukan teknologi, trik, atau teknik pemasaran yang lebih inovatif.

Sistem transportasi logistik merupakan sistem pengelolaan sumber daya yang berkaitan dengan alur kegiatan logistik. Beberapa kegiatan yang ada pada sistem transportasi logistik antara lain mengelola semua aspek masuk dan keluar kegiatan transportasi, termasuk pemrosesan pesanan, perencanaan muatan, pemilihan operator, mengatur rute dan jadwal kendaraan, mengatur dan melacak pengiriman, pembayaran tagihan pengiriman, dan audit [3]. Sistem transportasi logistik yang dapat mengakomodir kebutuhan perusahaan, diharapkan dapat memberikan parameter guna kemajuan perusahaan.

Untuk menentukan metode pada penelitian ini, penulis telah meninjau artikel ilmiah yang berkaitan dengan pengembangan sistem informasi yang dinamis dan fleksible. *Scrum* merupakan *framework* untuk pengembangan dan manajemen proyek perangkat lunak yang cukup populer. Konsep *scrum* adalah metode interaksi berkelanjutan yang dapat dilakukan untuk meningkatkan *product lifecycle* [4]. Metodologi *scrum* didasarkan pada pembagian proyek menjadi beberapa bagian *sprint*. Proses pengembangan metode *scrum* dilakukan secara berurutan sesuai prioritas yang ditetapkan. *Scrum* memiliki

sifat yang fleksible, cepat dalam arti pengerjaan jangka waktu pendek, dan kualitas produk yang baik [5].

Pada penelitian yang dilakukan oleh Kresna Dwi Prasetya, Suharjito dan Devriady Pratama tentang analisis perbandingan efektivitas model *scrum* dengan *waterfall* dalam pengembangan aplikasi, dinyatakan bahwa metode scrum memiliki keunggulan pada bidang kinerja waktu pengerjaan. Implementasi *scrum* membagi sumber daya menjadi beberapa tim. Karena ada banyak tim dalam proyek ini, setiap tim melakukan sprintnya sendiri dan tidak perlu diselesaikan secara bersamaan misalnya ketika salah satu tim scrum telah selesai dalam tahap pengumpulan kebutuhan, mereka dapat langsung menuju tahap pengembangan tanpa menunggu tim lain menyelesaikan tahapan pengumpulan kebutuhan. Di setiap tim dibuat ada product owner dari klien yang terlibat. Koordinasi dan komunikasi bisa langsung dengan *product owner* dan manajer. Product owner sudah terlibat sejak awal pada tahap perencanaan, sehingga diperoleh gambaran ruang lingkup yang lebih jelas, sehingga perencanaan lebih tepat dan efektif. Sedangkan metode waterfall membutuhkan waktu yang lebih, karena sebelum mengerjakan seluruh fitur pada proyek memerlukan persetujuan dari klien. Proses membuat dan memeriksa sampai persetujuan membutuhkan banyak waktu dan tenaga. Pengerjaan proyek waterfall juga harus dilakukan secara berurutan, jika ada satu proses yang masih belum selesai maka tidak boleh melompat ke proses berikutnya.

Berdasarkan permasalahan alur kerja proses logistik pada perusahaan Padma Group maka perlu dikembangkan sebuah sistem informasi manajemen

transportasi yang diberi nama *Transportation Management System*. Berdasarkan penelitian sebelumnya, pengembangan sistem akan digunakan metode *scrum*. Sistem ini dibangun berdasarkan kegiatan operasional pada Padma Group.

1.2 Rumusan Masalah

Berdasarkan latar belakang penulisan tersebut, maka timbul permasalahan sebagai berikut:

- 1 Bagaimana cara mengembangkan *Transportation Management System* menggunakan metode *scrum*?
- 2 Bagaimana mengukur kualitas perangkat lunak *Transportation Management System*?

1.3 Tujuan Penelitian

1. Membuat sistem informasi *Transportation Management System* berbasis *website* dan *android* dengan model *scrum*.
2. Mengukur kualitas perangkat lunak menggunakan ISO 9126.

1.4 Batasan Masalah

Batasan masalah dalam skripsi ini adalah :

1. Pengembangan sistem menggunakan model *scrum*.
2. Pengukuran kualitas perangkat lunak menggunakan ISO 9126 dengan karakteristik yang digunakan *functionality*, *reliability*, dan *usability*.

3. Pengembangan sistem sesuai dengan studi kasus yang ada di perusahaan Padma Group.
4. Fitur sistem terdapat 4 fitur utama yaitu menu pengelolaan keuangan, manajemen stok suku cadang kendaraan, manajemen pengiriman barang dan menu monitoring tracking kendaraan.

1.5 Manfaat Penelitian

Hasil yang diperoleh dari skripsi ini di harapkan bisa bermanfaat sebagai berikut:

1. Dapat mempermudah kegiatan operasional logistik perusahaan Padma Group.
2. Dapat menjadi referensi bagi mahasiswa yang ingin mengambil Tugas Akhir yang serupa permasalahannya.

1.6 Sistematika Penulisan

Sistematika penulisan ini adalah :

1. BAB 1 Pendahuluan

Dalam bab 1 terdiri dari latar belakang yang menjelaskan rasional atau justifikasi penelitian dilihat dari latar belakang pemilihan permasalahan yang diteliti; perumusan masalah yaitu merumuskan masalah secara lugas dan jelas; tujuan penelitian menyatakan target penelitian yang akan dicapai; batasan masalah dibuat sesuai dengan ruang lingkup penelitian

yang akan dilakukan dengan mempertimbangkan aspek-aspek metodologis, kelayakan dilapangan (dalam perancangan dan pembuatan program), dan ke-terbatasan yang ada pada penulis tanpa mengorbankan kebermanaan, konsep, atau judul yang diteliti; manfaat penelitian berisi tentang uraian dan harapan hasil penelitian; sistematika penulisan berisi tentang sistematika penulisan laporan skripsi.

2. BAB 2 Landasan Teori atau Kajian Teori atau Kajian Pustaka

Bagian ini berisi analisis berbagai teori dan hasil penelitian yang relevan dengan masalah yang akan diteliti. Dalam bagian ini peneliti melakukan sintesis terhadap teori yang relevan agar diperoleh legitimasi konseptual terhadap variabel yang akan diteliti.

3. BAB 3 Perancangan

Bab ini berisi tentang perancangan dari sistem yang dibuat. Perancangan ini meliputi: pokok-pokok bahasan dari perancangan, baik secara perangkat lunak atau perangkat keras yang dibuat.

BAB II

LANDASAN TEORI

2.1 Sistem Informasi

Putra & Sutjahjo menjelaskan pengertian sistem yaitu kumpulan dari beberapa fungsi yang saling berhubungan satu dengan yang lain [6]. Menurut Manurung sistem dikatakan utuh jika setiap bagian berhubungan satu sama lain dan saling beriteraksi sehingga mencapai tujuan tertentu [6]. Sistem memiliki karakteristik antara lain, memiliki komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*) dan sasaran (*objectives*) atau tujuan (*goal*) merupakan karakteristik sistem [6]. Informasi adalah kumpulan data yang sudah diolah menjadi sesuatu yang memiliki arti bagi penerimanya dan memiliki manfaat untuk pengambilan keputusan [6]. Sistem informasi adalah sebuah sistem yang dirancang untuk mengumpulkan, menyimpan, memproses data untuk menghasilkan informasi yang berguna bagi penerimanya. Sistem informasi memiliki fungsi untuk memberikan informasi tentang merancang dan mengatur operasional yang ada pada organisasi atau perusahaan, serta memberikan pengambilan keputusan yang tepat [7]. Menurut Burch dan Strater, sistem informasi kumpulan fungsi-fungsi yang bergabung secara sistematis yang dapat melaksanakan pengolahan data transaksi operasional, Menghasilkan informasi untuk mendukung manajemen dalam melaksanakan aktifitas perencanaan, pengendalian dan pengambilan keputusan dan menghasilkan laporan sebagai kepentingan organisasi [8]. Dari definisi

diatas terlihat bahwa sistem informasi merupakan satu kesatuan unsur yang bekerjasama dalam mengumpulkan, mengolah, menyimpan dan menyajikan informasi sehingga dapat digunakan dengan baik oleh pengguna [8].

2.2 Transportation Management System

Transportation Management System adalah aplikasi perangkat lunak yang membantu mengelola semua aspek keluar dan masuk kegiatan transportasi, termasuk pemrosesan pesanan, perencanaan muatan, pemilihan operator, pemilihan rute kendaraan, penjadwalan kendaraan, mengatur pengiriman, pembayaran tagihan pengiriman, dan audit kendaraan [3]. Elemen yang harus diperhatikan pada *Transportation Management System* adalah moda transportasi (darat, laut atau udara); informasi tentang ukuran, tipe, jumlah, lebar dan tinggi muatan; informasi tentang pengirim dan penerima; dan waktu pengiriman barang dan penerimaan barang. *Transportation Management System* juga harus memperhatikan kualitas layanan yang diberikan.

Untuk menjalankan proses transportasi dengan lancar, perusahaan banyak menggunakan sistem informasi yang saling berhubungan. *Transportation Management System* harus memiliki kemampuan untuk mengelola perangkat sederhana dan perangkat lunak yang memiliki kemungkinan terbatas untuk memperluas cakupan data yang diterima. *Transportation Management System* dapat melacak lokasi kendaraan melalui GPS sehingga dapat memberikan informasi tentang lokasi kendaraan berada di mana saat proses pengiriman. *Transportation Management System* memberikan informasi tentang pengemudi

yang melakukan pengiriman, hal ini dapat membantu untuk menghubungi pengemudi saat terjadi masalah pengiriman. Sistem ini mampu membantu dalam menentukan kendaraan apa yang akan dipakai untuk pengiriman sesuai dengan kondisi kendaraan, barang yang dimuat dan lokasi pengiriman; sehingga perusahaan dapat melakukan penghematan dalam hal pengiriman.

2.3 System Development Life Cycle

System Development Life Cycle (SDLC) merupakan tahap dalam melakukan pengembangan sistem. SDLC memiliki beberapa fase yaitu perencanaan, analisis, desain, implementasi, dan pemeliharaan [9]. Pada tahap perencanaan memiliki tujuan untuk mengidentifikasi kebutuhan sistem yang akan dikembangkan, menentukan jangka waktu pengembangan dan menentukan sumber daya yang akan digunakan. Pada tahap analisis memiliki tujuan untuk mengidentifikasi permasalahan dan kebutuhan untuk membangun sistem. Pada tahap desain berisi tentang penggambaran rancang model sistem yang sesuai dengan tahap analisis yang sudah dilakukan sebelumnya. Pada tahap implementasi akan dilakukan proses *coding* sesuai dengan rancangan yang sudah dibuat. Tahap pemeliharaan dilakukan supaya sistem yang telah dibuat dapat berjalan dengan lancar ataupun ada penyempurnaan sistem jika dirasa diperlukan.

Perangkat lunak dikembangkan untuk menyelesaikan beberapa tugas tertentu, oleh karena itu pengembangan perangkat lunak memiliki proses pengembangan yang kompleks dan ter integrasi, oleh karena itu sulit dalam mengawasi proses pengembangan perangkat lunak. Dalam pengembangan perangkat lunak

membutuhkan perencanaan yang matang sehingga menghasilkan perangkat lunak berkualitas lebih tinggi dengan biaya terendah sambil memenuhi tenggat waktu. Pemilihan SDLC sangat penting karena memaksimalkan dalam pembuatan perangkat lunak secara efektif; oleh karena itu, memilih SDLC yang tepat adalah pilihan manajemen dengan konsekuensi dan pre-eksekusi jangka panjang.

2.4 *Scrum*

Scrum merupakan metodologi yang mengakomodir pengembangan perangkat lunak yang fleksibel [10]. *Scrum* sering disebut dengan kerangka kerja atau langkah-langkah yang digunakan untuk mengimplementasikan pengembangan *agile* yang berfokus pada pengembangan sistem yang bersifat kompleks dan dapat berubah sesuai kebutuhan pengguna.

Scrum dinilai dapat mengatasi suatu masalah kompleks yang selalu berubah, dan juga dapat memberikan kualitas produk yang baik sesuai dengan keinginan pengguna secara produktif [11]. Selama pengerjaan proyek sedang berlangsung, pengguna dapat melakukan testing dan memberikan *feedback* lalu memberikan saran revisi pada proyek. Keuntungan menggunakan *scrum* antara lain:

1. Menghasilkan sistem yang lebih baik. Dikarenakan dalam proses pengerjaannya, proyek *scrum* dibagi menjadi beberapa proses kecil,

dimana proses-proses tersebut dilakukan pengujian. Sehingga hasil akhir proyek sudah teruji satu per satu.

2. Proses pengembangan yang cepat. Dikarenakan *scrum* memprioritaskan fitur utama, maka proyek dapat di presentasikan tanpa menunggu keseluruhan proyek terselesaikan.
3. Meningkatkan kepuasan pelanggan. Pengguna dapat melakukan uji coba serta memberikan masukan serta mengubah fokus utama dari proyek, permintaan dan komplain di tangani dengan cepat, sehingga hal ini menjadikan faktor utama kepuasan pengguna
4. Mengurangi resiko proyek gagal, alasannya antara lain karena dalam prosesnya telah mengalami perbaikan sedini mungkin.

Terdapat peran-peran tertentu dalam metodologi *scrum* yakni:

1. Master *scrum*, merupakan peranan yang bertugas untuk memastikan bahwa kinerja team *scrum* tetap bekerja sesuai jalur dan merencanakan langkah-langkah pengembangan proyek.
2. Produk owner, Peranan yang diharuskan memahami visi perusahaan sehingga proyek *scrum* dapat sesuai harapan perusahaan dan pengguna.
3. Tim pengembangan, terdiri dari programmer, desainer, data analis.

Menilik dari keuntungan penggunaan metodologi *scrum*, serta keberhasilan *scrum* dalam proses pengembangan sistem yang kompleks, menjadikan *scrum* dapat diaplikasikan ke dalam proyek TMS ini. Dengan sistem TMS yang cukup kompleks, pengorganisasian tugas-tugas menjadi

mudah di kelola Sistem kerja serta permintaan pada PT. Padmatirta group dinilai cocok karena pengguna dapat langsung dapat melihat kemajuan dan progress dalam waktu singkat.

Dalam implementasinya, metode scrum memiliki tahapan-tahapan yang meliputi:

1. Membentuk *team scrum*

Bekerja dalam tim memerlukan lebih dari satu orang per definisi. Sebagian besar tim bekerja dengan lancar, namun dinamika kelompok adalah masalah yang harus diwaspadai. Orang yang dominan cenderung mengambil alih tim dan dalam tim yang mengatur dirinya sendiri tanpa diawasi oleh manajer, hal ini bisa terjadi menjadi masalah yang cukup besar. Komunikasi dan serah terima adalah faktor kunci dalam membuat tim bekerja dan sebuah tim yang bekerja sama secara erat akan lebih mampu berkomunikasi dengan lebih baik dan memiliki lebih sedikit masalah dalam serah terima [12]. Dalam pengerjaan proyek, dibutuhkan tim yang mempunyai kompetensi yang beragam, mulai dari *developer*, *tester*, *support*, *designer*, serta *data analyst*. Pada tim *scrum* terdapat *scrum master*. *Scrum master* merupakan seseorang yang nantinya akan bertindak sebagai proyek manajer [12]. Manajer memiliki tugas untuk mengatur jalannya proyek. Manajer selalu memantau apakah pekerjaan yang dilakukan oleh anggota tim sesuai dengan perencanaan yang sudah dibuat.

2. Menunjuk produk *owner*

Seorang produk *owner* merupakan penerjemah dari team ke klien, agar proyek dapat sinkron dengan keinginan klien, maka seorang produk *owner* harus sering berinteraksi dan membahas mengenai proyek baik dengan team dan klien [12]. Pada kasus di Padma group yang ditunjuk *product owner* adalah data analis.

3. Membuat *product backlog*

Backlog merupakan daftar prioritas fitur yang masih harus diselesaikan untuk suatu sistem. *Product backlog* mewakili keseluruhan sistem dan *backlog* untuk setiap *sprint* [12]. Dalam product backlog ini, kepentingan dari user yang diinginkan dapat diselesaikan terlebih dahulu. Semakin penting *user story* suatu proyek, maka semakin tinggi juga di daftar *backlog*. *Product backlog* akan diperbaharui apabila ada proses *sprint* yang selesai, sehingga dapat membantu dalam mengerjakan fitur yang sesuai dengan prioritas.

4. Fase *Sprint*

Tujuan dari setiap *Sprint* adalah mengubah *user story* menjadi sebuah sistem [5]. *Sprint Backlog* bisa saja dipertahankan sebagai daftar perencanaan yang akan dikerjakan oleh tim. Selama *sprint*, tim *scrum* terpecah menjadi beberapa tim. Setiap *user story* dikerjakan oleh tim *scrum* dan harus selesai sesuai dengan waktu yang sudah ditentukan [5]. Tim ini terorganisir dan dikelola sendiri. Semua anggota tim berkontribusi dengan cara apa pun yang mereka bisa untuk menyelesaikan serangkaian

pekerjaan yang telah mereka sepakati bersama untuk diselesaikan di Sprint.

Dalam implementasi *sprint*, teknologi UML diterapkan. Karakteristik statis dan dinamis dari sistem dianalisis dan kemudian pengkodean dapat dilakukan [5]. Diagram model dapat digambar secara dangkal di papan atau bahkan di atas kertas (kasus umum). Untuk menjaga kelincahan dan pemrograman secepat mungkin. Untuk pendekatan ini, model UML digunakan.

5. *Sprint Review*

Di akhir setiap *sprint*, tim melakukan tinjauan Sprint, di mana fitur-fitur baru diperlihatkan kepada *product owner* atau pemangku kepentingan lainnya yang ingin memberikan *feedback* [5]. Proyek dievaluasi berdasarkan tujuan *sprint* yang ditentukan selama perencanaan *sprint*. Jenis umpan balik dalam pengembangan perangkat lunak ini dapat mengakibatkan peninjauan atau penambahan item ke *Product Backlog*, dan ini merupakan peluang untuk mengidentifikasi apa saja yang perlu ditingkatkan.

2.5 UML

UML (*Unified Modelling Language*) merupakan suatu pemodelan rancang bangun yang berbentuk diagram yang digunakan dalam pengembangan sistem. UML memiliki fungsi untuk membantu *programmer* dalam memahami sebuah rancangan sistem, sebelum dilakukannya proses pembuatan *coding* pada

pengembangan sistem. Diagram UML terdiri dari *usecase diagram*, *sequence diagram*, *activity diagram*, dan *class diagram*.





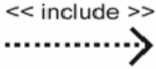
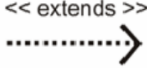
2.5.1 *Usecase diagram*


UML merupakan sebuah pemodelan yang digunakan untuk menjelaskan sistem agar mempermudah pemahaman user pada sistem yang sedang dibangun. Salah satu diagram penting yang digunakan untuk mengilustrasikan kebutuhan sistem adalah use case diagram. *Usecase diagram* mendeskripsikan interaksi dan kebutuhan antara user dengan sistem yang sedang dikembangkan secara visual. Selain itu, *usecase diagram* dapat mempresentasikan pemahaman alur bisnis suatu sistem dari sudut pandang user, serta mendeskripsikan proses utama dari sebuah sistem yang sedang dibangun.

Dalam pembuatan sebuah sistem, *usecase diagram* dibutuhkan untuk mendokumentasikan kebutuhan fungsionalitas sistem yang akan dibangun secara lengkap dan valid, mengidentifikasi faktor-faktor internal dan eksternal yang mempengaruhi sistem, dan mempresentasikan interaksi antara kebutuhan sistem dengan aktor.

Dalam membuat *usecase diagram* diperlukan identifikasi kebutuhan fungsionalitas dan batasan sistem. Terdapat beberapa komponen pada *usecase diagram*, diantaranya sebagai berikut:

Tabel 2. 1 Simbol usecase diagra

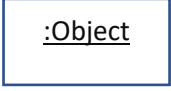


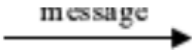

| No | Symbol | Elemen | Informasi |
|----|---|---------------------------------|---|
| 1 |  | <i>System Boundaries</i> | Merepresentasikan sebuah batasan scope pengembangan sistem yang telah didapatkan dari analisis kebutuhan. Disimbolkan dengan bentuk lingkaran |
| 2 |  | <i>Actor</i> | merupakan peranan yang diberikan kepada seseorang yang akan berinteraksi langsung dengan sistem. Actor bertanggung jawab terhadap proses input dan output yang diharapkan dari sebuah sistem. Disimbolkan dengan ikon user. |
| 3 |  | <i>Usecase</i> | fitur utama dari sebuah sistem. Setiap <i>usecase</i> merepresentasikan sebuah proses yang diharapkan dapat dicapai oleh sebuah sistem. Biasanya di simbolkan dengan bentuk elips |
| 4 |  | <i>Association Relationship</i> | Merepresentasikan penghubung antara actor dengan <i>usecase</i> , digambarkan dengan garis hitam lurus. |
| 5 |  | <i>Include Relationship</i> | Untuk menunjukkan sebuah <i>usecase</i> memiliki hubungan dengan <i>usecase</i> lain. Di visualkan dengan bentuk garis putus2 dengan panah di ujung nya. |
| 6 |  | <i>Extend Relationship</i> | Elemen ini menunjukkan suatu <i>usecase</i> merupakan tambahan dari <i>usecase</i> lain. Simbol ini di wakili dengan garis lurus dengan panah di ujungnya |

| No | Symbol | Elemen | Informasi |
|----|---|------------------------------------|---|
| 7 |  | <i>Generalization Relationship</i> | Hubungan antara objek utama dengan objek turunannya. di visualkan dengan bentuk garis dengan anak panah diujungnya. |

2.5.2 *sequence diagram*

Sequence diagram merupakan diagram yang digunakan untuk menampilkan bagaimana objek-objek saling berinteraksi serta bertukar pesan berdasarkan waktu pengirimannya. Sequence diagram merupakan pemodelan UML Kedua yang paling umum digunakan untuk merepresentasikan bagaimana objek saling berinteraksi [13]. Waktu berperan penting dalam mengatur pesan atau perintah dalam urutan yang benar. Sumbu vertical digunakan untuk merepresentasikan waktu, sedangkan sumbu horizontal untuk menampilkan. Tiap objek memiliki waktu yang digambarkan secara vertical yang disebut dengan lifeline. Sequence diagram digunakan untuk menunjukkan alur dalam sebuah proses, serta menampilkan interaksi objek – objek secara sekuensial. Beberapa elemen dari sequence diagram yakni:

Tabel 2. 2 Simbol sequence diagram

| No | Simbol | Elemen | informasi |
|----|---|-----------------|---|
| 1 |  | <i>Objek</i> | objek merupakan representasi kegiatan yang terjadi dari sistem. di visualisasikan dalam bentuk persegi. |
| 2 |  | <i>Lifeline</i> | mewakili waktu interaksi dari sebuah objek. Di visualisasikan dalam wujud garis putus-putus saat objek tidak aktif dan berbentuk garis persegi panjang yang tipis ketika objek aktif. |
| 3 |  | <i>Actor</i> | merepresentasikan pengguna yang berinteraksi dengan objek, di gambarkan sebagai persegi panjang atau ikon manusia. |
| 4 |  | <i>Messages</i> | komponen ini digunakan untuk menunjukkan komunikasi antara actor dengan objek. messages di visualisasikan sebagai anak panah. Arah anak panah menunjukkan sender dan receiver. |
| 5 |  | Activation box | elemen ini menunjukkan durasi waktu sebuah objek menyelesaikan proses. Elemen ini digambarkan dalam bentuk persegi panjang |





2.5.3 *activity diagram*

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang di rancang, bagaimana masing-masing aliran berawal, decision yang mungkin terjadi dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram membantu pengguna dalam memahami fungsi utama dari sebuah sistem, serta membantu menemukan kendala-kendala yang mungkin terjadi dalam waktu yang lebih singkat [14]. Dikaji melalui fungsinya, dapat di ketahui bahwa tujuan penggunaan activity diagram yakni:

1. Mendeskripsikan aliran aktifitas dari sebuah sistem dalam scope secara general
2. Menunjukkan urutan aktifitas satu ke aktifitas lainnya
3. Menggambarkan percabangan dan aliran dari sebuah sistem

Elemen-elemen yang digunakan dalam pembuatan activity diagram antara lain:






Tabel 2. 3 Simbol activity diagram

| No. | Simbol | Elemen | Informasi |
|-----|---|----------------|--|
| 1 |  | Start | Elemen ini mewakili awal mula suatu workflow dalam sebuah sistem. |
| 2 |  | End/final node | Merepresentasikan akhir dari sebuah workflow. Dalam suatu workflow dapat memiliki lebih dari satu final node |
| 3 |  | Activity | Mewakili sebuah aktifitas dalam suatu sistem. Disimbolkan dalam bentuk persegi dengan ujung yang tumpul. |
| 4 |  | Decision | Merupakan suatu titik yang menunjukkan percabangan dari sebuah kondisi aktifitas. |

2.5.4 Class diagram

Class diagram merupakan diagram yang membantu menunjukkan struktur dari hubungan / relasi yang terjadi pada sebuah sistem. Class diagram Memberikan dokumentasi sistem kepada developer serta memberikan gambaran struktur sistem untuk memudahkan perawatan [15]. Class diagram digunakan untuk meningkatkan pemahaman kebutuhan sistem dan struktur data serta menjelaskan secara rinci desain sistem. Pemodelan secara terperinci yang dimiliki class diagram, dapat menjadi bahan analisis bisnis untuk model sistem yang akan dibuat. Simbol dan elemen yang digunakan dalam pemodelan class diagram antara lain:

Tabel 2. 4 Simbol class diagram

| No. | Simbol | Elemen | Informasi |
|-----|---|-------------|--|
| 1. |  | Class | Merupakan atribut yang mendefinisikan sebuah kelas yang digambarkan sebagai persegi |
| 2 |  | Assosiation | Menunjukkan sebuah relasi antara 2 kelas serta mendeskripsikan hokum multiplisitas seperti one to one dan one to many, dilambangkan oleh sebuah garis . |
| 3 |  | Composition | Merupakan atribut yang menunjukkan suatu class yang memiliki hubungan dan masih satu bagian dari class lain, sehingga membutuhkan class lain untuk bergantung. Digambarkan sebagai garis dengan ujung berbentuk jajaran genjang. |
| 4 |  | Dependency | Menunjukkan operasi pada suatu kelas yang menggunakan kelas lain. Digambarkan sebagai sebuah panah bertitik-titik. |
| 5 |  | Aggregation | Menunjukkan relasi secara keseluruhan. Digambarkan dengan garis dengan jajaran genjang diujungnya. |

2.6 ISO 9126

ISO 9126 merupakan *framework* standar internasional yang digunakan untuk melakukan pengujian kualitas perangkat lunak, yang dibuat oleh International Organization for Standardization (ISO) dan International Electrotechnical Commission (IEC) [16]. ISO 9126 menentukan karakteristik

dan sub-karakteristik kualitas produk perangkat lunak. Memiliki sifat general, dan dapat diterapkan pada semua jenis produk perangkat lunak dengan disesuaikan dengan tujuan tertentu [17]. Karakteristik dan sub-karakteristik ISO 9126 mencakup fitur sistem yang luas dan mewakili model terperinci untuk mengevaluasi sistem perangkat lunak apa pun. ISO 9126 memiliki 6 karakteristik yaitu *functionality*, *reliability*, *usability*, *efficiency*, *maintainability*, dan *portability* [18].

2.6.1 *Functionality*

adalah Kemampuan perangkat lunak untuk menyediakan fungsi yang memenuhi kebutuhan pengguna dalam kondisi tertentu [17].

2.6.2 *Reliability*

adalah Kemampuan produk perangkat lunak untuk mempertahankan tingkat kinerjanya pada beberapa kondisi untuk periode waktu yang ditentukan [17].

2.6.3 *Usability*

adalah Kemampuan produk perangkat lunak mudah dipahami, dipelajari, digunakan pada kondisi penggunaan tertentu [17].

2.6.4 *Efficiency*

adalah Kemampuan produk perangkat lunak untuk memberikan kinerja yang diinginkan tergantung pada jumlah sumber daya yang digunakan [17].

2.6.5 *Maintainability*

adalah Kemampuan produk perangkat lunak dapat modifikasi yang mencakup koreksi, perbaikan atau perubahan sesuai dengan kebutuhan [17].

2.6.6 *Portability*

adalah Kemampuan produk perangkat lunak dapat dipindahkan dari satu lingkungan ke lingkungan lainnya. Lingkungan dapat mencakup organisasi, perangkat keras atau perangkat lunak [17].

Tabel 2. 5 Karakteristik dan sub-karakteristik ISO 9126

| Karakteristik | Sub karakteristik | Deskripsi |
|----------------------|--------------------------|--|
| <i>Functionality</i> | <i>Suitability</i> | Kemampuan perangkat lunak untuk menyediakan serangkaian fungsi yang sesuai untuk tugas-tugas tertentu dan tujuan pengguna. |
| | <i>Accurateness</i> | Kemampuan perangkat lunak dalam memberikan hasil yang akurat. |
| | <i>Interoperability</i> | Kemampuan perangkat lunak untuk berinteraksi dengan satu atau lebih sistem tertentu. |
| | <i>Security</i> | Kemampuan perangkat lunak untuk mencegah akses yang tidak diinginkan. |
| | <i>Compliance</i> | Kemampuan perangkat lunak dalam memenuhi standar dan kebutuhan sesuai peraturan yang berlaku. |
| <i>Reliability</i> | <i>Maturity</i> | Kemampuan perangkat lunak untuk menghindari kegagalan sebagai akibat dari kesalahan dalam perangkat lunak. |
| | <i>Fault tolerance</i> | Kemampuan perangkat lunak untuk mempertahankan kinerjanya jika terjadi kesalahan perangkat lunak. |
| | <i>Recoverability</i> | Kemampuan perangkat lunak untuk membangun kembali tingkat kinerja ketika terjadi kegagalan sistem. |
| <i>Usability</i> | <i>Understandability</i> | Kemampuan perangkat lunak dalam kemudahan untuk dipahami. |

| Karakteristik | Sub karakteristik | Deskripsi |
|------------------------|--------------------------|---|
| <i>Usability</i> | <i>Learnability</i> | Kemampuan perangkat lunak dalam kemudahan untuk dipelajari. |
| | <i>Operability</i> | Kemampuan perangkat lunak dalam kemudahan untuk dioperasikan. |
| | <i>Attractiveness</i> | Kemampuan perangkat lunak dalam menarik pengguna. |
| <i>Efficiency</i> | <i>Time behavior</i> | Kemampuan perangkat lunak dalam memberikan respon yang sesuai saat melakukan fungsinya. |
| | <i>Resource behavior</i> | Kemampuan perangkat lunak dalam menggunakan sumber daya yang dimilikinya ketika melakukan fungsi yang ditentukan. |
| <i>Maintainability</i> | <i>Analyzability</i> | Kemampuan perangkat lunak dalam menganalisa kekurangan atau penyebab kegagalan. |
| | <i>Changeability</i> | Kemampuan perangkat lunak untuk dimodifikasi. |
| | <i>Stability</i> | Kemampuan perangkat lunak untuk meminimalkan efek tak terduga dari modifikasi perangkat lunak. |
| | <i>Testability</i> | Kemampuan perangkat lunak untuk dimodifikasi dan divalidasi perangkat lunak lain. |
| <i>Portability</i> | <i>Adaptability</i> | Kemampuan perangkat lunak untuk diadaptasikan pada lingkungan yang berbeda. |

| Karakteristik | Sub karakteristik | Deskripsi |
|--------------------|------------------------|---|
| <i>Portability</i> | <i>Install ability</i> | Kemampuan perangkat lunak untuk dipasang dalam lingkungan yang berbeda. |
| | <i>Coexistence</i> | Kemampuan perangkat lunak untuk berdampingan dengan perangkat lunak lainnya dalam satu lingkungan dengan berbagi sumber daya. |
| | <i>Replaceability</i> | Kemampuan perangkat lunak untuk digunakan sebagai pengganti perangkat lunak lainnya. |

2.7 Skala Likert

Skala yang digunakan untuk mengukur persepsi, sikap atau pendapat seseorang atau kelompok mengenai sebuah peristiwa atau fenomena sosial berdasarkan definisi operasional yang telah ditetapkan. Perhitungan skala ini biasanya diaplikasikan dalam angket dan paling sering digunakan untuk riset berupa survei [19]. Skala likert digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Skala likert memiliki dua bentuk pertanyaan yaitu pertanyaan positif untuk mengukur minat positif, dan bentuk pertanyaan negatif untuk mengukur minat negatif.

Tabel 2. 6 Skala Likert

| Skala Likert | Nilai Skor Positif | Nilai Skor Negatif |
|---------------------|--------------------|--------------------|
| Sangat setuju | 5 | 1 |
| Setuju | 4 | 2 |
| Cukup | 3 | 3 |
| Tidak setuju | 2 | 4 |
| Sangat tidak setuju | 1 | 5 |

Sebelum melakukan perhitungan menggunakan skala likert, peneliti melakukan pengumpulan skor terhadap suatu fenomena. Adapun langkah-langkah untuk melakukan pengumpulan skor yaitu, 1) Peneliti mengumpulkan data yang cukup banyak, data tersebut memiliki relevansi dengan masalah yang sedang diteliti dan terdiri dari item yang cukup jelas disukai dan tidak disukai. 2) Item-item itu dicoba kepada sekelompok responden yang cukup representatif dari populasi yang ingin diteliti. 3) Responden mengecek tiap item, apakah memberikan umpan balik positif atau negatif. 4) Respons tersebut dikumpulkan dan jawaban yang memberikan indikasi positif diberi skor tertinggi. 5) Total skor dari masing-masing individu adalah penjumlahan dari skor masing-masing item dari individu tersebut. 6) Respons dianalisis untuk mengetahui item-item mana yang sangat nyata batasan antara skor tinggi dan skor rendah dalam skala total.

Skor yang sudah terkumpul berdasarkan survei yang sudah dilakukan, selanjutnya peneliti melakukan perhitungan menggunakan skala likert. Langkah-langkah melakukan perhitungan menggunakan skala likert yaitu 1)

Menghitung total skor berdasarkan poin-poin pertanyaan yang sudah dijawab oleh responden. 2) Menentukan interpretasi skor perhitungan dengan cara sebagai berikut:

X: Skor terendah likert x jumlah responden;

Y: Skor tertinggi likert x jumlah responden;

3) Menghitung indeks dengan cara berikut:

$$Indeks(\%) = \frac{Total\ Skor}{Y} \times 100\%$$

4) Menentukan interval penilaian :

Rumus interval:

$$I = \frac{100}{Jumlah\ Skor}$$

2.8 Pengujian *Blackbox*

Untuk mengetahui sistem sudah berjalan dengan baik, sistem harus melalui tahap pengujian agar memiliki kualitas yang teruji. Pengujian terhadap sistem penting untuk dilakukan untuk memeriksa kendala serta kesalahan agar tidak terjadi kerugian [20]. Terdapat beberapa teknik pengujian yang sering digunakan untuk memeriksa program, salah satunya adalah teknik *blackbox*. *Blackbox* testing atau yang sering disebut *Behavioral Testing*, adalah teknik pengujian yang dilakukan untuk melihat tugas sistem tanpa mengetahui pengkodean sistem yang dibuat. Dalam melakukan pengujian menggunakan

blackbox, penguji tidak perlu memiliki kemampuan membuat program, pengujian ini dapat dilakukan oleh siapapun. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan. Proses *Blackbox* Testing dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap formnya [20].

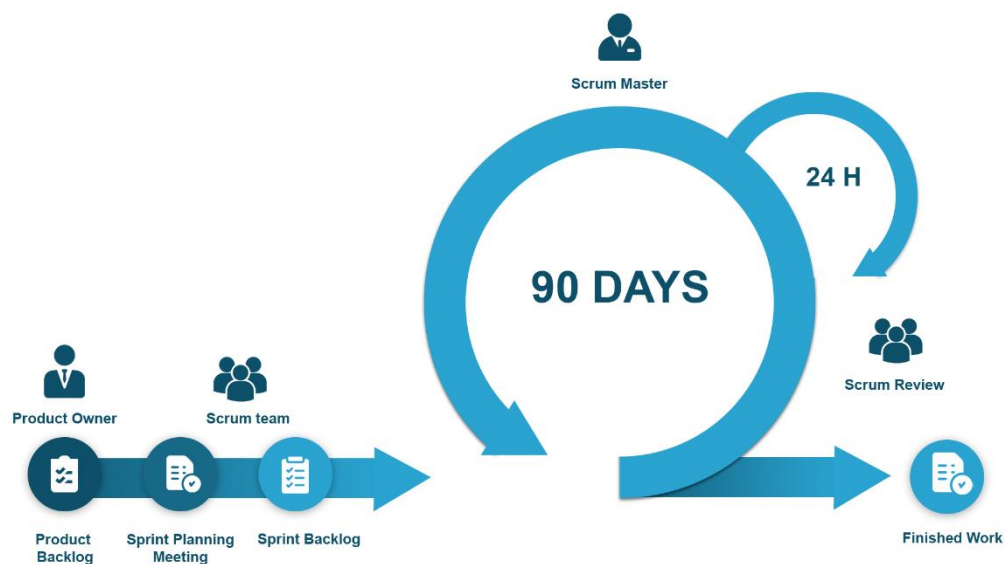
Terdapat beberapa keuntungan menggunakan teknik *blackbox* yakni antara lain Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu. Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan. Memungkinkan penguji dan pengembang bekerja secara independen tanpa mengganggu proses kerja satu sama lain [21]. Mempertimbangkan kondisi serta, sistem yang sedang berjalan pada PT. Padmatirta group, teknik *blackbox* dinilai sesuai, oleh sebab itu, teknik ini digunakan dalam pengujian ini.

BAB III

PERANCANGAN

Dari hasil Analisa yang dilakukan dapat disimpulkan bahwa Padma Group masih belum memiliki sistem manajemen transportasi yang sesuai dengan kebutuhan. Oleh karena itu *transportation management system* sangat dibutuhkan untuk meningkatkan efektifitas dan kualitas dalam kegiatan logistik.

Setelah identifikasi masalah sudah ditemukan, kemudian akan melakukan implemtasi metode *scrum*. Berikut adalah tahapan-tahapan yang dilakukan pada metode *scrum*:



Gambar 3. 1 Tahap-Tahap Scrum

3.1 Membuat tim *scrum*

Dalam pengerjaan proyek, dibutuhkan tim yang mempunyai kompetensi sebagai *programmer*, *data analyst*, *tester* dan *project manager*. Pada penelitian ini penulis bertindak sebagai *scrum master*, *data analyst*, *programmer* dan *tester*.



Gambar 3. 2 Tim Scrum

3.2 Menunjuk *Product Owner*

Seorang *product owner* merupakan penerjemah dari tim ke klien, agar proyek sesuai dengan keinginan klien, maka seorang produk *owner* harus sering berinteraksi dan membahas mengenai proyek baik dengan team dan klien. Pada kasus di Padma Group, *product owner* adalah manajer sistem analis.

3.3 Pembuatan *Backlog*

Setelah perencanaan sudah disetujui oleh pengguna dan tim proyek maka tim proyek mulai mengerjakan pembuatan *backlog* berdasarkan kebutuhan fungsional dan non fungsional. Kebutuhan fungsional berisi tentang proses-proses yang akan dilakukan oleh sistem tersebut. Sedangkan kebutuhan non fungsional merupakan spesifikasi kebutuhan sistem tersebut.

a. Kebutuhan fungsional meliputi:

1. Sistem dapat melakukan pencatatan keuangan secara sistematis.
2. Sistem dapat menampilkan laporan keuangan operasional.
3. Sistem dapat mencatat penerimaan suku cadang.
4. Sistem dapat menampilkan laporan stok suku cadang.
5. Sistem dapat mengelola perbaikan kendaraan.
6. Sistem dapat melakukan konfirmasi perbaikan kendaraan.

7. Sistem dapat mengetahui status dari perbaikan kendaraan.
 8. Sistem dapat mengelola pengiriman barang.
 9. Sistem dapat melakukan konfirmasi pengiriman barang.
 10. Sistem dapat melihat status surat tugas pengiriman barang.
 11. Sistem dapat menampilkan posisi kendaraan.
- b. Kebutuhan non fungsional meliputi:
1. Sistem berjalan di *platform website* dan *android*.
 2. Setiap *user* memiliki hak akses yang berbeda.
 3. Sistem terdapat pengecekan tipe data untuk meminimalisir kesalahan *input*.
 4. Sistem selalu aktif selama 24 jam.
 5. Sistem menampilkan data yang akurat.
 6. Sistem memiliki performa yang cepat dalam memproses data.
 7. Sistem mudah digunakan oleh pengguna.

Setelah kebutuhan fungsional dan non fungsional sudah ditentukan, proses berikutnya adalah pembuatan *backlog*.

Tabel 3. 1 Backlog

| No | Nama Backlog | Kepntingan (1-10) | Perkiraan Waktu (Hari) |
|----|-------------------------------------|----------------------|---------------------------|
| 1 | Membuat menu biaya operasional | 10 | 14 |
| 2 | Membuat menu perbaikan kendaraan | 10 | 21 |
| 3 | Membuat menu penerimaan suku cadang | 10 | 3 |
| 4 | Membuat menu pengiriman | 10 | 21 |

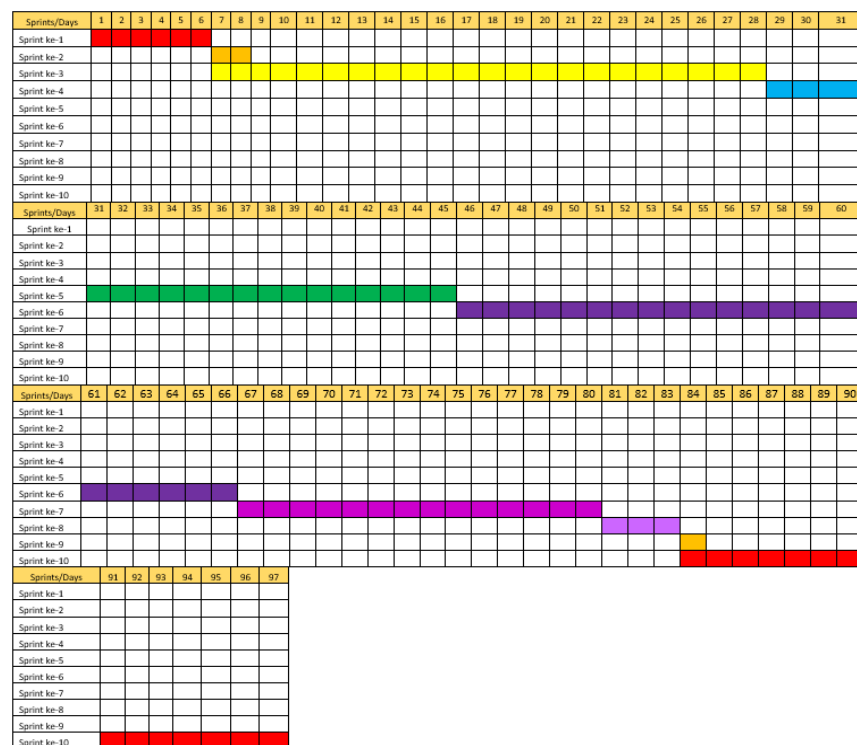
| No | Nama Backlog | Kepntingan (1-10) | Perkiraan Waktu (Hari) |
|----|--|----------------------|---------------------------|
| 5 | Membuat menu <i>tracking</i> kendaraan | 10 | 3 |
| 6 | Membuat menu laporan biaya operasional | 9 | 2 |
| 7 | Membuat menu laporan perbaikan kendaraan | 9 | 2 |
| 8 | Membuat menu laporan stok suku cadang | 9 | 2 |

3.4 Fase *Sprint*

Sprint adalah batasan waktu untuk menyelesaikan satu *product backlog*.

Sprint merupakan tahap ke dua, di mana pada tahap ini memiliki dua aktivitas yaitu:

3.4.1 *Sprint planning*



Gambar 3. 3 Sprint Timeline

Sprint planning yaitu tahap yang dilakukan setelah menentukan *product backlog* yang telah dibuat oleh tim pengembang. Tim akan melakukan rapat di setiap awal sprint untuk mengevaluasi *product backlog*, menentukan *product backlog* yang akan dikerjakan dan memberikan cara menyelesaikan *product backlog*. Pada tahapan ini proses *scrum* selanjutnya tim menentukan waktu yang dibutuhkan tiap anggota untuk melaksanakan setiap fiturnya, yang akan disajikan seperti berikut:

Tabel 3. 2 Tabel sprint ke 1

| No | Item Backlog | Task | Estimasi (Hari) |
|----|-----------------------------|---------------------------------|-----------------|
| 1 | Menu pengiriman barang | Membuat <i>usecase diagram</i> | 1 |
| | | Membuat <i>class diagram</i> | |
| | | Membuat <i>activity diagram</i> | |
| | | Membuat <i>sequence diagram</i> | |
| 2 | Menu tracking kendaraan | Membuat <i>usecase diagram</i> | 1 |
| | | Membuat <i>class diagram</i> | 1 |
| | | Membuat <i>activity diagram</i> | |
| | | Membuat <i>sequence diagram</i> | |
| 3 | Menu penerimaan suku cadang | Membuat <i>usecase diagram</i> | 1 |
| | | Membuat <i>class diagram</i> | |
| | | Membuat <i>activity diagram</i> | |
| | | Membuat <i>sequence diagram</i> | |

| No | Item Backlog | Task | Estimasi (Hari) |
|----|--------------------------|---------------------------------|-----------------|
| 4 | Menu perbaikan kendaraan | Membuat <i>usecase diagram</i> | 1 |
| | | Membuat <i>class diagram</i> | |
| | | Membuat <i>activity diagram</i> | |
| | | Membuat <i>sequence diagram</i> | |
| 5 | Menu biaya operasional | Membuat <i>usecase diagram</i> | 1 |
| | | Membuat <i>class diagram</i> | |
| | | Membuat <i>activity diagram</i> | |
| | | Membuat <i>sequence diagram</i> | |
| 6 | Menu laporan | Membuat <i>usecase diagram</i> | 1 |
| | | Membuat <i>class diagram</i> | |
| | | Membuat <i>activity diagram</i> | |
| | | Membuat <i>sequence diagram</i> | |

Tabel 3. 3 Tabel sprint ke 2

| No | Item Backlog | Task | Estimasi (Hari) |
|----|-----------------------------|-----------------------|-----------------|
| 1 | Menu pengiriman barang | Revisi desain diagram | 2 |
| 2 | Menu tracking kendaraan | Revisi desain diagram | |
| 3 | Menu penerimaan suku cadang | Revisi desain diagram | |
| 4 | Menu perbaikan kendaraan | Revisi desain diagram | |
| 5 | Menu biaya operasional | Revisi desain diagram | |
| 6 | Menu laporan | Revisi desain diagram | |

Tabel 3. 4 Tabel sprint ke 3

| No | Item Backlog | Task | Estimasi (Hari) |
|----|------------------------|-----------------|-----------------|
| 1 | Menu pengiriman barang | Desain database | 1 |
| 2 | Menu pengiriman barang | Desain form | 2 |

| No | Item Backlog | Task | Estimasi (Hari) |
|----|------------------------|-------------------|-----------------|
| 3 | Menu pengiriman barang | Proses coding | 14 |
| 4 | Menu pengiriman barang | Testing internal | 1 |
| 5 | Menu pengiriman barang | Pengerjaan revisi | 3 |

Tabel 3. 5 Tabel sprint ke 4

| No | Item Backlog | Task | Estimasi (Hari) |
|----|-------------------------|-------------------|-----------------|
| 1 | Menu tracking kendaraan | Desain form | 2 |
| 2 | Menu tracking kendaraan | Proses coding | |
| 3 | Menu tracking kendaraan | Testing internal | 1 |
| 4 | Menu tracking kendaraan | Pengerjaan revisi | |

Tabel 3. 6 Tabel sprint ke 5

| No | Item Backlog | Task | Estimasi (Hari) |
|----|-----------------------------|-------------------|-----------------|
| 1 | Menu penerimaan suku cadang | Desain database | 1 |
| 2 | Menu penerimaan suku cadang | Desain form | 2 |
| 3 | Menu penerimaan suku cadang | Proses coding | 7 |
| 4 | Menu penerimaan suku cadang | Testing internal | 1 |
| 5 | Menu penerimaan suku cadang | Pengerjaan revisi | 3 |

Tabel 3. 7 Tabel sprint ke 6

| No | Item Backlog | Task | Estimasi (Hari) |
|----|--------------------------|-------------------|-----------------|
| 1 | Menu perbaikan kendaraan | Desain database | 1 |
| 2 | Menu perbaikan kendaraan | Desain form | 2 |
| 3 | Menu perbaikan kendaraan | Proses coding | 14 |
| 4 | Menu perbaikan kendaraan | Testing internal | 1 |
| No | Item Backlog | Task | Estimasi (Hari) |
| 5 | Menu perbaikan kendaraan | Pengerjaan revisi | 3 |

Tabel 3. 8 Tabel sprint ke 7

| No | Item Backlog | Task | Estimasi (Hari) |
|----|------------------------|-------------------|-----------------|
| 1 | Menu biaya operasional | Desain database | 1 |
| 2 | Menu biaya operasional | Desain form | 2 |
| 3 | Menu biaya operasional | Proses coding | 7 |
| 4 | Menu biaya operasional | Testing internal | 1 |
| 5 | Menu biaya operasional | Pengerjaan revisi | 3 |

Tabel 3. 9 Tabel sprint ke 8

| No | Item Backlog | Task | Estimasi (Hari) |
|----|--------------|-------------------|-----------------|
| 1 | Menu laporan | Desain form | 2 |
| 2 | Menu laporan | Proses coding | |
| 3 | Menu laporan | Testing internal | 1 |
| 4 | Menu laporan | Pengerjaan revisi | |

Tabel 3. 10 Tabel sprint ke 9

| No | Item Backlog | Task | Estimasi (Hari) |
|----|-----------------------------|----------------------|-----------------|
| 1 | Menu pengiriman barang | User acceptance test | 1 |
| 2 | Menu tracking kendaraan | User acceptance test | |
| 3 | Menu penerimaan suku cadang | User acceptance test | |
| 4 | Menu perbaikan kendaraan | User acceptance test | |
| 5 | Menu biaya operasional | User acceptance test | |
| 6 | Menu laporan | User acceptance test | |

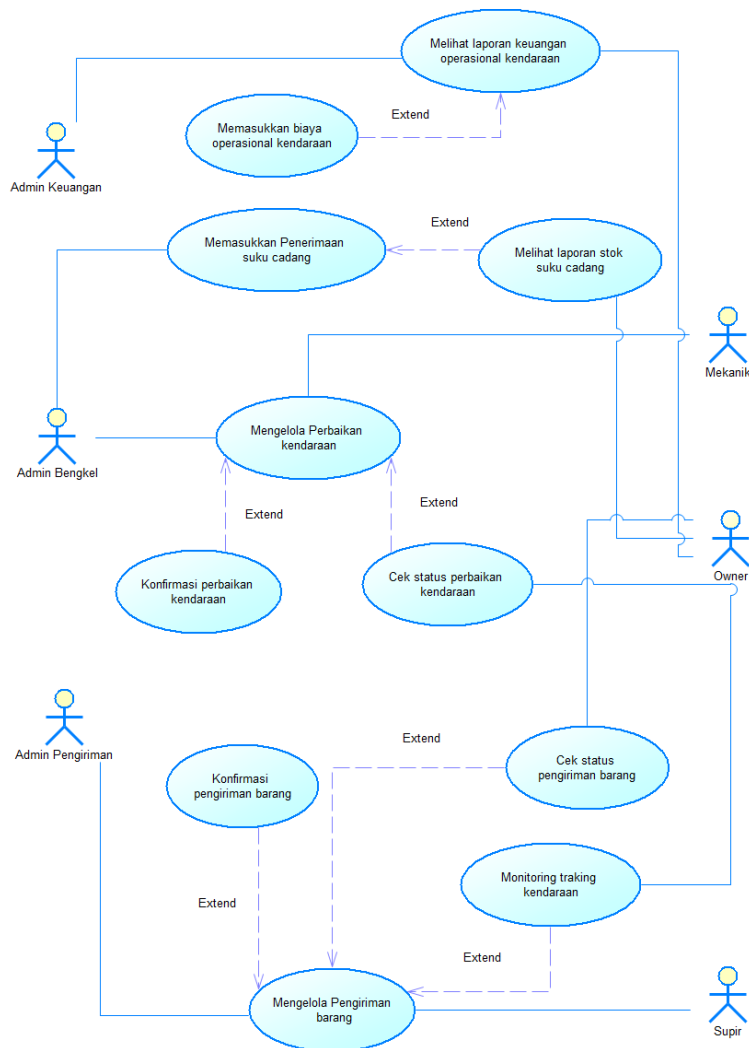
Tabel 3. 11 Tabel sprint ke 10

| No | Item Backlog | Task | Estimasi (Hari) |
|----|-----------------------------|---|-----------------|
| 1 | Menu pengiriman barang | Revisi menu pengiriman barang berdasarkan feedback UAT | 3 |
| 2 | Menu tracking kendaraan | Revisi menu tracking kendaraan berdasarkan feedback UAT | 2 |
| 3 | Menu penerimaan suku cadang | Revisi menu penerimaan suku cadang berdasarkan feedback UAT | 2 |
| 4 | Menu perbaikan kendaraan | Revisi menu perbaikan kendaraan berdasarkan feedback UAT | 3 |
| 5 | Menu biaya operasional | Revisi menu biaya operasional berdasarkan feedback UAT | 2 |
| 6 | Menu laporan | Revisi menu laporan berdasarkan feedback UAT | 2 |

3.4.2 Hasil Sprint

Berikut adalah hasil dari *product backlog* dan *sprint planning* yang telah dibuat sebelumnya:

a. *Usecase Diagram*



Gambar 3. 4 Usecase Diagram

Berdasarkan gambar 3.0 terdapat 6 aktor yang terlibat yaitu admin keuangan, admin bengkel, admin pengiriman, mekanik, owner dan supir. Terdapat 11 usecase yaitu melihat laporan keuangan operasional kendaraan, memasukkan biaya operasional kendaraan, memasukkan penerimaan suku cadang, melihat laporan stok suku cadang, mengelola perbaikan kendaraan, konfirmasi perbaikan kendaraan, cek status perbaikan kendaraan, konfirmasi pengiriman barang, cek status pengiriman barang, monitoring traking kendaraan dan mengelola pengiriman barang.

Penjelasan dari gambar usecase di atas sebagai berikut:

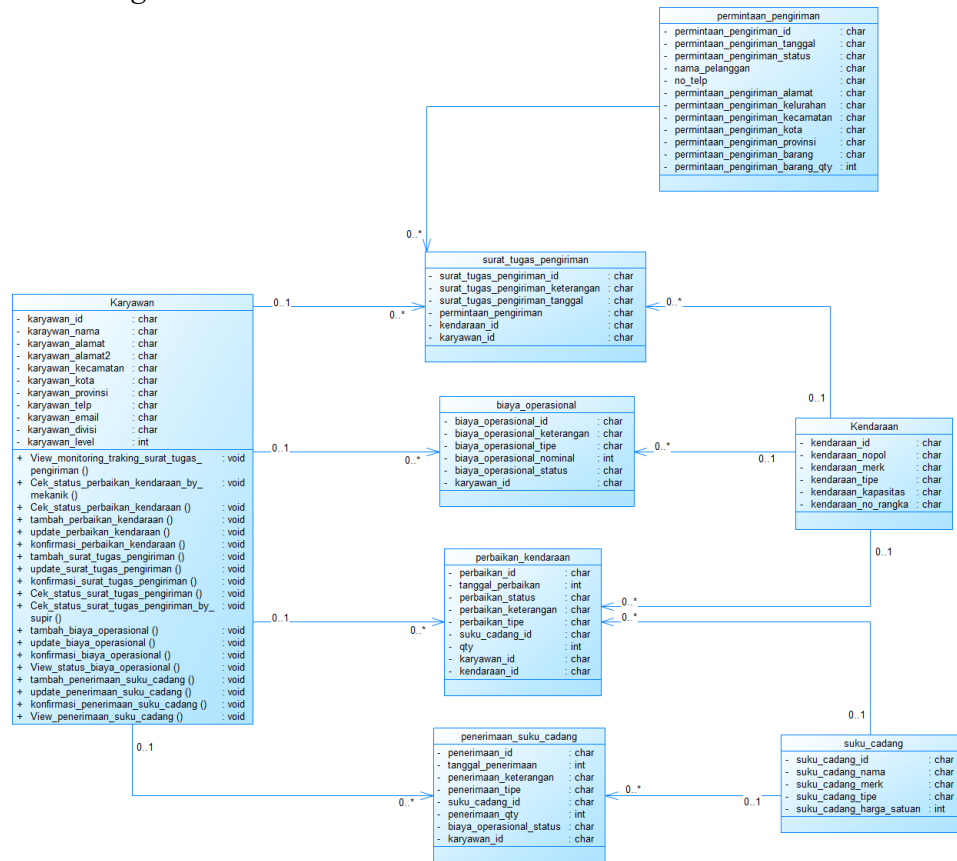
Pada usecase Melihat laporan keuangan operasional kendaraan, dijelaskan owner dan admin keuangan dapat melihat laporan keuangan operasional kendaraan. Laporan keuangan didapat dari proses memasukkan biaya operasional kendaraan, Berikutnya pada usecase Memasukkan biaya operasional kendaraan, admin keuangan dapat memasukkan data biaya operasional yang sesuai dengan pengeluaran biaya operasional yang dikeluarkan. Selanjutnya pada *usecase* Memasukkan penerimaan suku cadang, admin bengkel memasukkan data penerimaan suku cadang yang sesuai dengan penerimaan suku cadang yang diterima.

Pada Fitur *Usecase* Melihat laporan stok suku cadang, admin bengkel dan owner dapat melihat stok terakhir suku cadang yang ada di gudang. Untuk *usecase* Mengelola perbaikan kendaraan, admin bengkel dapat mengelola perbaikan kendaraan berupa input perbaikan kendaraan, konfirmasi perbaikan kendaraan dan cek status perbaikan kendaraan. *Usecase* Konfirmasi perbaikan kendaraan menjelaskan admin bengkel melakukan proses konfirmasi dokumen perbaikan kendaraan.

Cek status perbaikan kendaraan menerangkan, admin bengkel dan owner melakukan cek status dokumen perbaikan kendaraan. Selanjutnya untuk *usecase* Mengelola pengiriman barang, admin bengkel dapat mengelola pengiriman barang berupa input data pengiriman, konfirmasi pengiriman, cek status pengiriman dan melihat tracking kendaraan. pada *Usecase* Konfirmasi pengiriman barang, dapat dijelaskan bahwa admin pengiriman melakukan proses konfirmasi dokumen pengiriman. Dan untuk *usecase* Cek status pengiriman barang, admin pengiriman

dan owner melakukan cek status dokumen pengiriman barang. Dan untuk *usecase* Monitoring tracking kendaraan, pada fitur ini admin pengiriman dan owner dapat melihat posisi kendaraan terakhir berdasarkan dokumen pengiriman terakhir

b. Class diagram

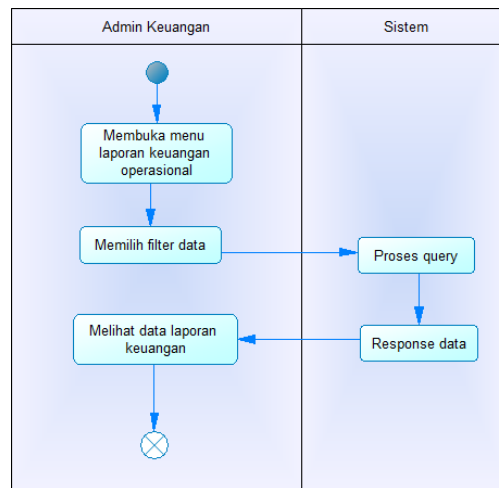


Gambar 3. 5 Class Diagram TMS

Pada gambar class diagram di atas terdapat class berupa karyawan, surat_tugas_pengiriman, permintaan_pengiriman, biaya_operasional, kendaraan, perbaikan_kendaraan, penerimaan_suku_cadang dan suku_cadang. Class surat_tugas_pengiriman memiliki relasi dengan permintaan_pengiriman melalui atribut permintaan_pengiriman_id, dengan karyawan melalui atribut karyawan_id, relasi dengan kendaraan melalui atribut kendaraan_id. Class biaya_operasional memiliki relasi dengan karyawan melalui atribut karyawan_id, relasi dengan kendaraan melalui atribut kendaraan_id. Class perbaikan_kendaraan memiliki

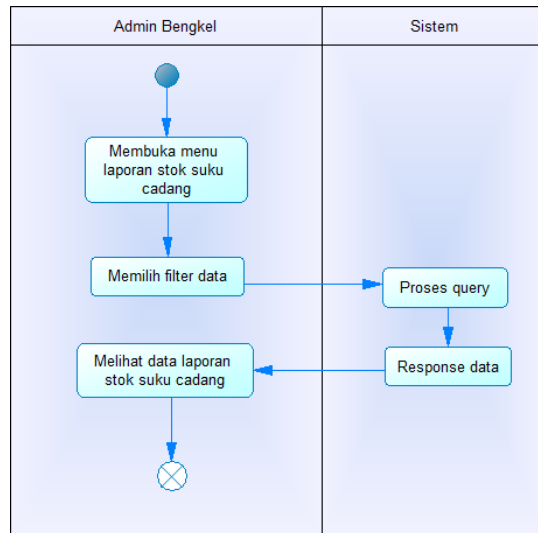
relasi dengan karyawan melalui atribut `karyawan_id`, relasi dengan kendaraan melalui atribut `kendaraan_id`, relasi dengan suku_cadang melalui atribut `suku_cadang_id`. Class `penerimaan_suku_cadang` memiliki relasi dengan karyawan melalui atribut `karyawan_id`, dengan suku_cadang melalui atribut `suku_cadang_id`.

c. Activity diagram



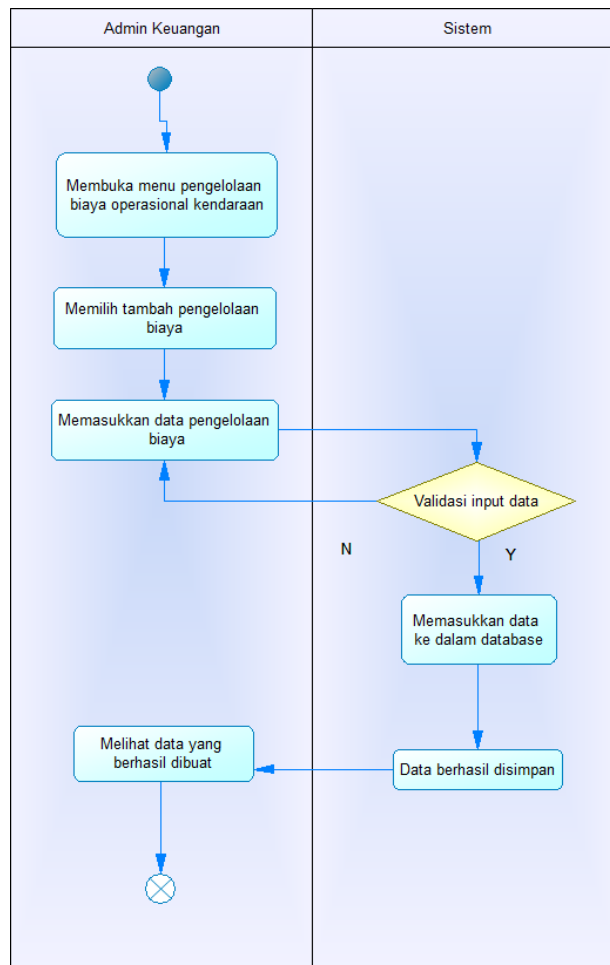
Gambar 3. 6 Activity Melihat Laporan Keuangan Operasional Kendaraan

Admin keuangan membuka menu laporan keuangan operasional. Pada menu tersebut admin diminta untuk memilih filter data untuk menampilkan data laporan. Setelah memilih filter data sistem akan memproses *query* untuk menampilkan hasil *response data*. Response data yang ditampilkan ke menu laporan keuangan sehingga admin dapat melihat data laporan keuangan.



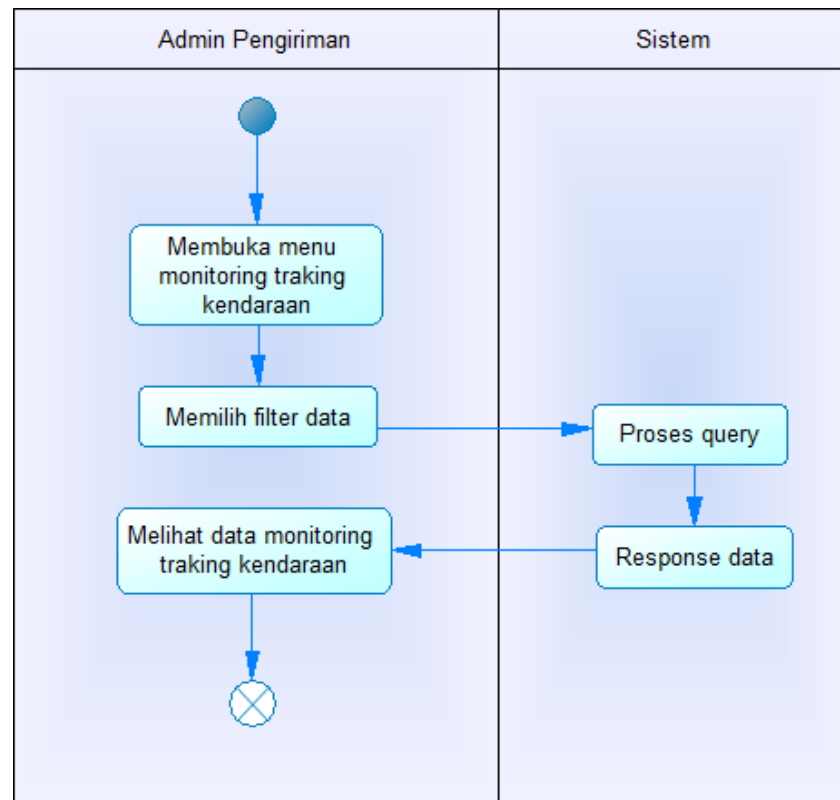
Gambar 3. 7 Activity Melihat Laporan Stok Suku Cadang

Admin bengkel membuka menu laporan stok suku cadang. Pada menu tersebut admin diminta untuk memilih filter data untuk menampilkan data laporan. Setelah memilih filter data sistem akan memproses *query* untuk menampilkan hasil *response data*. Response data yang ditampilkan ke menu laporan stok suku cadang sehingga admin dapat melihat data laporan stok.



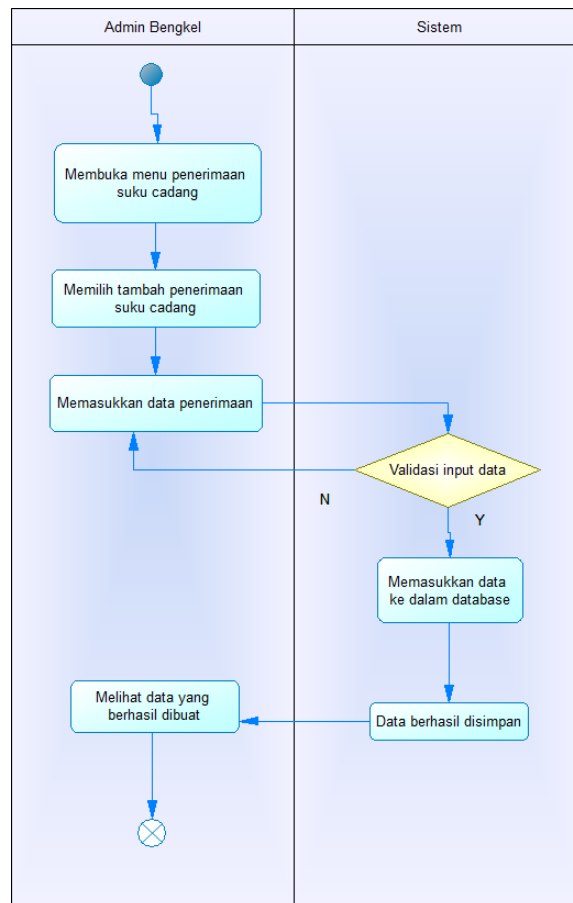
Gambar 3. 8 Activity Memasukkan Biaya Operasional

Admin keuangan membuka menu pengelolaan biaya operasional kendaraan. Admin memilih menu tambah pengelolaan biaya lalu memasukkan *input* data biaya. Setelah memilih admin memasukkan *input* data sistem akan mengecek validasi input data, jika input data benar maka akan lanjut ke proses memasukkan data ke database, jika gagal maka admin untuk mengecek apakah *input* data yang dilakukan sudah benar. Data yang berhasil dimasukkan ke database akan muncul dan admin dapat melihat rincian data yang berhasil dimasukkan.



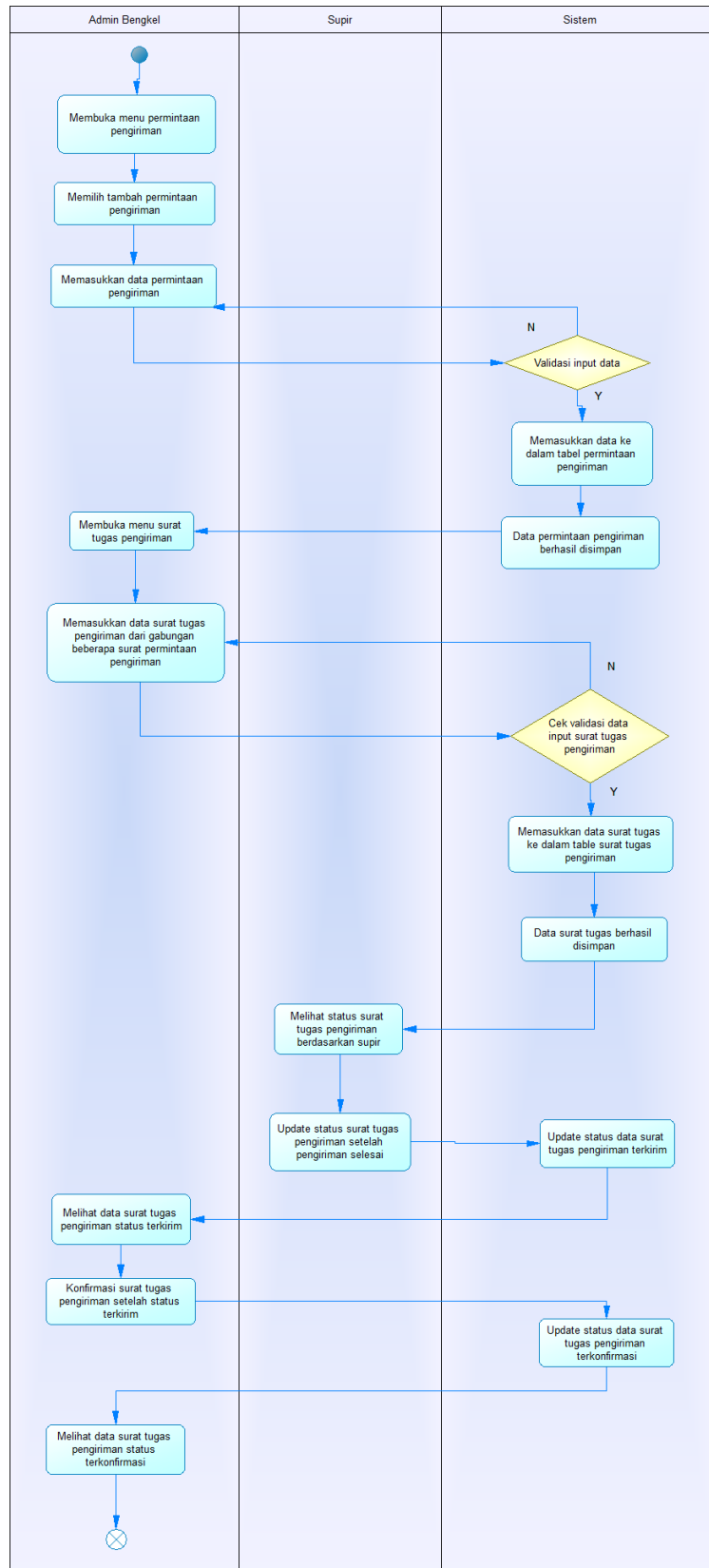
Gambar 3. 9 Activity Monitoring Traking Kendaraan

Admin pengiriman membuka menu monitoring *tracking* kendaraan. Pada menu tersebut admin diminta untuk memilih filter data untuk menampilkan data *tracking*. Setelah memilih filter data sistem akan memproses *query* untuk menampilkan hasil *response data*. Response data yang ditampilkan ke menu laporan stok suku cadang sehingga admin dapat melihat data *tracking* kendaraan.



Gambar 3. 10 Activity Penerimaan Suku Cadang

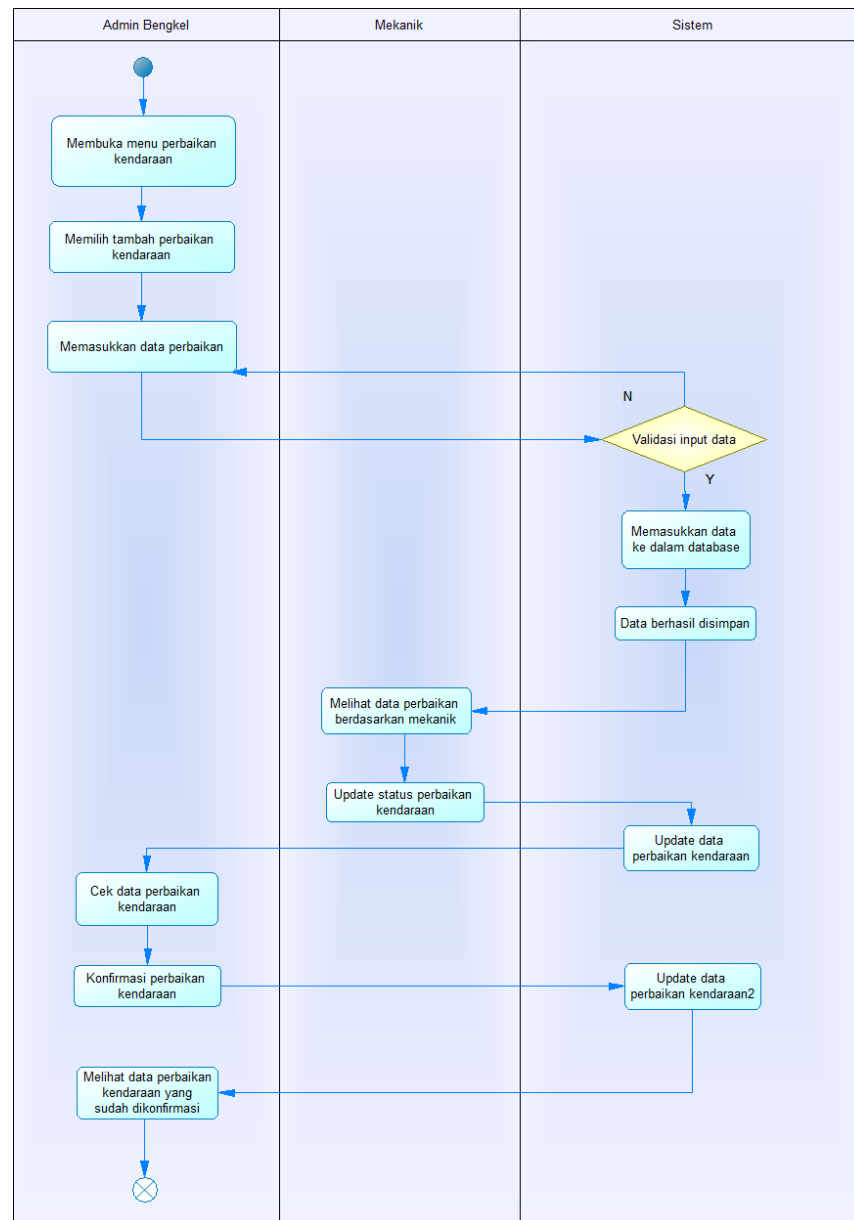
Admin bengkel membuka menu penerimaan suku cadang. Admin memilih menu tambah penerimaan lalu memasukkan *input* data penerimaan. Setelah memilih admin memasukkan *input* data sistem akan mengecek validasi input data, jika input data benar maka akan lanjut ke proses memasukkan data ke database, jika gagal maka admin untuk mengecek apakah *input* data yang dilakukan sudah benar. Data yang berhasil dimasukkan ke database akan muncul dan admin dapat melihat rincian data yang berhasil dimasukkan.



Gambar 3. 11 Activity Pengiriman Barang

Admin pengiriman membuka menu permintaan pengiriman. Admin memilih menu tambah permintaan pengiriman lalu memasukkan *input* data permintaan. Setelah memilih admin memasukkan *input* data sistem akan mengecek validasi input data, jika input data benar maka akan lanjut ke proses memasukkan data ke database, jika gagal maka admin untuk mengecek apakah *input* data yang dilakukan sudah benar. Data yang berhasil dimasukkan ke database akan muncul dan supir dapat melihat rincian data yang berhasil dimasukkan.

Supir melakukan pengiriman setelah itu supir melakukan update status pengiriman sudah selesai. Sistem mengupdate surat tugas pengiriman. Admin pengiriman melihat data surat tugas pengiriman yang sudah diperbaharui lalu admin pengiriman melakukan konfirmasi pengiriman surat tugas pengiriman, lalu sistem akan memperbaharui status surat tugas pengiriman. Admin pengiriman dapat melihat rincian detail surat tugas pengiriman yang sudah dilakukan konfirmasi.



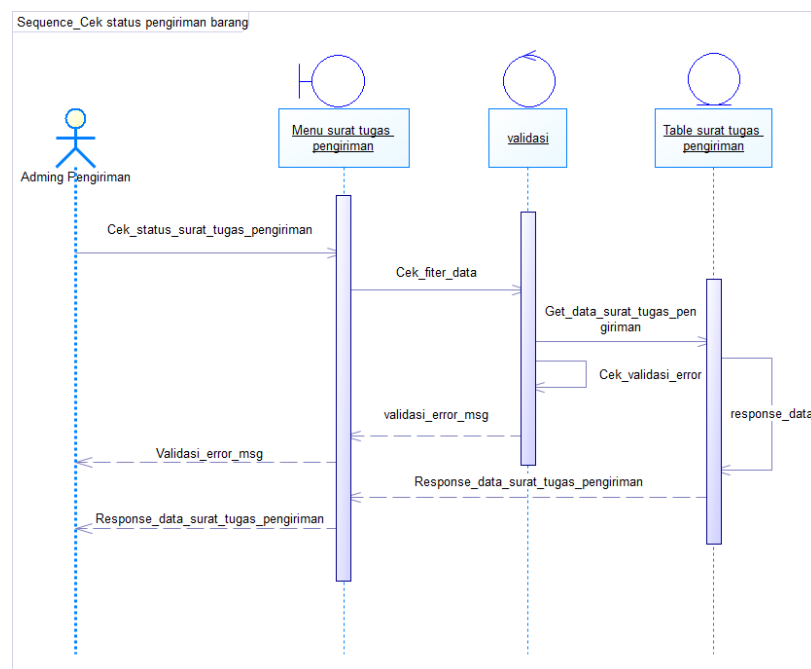
Gambar 3. 12 Activity Perbaikan Kendaraan

Admin bengkel membuka menu perbaikan kendaraan. Admin memilih menu tambah perbaikan kendaraan lalu memasukkan *input* data perbaikan. Setelah memilih admin memasukkan *input* data sistem akan mengecek validasi input data, jika input data benar maka akan lanjut ke proses memasukkan data ke database, jika gagal maka admin untuk mengecek apakah *input* data yang dilakukan sudah benar.

Data yang berhasil dimasukkan ke database akan muncul dan mekanik dapat melihat rincian data yang berhasil dimasukkan.

Mekanik melakukan perbaikan kendaraan setelah itu supir melakukan perubahan status perbaikan sudah selesai. Sistem memperbaharui data perbaikan kendaraan. Admin bengkel melihat data perbaikan kendaraan yang sudah diperbaharui lalu admin bengkel melakukan konfirmasi perbaikan kendaraan, lalu sistem akan memperbaharui status perbaikan kendaraan. Admin bengkel dapat melihat rincian detail perbaikan kendaraan yang sudah dilakukan konfirmasi.

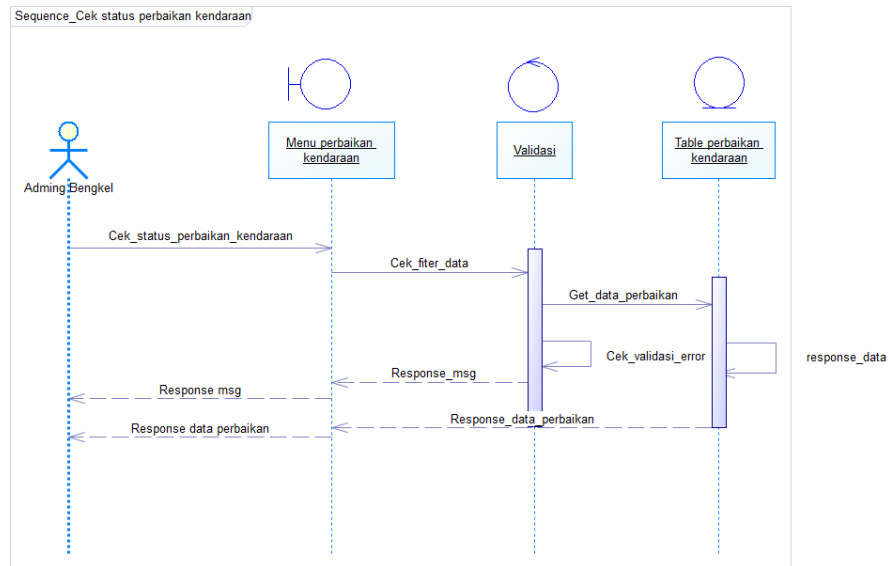
d. *Sequence diagram*



Gambar 3. 13 Sequence Cek Status Pengiriman Barang

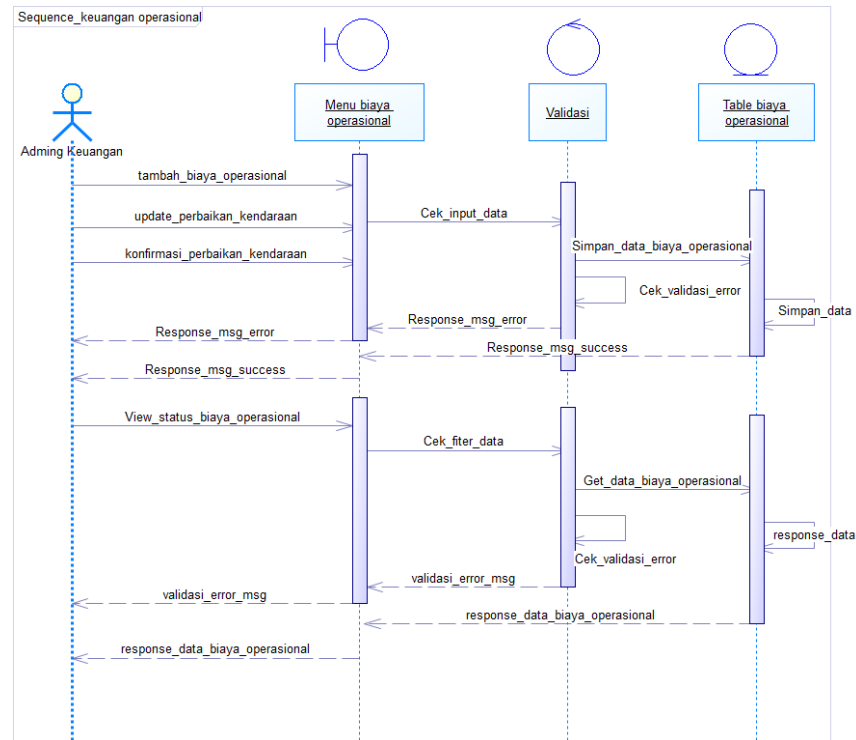
Admin pengiriman membuka menu surat tugas pengiriman. Admin memilih filter data untuk menampilkan data lalu melakukan cek status surat tugas pengiriman. Sistem akan mengecek filter data sebelum melanjutkan permintaan ke database tabel surat tugas pengiriman. Jika validasi cek filter data berhasil maka

akan mengirim permintaan untuk mendapatkan data surat tugas pengiriman pada database tabel surat tugas pengiriman. Sistem akan mengembalikan hasil *response data* dari database untuk ditampilkan di menu surat tugas pengiriman. Admin pengiriman dapat melihat status surat tugas pengiriman.



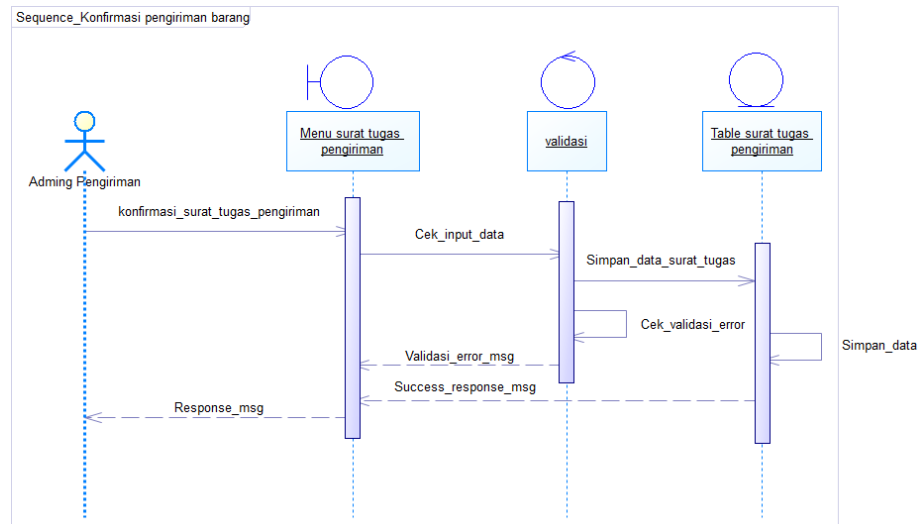
Gambar 3. 14 Sequence Cek Status Perbaikan Kendaraan

Admin bengkel membuka menu perbaikan kendaraan. Admin memilih filter data untuk menampilkan data lalu melakukan cek status perbaikan kendaraan. Sistem akan mengecek filter data sebelum melanjutkan permintaan ke database tabel perbaikan kendaraan. Jika validasi cek filter data berhasil maka akan mengirim permintaan untuk mendapatkan data perbaikan kendaraan pada database tabel perbaikan kendaraan. Sistem akan mengembalikan hasil *response data* dari database untuk ditampilkan di menu perbaikan kendaraan. Admin bengkel dapat melihat status perbaikan kendaraan.



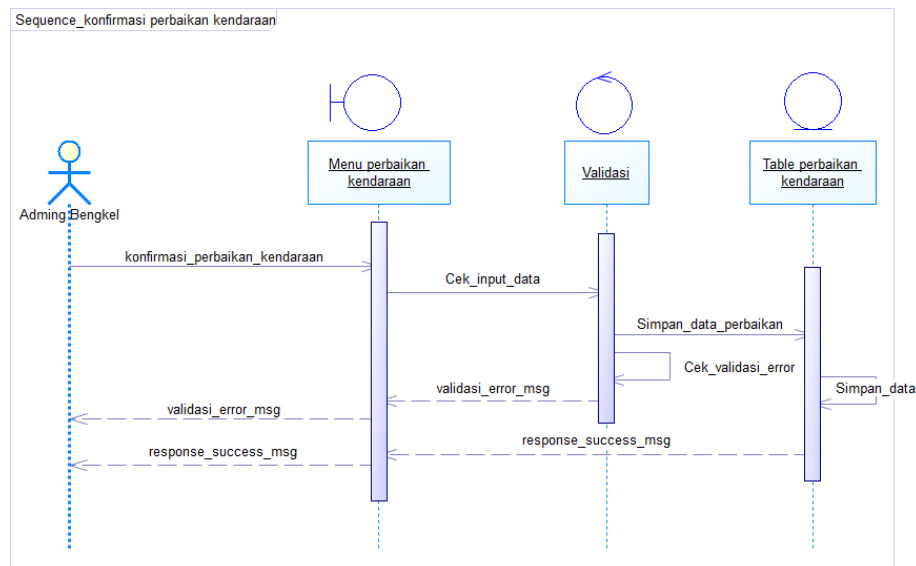
Gambar 3. 15 Sequence Keuangan Operasional

Admin keuangan membuka menu biaya operasional. Admin dapat melakukan tambah, update, dan konfirmasi. Sistem akan mengecek *input data* sebelum melanjutkan permintaan ke database tabel biaya operasional. Jika validasi cek *input data* berhasil maka akan mengirim permintaan untuk menyimpan, memperbaharui, maupun konfirmasi pada database tabel perbaikan kendaraan. Sistem akan mengembalikan hasil *response data* berhasil disimpan dari database untuk ditampilkan di menu biaya operasional. Admin keuangan dapat melihat data keuangan yang berhasil disimpan.



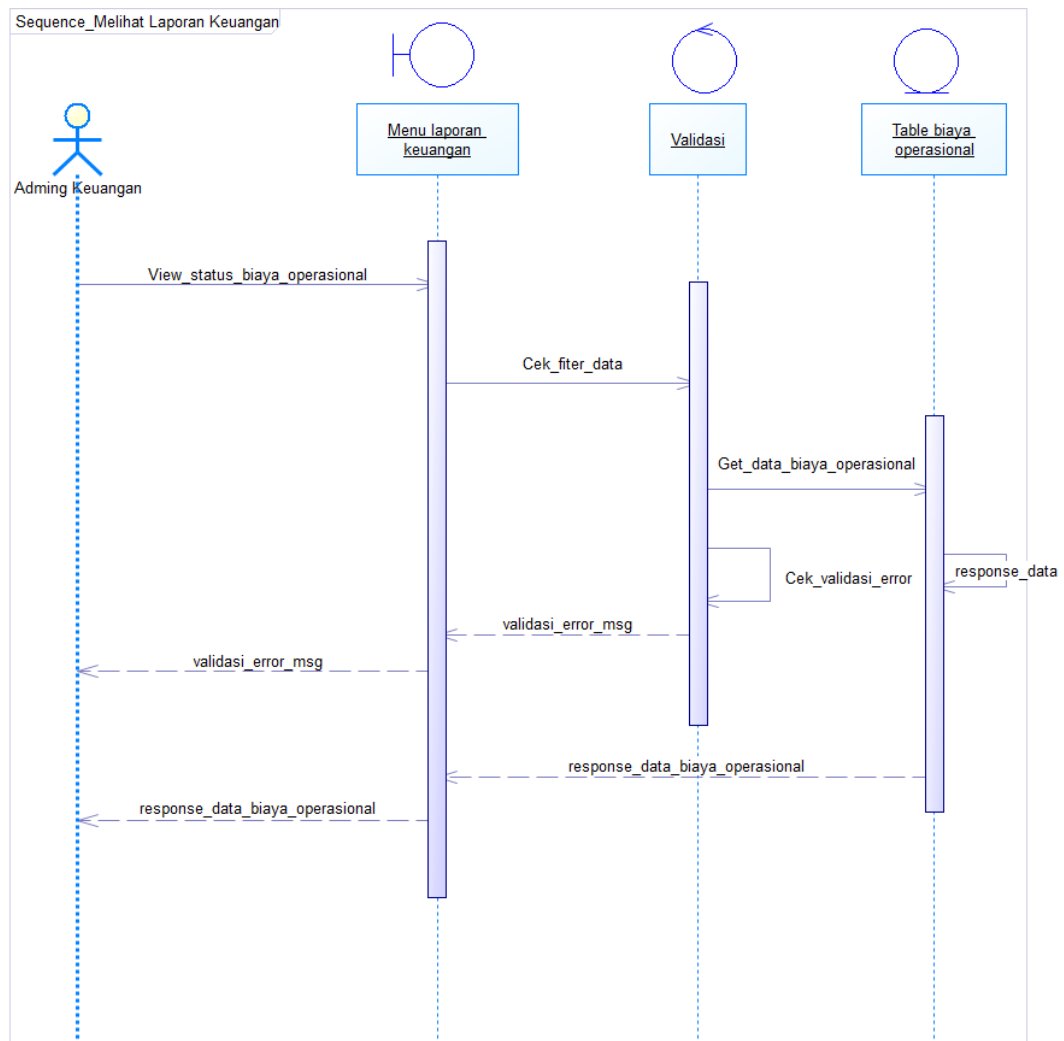
Gambar 3. 16 Sequence Konfirmasi Pengiriman Barang

Admin pengiriman membuka menu surat tugas pengiriman. Admin dapat melakukan konfirmasi surat tugas pengiriman. Sistem akan mengecek *input data* sebelum melanjutkan permintaan ke database tabel surat tugas pengiriman. Jika validasi cek *input data* berhasil maka akan mengirim permintaan untuk melakukan konfirmasi pada database tabel surat tugas pengiriman. Sistem akan mengembalikan hasil *response data* berhasil disimpan dari database untuk ditampilkan di menu surat tugas pengiriman. Admin pengiriman dapat melihat data surat tugas pengiriman yang berhasil disimpan.



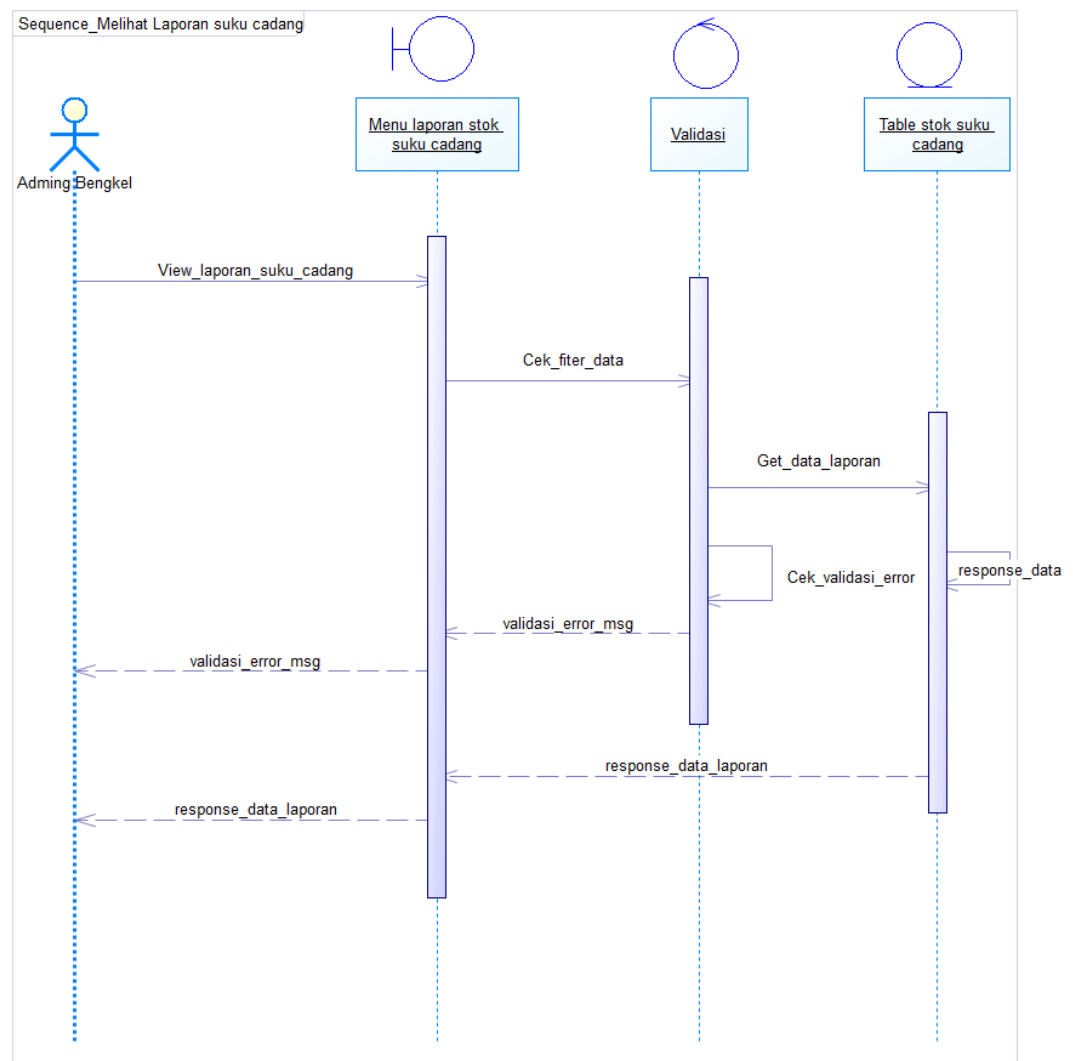
Gambar 3. 17 Sequence Konfirmasi Perbaikan Kendaraan

Admin bengkel membuka menu perbaikan kendaraan. Admin dapat melakukan konfirmasi perbaikan kendaraan. Sistem akan mengecek *input data* sebelum melanjutkan permintaan ke database tabel perbaikan kendaraan. Jika validasi cek *input data* berhasil maka akan mengirim permintaan untuk melakukan konfirmasi pada database tabel perbaikan kendaraan. Sistem akan mengembalikan hasil *response data* berhasil disimpan dari database untuk ditampilkan di menu perbaikan kendaraan Admin bengkel dapat melihat data perbaikan kendaraan yang berhasil disimpan.



Gambar 3. 18 Sequence Melihat Laporan Keuangan

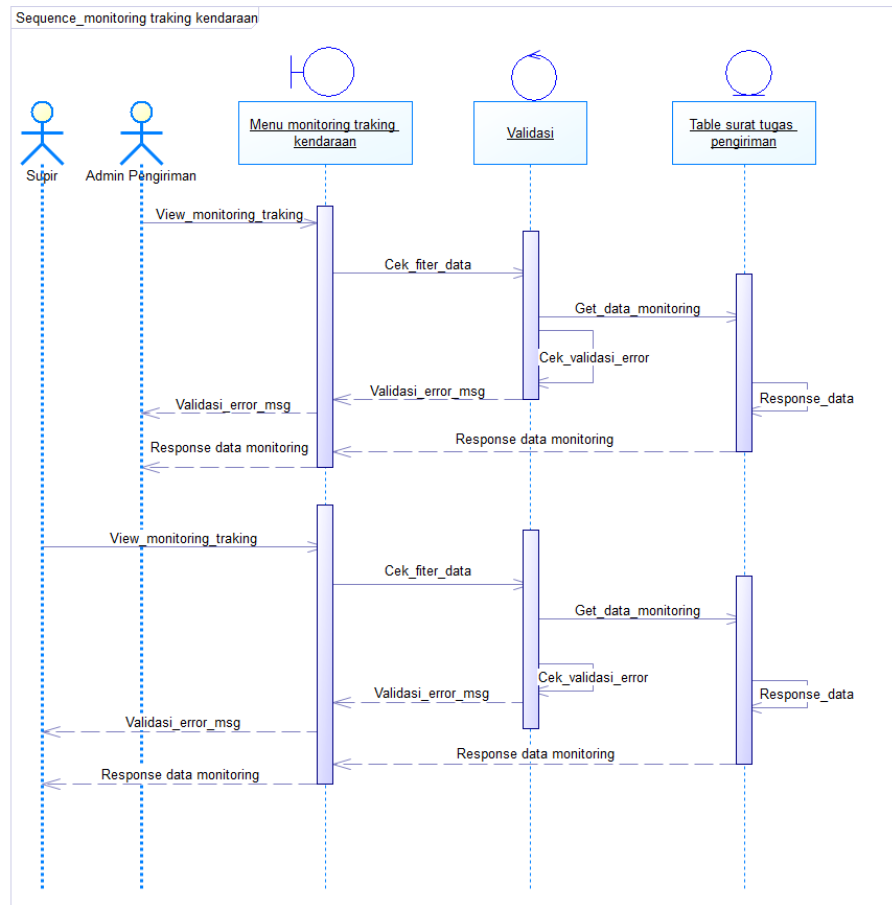
Admin keuangan membuka menu laporan keuangan. Admin memilih filter data untuk menampilkan data laporan keuangan. Sistem akan mengecek filter data sebelum melanjutkan permintaan ke database tabel biaya operasional. Jika validasi cek filter data berhasil maka akan mengirim permintaan untuk mendapatkan data laporan keuangan pada database tabel biaya operasional. Sistem akan mengembalikan hasil *response data* dari database untuk ditampilkan di menu laporan keuangan. Admin keuangan dapat melihat laporan keuangan.



Gambar 3. 19 Sequence Melihat Laporan Suku Cadang

Admin bengkel membuka menu laporan stok suku cadang. Admin memilih filter data untuk menampilkan data laporan stok. Sistem akan mengecek filter data sebelum melanjutkan permintaan ke database tabel surat tugas pengiriman. Jika validasi cek filter data berhasil maka akan mengirim permintaan untuk mendapatkan data laporan stok suku cadang pada database tabel stok suku cadang. Sistem akan mengembalikan hasil *response data* dari database untuk ditampilkan

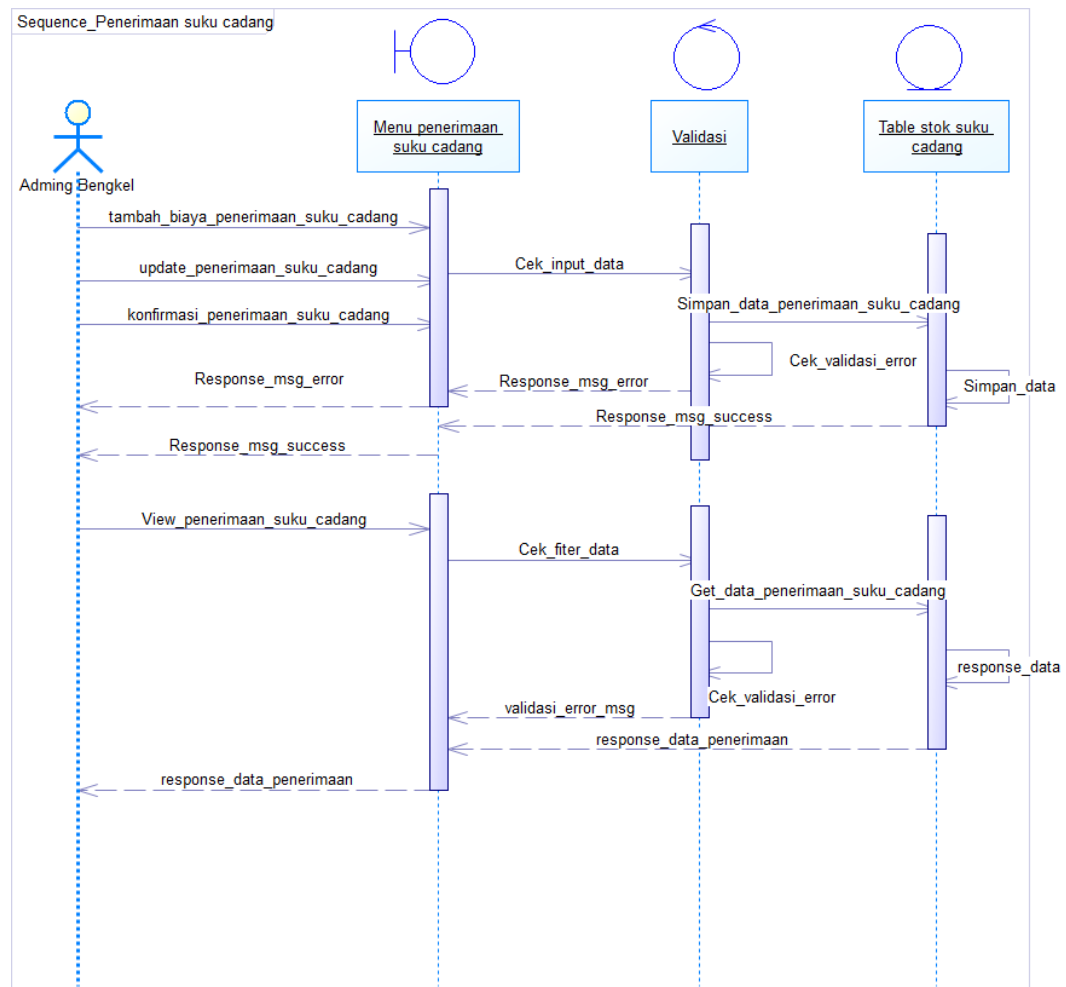
di menu laporan stok suku cadang. Admin bengkel dapat melihat laporan stok suku cadang.



Gambar 3. 20 Sequence Monitoring Traking Kendaraan

Admin pengiriman dan supir membuka menu monitoring *tracking* kendaraan. Admin dan supir memilih filter data untuk menampilkan data *tracking* kendaraan. Sistem akan mengecek filter data sebelum melanjutkan permintaan ke database tabel surat tugas pengiriman. Jika validasi cek filter data berhasil maka akan mengirim permintaan untuk mendapatkan data *tracking* kendaraan pada database tabel surat tugas pengiriman. Sistem akan mengembalikan hasil *response data* dari

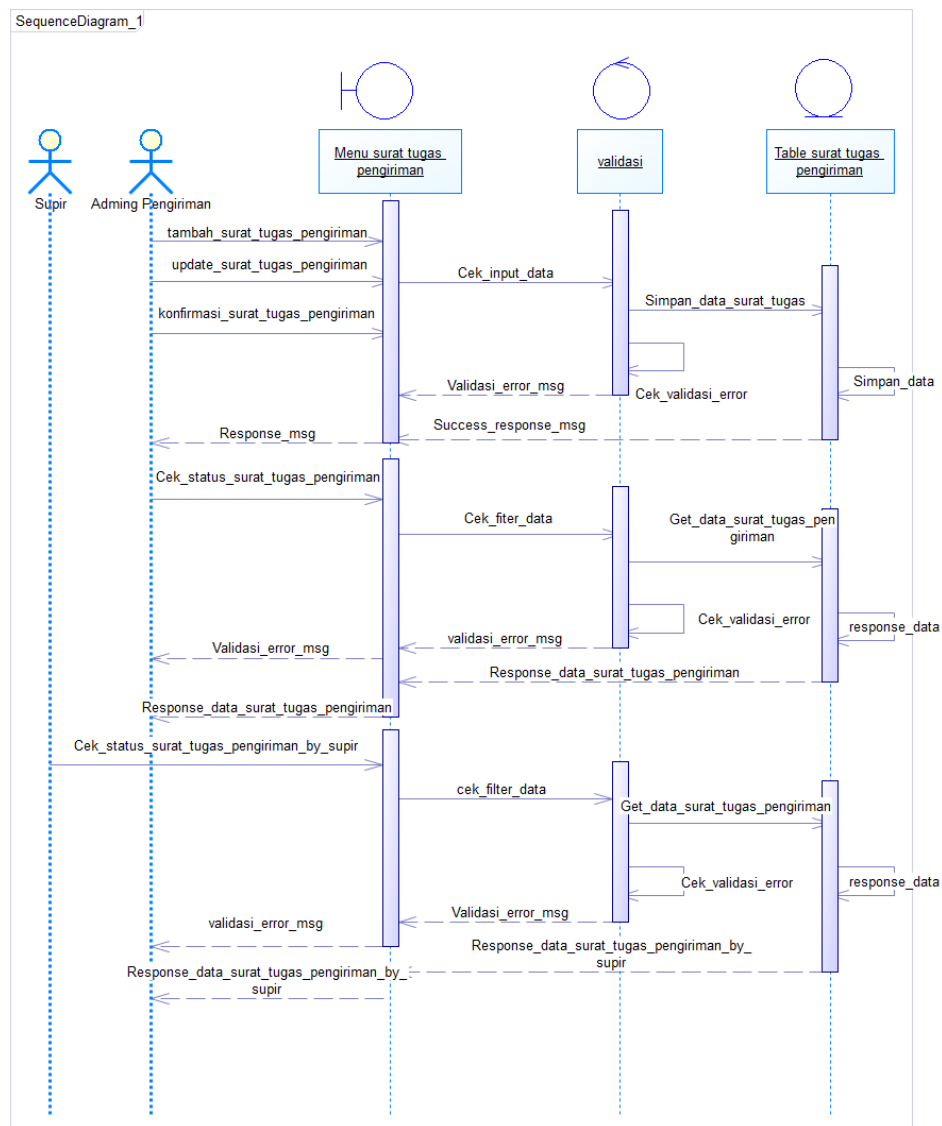
database untuk ditampilkan di menu monitoring *tracking* kendaraan. Admin pengirinan dan supir dapat melihat data *tracking* kendaraan.



Gambar 3. 21 Sequence Penerimaan Suku Cadang

Admin bengkel membuka menu penerimaan suku cadang. Admin dapat melakukan tambah, update, dan konfirmasi. Sistem akan mengecek *input data* sebelum melanjutkan permintaan ke database tabel stok suku cadang. Jika validasi cek *input data* berhasil maka akan mengirim permintaan untuk melakukan menyimpan, memperbaharui, maupun konfirmasi pada database tabel stok suku

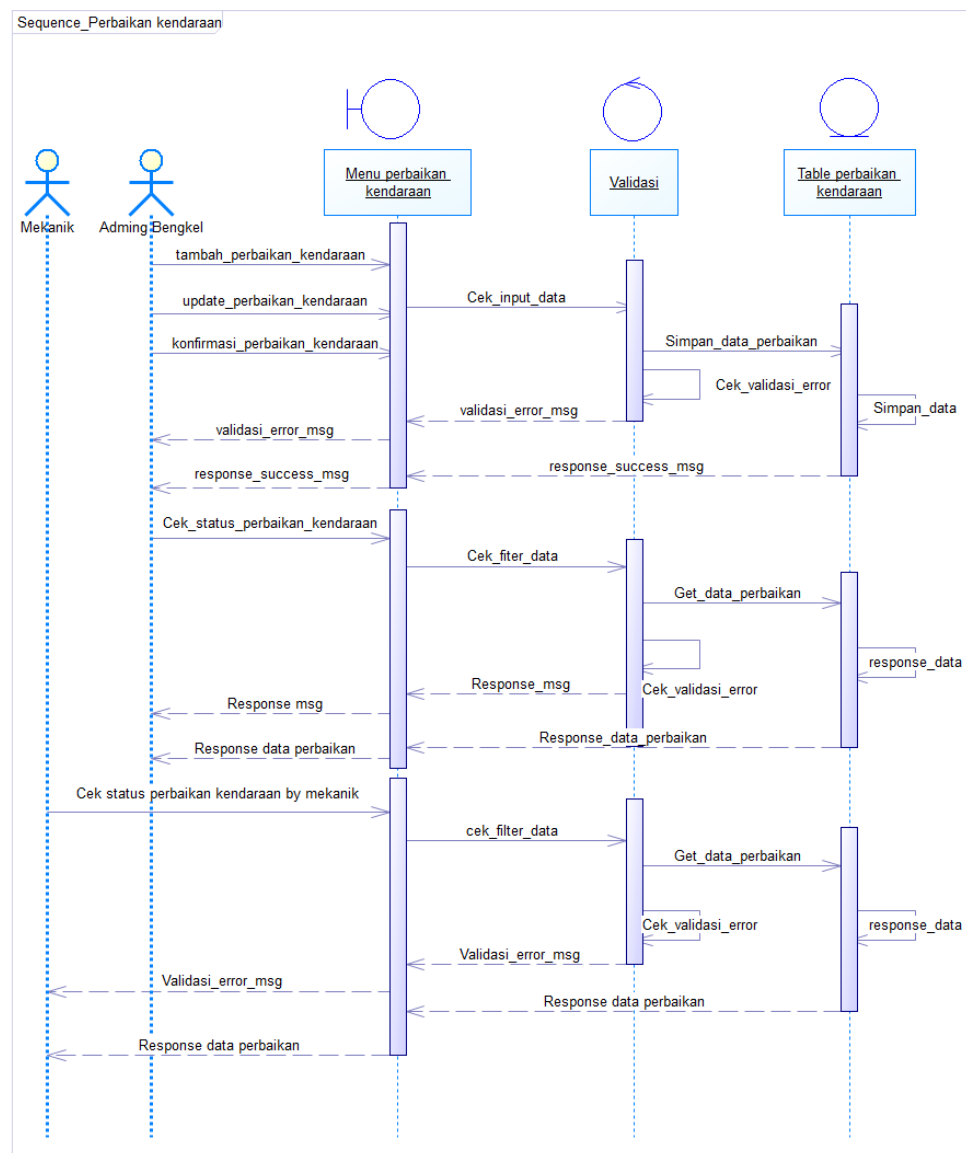
cadang. Sistem akan mengembalikan hasil *response data* berhasil disimpan dari database untuk ditampilkan di menu penerimaan suku cadang. Admin bengkel dapat melihat data penerimaan suku cadang yang berhasil disimpan.



Gambar 3. 22 Sequence Pengiriman Barang

Admin pengiriman membuka menu surat tugas pengiriman. Admin dapat melakukan tambah, update, dan konfirmasi. Sistem akan mengecek *input data* sebelum melanjutkan permintaan ke database tabel surat tugas pengiriman. Jika

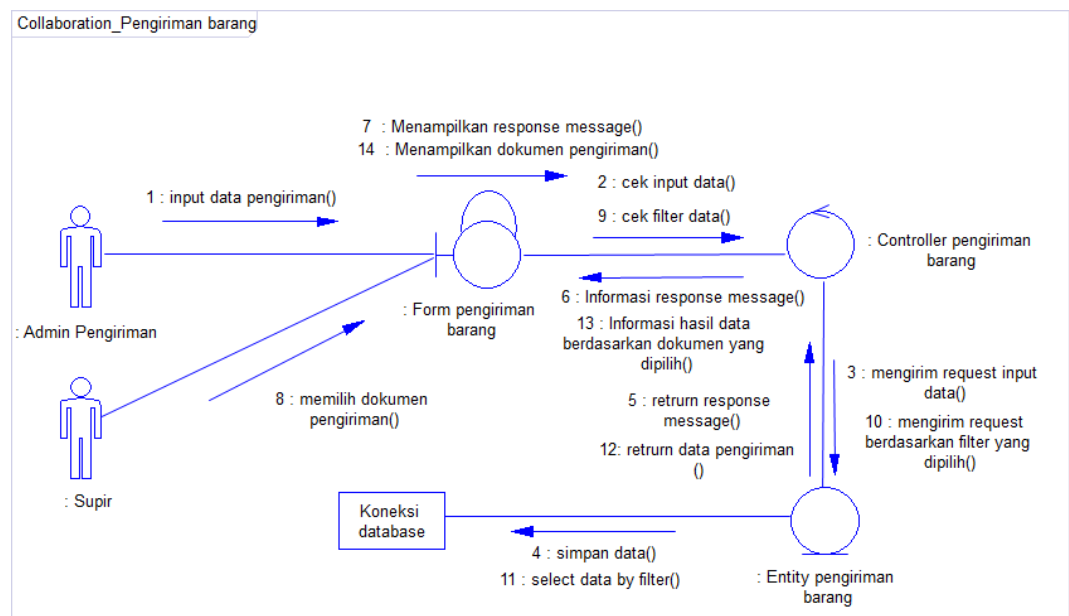
validasi cek *input data* berhasil maka akan mengirim permintaan untuk melakukan menyimpan, memperbaharui, maupun konfirmasi pada database tabel surat tugas pengiriman. Sistem akan mengembalikan hasil *response data* berhasil disimpan dari database untuk ditampilkan di menu surat tugas pengiriman. Admin pengiriman dapat melihat data surat tugas pengiriman yang berhasil disimpan. Supir juga dapat melihat data surat tugas pengiriman, untuk mengecek status surat tugas pengiriman yang dibawa oleh supir.



Gambar 3. 23 Sequence Perbaikan Kendaraan

Admin bengkel membuka menu perbaikan kendaraan. Admin dapat melakukan tambah, update, dan konfirmasi. Sistem akan mengecek *input data* sebelum melanjutkan permintaan ke database tabel perbaikan kendaraan. Jika validasi cek *input data* berhasil maka akan mengirim permintaan untuk melakukan menyimpan, memperbaharui, maupun konfirmasi pada database tabel perbaikan kendaraan. Sistem akan mengembalikan hasil *response data* berhasil disimpan dari database untuk ditampilkan di menu perbaikan kendaraan. Admin bengkel dapat melihat data perbaikan kendaraan yang berhasil disimpan. Mekanik juga dapat melihat data perbaikan kendaraan, untuk mengecek status perbaikan kendaraan yang dibawa oleh supir.

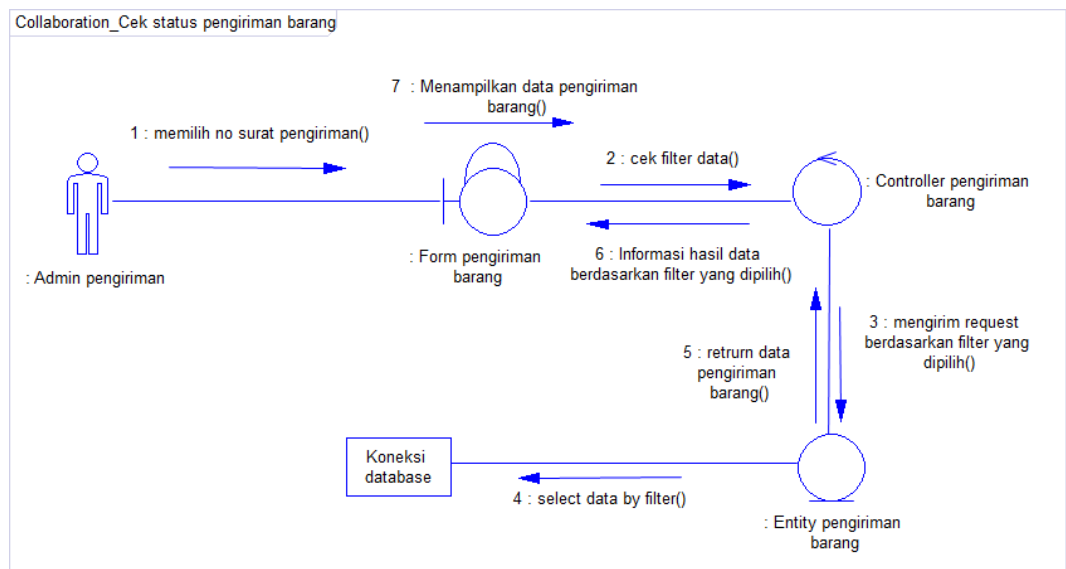
e. Collaboration diagram



Gambar 3. 24 collaboration diagram pengiriman barang

Admin pengiriman melakukan input data pengiriman pada *form* pengiriman barang. Inputa akan dikirim ke *controller* pengiriman barang untuk dilakukan pengecekan input data. Jika pengecekan berhasil selanjutnya inputan data dilanjutkan ke *entitiy* pengiriman barang untuk

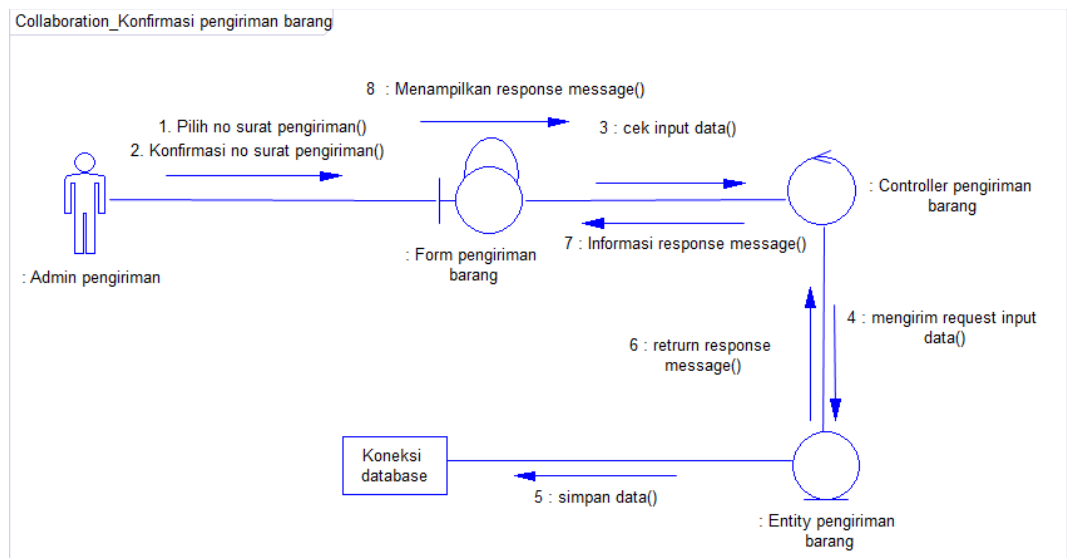
selanjutnya data inputan tersebut disimpan ke database. Data yang berhasil disimpan *entitiy* pengiriman barang akan mengembalikan *response message* dan dikirim ke *controller* pengiriman barang. *Controller* pengiriman barang menampilkan hasil *response message* pada *form* pengiriman barang. Supir dapat melihat data pengiriman barang yang berhasil disimpan sebelumnya. *Controller* akan mengecek *filter data*, jika pengecekan berhasil selanjutnya *controller* akan mengirim *request filter data* ke *entity* pengiriman barang untuk menampilkan data sesuai dengan *filter*. *Entitiy* pengiriman barang mengirim hasil *response data* ke *controller* pengiriman barang. *Controller* pengiriman barang akan menampilkan hasil *response data* ke menu pengiriman barang supaya dapat dilihat oleh supir.



Gambar 3. 25 collaboration diagram cek status pengiriman barang

Admin pengiriman dapat melihat data status pengiriman barang yang berhasil disimpan sebelumnya dengan cara memilih no surat pengiriman lalu

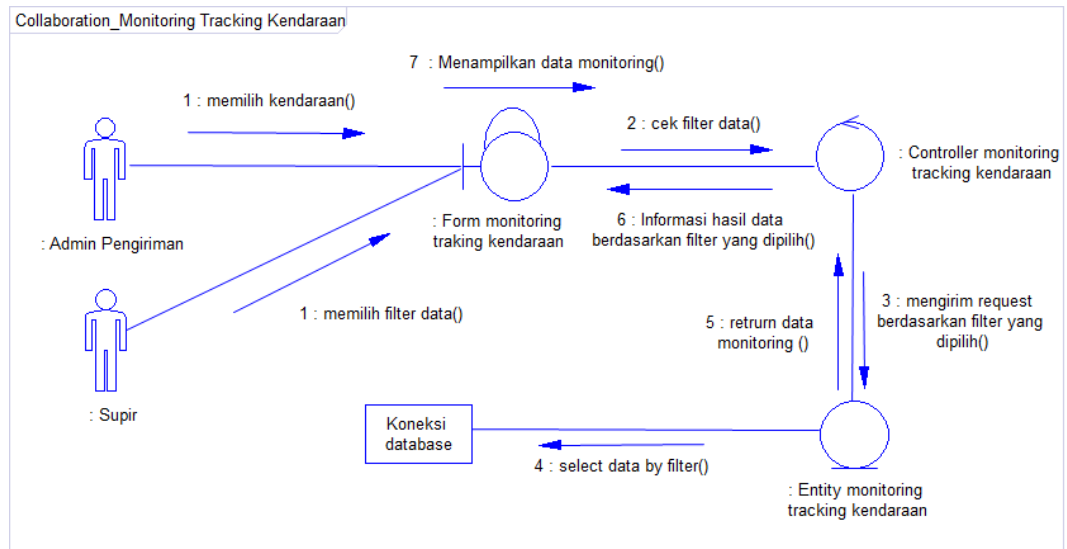
sistem akan memproses *request* untuk menampilkan data status no surat pengiriman. Controller akan mengecek *filter data*, jika pengecekan berhasil selanjutnya *controller* akan mengirim *request filter data* ke *entity* pengiriman barang untuk menampilkan data sesuai dengan *filter*. Entitit pengiriman barang mengirim hasil *response data* ke *controller* pengiriman barang. *Controller* pengiriman barang akan menampilkan hasil *response data* ke menu pengiriman barang.



Gambar 3. 26 collaboration diagram konfirmasi pengiriman barang

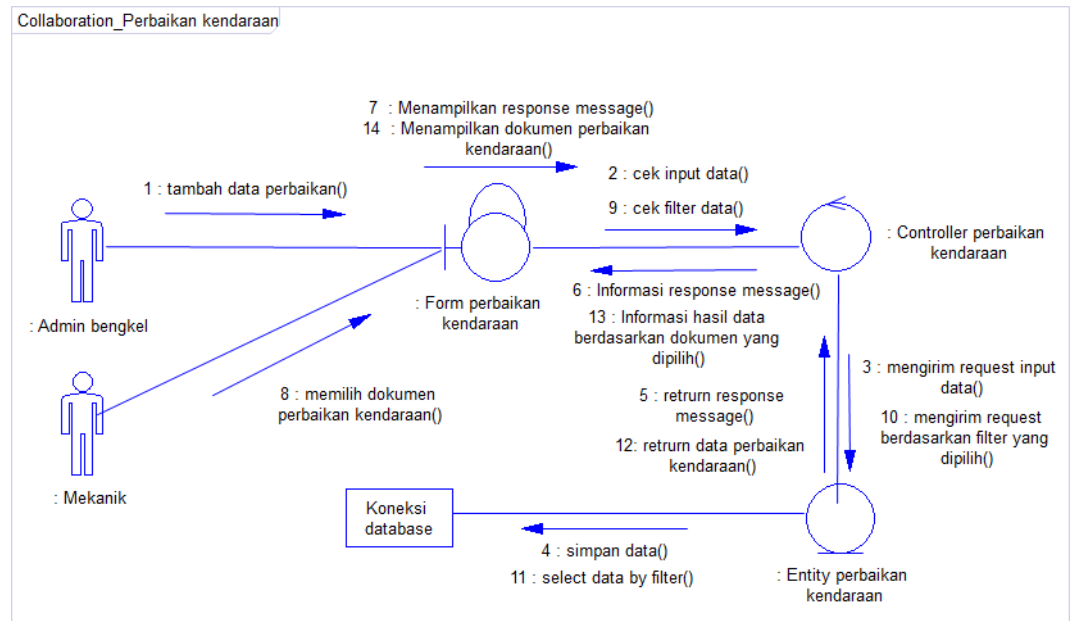
Admin pengiriman memilih no surat pengiriman lalu melakukan konfirmasi pengiriman pada *form* pengiriman barang. Input data akan dikirim ke *controller* pengiriman barang untuk dilakukan pengecekan input data. Jika pengecekan berhasil selanjutnya inputan data dilanjutkan ke *entitit* pengiriman barang untuk selanjutnya data inputan tersebut disimpan ke database. Data yang berhasil disimpan *entitit* pengiriman barang akan mengembalikan *response message* dan dikirim ke *controller* pengiriman

barang. *Controller* pengiriman barang menampilkan hasil *response message* pada *form* pengiriman barang.



Gambar 3. 27 collaboration diagram monitoring tracking kendaraan

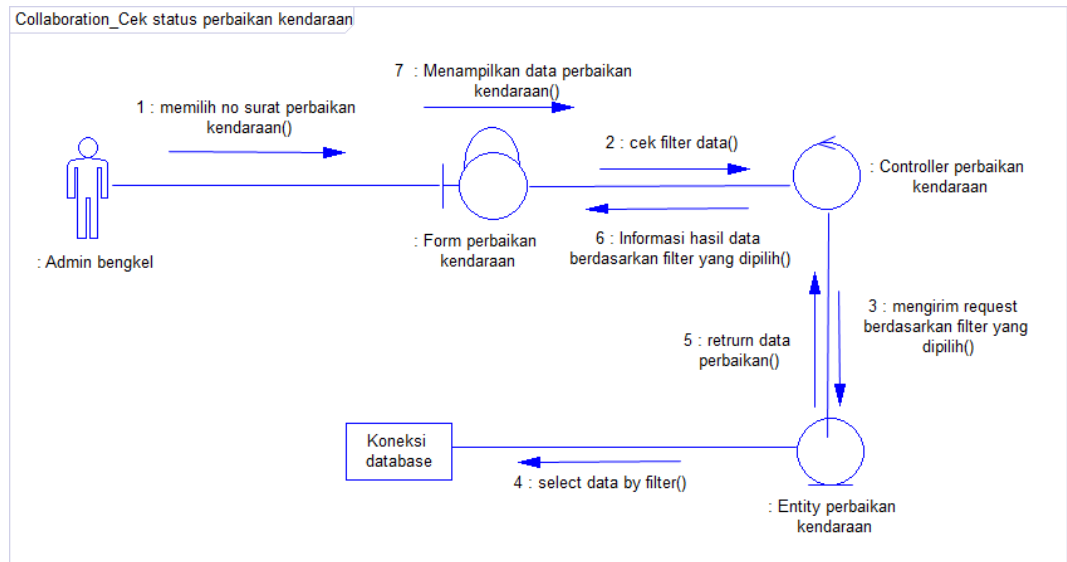
Admin pengiriman dan supir memilih kendaraan lalu sistem akan memproses *request* untuk menampilkan data *tracking* kendaraan. Controller akan mengecek *filter data*, jika pengecekan berhasil selanjutnya *controller* akan mengirim *request filter data* ke *entity monitoring tracking* kendaraan untuk menampilkan data sesuai dengan *filter*. *Entitiy monitoring tracking* kendaraan mengirim hasil *response data* ke *controller monitoring tracking* kendaraan. *Controller monitoring tracking* kendaraan akan menampilkan hasil *response data* tentang *tracking* kendaraan ke menu *monitoring tracking* kendaraan.



Gambar 3. 28 collaboration diagram perbaikan kendaraan

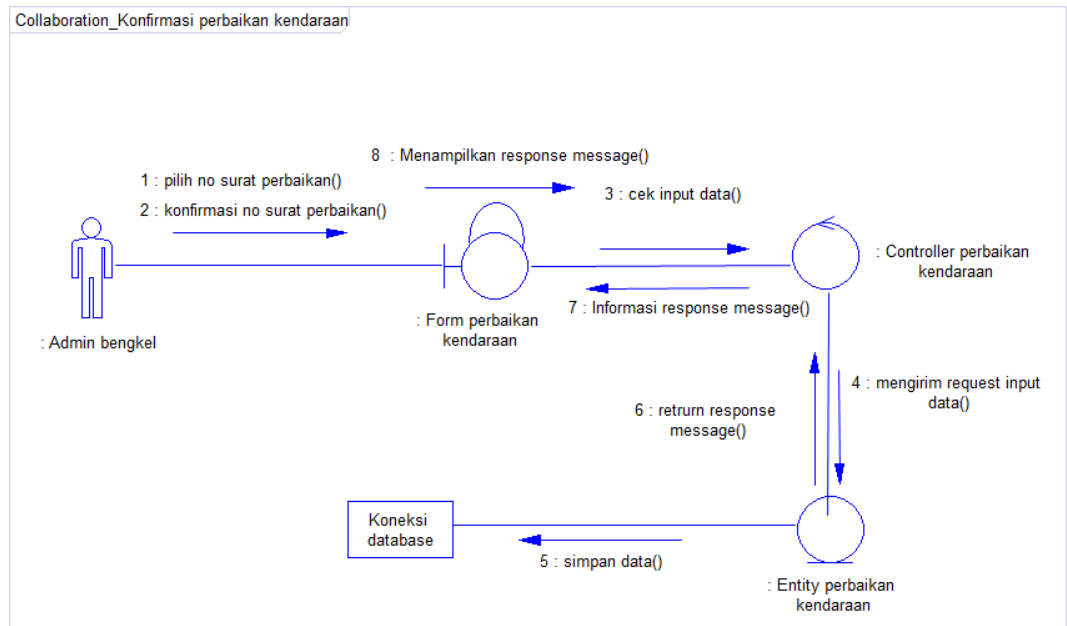
Admin bengkel melakukan input data perbaikan pada *form* perbaikan kendaraan. Input data akan dikirim ke *controller* perbaikan kendaraan untuk dilakukan pengecekan input data. Jika pengecekan berhasil selanjutnya inputan data dilanjutkan ke *entitiy* perbaikan kendaraan untuk selanjutnya data inputan tersebut disimpan ke database. Data yang berhasil disimpan *entitiy* pengiriman barang akan mengembalikan *response message* dan dikirim ke *controller* perbaikan kendaraan. *Controller* perbaikan kendaraan menampilkan hasil *response message* pada *form* perbaikan kendaraan. Supir dapat melihat data perbaikan kendaraan yang berhasil disimpan sebelumnya. *Controller* akan mengecek *filter data*, jika pengecekan berhasil selanjutnya *controller* akan mengirim *request filter data* ke *entitiy* perbaikan kendaraan untuk menampilkan data sesuai dengan *filter*. *Entitiy* perbaikan kendaraan mengirim hasil *response data* ke *controller* perbaikan kendaraan. *Controller* perbaikan kendaraan akan

menampilkan hasil *response data* ke menu perbaikan kendaraan supaya dapat dilihat oleh supir.



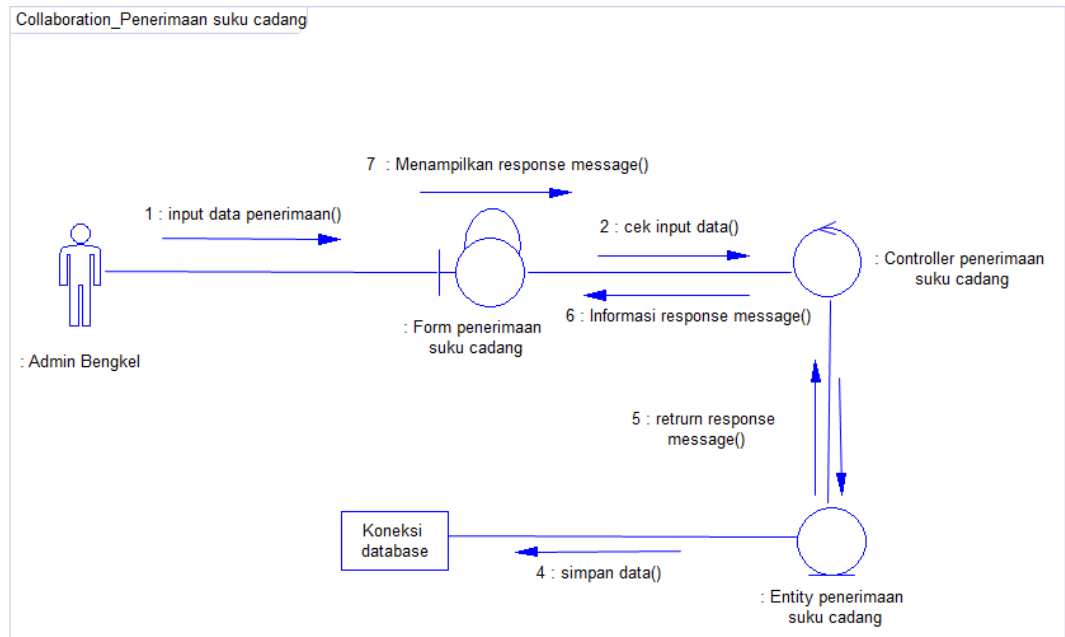
Gambar 3. 29 collaboration diagram cek status perbaikan kendaraan

Admin bengkel dapat melihat data status perbaikan kendaraan yang berhasil disimpan sebelumnya dengan cara memilih no surat perbaikan kendaraan lalu sistem akan memproses *request* untuk menampilkan data status no surat perbaikan kendaraan. Controller akan mengecek *filter data*, jika pengecekan berhasil selanjutnya *controller* akan mengirim *request filter data* ke *entity* perbaikan kendaraan untuk menampilkan data sesuai dengan *filter*. Entitiy perbaikan kendaraan mengirim hasil *response data* ke *controller* perbaikan kendaraan. *Controller* perbaikan kendaraan akan menampilkan hasil *response data* ke menu perbaikan kendaraan.



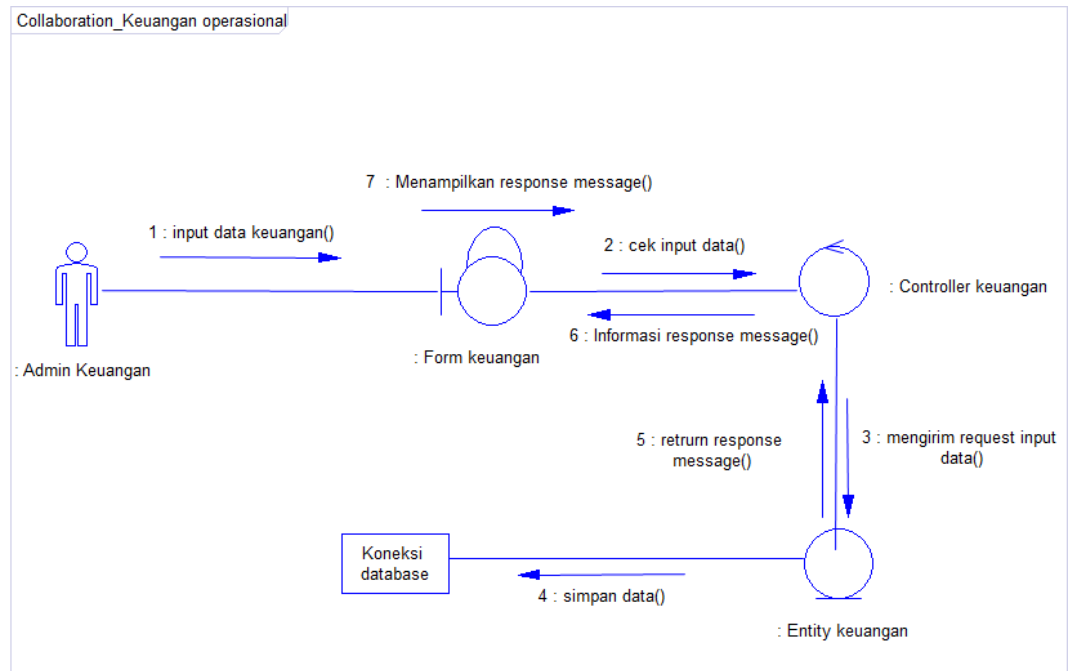
Gambar 3. 30 collaboration diagram konfirmasi perbaikan kendaraan

Admin bengkel memilih no surat perbaikan kendaraan lalu melakukan konfirmasi perbaikan kendaraan pada *form* perbaikan kendaraan. Input data akan dikirim ke *controller* perbaikan kendaraan untuk dilakukan pengecekan input data. Jika pengecekan berhasil selanjutnya inputan data dilanjutkan ke *entitiy* perbaikan kendaraan untuk selanjutnya data inputan tersebut disimpan ke database. Data yang berhasil disimpan *entitiy* perbaikan kendaraan akan mengembalikan *response message* dan dikirim ke *controller* perbaikan kendaraan. *Controller* perbaikan kendaraan menampilkan hasil *response message* pada *form* perbaikan kendaraan.



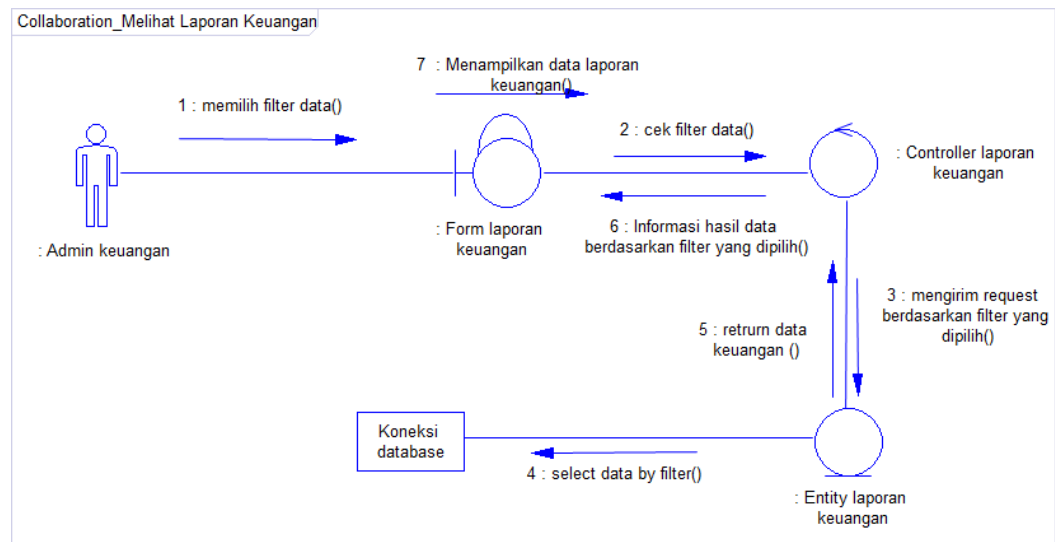
Gambar 3. 31 collaboration diagram penerimaan suku cadang

Admin bengkel melakukan input data penerimaan suku cadang pada *form* penerimaan suku cadang. Input data akan dikirim ke *controller* penerimaan suku cadang untuk dilakukan pengecekan input data. Jika pengecekan berhasil selanjutnya inputan data dilanjutkan ke *entitiy* penerimaan suku cadang untuk selanjutnya data inputan tersebut disimpan ke database. Data yang berhasil disimpan *entitiy* penerimaan suku cadang akan mengembalikan *response message* dan dikirim ke *controller* penerimaan suku cadang. *Controller* penerimaan suku cadang menampilkan hasil *response message* pada *form* penerimaan suku cadang.



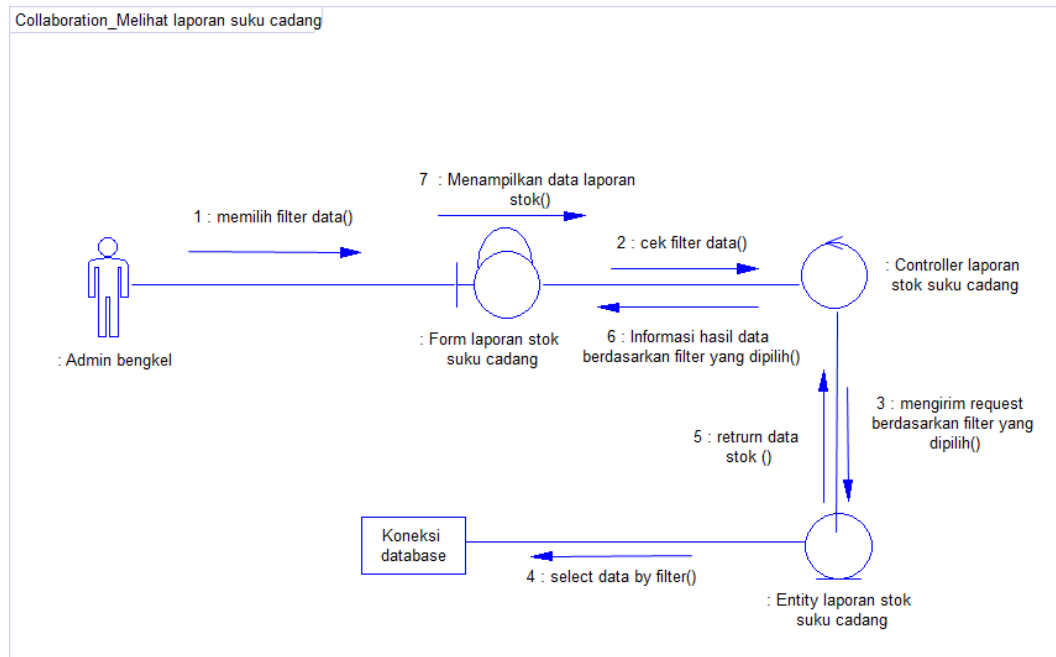
Gambar 3. 32 collaboration diagram keuangan operasional

Admin bengkel melakukan input data keuangan pada *form* keuangan. Input data akan dikirim ke *controller* keuangan untuk dilakukan pengecekan input data. Jika pengecekan berhasil selanjutnya inputan data dilanjutkan ke *entitiy* keuangan untuk selanjutnya data inputan tersebut disimpan ke database. Data yang berhasil disimpan *entitiy* keuangan cadang akan mengembalikan *response message* dan dikirim ke *controller* keuangan. *Controller* keuangan menampilkan hasil *response message* pada *form* keuangan.



Gambar 3. 33 collaboration diagram melihat laporan keuangan

Admin keuangan dapat melihat data laporan keuangan dengan cara admin memilih filter data terlebih dahulu. *Controller* akan mengecek *filter data*, jika pengecekan berhasil selanjutnya *controller* akan mengirim *request filter data* ke *entity* laporan keuangan untuk menampilkan data sesuai dengan *filter*. *Entitiy* laporan keuangan mengirim hasil *response data* ke *controller* laporan keuangan. *Controller* laporan keuangan akan menampilkan hasil *response data* ke menu laporan keuangan supaya laporan dapat dilihat oleh admin keuangan.



Gambar 3. 34 collaboration diagram melihat laporan suku cadang

Admin bengkel dapat melihat data laporan stok suku cadan dengan cara admin memilih filter data terlebih dahulu. *Controller* akan mengecek *filter data*, jika pengecekan berhasil selanjutnya *controller* akan mengirim *request filter data* ke *entity* laporan suku cadang untuk menampilkan data sesuai dengan *filter*. *Entitiy* laporan suku cadang mengirim hasil *response data* ke *controller* laporan suku cadang. *Controller* laporan suku cadang akan menampilkan hasil *response data* ke menu laporan suku cadang supaya laporan dapat dilihat oleh admin bengkel.

f. Tabel pengujian

Tabel 3. 12 Tabel pengujian menu biaya operasional

| No | Deskripsi | Hasil yang diharapkan | status |
|----|---|---|--------|
| 1 | Memasukkan tipe biaya, tanggal dokumen, keterangan, nomer akun dan nominal selanjutnya tekan tombol simpan. | Sistem akan menyimpan data yang sudah dimasukkan. | |
| 2 | Tidak mengisi data tipe biaya, tanggal dokumen, keterangan, nomer akun dan nominal selanjutnya tekan tombol simpan. | Sistem akan menolak karena data yang tidak diisi merupakan data yang wajib diisi | |
| 3 | Mengisi data tidak sesuai dengan tipe data yang sudah ditentukan. | Sistem akan menolak karena data yang diinputkan harus sesuai dengan tipe data yang sudah ditentukan | |

Tabel 3. 13 Tabel pengujian menu perbaikan kendaraan

| No | Deskripsi | Hasil yang diharapkan | status |
|----|---|--|--------|
| 1 | Memasukkan tanggal dokumen, keterangan, suku cadang, nopol kendaraan, dan nominal selanjutnya tekan tombol simpan. | Sistem akan menyimpan data yang sudah dimasukkan. | |
| 2 | Tidak mengisi tanggal dokumen, keterangan, suku cadang, nopol kendaraan, dan nominal selanjutnya tekan tombol simpan. | Sistem akan menolak karena data yang tidak diisi merupakan data yang wajib diisi | |

| No | Deskripsi | Hasil yang diharapkan | status |
|----|---|---|--------|
| 3 | Mengisi data tidak sesuai dengan tipe data yang sudah ditentukan. | Sistem akan menolak karena data yang diinputkan harus sesuai dengan tipe data yang sudah ditentukan | |

Tabel 3. 14 Tabel pengujian menu penerimaan suku cadang

| No | Deskripsi | Hasil yang diharapkan | status |
|----|---|---|--------|
| 1 | Memasukkan tanggal dokumen, keterangan, suku cadang, dan jumlah suku cadang selanjutnya tekan tombol simpan. | Sistem akan menyimpan data yang sudah dimasukkan. | |
| 2 | Tidak mengisi tanggal dokumen, keterangan, suku cadang, nopol kendaraan, dan nominal selanjutnya tekan tombol simpan. | Sistem akan menolak karena data yang tidak diisi merupakan data yang wajib diisi | |
| 3 | Mengisi data tidak sesuai dengan tipe data yang sudah ditentukan. | Sistem akan menolak karena data yang diinputkan harus sesuai dengan tipe data yang sudah ditentukan | |

Tabel 3. 15 Tabel pengjian menu pengiriman barang

| No | Deskripsi | Hasil yang diharapkan | status |
|----|---|---|--------|
| 1 | Memasukkan tanggal dokumen, keterangan, pelanggan, nopol kendaraan, pengemudi, barang, dan jumlah barang selanjutnya tekan tombol simpan. | Sistem akan menyimpan data yang sudah dimasukkan. | |

| No | Deskripsi | Hasil yang diharapkan | status |
|----|--|---|--------|
| 2 | Tidak mengisi tanggal dokumen, keterangan, pelanggan, nopol kendaraan, pengemudi, barang, dan jumlah barang selanjutnya tekan tombol simpan. | Sistem akan menolak karena data yang tidak diisi merupakan data yang wajib diisi | |
| 3 | Mengisi data tidak sesuai dengan tipe data yang sudah ditentukan. | Sistem akan menolak karena data yang diinputkan harus sesuai dengan tipe data yang sudah ditentukan | |

Tabel 3. 16 Tabel menu tracking kendaraan

| No | Deskripsi | Hasil yang diharapkan | status |
|----|--|--|--------|
| 1 | Memasukkan filter nopol kendaraan. | Sistem akan menampilkan status kendaraan saat ini. | |
| 2 | Tidak memasukkan filter nopol kendaraan. | Sistem tidak akan menampilkan status kendaraan. | |

Tabel 3. 17 Tabel laporan biaya operasional

| No | Deskripsi | Hasil yang diharapkan | status |
|----|---|---|--------|
| 1 | Memasukkan filter tanggal dan tipe biaya operasional. | Sistem akan menampilkan data laporan biaya operasional. | |
| 2 | Tidak memasukkan filter tanggal biaya operasional. | Sistem tidak akan menampilkan data laporan biaya operasional. | |

Tabel 3. 18 Tabel menu laporan perbaikan kendaraan

| No | Deskripsi | Hasil yang diharapkan | status |
|----|---|---|--------|
| 1 | Memasukkan filter tanggal dan tipe perbaikan kendaraan. | Sistem akan menampilkan data laporan perbaikan kendaraan. | |
| 2 | Tidak memasukkan filter tanggal perbaikan kendaraan. | Sistem tidak akan menampilkan data laporan perbaikan kendaraan. | |

Tabel 3. 19 Tabel menu laporan stok suku cadang

| No | Deskripsi | Hasil yang diharapkan | status |
|----|------------------------------------|--|--------|
| 1 | Memasukkan filter tipe stok. | Sistem akan menampilkan data laporan stok suku cadang berdasarkan tipe stok. | |
| 2 | Tidak memasukkan filter tipe stok. | Sistem akan menampilkan semua data laporan stok suku cadang. | |

Sistem yang sudah dilakukan pengujian di lapangan oleh pengguna, akan dilakukan evaluasi untuk mencari kekurangan sistem. Jika terdapat kekurangan, pengguna akan memberikan umpan balik kepada tim proyek mengerjakan kekurangan tersebut sampai sistem sudah sempurna sesuai dengan kebutuhan pengguna.

3.5 Sprint Review

Sprint review adalah hasil dari setiap *sprint* yang telah selesai berupa menu sistem yang sudah berjalan sesuai dengan kebutuhan. Setelah menu yang dikerjakan sudah selesai, maka proses selanjutnya adalah melakukan pengujian. Saat dilakukan pengujian dan terdapat hal yang tidak sesuai

kebutuhan maka kegiatan sprint review juga terdapat kegiatan melakukan revisi sistem, jika terdapat bagian yang tidak sesuai dengan kebutuhan terbaru. Setelah proses pengujian dan revisi selesai maka langkah selanjutnya memperkirakan product backlog item yang perlu dikerjakan di sprint selanjutnya.

Tabel 3. 20 Menu biaya operasional

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|-------------------------|---|---|---|---|---|---|
| <i>Functionality</i> | <i>Suitability</i> | Apakah menu biaya operasional tersebut dapat melakukan fungsi yang sudah ditentukan? | | | | | |
| | <i>Accurateness</i> | Apakah hasil pengolahan data pada menu biaya operasional tersebut sesuai dengan yang diharapkan? | | | | | |
| | <i>Interoperability</i> | Apakah menu biaya operasional tersebut dapat berinteraksi dengan perangkat lunak lainnya? | | | | | |
| | <i>Security</i> | Apakah menu biaya operasional tersebut mampu mengantisipasi / mencegah akses yang tidak sah ? | | | | | |
| | <i>Compliance</i> | Apakah menu biaya operasional tersebut mengikuti aturan standar aplikasi atau regulasi peraturan yang berlaku ? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|--------------------|------------------------|---|---|---|---|---|---|
| <i>Reliability</i> | <i>Maturity</i> | Apakah perangkat lunak mampu menghindari dari kesalahan penerapan perangkat lunak ? | | | | | |
| | <i>Fault tolerance</i> | Apakah menu biaya operasional tersebut mampu mempertahankan tingkat kinerjanya dalam kasus kesalahan dari perangkat lunak dan perangkat keras ? | | | | | |
| | <i>Recoverability</i> | Dapatkah menu biaya operasional tersebut memulihkan data kembali jika terjadi kegagalan? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|------------------|--------------------------|---|---|---|---|---|---|
| <i>Usability</i> | <i>Understandability</i> | Apakah para pengguna menu biaya operasional tersebut dapat memahami cara menggunakan menu biaya operasional tersebut dengan mudah ? | | | | | |
| | <i>Learnability</i> | Apakah langkah-langkah operasional menu biaya operasional tersebut dapat dipelajari dengan mudah ? | | | | | |
| | <i>Operability</i> | Apakah menu biaya operasional tersebut dapat digunakan hanya dengan menggunakan sumber daya seadanya ? | | | | | |
| | <i>Attractiveness</i> | Apakah menu biaya operasional tersebut memiliki antar muka (interface) yang menarik ? | | | | | |

Tabel 3. 21 Menu perbaikan kendaraan

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|-------------------------|--|---|---|---|---|---|
| <i>Functionality</i> | <i>Suitability</i> | Apakah menu perbaikan kendaraan tersebut dapat melakukan fungsi yang sudah ditentukan? | | | | | |
| | <i>Accurateness</i> | Apakah hasil pengolahan data pada menu perbaikan kendaraan tersebut sesuai dengan yang diharapkan? | | | | | |
| | <i>Interoperability</i> | Apakah menu perbaikan kendaraan tersebut dapat berinteraksi dengan perangkat lunak lainnya? | | | | | |
| | <i>Security</i> | Apakah menu perbaikan kendaraan tersebut mampu mengantisipasi / mencegah akses yang tidak sah ? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|------------------------|---|---|---|---|---|---|
| <i>Functionality</i> | <i>Compliance</i> | Apakah menu perbaikan kendaraan tersebut mengikuti aturan standar aplikasi atau regulasi peraturan yang berlaku ? | | | | | |
| | <i>Maturity</i> | Apakah perangkat lunak mampu menghindari dari kesalahan penerapan perangkat lunak ? | | | | | |
| | <i>Fault tolerance</i> | Apakah menu perbaikan kendaraan tersebut mampu mempertahankan tingkat kinerjanya dalam kasus kesalahan dari perangkat lunak dan perangkat keras ? | | | | | |
| | <i>Recoverability</i> | Dapatkah menu perbaikan kendaraan tersebut memulihkan data kembali jika terjadi kegagalan? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|------------------|--------------------------|---|---|---|---|---|---|
| <i>Usability</i> | <i>Understandability</i> | Apakah para pengguna menu perbaikan kendaraan tersebut dapat memahami cara menggunakan menu perbaikan kendaraan tersebut dengan mudah ? | | | | | |
| | <i>Learnability</i> | Apakah langkah-langkah operasional menu perbaikan kendaraan tersebut dapat dipelajari dengan mudah ? | | | | | |
| | <i>Operability</i> | Apakah menu perbaikan kendaraan tersebut dapat digunakan hanya dengan menggunakan sumber daya seadanya ? | | | | | |
| | <i>Attractiveness</i> | Apakah menu perbaikan kendaraan tersebut memiliki antar muka (interface) yang menarik ? | | | | | |

Tabel 3. 22 Menu penerimaan suku cadang

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|-------------------------|---|---|---|---|---|---|
| <i>Functionality</i> | <i>Suitability</i> | Apakah menu penerimaan suku cadang tersebut dapat melakukan fungsi yang sudah ditentukan? | | | | | |
| | <i>Accurateness</i> | Apakah hasil pengolahan data pada menu penerimaan suku cadang tersebut sesuai dengan yang diharapkan? | | | | | |
| | <i>Interoperability</i> | Apakah menu penerimaan suku cadang tersebut dapat berinteraksi dengan perangkat lunak lainnya? | | | | | |
| | <i>Security</i> | Apakah menu penerimaan suku cadang tersebut mampu mengantisipasi / mencegah akses yang tidak sah ? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|------------------------|--|---|---|---|---|---|
| <i>Functionality</i> | <i>Compliance</i> | Apakah menu penerimaan suku cadang tersebut mengikuti aturan standar aplikasi atau regulasi peraturan yang berlaku ? | | | | | |
| <i>Reliability</i> | <i>Maturity</i> | Apakah perangkat lunak mampu menghindari dari kesalahan penerapan perangkat lunak ? | | | | | |
| | <i>Fault tolerance</i> | Apakah menu penerimaan suku cadang tersebut mampu mempertahankan tingkat kinerjanya dalam kasus kesalahan dari perangkat lunak dan perangkat keras ? | | | | | |
| | <i>Recoverability</i> | Dapatkah menu penerimaan suku cadang tersebut memulihkan data kembali jika terjadi kegagalan? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|------------------|--------------------------|---|---|---|---|---|---|
| <i>Usability</i> | <i>Understandability</i> | Apakah para pengguna menu penerimaan suku cadang tersebut dapat memahami cara menggunakan menu penerimaan suku cadang tersebut dengan mudah ? | | | | | |
| | <i>Learnability</i> | Apakah langkah-langkah operasional menu penerimaan suku cadang tersebut dapat dipelajari dengan mudah ? | | | | | |
| | <i>Operability</i> | Apakah menu penerimaan suku cadang tersebut dapat digunakan hanya dengan menggunakan sumber daya seadanya ? | | | | | |
| | <i>Attractiveness</i> | Apakah menu penerimaan suku cadang tersebut memiliki antar muka (interface) yang menarik ? | | | | | |

Tabel 3. 23 Menu Pengiriman Barang

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|-------------------------|--|---|---|---|---|---|
| <i>Functionality</i> | <i>Suitability</i> | Apakah menu pengiriman barang tersebut dapat melakukan fungsi yang sudah ditentukan? | | | | | |
| | <i>Accurateness</i> | Apakah hasil pengolahan data pada menu pengiriman barang tersebut sesuai dengan yang diharapkan? | | | | | |
| | <i>Interoperability</i> | Apakah menu pengiriman barang tersebut dapat berinteraksi dengan perangkat lunak lainnya? | | | | | |
| | <i>Security</i> | Apakah menu pengiriman barang tersebut mampu mengantisipasi / mencegah akses yang tidak sah ? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|------------------------|---|---|---|---|---|---|
| <i>Functionality</i> | <i>Compliance</i> | Apakah menu pengiriman barang tersebut mengikuti aturan standar aplikasi atau regulasi peraturan yang berlaku ? | | | | | |
| | <i>Maturity</i> | Apakah perangkat lunak mampu menghindari dari kesalahan penerapan perangkat lunak ? | | | | | |
| | <i>Fault tolerance</i> | Apakah menu pengiriman barang tersebut mampu mempertahankan tingkat kinerjanya dalam kasus kesalahan dari perangkat lunak dan perangkat keras ? | | | | | |
| | <i>Recoverability</i> | Dapatkah menu pengiriman barang tersebut memulihkan data kembali jika terjadi kegagalan? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|------------------|--------------------------|---|---|---|---|---|---|
| <i>Usability</i> | <i>Understandability</i> | Apakah para pengguna menu pengiriman barang tersebut dapat memahami cara menggunakan menu pengiriman barang tersebut dengan mudah ? | | | | | |
| | <i>Learnability</i> | Apakah langkah-langkah operasional menu pengiriman barang tersebut dapat dipelajari dengan mudah ? | | | | | |
| | <i>Operability</i> | Apakah menu pengiriman barang tersebut dapat digunakan hanya dengan menggunakan sumber daya seadanya ? | | | | | |
| | <i>Attractiveness</i> | Apakah menu pengiriman barang tersebut memiliki antar muka (interface) yang menarik ? | | | | | |

Tabel 3. 24 Menu Monitoring Kendaraan

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|-------------------------|--|---|---|---|---|---|
| <i>Functionality</i> | <i>Suitability</i> | Apakah menu traking kendaraan tersebut dapat melakukan fungsi yang sudah ditentukan? | | | | | |
| | <i>Accurateness</i> | Apakah hasil pengolahan data pada menu traking kendaraan tersebut sesuai dengan yang diharapkan? | | | | | |
| | <i>Interoperability</i> | Apakah menu traking kendaraan tersebut dapat berinteraksi dengan perangkat lunak lainnya? | | | | | |
| | <i>Security</i> | Apakah menu traking kendaraan tersebut mampu mengantisipasi / mencegah akses yang tidak sah ? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|------------------------|---|---|---|---|---|---|
| <i>Functionality</i> | <i>Compliance</i> | Apakah menu traking kendaraan tersebut mengikuti aturan standar aplikasi atau regulasi peraturan yang berlaku ? | | | | | |
| | <i>Maturity</i> | Apakah perangkat lunak mampu menghindari dari kesalahan penerapan perangkat lunak ? | | | | | |
| | <i>Fault tolerance</i> | Apakah menu traking kendaraan tersebut mampu mempertahankan tingkat kinerjanya dalam kasus kesalahan dari perangkat lunak dan perangkat keras ? | | | | | |
| | <i>Recoverability</i> | Dapatkah menu traking kendaraan tersebut memulihkan data kembali jika terjadi kegagalan? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|------------------|--------------------------|---|---|---|---|---|---|
| <i>Usability</i> | <i>Understandability</i> | Apakah para pengguna menu traking kendaraan tersebut dapat memahami cara menggunakan menu traking kendaraan tersebut dengan mudah ? | | | | | |
| | <i>Learnability</i> | Apakah langkah-langkah operasional menu traking kendaraan tersebut dapat dipelajari dengan mudah ? | | | | | |
| | <i>Operability</i> | Apakah menu traking kendaraan tersebut dapat digunakan hanya dengan menggunakan sumber daya seadanya ? | | | | | |
| | <i>Attractiveness</i> | Apakah menu traking kendaraan tersebut memiliki antar muka (interface) yang menarik ? | | | | | |

Tabel 3. 25 Menu laporan biaya operasional

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|-------------------------|--|---|---|---|---|---|
| <i>Functionality</i> | <i>Suitability</i> | Apakah menu laporan biaya operasional tersebut dapat melakukan fungsi yang sudah ditentukan? | | | | | |
| | <i>Accurateness</i> | Apakah hasil pengolahan data pada menu laporan biaya operasional tersebut sesuai dengan yang diharapkan? | | | | | |
| | <i>Interoperability</i> | Apakah menu laporan biaya operasional tersebut dapat berinteraksi dengan perangkat lunak lainnya? | | | | | |
| | <i>Security</i> | Apakah menu laporan biaya operasional tersebut mampu mengantisipasi / mencegah akses yang tidak sah ? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|------------------------|---|---|---|---|---|---|
| <i>Functionality</i> | <i>Compliance</i> | Apakah menu laporan biaya operasional tersebut mengikuti aturan standar aplikasi atau regulasi peraturan yang berlaku ? | | | | | |
| | <i>Maturity</i> | Apakah perangkat lunak mampu menghindari dari kesalahan penerapan perangkat lunak ? | | | | | |
| | <i>Fault tolerance</i> | Apakah menu laporan biaya operasional tersebut mampu mempertahankan tingkat kinerjanya dalam kasus kesalahan dari perangkat lunak dan perangkat keras ? | | | | | |
| | <i>Recoverability</i> | Dapatkah menu laporan biaya operasional tersebut memulihkan data kembali jika terjadi kegagalan? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|------------------|--------------------------|---|---|---|---|---|---|
| <i>Usability</i> | <i>Understandability</i> | Apakah para pengguna menu laporan biaya operasional tersebut dapat memahami cara menggunakan menu laporan biaya operasional tersebut dengan mudah ? | | | | | |
| | <i>Learnability</i> | Apakah langkah-langkah operasional menu laporan biaya operasional tersebut dapat dipelajari dengan mudah ? | | | | | |
| | <i>Operability</i> | Apakah menu laporan biaya operasional tersebut dapat digunakan hanya dengan menggunakan sumber daya seadanya ? | | | | | |
| | <i>Attractiveness</i> | Apakah menu laporan biaya operasional tersebut memiliki antar muka (interface) yang menarik ? | | | | | |

Tabel 3. 26 Menu laporan perbaikan kendaraan

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|-------------------------|--|---|---|---|---|---|
| <i>Functionality</i> | <i>Suitability</i> | Apakah menu laporan perbaikan kendaraan tersebut dapat melakukan fungsi yang sudah ditentukan? | | | | | |
| | <i>Accurateness</i> | Apakah hasil pengolahan data pada menu laporan perbaikan kendaraan tersebut sesuai dengan yang diharapkan? | | | | | |
| | <i>Interoperability</i> | Apakah menu laporan perbaikan kendaraan tersebut dapat berinteraksi dengan perangkat lunak lainnya? | | | | | |
| | <i>Security</i> | Apakah menu laporan perbaikan kendaraan tersebut mampu mengantisipasi / mencegah akses yang tidak sah ? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|------------------------|---|---|---|---|---|---|
| <i>Functionality</i> | <i>Compliance</i> | Apakah menu laporan perbaikan kendaraan tersebut mengikuti aturan standar aplikasi atau regulasi peraturan yang berlaku ? | | | | | |
| | <i>Maturity</i> | Apakah perangkat lunak mampu menghindari dari kesalahan penerapan perangkat lunak ? | | | | | |
| | <i>Fault tolerance</i> | Apakah menu laporan perbaikan kendaraan tersebut mampu mempertahankan tingkat kinerjanya dalam kasus kesalahan dari perangkat lunak dan perangkat keras ? | | | | | |
| | <i>Recoverability</i> | Dapatkah menu laporan perbaikan kendaraan tersebut memulihkan data kembali jika terjadi kegagalan? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|------------------|--------------------------|---|---|---|---|---|---|
| <i>Usability</i> | <i>Understandability</i> | Apakah para pengguna menu laporan perbaikan kendaraan tersebut dapat memahami cara menggunakan menu laporan perbaikan kendaraan tersebut dengan mudah ? | | | | | |
| | <i>Learnability</i> | Apakah langkah-langkah operasional menu laporan perbaikan kendaraan tersebut dapat dipelajari dengan mudah ? | | | | | |
| | <i>Operability</i> | Apakah menu laporan perbaikan kendaraan tersebut dapat digunakan hanya dengan menggunakan sumber daya seadanya ? | | | | | |
| | <i>Attractiveness</i> | Apakah menu laporan perbaikan kendaraan tersebut memiliki antar muka (interface) yang menarik ? | | | | | |

Tabel 3. 27 Menu laporan stok suku cadang

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|-------------------------|---|---|---|---|---|---|
| <i>Functionality</i> | <i>Suitability</i> | Apakah menu laporan stok suku cadang tersebut dapat melakukan fungsi yang sudah ditentukan? | | | | | |
| | <i>Accurateness</i> | Apakah hasil pengolahan data pada menu laporan stok suku cadang tersebut sesuai dengan yang diharapkan? | | | | | |
| | <i>Interoperability</i> | Apakah menu laporan stok suku cadang tersebut dapat berinteraksi dengan perangkat lunak lainnya? | | | | | |
| | <i>Security</i> | Apakah menu laporan stok suku cadang tersebut mampu mengantisipasi / mencegah akses yang tidak sah ? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|----------------------|------------------------|--|---|---|---|---|---|
| <i>Functionality</i> | <i>Compliance</i> | Apakah menu laporan stok suku cadang tersebut mengikuti aturan standar aplikasi atau regulasi peraturan yang berlaku ? | | | | | |
| | <i>Maturity</i> | Apakah perangkat lunak mampu menghindari dari kesalahan penerapan perangkat lunak ? | | | | | |
| | <i>Fault tolerance</i> | Apakah menu laporan stok suku cadang tersebut mampu mempertahankan tingkat kinerjanya dalam kasus kesalahan dari perangkat lunak dan perangkat keras ? | | | | | |
| | <i>Recoverability</i> | Dapatkah menu laporan stok suku cadang tersebut memulihkan data kembali jika terjadi kegagalan? | | | | | |

| Karakteristik | Sub karakteristik | Pertanyaan | 1 | 2 | 3 | 4 | 5 |
|------------------|--------------------------|---|---|---|---|---|---|
| <i>Usability</i> | <i>Understandability</i> | Apakah para pengguna menu laporan stok suku cadang tersebut dapat memahami cara menggunakan menu laporan stok suku cadang tersebut dengan mudah ? | | | | | |
| | <i>Learnability</i> | Apakah langkah-langkah operasional menu laporan stok suku cadang tersebut dapat dipelajari dengan mudah ? | | | | | |
| | <i>Operability</i> | Apakah menu laporan stok suku cadang tersebut dapat digunakan hanya dengan menggunakan sumber daya seadanya ? | | | | | |
| | <i>Attractiveness</i> | Apakah menu laporan stok suku cadang tersebut memiliki antar muka (interface) yang menarik ? | | | | | |

DAFTAR PUSTAKA

- [1] M. K. S. K. Janusz Grabara, "THE ROLE OF INFORMATION SYSTEMS IN TRANSPORT LOGISTICS," *THE ROLE OF INFORMATION SYSTEMS IN TRANSPORT LOGISTICS*, p. 2, 2014.
- [2] E. M. B. L. M. Q. K. M. Marouane El Midaoui, "Logistics tracking system based on decentralized IoT and blockchain platform," *Logistics tracking system based on decentralized IoT and blockchain platform*, p. 422, 2021.
- [3] M. Sattayathamrongthian, "Business's Transportation Management System Technology," *Business's Transportation Management System Technology*, p. 2450, 2022.
- [4] S. D. P. Kresna Dwi Prasetya, "Effectiveness Analysis of Distributed Scrum Model Compared to Waterfall approach in Third-Party Application Development," *Effectiveness Analysis of Distributed Scrum Model Compared to Waterfall approach in Third-Party Application Development*, p. 103, 2020.
- [5] P. R. Fidel Kussunga, "Proposal of a Visual Environment to Support Scrum," *Proposal of a Visual Environment to Support Scrum*, 2019.
- [6] Irwanto, "Perancangan Sistem Informasi Sekolah Kejuruan dengan Menggunakan Metode Waterfall (Studi Kasus SMK PGRI 1 Kota Serang-Banten)," *Perancangan Sistem Informasi Sekolah Kejuruan dengan Menggunakan Metode Waterfall (Studi Kasus SMK PGRI 1 Kota Serang-Banten)*, p. 88, 2021.
- [7] S. M. Sutiyono, "MEMBANGUN SISTEM INFORMASI PENDAFTARAN SISWA BARU," *MEMBANGUN SISTEM INFORMASI PENDAFTARAN SISWA BARU*, p. 51, 2020.
- [8] E. A. Adnan Buyung Nasution, "IMPLEMENTASI SISTEM INFORMASI QUALITY CONTROL PADA PRODUKSI GRANIT TILE BERBASIS WEB (STUDI KASUS PT. JUI SHIN INDONESIA)," *IMPLEMENTASI SISTEM INFORMASI QUALITY CONTROL PADA PRODUKSI GRANIT TILE BERBASIS WEB (STUDI KASUS PT. JUI SHIN INDONESIA)*, p. 38, 2017.
- [9] B. Reihansyah, "PENGEMBANGAN SISTEM INFORMASI RUMAH SAKIT BADAN LAYANAN," *PENGEMBANGAN SISTEM INFORMASI RUMAH SAKIT BADAN LAYANAN*, p. 5, 2019.
- [10] S. Sachdeva, "Scrum Methodology," *International Journal Of Engineering And Computer Science*, p. 9, 2016.
- [11] M. T. S. M. Shon Hadji, "IMPLEMENTASI METODE SCRUM PADA PENGEMBANGAN APLIKASI DELIVERY ORDER BERBASIS WEBSITE (STUDI KASUS PADA RUMAH MAKAN LOMBOK IDJO SEMARANG)," *KONFERENSI ILMIAH MAHASISWA UNISSULA*, p. 12, 2019.

- [12] M. Blom, "Is Scrum and XP suitable for CSE Development?," *Is Scrum and XP suitable for CSE Development?*, p. 1512, 2012.
- [13] S. Al-Fedaghi, "Diagramming the Class Diagram: Toward a Unified Modeling Methodology," *Diagramming the Class Diagram: Toward a Unified Modeling Methodology*, p. 30, 2017.
- [14] Hamayoon Ghafory, "The Effectiveness of use case And Activity Diagram on Software Requirement Inspection: An Experimental Study," *The Effectiveness of use case And Activity Diagram on Software Requirement Inspection: An Experimental Study*, p. 36, 2018.
- [15] A. Hafeez, "Importance and Impact of Class Diagram in Software Development," *Importance and Impact of Class Diagram in Software Development*, p. 3, 2019.
- [16] P. D. A. Pamungkas, "ISO 9126 untuk Pengujian Kualitas Aplikasi Perpustakaan Senayan Library Management System (SLiMS)," *ISO 9126 untuk Pengujian Kualitas Aplikasi Perpustakaan Senayan Library Management System (SLiMS)*, p. 466, 2018.
- [17] Padayachee, "ISO 9126 external systems quality characteristics, sub-characteristics and domain specific criteria for evaluating e-Learning systems," *ISO 9126 external systems quality characteristics, sub-characteristics and domain specific criteria for evaluating e-Learning systems*, p. 2, 2010.
- [18] Z. Mansor, "Empirical Study of Cost Management Success Determinants in Agile Based Software Development Project: A Rasch Measurement Model Analysis," *Empirical Study of Cost Management Success Determinants in Agile Based Software Development Project: A Rasch Measurement Model Analysis*, p. 130, 2013.
- [19] D. Taluke, "ANALISIS PREFERENSI MASYARAKAT DALAM PENGELOLAAN EKOSISTEM MANGROVE DI PESISIR PANTAI KECAMATAN LOLODA KABUPATEN HALMAHERA BARAT," *ANALISIS PREFERENSI MASYARAKAT DALAM PENGELOLAAN EKOSISTEM MANGROVE DI PESISIR PANTAI KECAMATAN LOLODA KABUPATEN HALMAHERA BARAT*, p. 534, 2019.
- [20] D. S. S. A. H. A. P. A. S. Fadhila Cahya Ningrum, "Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik," *Jurnal Informatika Universitas Pamulang*, p. 6, 2019.
- [21] T. S. Jaya, "Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis," *Jurnal Pengembangan IT (JPIT)*, p. 4, 2018.
- [22] W. A. L. B. J. M. S. C. Edwin Arturo Quintero Torres, "Design of an environmental management information system for the Universidad Distrita," *Design of an environmental management information system for the Universidad Distrita*, p. 530, 2021.

- [23] J. Heimicke, "A framework for generating agile methods for product development," *A framework for generating agile methods for product development*, p. 786, 2021.
- [24] T. J. G. Mahnaz Afshari, "Quality of agile adoption in global software development: An assessment model," *Quality of agile adoption in global software development: An assessment model*, p. 368, 2020.
- [25] I. Mahendra, "AGILE DEVELOPMENT METHODS DALAM PENGEMBANGAN," *AGILE DEVELOPMENT METHODS DALAM PENGEMBANGAN*, p. 13, 2018.