

**PENGEMBANGAN SISTEM INFORMASI POINT OF SALES
UNTUK BENGKEL BERBASIS CLOUD COMPUTING
(STUDI KASUS : BENGKEL MAS PUR BATURAJA)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Bella Hersy Lavenia

NIM: 165150700111001



**PROGRAM STUDI TEKNOLOGI INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2020

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI *POINT OF SALES* UNTUK BENGKEL BERBASIS *CLOUD COMPUTING* (STUDI KASUS: BENGKEL MAS PUR BATURAJA)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun Oleh :
Bella Hersy Lavenia
NIM: 165150707111001

Skripsi ini telah diuji dan dinyatakan lulus pada
7 Januari 2020
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Widhy Hayuhardhika N.P., S.Kom., M.Kom.
NIK: 2017128704092001

Dosen Pembimbing II

Buce Trias H., S.Kom., M.Kom.
NIP: 198904262019031009

Mengetahui



Ketua Jurusan Sistem Informasi

Dr. Eng. Herman Tolle, S.T., M.T.
NIP: 197408232000121001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70)

Malang, 7 Januari 2020



Penulis

Email: bellahersylavenia@outlook.com



PRAKATA

Dengan menyebut nama Allah SWT Yang Maha Pengasih dan Penyayang karena berkat Rahmat, hidayah dan Karunia-Nya yang diberikan sehingga penulis dapat menyelesaikan skripsi berjudul “Pengembangan Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* (Studi Kasus: Bengkel Mas Pur Baturaja)”.

Dalam penyusunan dan penulisan naskah skripsi ini penulis mendapatkan banyak bantuan, bimbingan serta dukungan dari berbagai pihak baik secara moril maupun materil, sehingga saya ingin menyampaikan rasa terima kasih kepada:

1. Kedua orang tua dan keluarga yang selalu memberikan dukungan doa, moril, materil dan kasih sayang sehingga penulis mampu melaksanakan skripsi dengan lancar.
2. Bapak Widhy Hayuhardhika Nugraha Putra., S.Kom., M.Kom. selaku pembimbing 1 dan bapak Buce Trias Hanggara, S.Kom., M.Kom. selaku pembimbing 2 yang telah memberi bimbingan, arahan dan saran kepada penulis dalam proses pengerjaan skripsi dari awal hingga akhir.
3. Bapak Issa Arwani, S.Kom., M.Sc. selaku Ketua Program Studi Sistem Informasi.
4. Bapak dan Ibu Dosen Fakultas Ilmu Komputer, Universitas Brawijaya yang telah memberikan ilmu kepada penulis selama proses perkuliahan hingga penulis mampu menyelesaikan skripsi.
5. Yuninda, Novia, Kartika, Retno, Roisul, Arizal, Farkhan, Kevin, mbak Melly, dan mbak Serly selaku sahabat-sahabat yang telah membantu penulis dalam menjalani masa penyusunan skripsi.
6. Teman-teman Program Studi Teknologi Informasi Fakultas Ilmu Komputer Universitas Brawijaya yang tidak dapat saya sebutkan namanya satu per satu, yang selalu memberikan semangat, dorongan, bantuan pikiran dan menjadi teman yang baik dan menyenangkan selama masa perkuliahan.
7. Seluruh pihak yang telah membantu kelancaran penulisan skripsi yang tidak dapat disebutkan satu persatu.

Semoga Allah SWT melimpahkan rahmat berlipat ganda kepada seluruh pihak yang membantu dan berkontribusi pada proses penelitian ini. Tidak lupa saya sampaikan bahwa penelitian yang telah dilakukan memiliki banyak kekurangan, diskusi mengenai penelitian lanjutan, saran, dan kritik yang membangun merupakan beberapa hal yang saya harapkan dapat disampaikan kepada saya.

Malang, 7 Januari 2020

Penulis

Email: bellahersylavenia@outlook.com

ABSTRAK

Bella Hersy Lavenia, Pengembangan Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* (Studi Kasus: Bengkel Mas Pur Baturaja)

Pembimbing: Widhy Hayuhardhika Nugraha Putra, S.Kom., M.Kom. dan Buce Trias Hanggara, S.Kom., M.Kom.

Bengkel Mas Pur Baturaja merupakan perusahaan yang bergerak dibidang jasa perbaikan badan kendaraan khususnya roda empat. Sejak awal berdiri perusahaan ini menggunakan pengolahan data manual yang mengharuskan setiap transaksi dicatat dan dibukukan. Dalam pengelolaannya, perusahaan ini memiliki permasalahan yaitu proses transaksi sering kali tidak tercatat sehingga mengakibatkan tidak lengkapnya data-data perusahaan. Dari permasalahan tersebut, dibuatlah sebuah sistem informasi *point of sales* untuk bengkel. Sistem informasi *point of sales* untuk bengkel merupakan sistem informasi berbasis *cloud computing* yang dapat diakses oleh pengguna melalui penelusur web, pengguna akan dengan mudah merekap dan mencatat setiap transaksi penjualan dan pembelian yang dilakukan perusahaan. Pengembangan sistem informasi *point of sales* untuk bengkel menggunakan *framework* CodeIgniter, bahasa pemrograman PHP, HTML, CSS, *javascript* dan basis data MySQL, untuk implementasi *cloud computing* akan digunakan jenis layanan *Infrastructure as a Service (IaaS)* yaitu layanan AWS EC2 dari *Amazon Web Service* dengan jenis *instance t2.micro* dan sistem operasi *Ubuntu Server 16.04*. Pengujian yang dilakukan pada sistem ini akan menggunakan pengujian fungsional dan pengujian non-fungsional. Pengujian fungsional pada sistem menghasilkan 15 kasus uji dengan 14 dinyatakan valid sesuai dengan kebutuhan pengguna. Pengujian non-fungsional akan menguji kompatibilitas sistem terhadap beberapa penelusur web dengan menggunakan perangkat lunak *sorbsite* menghasilkan 2 *critical issues* pada *chrome* dan *firefox*, 2 *major issues* dan 10 *minor issues*. Berdasarkan hasil pengujian yang telah dilakukan disimpulkan bahwa sistem informasi *point of sales* untuk bengkel berbasis *cloud computing* layak untuk digunakan di Bengkel Mas Pur Baturaja.

Kata Kunci: *point of sales, cloud computing, Infrastructure as a Service, CodeIgniter, PHP, MySQL, AWS EC2.*

ABSTRACT

Bella Hersy Lavenia, Development of Point of Sales Information System for Car Services Shop Cloud Computing Based (Case Study: Bengkel Mas Pur Baturaja)

Mentors: Widhy Hayuhardhika Nugraha Putra, S.Kom., M.Kom. and Buce Trias Hanggara, S.Kom., M.Kom.

Bengkel Mas Pur Baturaja is a company that focuses on vehicle body repair services, especially for four-wheel vehicle. In the very beginning, Bengkel Mas Pur Baturaja uses manual data processing which requires that every transaction must be recorded. In this management, this company has many problem including unregistered transactions and incomplete transaction data. From these problems, a point of sales information system for car services shop was made. Point of sales information system for car services shop is a cloud computing-based system that can be accessed by users through a web browser, users will be easily to record every transaction by the company. The development of point of sales information systems for a workshops is using the CodeIgniter framework, a programming languages like PHP, HTML, CSS, Javascript and used MySQL as database. The implementation for cloud computing will be used Infrastructure service (IaaS) as service type that is AWS EC2 from Amazon Web Services as a cloud provider and the instance is t2.micro with Ubuntu Server 16.04 as operating system. Tests carried out on this system will use functional testing and non-fuctional testing. Functional testing on the system has 15 test cases with 14 declared valid according to user requirements. Non-functional testing will test the system's compatibility with some web browser using sortsite software to produce 2 critical issues on Chrome and Firefox, 2 major issues and 10 minor issues. Based on the results of tests that have been conducted, it has been concluded that the point of sales information system for car services shop cloud computing-based is feasible to be used in Bengkel Mas Pur Baturaja.

Keywords: *point of sales, cloud computing, Infrastructure as a Service, CodeIgniter, PHP, MySQL, AWS EC2.*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Manfaat	2
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Bengkel Mas Pur Baturaja	6
2.3 Visi dan Misi	7
2.3.1 Visi	7
2.3.2 Misi	7
2.4 Sistem Informasi	7
2.5 Point of Sales	8
2.6 Analisis dan Desain Sistem	8
2.7 Pengembangan Perangkat Lunak	9
2.7.1 System Development Life Cycle	9
2.7.2 Waterfall Model	12
2.8 Business Process Model and Notation	13
2.8.1 Flow Object	13



2.8.2 Data	15
2.8.3 Connecting Object	15
2.8.4 Swimlanes	16
2.8.5 Artifacts	16
2.9 Object Oriented Analysis and Design (OOAD)	16
2.10 Unified Modelling Language (UML)	17
2.10.1 Use case Diagram	17
2.10.2 Class Diagram	18
2.10.3 Activity Diagram	19
2.10.4 Sequence Diagram	20
2.11 Physical Data Model	21
2.12 Implementasi	22
2.12.1 PHP	22
2.12.2 MVC	23
2.12.3 MySQL	23
2.13 CodeIgniter	23
2.14 Cloud Computing	24
2.15 Amazon Web Service Cloud	25
2.16 Pengujian	25
2.16.1 Pengujian Fungsional	26
2.16.2 Pengujian Non-Fungsional	26
BAB 3 METODOLOGI	27
3.1 Studi Literatur	27
3.2 Analisis Kebutuhan	28
3.3 Perancangan	28
3.4 Implementasi	28
3.4.1 Implementasi Point of Sales	28
3.4.2 Implementasi Cloud Computing	28
3.5 Pengujian	28
3.6 Kesimpulan dan Saran	29
BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM	30
4.1 Pemodelan Proses Bisnis	30

4.1.1	Pemodelan Proses Bisnis As-Is	30
4.1.2	Pemodelan Proses Bisnis To-Be	32
4.2	Analisis Kebutuhan	34
4.2.1	Identifikasi Aktor	34
4.2.2	Kebutuhan Fungsional.....	35
4.2.3	Kebutuhan Non-Fungsional	37
4.2.4	Pemodelan Use Case	38
4.3	Spesifikasi Use Case	39
4.3.1	Spesifikasi Use Case Melakukan Login	39
4.3.2	Spesifikasi Use Case Memasukan Data Penjualan	40
4.3.3	Spesifikasi Use Case Memasukan Data Pembelian	41
4.3.4	Spesifikasi Use Case Memasukan Data Gaji Karyawan	41
4.3.5	Spesifikasi Use Case Melihat Stok Suku Cadang	42
4.3.6	Spesifikasi Use Case Melihat Laporan Pemasukan	43
4.3.7	Spesifikasi Use Case Melihat Laporan Pengeluaran	43
4.3.8	Spesifikasi Use Case Melakukan <i>Logout</i>	44
4.4	Activity Diagram.....	44
4.4.1	Activity Diagram Melakukan Login.....	45
4.4.2	Activity Diagram Memasukan Data Penjualan.....	45
4.4.3	Activity Diagram Memasukan Data Pembelian	47
4.4.4	Activity Diagram Memasukan Data Gaji Karyawan	48
4.4.5	Activity Diagram Melihat Stok Suku Cadang.....	49
4.4.6	Activity Diagram Melihat Laporan Pemasukan.....	49
4.4.7	Activity Diagram Melihat Laporan Pengeluaran	50
4.4.8	Activity Diagram Melakukan Logout	51
4.5	Sequence Diagram	51
4.5.1	Sequence Diagram Melakukan Login.....	51
4.5.2	Sequence Diagram Memasukan Data Penjualan.....	52
4.5.3	Sequence Diagram Memasukan Data Pembelian.....	53
4.5.4	Sequence Diagram Memasukan Data Gaji Karyawan.....	53
4.5.5	Sequence Diagram Melihat Stok Suku Cadang.....	54
4.5.6	Sequence Diagram Melihat Laporan Pemasukan	55



4.5.7 Sequence Diagram Melihat Laporan Pengeluaran	55
4.5.8 Sequence Diagram Melakukan Logout	56
4.6 Class Diagram.....	57
4.7 Physical Data Model	58
4.8 Perancangan Antarmuka	59
4.8.1 Perancangan Antarmuka Halaman Login.....	60
4.8.2 Perancangan Antarmuka Halaman Penjualan	60
4.8.3 Perancangan Antarmuka Halaman Pembelian	61
4.8.4 Perancangan Antarmuka Halaman Gaji Karyawan	62
4.8.5 Perancangan Antarmuka Halaman Stok Suku Cadang.....	64
4.8.6 Perancangan Antarmuka Halaman Pemasukan.....	64
4.8.7 Perancangan Antarmuka Halaman Pengeluaran	65
4.8.8 Perancangan Antarmuka Logout.....	67
BAB 5 IMPLEMENTASI DAN PENGUJIAN SISTEM	68
5.1 Implementasi Sistem	68
5.1.1 Spesifikasi Lingkungan Implementasi	68
5.1.2 Implementasi Kode Program	69
5.1.3 Implementasi Antarmuka	75
5.1.4 Implementasi Cloud Computing	81
5.2 Pengujian Sistem.....	85
5.2.1 Pengujian Fungsional	85
5.2.2 Pengujian Non-Fungsional	89
BAB 6 PENUTUP	91
6.1 Kesimpulan.....	91
6.2 Saran	92
DAFTAR REFERENSI	93
LAMPIRAN A HASIL WAWANCARA.....	95
LAMPIRAN B KONFIGURASI AWS EC2	97
LAMPIRAN C HASIL PENGUJIAN VALIDASI	108



DAFTAR TABEL

Tabel 2.1 Notasi Event, Activities dan Gateways 14

Tabel 2.2 Notasi Data Object, Data Inputs, Data Outputs dan Data Stores 15

Tabel 2.3 Notasi Sequence Flow, Message Flow, Associations, dan Data Associations..... 15

Tabel 2.4 Notasi Pool dan Lanes 16

Tabel 2.5 Notasi Group dan Text Annotation 16

Tabel 2.6 Simbol-simbol pada Use Case Diagram 18

Tabel 2.7 Simbol-simbol pada Class Diagram 19

Tabel 2.8 Simbol-simbol Activity Diagram 19

Tabel 2.9 Simbol-simbol pada Sequence Diagram 21

Tabel 4.1 Identifikasi Aktor 34

Tabel 4.2 Kebutuhan Fungsional Sistem Aktor Adminstrasi Keuangan 35

Tabel 4.3 Kebutuhan Fungsional Sistem Aktor Manajer 36

Tabel 4.4 Kebutuhan Non-Fungsional 38

Tabel 4.5 Spesifikasi Use Case Melakukan Login 39

Tabel 4.6 Spesifikasi Use Case Memasukan Data Penjualan 40

Tabel 4.7 Spesifikasi Use Case Memasukan Data Pembelian 41

Tabel 4.8 Spesifikasi Use Case Memasukan Data Gaji Karyawan 41

Tabel 4.9 Spesifikasi Use Case Melihat Stok Suku Cadang 42

Tabel 4.10 Spesifikasi Use Case Melihat Laporan Pemasukan 43

Tabel 4.11 Spesifikasi Use Case Melihat Laporan Pengeluaran 43

Tabel 4.12 Spesifikasi Use Case Melakukan Logout 44

Tabel 5.1 Spesifikasi Lingkungan Perangkat Keras 68

Tabel 5.2 Spesifikasi Lingkungan Perangkat Lunak 68

Tabel 5.3 Kode Program Penjualan 69

Tabel 5.4 Kode Program Gaji Karyawan 72

Tabel 5.5 Kode Program Pemasukan 74

Tabel 5.6 Spesifikasi Instance AWS EC2 81

Tabel 5.7 Komponen Perangkat Lunak 82

Tabel 5.8 Uji Validasi Melakukan Login 86



Tabel 5.9 Uji Validasi Memasukan Data Penjualan..... 87

Tabel 5.10 Uji Validasi Memasukan Data Pembelian..... 88

Tabel 5.11 Uji Validasi Memasukan Data Gaji Karyawan..... 88



DAFTAR GAMBAR

Gambar 2.1 Logo Bengkel Mas Pur Baturaja	6
Gambar 2.2 Waterfall Model	12
Gambar 2.3 Contoh Penggunaan Use Case Diagram	17
Gambar 2.4 Contoh Penggunaan Class Diagram	18
Gambar 3.1 Diagram Alir Metode Penelitian Sistem Informasi Point of Sales Untuk Bengkel	27
Gambar 4.1 Proses Bisnis Saat ini Penjualan	31
Gambar 4.2 Proses Bisnis Usulan Penjualan	33
Gambar 4.3 Aturan Penomoran Identifikasi Aktor	34
Gambar 4.4 Aturan Penomoran Kebutuhan Fungsional	35
Gambar 4.5 Aturan Penomoran Kebutuhan Non-Fungsional	37
Gambar 4.6 Use Case Sistem Informasi Point of Sales untuk Bengkel	38
Gambar 4.7 Activity Diagram Melakukan Login	45
Gambar 4.8 Activity Diagram Memasukan Data Penjualan	46
Gambar 4.9 Activity Diagram Memasukan Data Pembelian	47
Gambar 4.10 Activity Diagram Memasukan Gaji Karyawan	48
Gambar 4.11 Activity Diagram Melihat Stok Suku Cadang	49
Gambar 4.12 Activity Diagram Melihat Laporan Pemasukan	50
Gambar 4.13 Activity Diagram Melihat Laporan Pengeluaran	50
Gambar 4.14 Activity Diagram Melakukan Logout	51
Gambar 4.15 Sequence Diagram Melakukan Login	52
Gambar 4.16 Sequence Diagram Memasukan Data Penjualan	52
Gambar 4.17 Sequence Diagram Memasukan Data Pembelian	53
Gambar 4.18 Sequence Diagram Memasukan Data Gaji Karyawan	54
Gambar 4.19 Sequence Diagram Melihat Stok Suku Cadang	54
Gambar 4.20 Sequence Diagram Melihat Laporan Pemasukan	55
Gambar 4.21 Sequence Diagram Melihat Laporan Pengeluaran	56
Gambar 4.22 Sequence Diagram Melakukan Logout	56
Gambar 4.23 Class Diagram Sistem Informasi Point of Sales untuk Bengkel	57

Gambar 4.24 Physical Data Model Sistem Informasi Point of Sales untuk Bengkel	59
Gambar 4.25 Perancangan Antarmuka Halaman Login.....	60
Gambar 4.26 Perancangan Antarmuka Halaman Penjualan	61
Gambar 4.27 Perancangan Antarmuka Halaman Pembelian	62
Gambar 4.28 Perancangan Antarmuka Halaman Gaji Karyawan – Aktif.....	63
Gambar 4.29 Perancangan Antarmuka Halaman Gaji Karyawan – Tidak Aktif	63
Gambar 4.30 Perancangan Antarmuka Halaman Stok Suku Cadang.....	64
Gambar 4.31 Perancangan Antarmuka Halaman Pemasukan.....	65
Gambar 4.32 Perancangan Antarmuka Halaman Pengeluaran – Pembelian	66
Gambar 4.33 Perancangan Antarmuka Halaman Pengeluaran – Gaji Karyawan .	66
Gambar 4.34 Perancangan Antarmuka Logout.....	67
Gambar 5.1 Antarmuka Halaman Login.....	76
Gambar 5.2 Antarmuka Halaman Penjualan	76
Gambar 5.3 Antarmuka Halaman Pembelian	77
Gambar 5.4 Antarmuka Halaman Gaji Karyawan – Aktif.....	77
Gambar 5.5 Antarmuka Halaman Gaji Karyawan – Tidak Aktif	78
Gambar 5.6 Antarmuka Halaman Stok Suku Cadang.....	78
Gambar 5.7 Antarmuka Halaman Pemasukan.....	79
Gambar 5.8 Antarmuka Halaman Pengeluaran – Pembelian	79
Gambar 5.9 Antarmuka Halaman Pengeluaran – Gaji Karyawan	80
Gambar 5.10 Antarmuka Logout.....	80
Gambar 5.11 Instance AWS EC2	81
Gambar 5.12 Login pada Instance Menggunakan Protokol SSH.....	82
Gambar 5.13 Pemasangan Apache2 pada Ubuntu 16.04	83
Gambar 5.14 Pemasangan PHP7 pada Ubuntu 16.04	83
Gambar 5.15 Pemasangan MySQL pada Ubuntu 16.04.....	84
Gambar 5.16 Konfigurasi Akses Folder Kode Program	84
Gambar 5.17 Pemasangan PhpMyAdmin pada Ubuntu 16.04	85
Gambar 5.18 Aturan Penomoran Pengujian Fungsional	86
Gambar 5.19 Hasil Uji Kompatibilitas menggunakan Perangkat Lunak Sortsite. .	89
Gambar 5.20 Petunjuk Kehilangan Konten atau Fungsionalitas.....	90

Gambar 5.21 Petunjuk Masalah Berat pada Tata Letak atau Kinerja..... 90

Gambar 5.22 Petunjuk Masalah Ringan pada Tata Letak atau Kinerja..... 90



DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA.....	95
A.1 Hasil Wawancara Dengan Administrasi Keuangan.....	95
A.2 Hasil Wawancara Dengan Manajer.....	96
LAMPIRAN B KONFIGURASI AWS EC2.....	97
B.1 Konfigurasi Awal AWS EC2.....	97
B.1.1 Memilih Sistem Operasi.....	97
B.1.2 Memilih Jenis Instance.....	97
B.1.3 Konfigurasi Detail Instance.....	98
B.1.4 Konfigurasi Memori.....	98
B.1.5 Konfigurasi Tags.....	98
B.1.6 Konfigurasi Security Group.....	99
B.1.7 Membuat Key Pair.....	99
B.2 Konfigurasi Perangkat Lunak Pendukung.....	100
B.2.1 Pemasangan Apache2.....	100
B.2.2 Pemasangan PHP7.....	102
B.2.3 Pemasangan MySQL.....	103
B.2.4 Pemasangan PhpMyAdmin.....	104
B.3 Konfigurasi simaspur.com.....	106
B.3.1 Pembuatan Direktori.....	106
B.3.2 Konfigurasi simaspur.conf.....	106
B.3.3 Konfigurasi User pada Group www-data.....	107
LAMPIRAN C HASIL PENGUJIAN VALIDASI.....	108
C.1 Hasil Pengujian Validasi pada Administrator Keuangan.....	108
C.2 Hasil Pengujian Validasi pada Manajer.....	113

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Semakin pesatnya perkembangan teknologi informasi, mendorong untuk setiap lapisan kehidupan, baik bisnis maupun kehidupan sehari-hari untuk beradaptasi dengan keadaan tersebut. Teknologi yang tengah pesat perkembangannya adalah sistem informasi karena memiliki banyak fungsi diantaranya untuk mengumpulkan data, lalu memprosesnya menjadi informasi yang dapat didistribusikan kepada pengguna (Hall, 2008). Secara teknis suatu organisasi akan lebih mudah dalam pengumpulan, pemrosesan, penyimpanan dan pendistribusian suatu informasi yang digunakan dalam pengambilan keputusan dan pengendalian organisasinya menggunakan sistem informasi (Laudon, 2014). Singkatnya suatu sistem informasi dapat sangat membantu suatu organisasi dengan fungsi-fungsinya yang dapat mempermudah dalam pengambilan keputusan.

Bengkel Mas Pur Baturaja merupakan sebuah perusahaan yang bergerak dibidang jasa perbaikan badan kendaraan khususnya roda empat. Perusahaan ini didirikan oleh Purwanto pada tahun 1995 di Kota Baturaja. Berbagai jenis kendaraan yang diperbaiki berasal dari perseorangan, *dealer-dealer* mobil, dan perusahaan-perusahaan di dalam maupun luar Kota Baturaja. Sejak awal berdiri perusahaan ini menggunakan pengolahan data manual yang mengharuskan setiap transaksi dicatat dan dibukukan. Dalam pengelolannya, perusahaan ini memiliki permasalahan yaitu proses transaksi sering kali tidak tercatat sehingga mengakibatkan tidak lengkapnya data-data perusahaan. Hal ini juga mengakibatkan sulitnya mengambil keputusan untuk kemajuan perusahaan. Selain itu penyimpanan data dalam bentuk buku sangat rawan terhadap kerusakan. Kerusakan bisa disebabkan oleh banyak hal seperti kerusakan dikarenakan alam ataupun kehilangan. Penyimpanan data dengan memanfaatkan teknologi akan memberikan kemudahan bagi perusahaan dalam mengolah data dan menjamin keamanan data. Dengan data yang terjamin akan dapat meningkatkan kualitas pelayanan pada pelanggan dan menambah nilai perusahaan.

Untuk mengembangkan suatu sistem, diperlukan persiapan infrastruktur guna menunjang berjalannya sistem. Infrastruktur tersebut berupa CPU, RAM, *storage*, *bandwith* serta konfigurasi lainnya. Namun terdapat permasalahan sumber daya perusahaan yang terbatas, maka dalam penelitian ini akan menggunakan *cloud computing* untuk mengatasinya. *Cloud computing* adalah jenis teknologi baru membangun komputasi utilitas, virtualisasi, berorientasi layanan arsitektur dan komputasi paralel (Namasudra, et al., 2017). Dengan menggunakan *cloud computing*, perusahaan tidak perlu menyediakan infrastruktur guna menunjang sistem yang akan dikembangkan. Selain itu, dapat menghemat tempat dan mengurangi pengeluaran untuk biaya pemeliharaan sistem.

Berdasarkan masalah di atas maka peneliti akan mengembangkan “Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing*” untuk mempermudah dan memperjelas sistem pengolahan Bengkel Mas Pur Baturaja. Sistem ini akan memuat beberapa fitur seperti transaksi, laporan keuangan dan bukti pembayaran yang bisa dicetak jika diinginkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, didapatkan perumusan masalah sebagai berikut:

1. Bagaimana analisis dan perancangan Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* pada Bengkel Mas Pur Baturaja?
2. Bagaimana implementasi Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* pada Bengkel Mas Pur Baturaja?
3. Bagaimana pengujian Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* pada Bengkel Mas Pur Baturaja?

1.3 Tujuan

Tujuan yang akan dicapai dari penelitian ini adalah:

1. Mendapatkan hasil analisis dan rancangan Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* pada Bengkel Mas Pur Baturaja.
2. Mendapatkan hasil implementasi Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* pada Bengkel Mas Pur Baturaja.
3. Mendapatkan hasil pengujian Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* pada Bengkel Mas Pur Baturaja.

1.4 Manfaat

Manfaat yang akan diperoleh dari pengembangan sistem informasi ini adalah:

1. Dapat memberikan kemudahan dalam membuat laporan keuangan dan laporan transaksi bengkel.
2. Dapat memberikan kemudahan dalam pengolahan data bengkel.
3. Dapat memberikan kemudahan dalam melihat kemajuan bengkel melalui pendataan laporan keuangan dan transaksi bengkel.
4. Dapat memberikan kemudahan dalam pencatatan setiap transaksi dalam bengkel.

1.5 Batasan Masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka penelitian ini hanya akan membahas mengenai hal-hal berikut:

1. Penelitian ini akan melakukan analisis, desain, implementasi dan pengujian pada Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* pada Bengkel Mas Pur Baturaja.
2. Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* hanya ditujukan untuk mengelolah Bengkel Mas Pur Baturaja.
3. Penelitian ini akan diimplementasikan menggunakan *framework* Codeigniter dengan bahasa pemrograman PHP dan layanan *Amazon Web Service Cloud*.

1.6 Sistematika Pembahasan

Sistematika pembahasan dalam penelitian ini akan disusun sebagai berikut:

1. BAB 1 PENDAHULUAN

Pembahasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika pembahasan.

2. BAB 2 LANDASAN KEPUSTAKAAN

Pencarian literatur dasar untuk menguatkan topik penelitian yang akan dilaksanakan dengan informasi penelitian sebelumnya. Teori harus berhubungan dengan topik yang akan diteliti.

3. BAB 3 METODOLOGI

Menguraikan mengenai alur dalam menjalankan penelitian di antaranya terdapat studi literatur, pengumpulan data, analisa persyaratan dan perancangan sistem.

4. BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM

Menjabarkan pembahasan berkaitan dengan penggambaran arsitektur, pemodelan bisnis, identifikasi aktor dan persyaratan fungsional sistem berdasarkan aktornya. Serta mengilustrasikan terkait perancangan sistem yang berisi *activity diagram*, *sequence diagram*, pemodelan kelas, serta rancangan antarmuka.

5. BAB 5 IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini membahas mengenai source code dari implementasi hasil perancangan dalam bentuk file PHP serta juga akan membahas mengenai proses dan hasil pengujian terhadap sistem yang telah direalisasikan.

6. BAB 6 PENUTUP

Menarik hasil penelitian berupa kesimpulan berdasarkan perancangan dan pengujian rancangan sistem. Selain itu juga membahas saran agar mengembangkan sistem menjadi kompleks.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Sebelumnya telah ada penelitian mengenai pengembangan sistem *point of sales* yaitu salah satunya oleh (Berner, 2003) dengan judul *About the Development of Point of Sale System: an Experience Report*. Dalam penelitiannya Berner mengembangkan sistem *point of sales* untuk divisi asuransi jiwa dari sebuah bank besar

Dalam penelitiannya, Berner menjelaskan tujuan dari pembuatan sistem ini adalah untuk membantu penjualan proses asuransi jiwa pada bank tersebut. Bank ini dipersiapkan untuk memasuki pasar asuransi jiwa dengan portofolio produk yang lengkap. Agar tidak membangun organisasi penjualan baru, infrastruktur yang ada (kantor cabang) seharusnya digunakan. Karena itu, sistem POS harus diaktifkan untuk membantu pegawai bank untuk menawarkan dan menjual asuransi jiwa kepada pelanggan bank. Sistem POS diwujudkan dalam *multi-tier* umum arsitektur dengan basis data relasional di latar belakang, CORBA sebagai *middleware*, dan *Java* sebagai bahasa pemrograman utama untuk menyambungkan keduanya, komponen bisnis pada *middleware*, dan komponen klien dalam bentuk *thin* klien. *Front end* sistem POS akan berjalan di penelusur web untuk memudahkan pegawai dalam mengaksesnya.

Penelitian selanjutnya adalah dari (Hewitt, 2008) yang berjudul *ORGs for Scalable, Robust, Privacy-Friendly Client Cloud Computing*. Hewitt menjelaskan bahwa dalam arsitektur *multicore*, banyak prosesor yang di buat menjadi sebuah *chip* yang nantinya dapat secara fisik dihubungkan menjadi paket *multichip*. Sebuah *core* memiliki komunikasi *bandwidth low latency* yang sangat tinggi. *Chip* yang diproduksi oleh intel maupun AMD saat ini hanya memiliki empat *core* pada sebuah *chip* sedangkan proyeksi industri membutuhkan puluhan *core* untuk memenuhi kebutuhan. Dalam paradigma *cloud computing* informasi disimpan secara permanen pada server di internet dan disimpan secara sementara pada komputer klien. *Multicore* bermaksud mengubah *cloud computing* pada klien dengan cara memperluas *cloud* pada klien dan menjamin privasi informasi.

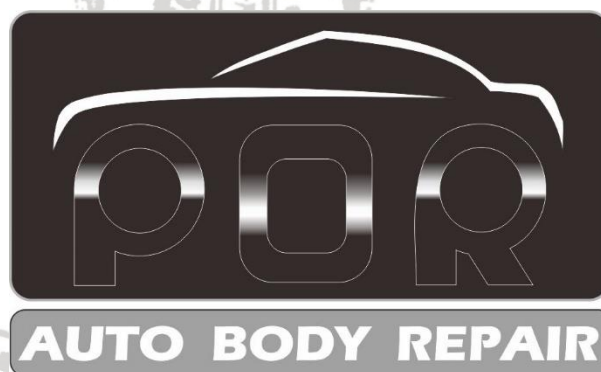
Dalam penelitian yang berjudul *Cloud Computing Technologies in "Smart City" Projects* oleh (Wieclaw, et al., 2017) memanfaatkan teknologi *cloud computing* dalam analisis desain kelas *smart city*. Sebagian besar penerapan teknologi *cloud computing* dikombinasikan dengan teknologi seperti GRID, IoT, *Big Data*, OLAP, dan GIS. Pada saat yang sama perlu dipertimbangkan solusi yang kompleks serta sistematis mengingat kekhasan teknologi yang dikembangkan secara praktis itu tidak ada. Untuk memperkenalkan konsep *smart city* perlu untuk mewujudkan sepenuhnya pendekatan sistem, mempertimbangkan kekhasan TI yang ada, infrastruktur dan layanan yang sesuai. Dalam kasus seperti itu kepentingan berbagai organisasi yang termotivasi, perusahaan-perusahaan dan sejumlah besar penduduk kota harus diperhitungkan sejauh perbedaan hak akses sumber daya informasi. Arsitektur lapisan informasi kelas sistem *smart city* untuk akumulasi dan

pemrosesan data yang beragam yang diperoleh dari penyedia dan dianalisis selama penelitian. Karakteristik dan realisasi teknologi informasi berdasarkan *cloud computing* yang digunakan dalam desain *smart city* ditinjau dalam penyelidikan lebih lanjut untuk memodelkan sistem mengoperasikan dan mengembangkan prototipe untuk individu dan *heat cost accounting* serta operasional *mobile* akun pelanggan untuk layanan yang diterima. Semua itu perlu untuk membangun kemungkinan peningkatan fungsi dan integrasi dengan subsistem lain dari "*Smart City Ternopil design portofolio*".

2.2 Bengkel Mas Pur Baturaja

Bengkel Mas Pur Baturaja adalah perusahaan yang bergerak dibidang jasa perbaikan badan kendaraan khususnya roda empat. Berdiri pada tahun 1995 dan berlokasi di Jalan Ahmad Yani KM 6 Lorong Sundawan No. 5 Kemelak kota Baturaja provinsi Sumatera Selatan, nama perusahaan diambil dari nama pendiri bengkel tersebut yaitu Bapak Purwanto yang berperan sebagai pendiri, pemilik sekaligus pimpinan di perusahaan ini. Bengkel Mas Pur Baturaja merupakan salah satu bengkel lama yang masih tetap menjaga kualitas hasil kerja dan masih dipercaya sampai saat ini.

Berbagai jenis kendaraan yang di perbaiki oleh Bengkel Mas Pur Baturaja berasal dari perseorangan, *dealer-dealer* mobil, dan perusahaan-perusahaan di daerah dalam maupun luar kota Baturaja, Bengkel Mas Pur Baturaja mengusung motto "*Mengutamakan kualitas bukan kuantitas*" sudah menjadi komitmen perusahaan untuk dapat memberikan pelayanan *service* maupun *after service* terbaik bagi pelanggan. Logo dari Bengkel Mas Pur Baturaja dapat dilihat pada Gambar 2.1.



Gambar 2.1 Logo Bengkel Mas Pur Baturaja

Sumber: Dokumen Perusahaan

Seiring bertambahnya usia, perusahaan ini terus meningkatkan kualitas untuk menjaga kepercayaan pelanggan. Meperbarui bahan dan peralatan kerja, menambah kualitas sumber daya manusia serta mengadakan *training* tahunan bagi karyawan untuk meningkatkan hasil kerja.

2.3 Visi dan Misi

2.3.1 Visi

Visi merupakan suatu gambaran mengenai perencanaan masa depan yang diinginkan dengan melihat potensi dan kebutuhan usaha. Visi dari Bengkel Mas Pur Baturaja adalah

“Menjadi perusahaan yg terkemuka di Sumatera Selatan dalam bidang body repair dan menjadi acuan bagi perusahaan sejenis serta membangun ikatan antara sesama perusahaan body repair.”

2.3.2 Misi

Dalam mewujudkan visi diatas diperlukan langkah nyata yang diwujudkan dalam misi berikut

1. Memberikan pengalaman yang mengesankan bagi pelanggan melalui hasil kerja yang berkualitas.
2. Mengoptimalkan sumber daya manusia yang bertumbuh dalam kreativitas dan kemampuan.
3. Memberikan hasil yang maksimal dan tingkat resiko yang minimal kepada perusahaan sehingga memberikan keuntungan kepada *stakeholder* dan kesejahteraan karyawan.

2.4 Sistem Informasi

Sistem informasi merupakan penggabungan antara manusia, seperangkat teknologi, kebijakan dan prosedur terorganisasi yang menyimpan, mengambil, mengubah dan memisahkan informasi dalam sebuah organisasi secara terstruktur (O'Brien, 2014). Cara komunikasi yang digunakan antara satu dengan yang lainnya adalah pengguna mengakses sistem informasi dari perangkat keras, prosedur dan instruksi pemrosesan informasi dalam hal ini perangkat lunak, saluran komunikasi berupa jaringan serta data yang tersimpan sebagai sumber data.

Alasan mengapa sebuah organisasi harus menggunakan sistem informasi adalah (O'Brien, 2014):

1. Operasional bisnis organisasi akan sangat terbantu dalam proses dan kegiatannya.
2. Manajer dan karyawan akan sangat mudah dalam pengambilan keputusan.
3. Mendukung terciptanya strategi bagi keuntungan kompetitif.

2.5 Point of Sales

Point of sales (POS) merupakan sebuah skema dalam menyelesaikan proses transaksi penjualan. Pada POS, pedagang menghitung jumlah yang harus dibayar oleh pelanggan, menunjukkan jumlah itu, dapat menyiapkan faktur untuk pelanggan dan menunjukkan pilihan bagi pelanggan untuk melakukan pembayaran. Pada saat itu juga merupakan waktu di mana pelanggan melakukan pembayaran kepada pedagang dengan imbalan barang atau setelah penyediaan layanan. Setelah menerima pembayaran, pedagang dapat mengeluarkan tanda terima untuk transaksi, yang biasanya dicetak tetapi juga dapat dikeluarkan atau dikirim secara elektronik (Dyson, 2015). Bisnis semakin mengadopsi sistem POS, dan salah satu alasan yang paling jelas dan meyakinkan adalah bahwa sistem POS menghilangkan kebutuhan akan label harga. Harga jual terkait dengan kode produk suatu barang saat menambahkan stok, sehingga kasir hanya perlu memindai kode ini untuk memproses penjualan. Jika ada perubahan harga, ini juga dapat dengan mudah dilakukan melalui jendela inventaris. Keuntungan lain termasuk kemampuan untuk menerapkan berbagai jenis diskon, skema loyalitas untuk pelanggan, dan kontrol stok yang lebih efisien, dan fitur-fitur ini adalah ciri khas dari hampir semua sistem e-POS modern. Saat ini POS bisa memanfaatkan jaringan internet dengan adanya *cloud computing*. Dengan menyewa server komputer di internet, memungkinkan data-data tersebut dapat diakses kapan dan di mana saja selama ada koneksi internet. Lebih praktis dan mudah bagi pebisnis yang sering mobile atau memiliki banyak cabang usaha di tempat lain.

Berikut beberapa keuntungan dengan menggunakan *point of sales*:

1. Dapat menganalisis data penjualan, mencari tahu seberapa baik semua item yang dijual untuk menyesuaikan order pembelian selanjutnya.
2. Dapat mempertahankan rekapitulasi penjualan untuk membantu menyesuaikan keputusan order pembelian untuk tren pembelian konsumen musiman.
3. Dapat meningkatkan akurasi harga dengan mengintegrasikan bar-code scanner dan kemampuan otorisasi kartu kredit dengan sistem POS.

2.6 Analisis dan Desain Sistem

Analisis sistem merupakan kegiatan guna melihat sistem telah berjalan dengan baik atau belum, melihat kualitas sistem yang selanjutnya mendokumentasikan kebutuhan yang akan dipenuhi dalam sistem yang baru (Rosa & Shalahudin, 2016).

Pada prosesnya seringkali pengguna kesulitan dalam mendefinisikan kebutuhan mereka maka solusinya adalah dengan menjalankan proses analisis dan desain secara bersamaan. Pendefinisian kebutuhan pengguna dirasa akan lebih muda apabila sudah ada antarmuka.

Tercapainya kebutuhan fungsional dengan cara melakukan perancangan sebuah sistem berupa desain pengembangan perangkat lunak. Pemenuhan target desain perangkat lunak dari segi performa, waktu, dan biaya perangkat yang harus memenuhi ketentuan yang telah disepakati.

2.7 Pengembangan Perangkat Lunak

Kondisi dimana suatu perangkat lunak dikembangkan dengan tujuan bisnis, perangkat lunak yang disatukan dengan perangkat lain. Perangkat lunak yang dikembangkan dengan cara professional hasilnya nanti akan dimanfaatkan oleh orang lain selain pengembang itu sendiri, dibutuhkan pemeliharaan dan perkembangan mengikuti *trend* teknologi (Sommerville, 2011)

Pengembangan perangkat lunak juga biasa disebut dengan rekayasa perangkat lunak (*software engineering*) adalah pembangunan dengan menggunakan konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang dapat bekerja secara efisien dan bernilai ekonomis yang dipercaya.

2.7.1 System Development Life Cycle

System Development Life Cycle (SDLC) merupakan siklus pengembangan suatu sistem menggunakan metodologi atau model yang digunakan untuk mengembangkan sistem perangkat lunak. Dalam membuat perangkat lunak harus melalui beberapa tahap agar dapat menghasilkan perangkat lunak yang baik dan berkualitas (Rosa & Shalahudin, 2016).

2.7.1.1 Model System Development Life Cycle (SDLC)

Beberapa macam SDLC yang diterapkan kedalam pengembangan perangkat lunak sebagai berikut (Pressman, 2010):

1. *Specialized Process Models*

Cara kerja model ini adalah dengan mengambil satu atau lebih ciri khas suatu model pengembangan sistem yang lebih sederhana dan mudah diterapkan pada saat pemilihan pendekatan pengembangan perangkat lunak yang akan dijabarkan lebih spesifik. Model ini lebih cocok untuk menyelesaikan tujuan utama yang lebih spesifik (Pressman, 2010). Beberapa macam dari *Specialized Process Model* sebagai berikut (Pressman, 2010):

- 1) *Component-Based Development*, model ini banyak menggunakan ciri khas dari model spiral dalam pengembangan perangkat lunak dengan pendekatan iteratif. Namun model ini akan membangun sebuah perangkat lunak sebagai modul ataupun kelas dengan berbagai komponen perangkat lunak yang dapat menyesuaikan.
- 2) *The Formal Methods Model*, pada model ini pengembang dapat menentukan, mengembangkan dan memverifikasi sistem dengan menggunakan rumus-rumus matematika.

- 3) *Aspect-Oriented Software Development* menerapkan ciri khas yang bersifat iteratif. Model ini akan menggambarkan, merancang, mengkhususkan serta mengkonstruksi sebuah aspek.

2. *Prescriptive Model*

Prescriptive Model adalah model pengembangan perangkat lunak yang menggunakan aliran kerja yang setiap elemen didalamnya saling berhubungan satu sama lain dalam menyelesaikan pekerjaannya (Pressman, 2010). Berikut contoh dari *Prescriptive Model* adalah (Pressman, 2010):

- 1) *The Waterfall Model* merupakan model pengembangan perangkat lunak dengan menggunakan konsep terstruktur, berurutan dan sistematis. Tahapannya dimulai dengan berkomunikasi dengan pengguna untuk mengetahui spesifikasi kebutuhan pengguna lalu melanjutkan tahapan-tahapan perancangan, pemodelan, konstruksi dan penyerahan sistem pada klien dan yang terakhir adalah mendukung keberlanjutan perangkat lunak yang telah selesai dikembangkan.
- 2) *Incremental Process Models* adalah model dengan jalan proses paralel dan linear akan digabungkan dari suatu proses. Cara pengimplementasian model ini dengan cara melalui sebuah urutan linear dengan bertahap. Aliran kegiatan yang bercirikan evolusioner dengan cara menambahkan satu persatu akan menghasilkan penambahan pada runtutan linear ini.
- 3) *Evolutionary Process Models*, model ini memakai cara penyusunan perangkat lunak dengan cara tidak sistematis. Sehingga ditengah penyusunan bisa jadi ada penambahan fitur atau komponen sesuai dengan kebutuhan perangkat lunak.
- 4) *Concurrent Models* memungkinkan tim perangkat lunak untuk mewakili elemen yang berulang dan bersamaan dari salah satu model proses. Sebagai contoh, aktivitas pemodelan yang didefinisikan untuk model spiral dicapai dengan menerapkan satu atau lebih dari tindakan rekayasa perangkat lunak berikut: prototyping, analisis, dan desain.
- 5) *A Final Word on Evolutionary Process*, maksud dari model ini adalah untuk mengembangkan perangkat lunak berkualitas tinggi secara iteratif atau bertahap. Namun, dimungkinkan untuk menggunakan proses evolusi untuk menekankan fleksibilitas, ekstensibilitas, dan kecepatan perkembangan. Tantangan bagi tim perangkat lunak dan manajernya adalah untuk membangun keseimbangan yang tepat antara proyek dan parameter produk serta kepuasan pelanggan (penentu utama kualitas perangkat lunak).

3. Unified Process

Unified Process adalah model pengembangan perangkat lunak yang berfokus pada hubungan ciri khas sistem dari penjelasan klien. Pemrograman berorientasi objek akan sangat cocok dikembangkan bersama dengan model ini karena akan sangat mengedepankan peran dari analisis kebutuhan dan perancangan sistem dalam pengembangan perangkat lunak yang diminta oleh pengguna.

2.7.1.2 Tahapan-tahapan dalam sistem Development Life Cycle (SDLC)

Tahap-tahap dalam SDLC secara umum terdiri dari (Rosa & Shalahudin, 2016):

1. Inisiasi (*Initiation*)

Dalam tahap ini akan melakukan kegiatan dengan hasil dokumen proyek perangkat lunak (Rosa & Shalahudin, 2016).

2. Pengembangan Konsep sistem (*System Concept Development*)

Tahap ini akan mendefinisikan dokumen analisis manfaat, lingkup sistem, biaya dan manajemen rencana serta pembelajaran kemudahan sistem (Rosa & Shalahudin, 2016).

3. Perencanaan (*Planning*)

Pengembangan konsep penggunaan sebuah proyek serta dokumen perencanaan lainnya. Mendeskripsikan informasi dasar dari sumber daya yang dibutuhkan dalam proyek (Rosa & Shalahudin, 2016).

4. Analisa Kebutuhan

Analisa kebutuhan pengguna sistem dan mengembangkan kebutuhan pengguna serta membuat dokumen kebutuhan sistem (Rosa & Shalahudin, 2016).

5. Desain (*Design*)

Mengubah kebutuhan detail sistem menjadi kebutuhan yang lebih lengkap, kemudian dokumen sistem berfokus pada pemenuhan fungsi-fungsi yang dibutuhkan (Rosa & Shalahudin, 2016).

6. Pengembangan (*Development*)

Mentransformasi desain ke sistem informasi yang lebih nyata termasuk cara memperoleh dan melakukan instalasi pada lingkungan sistem yang dibutuhkan, membuat basis data dan mempersiapkan prosedur kasus pengujian; mempersiapkan materi pengujian, pengkodean, penggabungan, memperbaiki program, serta pemeriksaan pengujian (Rosa & Shalahudin, 2016).

7. Integrasi dan pengujian (*Integration and test*)

Pengimplementasian sistem bahwa telah memenuhi kebutuhan yang telah didefinisikan pada dokumen kebutuhan fungsional. Menghasillkan laporan analisisa pengujian (Rosa & Shalahudin, 2016).

8. Implementasi (*Implementation*)

Implementasi perangkat lunak pada lingkungan pengguna dan menjalankan solusi dari permasalahan yang diidentifikasi dari fase integrase dan pengujian (Rosa & Shalahudin, 2016)

9. Operasi dan Pemeliharaan (*Operation and Maintenance*)

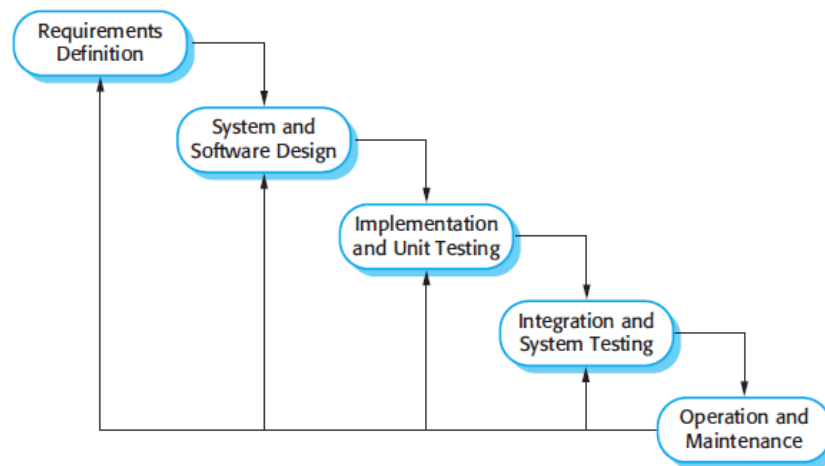
Mendeskripsikan mengenai kegiatan pengorperasian dan pemeliharaan sistem pada lingkungan pengguna, yang didalamnya ada implementasi tahap akhir dan menuju proses peninjauan (Rosa & Shalahudin, 2016)

10. Disposisi (*Dispotition*)

Menjelaskan mengenai aktifitas terakhir dari pengembangan perangkat lunak dan pembangunan data yang sesuai dengan aktifitas pengguna (Rosa & Shalahudin, 2016).

2.7.2 Waterfall Model

Menurut Sommerville (Sommerville, 2011) *Waterfall model* memiliki beberapa tahapan, sebagaimana terdapat pada Gambar 2.2.



Gambar 2.2 Waterfall Model

Sumber : (Sommerville, 2011).

Berikut tahapan-tahapan yang menjadi alur kerja pada *waterfall model* dalam pengembangan sebuah sistem:

1. Analisis dan definisi persyaratan

Pada tahap ini layanan, kendala, dan tujuan sistem ditetapkan melalui hasil wawancara dengan pengguna sistem yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.



2. Perancangan sistem dan perangkat lunak

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem. Desain perangkat lunak melibatkan mengidentifikasi dan menggambarkan abstraksi sistem perangkat lunak mendasar dan hubungannya.

3. Implementasi dan pengujian unit

Selama tahap ini, perancangan perangkat lunak diwujudkan sebagai satu set program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasi.

4. Integrasi dan pengujian sistem

Masing-masing unit program atau program diintegrasikan dan diuji sebagai sistem yang lengkap untuk memastikan bahwa persyaratan perangkat lunak telah dipenuhi. Setelah pengujian, sistem perangkat lunak dapat dikirim ke pelanggan.

5. Operasi dan pemeliharaan

Tahap ini adalah fase siklus hidup terpanjang. Sistem akan dilakukan instalasi dan digunakan secara nyata. Proses *maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya. Tujuan dari tahap ini untuk meningkatkan implementasi unit sistem dan meningkatkan layanan sistem sebagai kebutuhan baru.





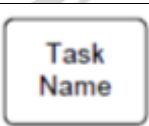

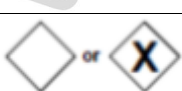
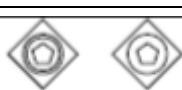




2.8 Business Process Model and Notation

Business Process Model and Notation (BPMN) suatu pemodelan hubungan antar gambaran awal prosedur bisnis serta hasil dari prosedur yang sudah diterapkan (Object Management Group, Inc., 2011). Tujuan dari pembuatan BPMN agar dapat menghasilkan urutan proses bisnis yang lebih mudah dipahami.

2.8.1 Flow Object

Flow object terbagi atas *events*, *activities*, serta *gateways*. *Events* merepresentasikan aktivitas pada sebuah proses atau koreografi. Sementara *activity* merupakan aktivitas pengguna dalam menjalankan suatu proses. Sedangkan *gateways* sebagai pemantau pada alur *sequence flow*. Penjelasan lebih lengkapnya dijelaskan dalam Tabel 2.1.

Tabel 2.1 Notasi Event, Activities dan Gateways

Notasi	Nama	Deskripsi
	<i>Start</i>	Start sebagai simbol suatu proses yang akan dimulai
	<i>Intermediate</i>	<i>Intermediate</i> sebagai simbol antara <i>start</i> dan <i>end event</i> .
	<i>End</i>	<i>End</i> sebagai simbol ketika suatu proses berakhir.
	<i>Activity</i>	<i>Activity</i> sebagai simbol kegiatan yang dikerjakan oleh perusahaan selama proses berlangsung.
	<i>Task Name</i>	<i>Task name</i> sebagai simbol aktivitas atomic yang ada pada proses
	<i>Gateway</i>	<i>Gateway</i> sebagai simbol yang memantau alur <i>sequence flow</i> pada suatu proses.
	<i>Exclusive</i>	<i>Exclusive</i> sebagai simbol untuk membuat jalur <i>alternative</i> .
	<i>Event – Based</i>	<i>Event Based</i> sebagai simbol untuk memilih jalur yang didasarkan dari <i>event</i> .
	<i>Parallel Event- Based</i>	<i>Parallel Event- Based</i> simbol untuk memulai proses baru dalam sebuah proses.
	<i>Inclusive</i>	<i>Inclusive</i> simbol untuk membuat jalur alternatif atau jalur paralel dalam proses.
	<i>Complex</i>	<i>Complex</i> simbol untuk menggambarkan fungsi yang rumit.
	<i>Parallel</i>	<i>Parallel</i> simbol untuk membuat jalur yang berjalan paralel atau bersamaan dalam satu waktu.






Sumber: (Object Management Group, Inc., 2011)



2.8.2 Data

Data memiliki empat notasi di antaranya *data objects*, *data inputs*, *data outputs* dan *data stores* yang dapat dilihat pada Tabel 2.2.

Tabel 2.2 Notasi Data Object, Data Inputs, Data Outputs dan Data Stores





Notasi	Nama	Deskripsi
	<i>Data Object</i>	sebagai simbol yang memuat informasi kegiatan yang akan dilakukan beserta hasilnya.
	<i>Data Object (Collection)</i>	sebagai simbol kumpulan informasi.
	<i>Data Input</i>	sebagai simbol masukkan informasi untuk seluruh proses yang berjalan.
	<i>Data Output</i>	sebagai simbol informasi yang telah diperoleh.
	<i>Data Store</i>	sebagai simbol pengambilan atau perbaruan data yang sebelumnya sudah disimpan

Sumber: (Object Management Group, Inc., 2011)

2.8.3 Connecting Object

Connecting object berfungsi sebagai penghubung dari *flow object* dan bisa juga informasi lain. Terdapat empat *connecting object* sebagai berikut yang dijelaskan pada Tabel 2.3.

Tabel 2.3 Notasi Sequence Flow, Message Flow, Associations, dan Data Associations

Notasi	Nama	Deskripsi
	<i>Sequence Flow</i>	sebagai simbol ilustrasi alur aktivitas yang dijalankan.
	<i>Message Flow</i>	sebagai simbol ilustrasi alur untuk mengirim dan menerima pesan antara dua entitas.
	<i>Associations</i>	sebagai simbol asosiasi antara data, <i>flow object</i> serta beberapa grafis BPMN.
	<i>Data Associations</i>	sebagai simbol yang mengoeksikan informasi dari <i>data object</i> serta <i>input output</i> .



Sumber: (Object Management Group, Inc., 2011)



2.8.4 Swimlanes

Swimlanes terdiri *pool* dan *lanes*, yang digunakan dalam mengelompokkan aktivitas sesuai dengan proses bisnis kemudian dijalankan melalui kegiatan tertentu. Berikut Tabel 2.4 merupakan penjelesan dari macam-macam *swimlanes*.

Tabel 2.4 Notasi Pool dan Lanes



Notasi	Nama	Deskripsi
	<i>Pool</i>	<i>Pool</i> merupakan partisi pada suatu proses.
	<i>Lane</i>	Bagian dari partisi di dalam suatu <i>pool</i> tujuannya sebagai pengatur atau pengumpulan proses kegiatan.

Sumber: (Object Management Group, Inc., 2011)

2.8.5 Artifacts

Artifacts merepresentasikan suatu informasi pendukung yang berhubungan dengan proses yang terdapat pada proses yang berjalan. Tabel 2.5 *artifacts* terbagi menjadi *group* dan *text annotation* seperti di bawah ini

Tabel 2.5 Notasi Group dan Text Annotation

Notasi	Nama	Deskripsi
	<i>Group</i>	<i>Group</i> sebagai penanda data proses yang telah dikelompokkan pada satu kelompok yang sama.
	<i>Text Annotation</i>	<i>Text Annotation</i> adalah alur yang menyediakan suatu data berupa keterangan lain yang ditujukan untuk memperjelas proses.

Sumber: Object Management Group (2011)

2.9 Object Oriented Analysis and Design (OOAD)

Menurut (Kendall & Kendall, 2011) OOAD adalah sebuah pendekatan dalam memecahkan sebuah masalah dengan menggunakan model yang dibuat menurut konsep dari lingkungan sekitar. Berikut beberapa konsep dalam merancang sebuah sistem dengan menggunakan pendekatan OOAD menurut (Whitten & Bentley, 2007) adalah

1. *Object* merupakan sesuatu yang dapat dilihat, disentuh atau dirasakan dan pelanggan menyimpan data mengenai sesuatu tersebut.
2. Atribut merupakan data yang menggambarkan ciri khas dari sebuah *object*.
3. *Object instance* merupakan setiap orang khusus, tempat, sesuatu atau sebuah kejadian dan juga nilai atribut dari *object*.
4. *Behavior* merupakan kumpulan fungsi-fungsi yang dapat dilakukan oleh *object*.
5. Enkapsulasi atau pengemasan beberapa *item* ke dalam sebuah unit.

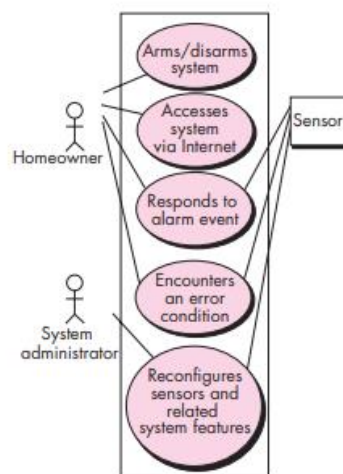
2.10 Unified Modelling Language (UML)

Unified Model Language (UML) sebutan bahasa standar untuk pemodelan sistem berorientasi objek (Whitten & Bentley, 2007). *Unified Model Language* (UML) adalah bahasa umum yang sering digunakan untuk:

1. *Visualizing* adalah menggambarkan pemodelan grafis yang sesuai.
2. *Specifying* adalah membangun pemodelan yang sesuai, tidak ambigu, serta lengkap.
3. *Constructing* adalah membuat pemodelan agar terhubung langsung ke berbagai bahasa pemrograman.
4. *Documenting* adalah pendokumentasian dari arsitektur sistem dan seluruh rincian informasi yang telah dikumpulkan.

2.10.1 Use case Diagram



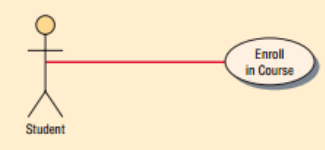
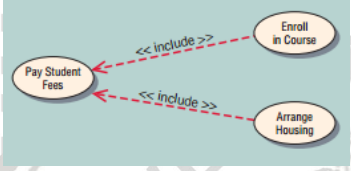
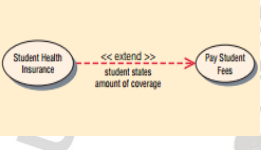
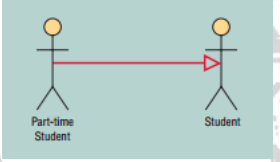
Use case diagram digunakan sebagai penggambaran hubungan antara sistem dengan pengguna berdasarkan alur yang telah dibuat (Whitten & Bentley, 2007). Contoh penggunaan *use case* dapat dilihat pada Gambar 2.3 dan untuk penjelasan setiap notasi akan dijelaskan pada Tabel 2.6.



Gambar 2.3 Contoh Penggunaan Use Case Diagram

Sumber: (Pressman, 2010)

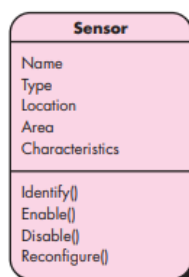
Tabel 2.6 Simbol-simbol pada Use Case Diagram

Simbol	Deskripsi
	<i>Use case</i> menggambarkan fitur yang bisa dijalankan pengguna.
	<i>Actor</i> sebagai pengguna yang dapat mengoperasikan sistem.
	<i>Communicates Relationship</i> berfungsi sebagai penghubung antara aktor serta <i>use case</i> .
	<i>Include Relationship</i> mengilustrasikan kondisi pada <i>use case</i> tambahan memerlukan <i>use case</i> lain dalam menjalankannya.
	<i>Extend Relationship</i> merupakan <i>use case</i> yang dapat berdiri sendiri meskipun ada tambahan <i>use case</i> lain.
	<i>Generalizes relationship</i> merupakan penghubung untuk suatu objek dengan objek bagiannya.

Sumber: (Kendall & Kendall, 2011)

2.10.2 Class Diagram

Class Diagram untuk menggambarkan struktur objek statis dalam sebuah sistem (Whitten & Bentley, 2007). Pada diagram ini juga menotasikan hubungan setiap kelas yang terdapat dalam sistem. Gambar 2.4 merupakan contoh penggunaan *class diagram* dan notasinya akan dijelaskan pada Tabel 2.7.



Gambar 2.4 Contoh Penggunaan Class Diagram

Sumber: (Pressman, 2010)



Tabel 2.7 Simbol-simbol pada Class Diagram

Simbol	Deskripsi
<p>Inheritable <u>Attributes</u> and <u>Behaviors</u></p>	<p><i>Attribute</i> sebagai informasi penginisialisasi pada kelas.</p> <p><i>Behavior</i> adalah fungsi, metode atau operasi yang akan dijalankan.</p>
	<p><i>Inheritance</i> menggambarkan ketika ada kelas turunan dari kelas lain.</p>
	<p><i>Association</i> menggambarkan objek pada suatu kelas dengan kelas lain yang saling terhubung.</p>
	<p><i>Agregation</i> menggambarkan objek dari suatu kelas merupakan bagian kelas lain.</p>
	<p><i>Composition</i> menggambarkan keterkaitan suatu kelas saling berpengaruh terhadap kelas lain.</p>

Sumber: (Whitten & Bentley, 2007)

2.10.3 Activity Diagram


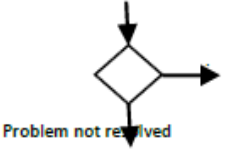
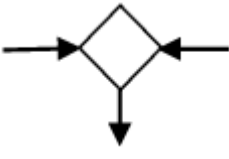
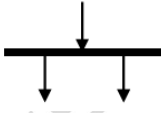
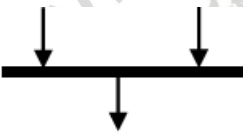

Activity diagram merupakan rangkaian runtutan alur kegiatan *use case* (Whitten & Bentley, 2007). Selain itu, menjelaskan alur aktivitas sistem dari awal hingga berakhir serta terdiri atas notasi yaitu *initial node*, *action*, *flow*, *decision*, *merge*, *fork*, *join* dan *activity final*. Untuk mempartisi aksi yang terjadi berdasarkan pengguna dapat menggunakan *swimlane*.

Berikut ini adalah simbol-simbol dalam *activity diagram* terdapat pada Tabel 2.8.

Tabel 2.8 Simbol-simbol Activity Diagram

No	Simbol	Nama	Keterangan
1		<i>Initial node</i>	<i>Initial node</i> berbentuk lingkaran penuh sebagai awal suatu proses dijalankan.
2		<i>Actions</i>	<i>Actions</i> berbentuk oval yang mengilustrasikan urutan kejadian.

Tabel 2.8 Simbol-simbol Activity Diagram (Lanjutan)

No	Simbol	Nama	Keterangan
3		Flow (alur)	Flow berbentuk panah sebagai alur antar-actions.
4		Decision	Decision memiliki satu alur masuk dan dua atau lebih alur keluar sesuai situasi kejadian.
5		Merge	Merge memiliki dua atau lebih alur masuk dan satu alur keluar untuk penggabungan alur yang terpisah.
6		Fork	Fork digunakan untuk memisahkan aktifitas yang memiliki dua keluaran.
7		Join	Join digunakan untuk menyatukan dua aktifitas yang memiliki keluaran yang sama setelah dipisahlan oleh fork.
8		Activity final	Activity final sebagai tanda proses telah selesai.



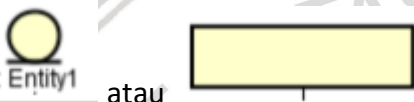





Sumber: (Whitten & Bentley, 2007)

2.10.4 Sequence Diagram

Sequence diagram menunjukkan hubungan antara objek selama menjalankan sistem (Pressman, 2010). Penggambaran ini menotasikan alur algoritma atau pesan yang berjalan pada sistem. Pesan dapat berupa sinyal atau panggilan terhadap suatu operasi. Tabel 2.10 merupakan simbol-simbol dari sequence diagram.



Tabel 2.9 Simbol-simbol pada Sequence Diagram

Simbol	Deskripsi
	Aktor, orang, individu, proses atau sistem lain yang berhubungan dengan sistem nantinya.
	Lifelines digambarkan garis vertikal terputus yang mendefinisikan urutan kejadian.
	Entity memiliki atribut data yang dapat menyimpan perubahan data.
	Boundary menggambarkan tampilan sistem yang menghubungkan antara pengguna dan sistem.
	Control menggambarkan aktivitas yang dijalankan.
	Message digambarkan oleh panah dari kiri ke kanan untuk mengirimkan pesan.
	Return Values digambarkan dengan garis panah putus-putus dari arah kanan ke arah kiri sebagai hasil dari pengiriman pesan.
	Bar aktivasi menggambarkan periode waktu yang dibutuhkan ketika mengirimkan pesan.

Sumber: (Rosa & Shalahudin, 2016)

2.11 Physical Data Model

Physical Data Model (PDM) menggambarkan relasi tabel pada database dengan menggambarkan tabel, kolom dan atribut (Ambysoft Inc., 2003). Pemodelan data dapat dibuat berdasarkan ketentuan sebagai berikut (Booch, et al., 1998):

1. Mendefinisikan setiap kelas untuk berupa *physical data model*. Strategi ini memiliki kekurangan yakni dapat menemukan kesulitan ketika rancangan *class diagram* ini nantinya diperlukan sebuah perubahan.
2. Berikutnya dengan mengelompokkan kelas turunan dan induk menjadi tingkatan sejajar. Selanjutnya setiap kelas ini akan didefinisikan menjadi *physical data model*. Namun, memiliki kekurangan yakni berdampak pada kapasitas penyimpanan yang akan cepat berkurang.
3. Strategi terakhir adalah dengan memisahkan subclass dengan superclass dan mengidentifikasi dalam *physical data model*. Tetapi memiliki kekurangan yakni memerlukan proses dalam pengambilan data melalui basis data menggunakan mekanisme *join table*.

2.12 Implementasi

Implementasi dalam pengembangan Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* akan menggunakan *framework CodeIgniter* dengan basis data MySQL dan PHP sebagai bahasa pemrogramannya serta menggunakan konsep pemrograman berorientasi objek.

2.12.1 PHP

Hypertext Preprocessor (PHP) merupakan bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat ditambahkan ke HTML (Supono, 2018).

Penggunaan PHP memiliki beberapa keunggulan diantaranya (Solichin, 2016)

1. PHP dapat diunduh dan digunakan secara gratis
2. PHP berlisensi GNU *General Public License* (GPL) yaitu semua versi PHP selalu didistribusikan secara gratis.
3. Performa handal dengan spesifikasi *server* yang tidak mahal, dapat melajani jutaan pengaksesan.
4. Dukungan basisdata karena mendukung penggunaan MySQL, Oracle, Informix, Sybase hingga SQLite.
5. Menyediakan pustaka bawaan yang berkaitan dengan web. PHP memiliki pustaka untuk menampilkan PDF langsung di *browser*, mengelola *file*, mengenkripsi *file*, komperesi dan penanganan *sessions*.
6. PHP dapat dijalankan hampir di semua *platform* seperti Linux, Unix, Windows, Mac OS, dan bahkan dapat berjalan di sistem Android melalui proyek DroidPHP.
7. Perintah mudah dipelajari karena diadopsi dari Bahasa pemrograman populer seperti Java, Perl, C dan C++.

2.12.2 MVC

Merupakan salah satu dari *Architectural Design Pattern*. Saat ini banyak kerangka kerja yang menggunakan konsep ini dengan dipadukan pemrograman berorientasi objek.

1. Model

Bertugas menghubungkan projek dengan basis data dengan menggunakan fungsi-fungsi seperti membuat, menghapus, memperbarui data serta mencari dan mengambil data (Hidayatullah & Kawistara, 2014).

2. View

Memuat halaman HTML, CSS, *javascript*, *ajax* dan lain-lain yang memiliki hubungan langsung dengan pengguna, pengguna akan berinteraksi dengan sistem melalui *view* (Hidayatullah & Kawistara, 2014).

3. Controller

Penghubung sekaligus pemegang kendali terhadap model dan view yang bertugas memproses data atau alur logika serta *error handler*, validasi atau *check* terhadap beberapa masukan (Hidayatullah & Kawistara, 2014).

2.12.3 MySQL

My Structure Query Language (MySQL) berupa suatu sistem kerap kali dimanfaatkan dalam pengelolaan *database*. MySQL dapat dikategorikan menjadi *Database Management System* (DBMS) (Anhar, 2010)

Database dimanfaatkan agar dapat menyimpan data pada sistem ini. MySQL sesuai untuk diterapkan ini karena dapat disimpan pada *server* sehingga data yang ada, bisa digunakan oleh *multi-user* dalam internal perusahaan.

2.13 CodeIgniter

CodeIgniter merupakan *framework* web yang menggunakan bahasa pemrograman PHP. *Framework* ini dapat mempermudah *developer* untuk mengembangkan sistem yang telah dibangun dengan berbasis web (Raharjo, 2015). Kelebihan penggunaan *framework* di antaranya seperti di bawah ini:

1. *CodeIgniter* merupakan *framework open source*.
2. *CodeIgniter* menggunakan pola desain Model View Controller (MVC), sehingga satu file tidak berisi banyak code dan membuat kode lebih mudah dibaca, dipahami dan dipelihara kedepannya.
3. *CodeIgniter* memiliki ukuran yang kecil jika dibandingkan dengan *framework* lainnya. *CodeIgniter* yang memiliki ukuran kurang lebih 2 MB dan Dokumentasinya juga hanya memiliki ukuran 6 MB.
4. Hasil pembuatan sistem dapat dieksekusi lebih efektif, efisien serta dikembangkan berdasarkan keperluan.
5. Menyediakan dokumentasi tentang kelas dan fungsi pada paket distribusi.

2.14 Cloud Computing

Cloud computing merupakan sebuah paradigma dimana suatu informasi dapat disimpan secara permanen dalam sebuah *server* di internet dan tersimpan secara sesaat pada komputer klien termasuk juga didalamnya terdapat *desktop*, komputer *tablet*, *notebook*, *handheld*, sensor-sensor, dan sebagainya, sehingga dapat menghemat dari segi tempat, biaya dan *hardware* (Hewitt, 2008). *Cloud computing* sangat bergantung pada internet, karena semua proses komputasinya berada pada jaringan internet. Menurut (Namasudra, et al., 2017) *cloud computing* memiliki beberapa keuntungan yaitu:

1. *Reduced IT cost.*

Cloud computing dapat mengurangi biaya pengelolaan dan pemeliharaan sistem IT, pengguna dapat menggunakan sumber daya yang disediakan oleh penyedia layanan cloud tanpa perlu membeli sistem yang mahal.

2. *Scalability.*

Dalam *cloud computing* sistem dapat meningkatkan dan menurunkan sumber daya sesuai dengan situasi dan kebutuhan pengguna.

3. *Business continuity.*

Bagian terpenting dari perencanaan keberlangsungan bisnis adalah untuk melindungi data dan sistem. Dalam beberapa kasus, data mungkin menghadapi banyak bencana alam, data harus disimpan di *server cloud* saat dicadangkan.

4. *Flexibility of work practices.*

Cloud computing memungkinkan pengguna lebih fleksibel, dan juga memungkinkan pelanggan untuk mengakses data dari mana saja.

5. *Almost unlimited storage.*

Dalam *cloud computing*, pengguna tidak perlu khawatir tentang ruang penyimpanan karena terdapat ruang penyimpanan yang hampir tidak terbatas pada *cloud computing*.

6. *Speed.*

Pengguna dapat mengakses *server* dimana saja dengan cepat, dengan catatan pengguna telah tersambung pada jaringan internet.

Cloud computing memiliki tiga jenis layanan, yaitu *software as a service* (SaaS), *platform as a service* (PaaS), dan *infrastructure as a service* (IaaS) :

1. *Software as a Service (SaaS)*

SaaS *cloud computing* merupakan layanan dimana pengguna dapat menggunakan layanan berupa perangkat lunak siap pakai yang disediakan oleh penyedia layanan *cloud*. Pengguna cukup tau bahwa layanan bisa berjalan dan bisa digunakan dengan baik. Contoh layanan SaaS adalah Gmail, Facebook, Adobe Creative Cloud dan masih banyak lagi.

2. Platform as a Service (PaaS)

Paas *cloud computing* merupakan layanan penyewaan server dan lingkungannya serta pemeliharannya, pengguna cukup fokus pada aplikasi yang sedang dikembangkan tanpa perlu memikirkan pemeliharaan *server* karena hal tersebut sudah menjadi tanggung jawab dari penyedia layanan *cloud*.

3. Infrastructure as a Service (IaaS)

IaaS *cloud computing* merupakan layanan penyewaan infrastruktur IT (unit komputasi, *storage*, *memory*, *network*, dan sebagainya). Dalam layanan ini pengguna dapat mendefinisikan sendiri berapa besar unit yang akan disewa, sehingga pengguna seperti menyewa sebuah komputer kosong yang belum dikonfigurasi. Contoh penyedia layanan IaaS adalah AWS EC2, Rackspace Cloud, Windows Azure dan sebagainya.

2.15 Amazon Web Service Cloud

Amazon.com awalnya merupakan sebuah toko buku *on-line*, namun beberapa tahun kemudian Amazon mengembangkan AWS (*Amazon Web Service*) yang menyediakan layanan *cloud computing*, yang nantinya setiap fungsi dapat diakses dengan panggilan *web services*. AWS *Cloud* adalah salah satu pendahulu dari konsep *cloud computing* bersama Google lalu diikuti oleh Salesforce. Salah satu layanan AWS *cloud* yang akan digunakan pada penelitian ini adalah AWS EC2 yakni layanan web yang memberikan kapasitas komputasi yang aman dan berukuran fleksibel di *cloud*. AWS EC2 merupakan jenis layanan *cloud computing* IaaS (*Infrastructure as a Service*) yang dirancang agar *cloud computing* berskala web lebih mudah bagi pengembang.

Beberapa kelebihan saat menggunakan layanan AWS EC2 (Amazon Web Services, 2019) sebagai berikut:

1. Akses penyebaran cepat, pengguna dapat mengkonfigurasi sebuah infrastruktur dilakukan dalam hitungan menit.
2. Biaya rendah, AWS menyediakan pola pembayaran dimana pengguna hanya membayar apa yang digunakan saja.
3. Aman, AWS menjamin dan melindungi data dan privasi penggunanya.
4. Mudah dalam mengukur penggunaan sumber daya sehingga menghindari kelebihan atau kekurangan sumber daya.

2.16 Pengujian

Kebutuhan fungsional dan non-fungsional yang telah didefinisikan akan dilakukan pengujian sebagai dasar dari verifikasi dan validasi apakah sistem sudah sesuai dengan analisis kebutuhan. Menurut (Pressman, 2010) Pengujian ini dilakukan untuk mengetahui seberapa banyak kesalahan yang mungkin timbul saat sistem dirancang dan dikembangkan. Pengujian fungsional dan non-fungsional

merupakan metode atau teknik pengujian yang akan digunakan untuk menguji sistem informasi *point of sales* untuk bengkel.

2.16.1 Pengujian Fungsional

Pengujian ini akan menguji perangkat lunak dari segi kebutuhan fungsional tanpa menguji desain dan kode program. Pengujian ini dilakukan apabila sistem telah selesai dikembangkan, dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi. Uji validasi dilakukan dengan membuat kasus uji yang akan mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi. Kasus uji yang dibuat untuk melakukan uji validasi harus dibuat dengan menyertakan kasus salah tidak hanya kasus benar saja, misalnya untuk kasus proses *login* maka kasus uji yang akan dibuat harus seperti berikut (Rosa & Shalahudin, 2016).

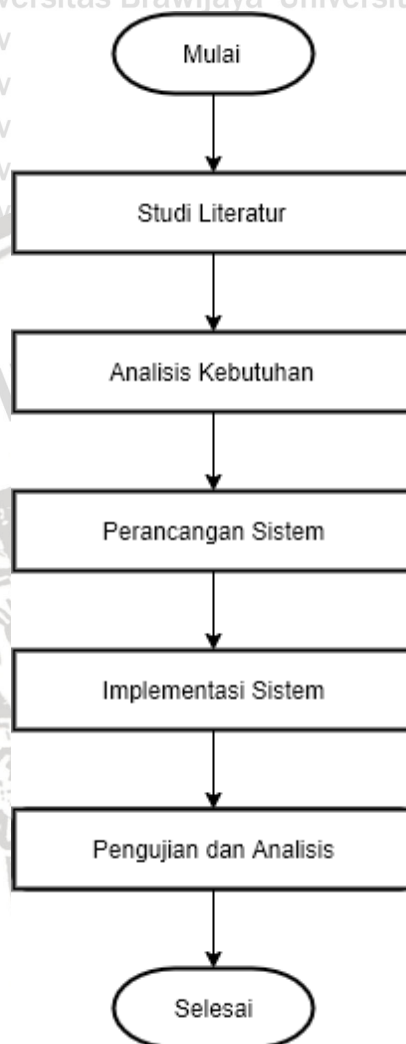
1. Jika pengguna memasukkan *username* dan *password* yang benar.
2. Jika pengguna memasukkan *username* dan *password* yang salah, misalkan *username* benar dan *password* salah atau sebaliknya atau keduanya salah.

2.16.2 Pengujian Non-Fungsional

Pengujian non-fungsional merupakan pengujian terhadap kebutuhan non-fungsional yang telah didefinisikan pada analisis kebutuhan. Pengujian ini sebenarnya adalah mengidentifikasi penelusur web yang digunakan, Pengaksesan koneksi internet, waktu eksekusi proram serta beberapa informasi sejenisnya. Dalam penelitian ini akan dilakukan serangkaian pengujian validasi kompatibilitas yang menganalisa kesesuaian antarmuka terhadap beberapa penelusur web yang umum digunakan pengguna. Target utama dari pengujian ini untuk menemukan kesalahan atau masalah eksekusi yang dapat ditelusuri melalui berbagai konfigurasi (Pressman, 2010).

BAB 3 METODOLOGI

Metodologi membahas runtutan alur sebagai pedoman untuk merancang dan membangun Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing*. Gambar 3.1 adalah diagram alir tahapan metode penelitian :



Gambar 3.1 Diagram Alir Metode Penelitian Sistem Informasi Point of Sales Untuk Bengkel

Sumber: Penulis (2019).

3.1 Studi Literatur

Penerapan pencarian studi literatur dilakukan agar dapat lebih mengetahui secara mendalam terkait penelitian yang dilakukan untuk menjadi dasar memulai penelitian. Sehingga dari tahap ini dapat mencapai tujuan penelitian yang diharapkan. Dalam tahapan ini, melakukan pengkajian dengan cara memahami berbagai sumber penelitian yang pernah dilakukan sebelumnya. Referensi pada penelitian ini berdasarkan pencarian data melalui jurnal dan buku sesuai dengan topik penelitian yang akan dilakukan.

3.2 Analisis Kebutuhan

Membuat analisa data dari perolehan wawancara tujuannya agar memenuhi persyaratan sistem yang akan dibuat. Di samping itu untuk menggambarkan prosedur bisnis. Selanjutnya akan mengidentifikasi persyaratan fungsional dan nonfungsional sistem. Dari identifikasi tersebut, selanjutnya menggambarkan *use case* agar dapat menunjukkan hubungan antara sistem beserta aktor yang menjalankannya.

3.3 Perancangan

Merancang sistem berdasarkan hasil analisa dalam metode sebelumnya. Tujuan dari tahap ini untuk memperoleh penggambaran perancangan *activity diagram*, *sequence diagram*, *class diagram*, *physical data model* dan perancangan tampilan antarmuka untuk pengguna.

3.4 Implementasi

Pada tahap ini sistem akan dilakukan proses pembuatan Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* yang sesuai dengan perancangan sistem yang telah dibuat sebelumnya.

3.4.1 Implementasi Point of Sales

Sistem dikembangkan dengan menggunakan *framework CodeIgniter* dan bahasa pemrograman PHP. Hasil dari proses ini adalah Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* yang berisi fitur-fitur sesuai dengan tahap analisis persyaratan yang telah dibuat sebelumnya. Dalam implementasi nantinya akan digunakan bahasa pemrograman *Javascript* sebagai bahasa penunjang untuk mengatur *view* atau tampilan sistem.

3.4.2 Implementasi Cloud Computing

Pada tahap ini setelah sistem telah selesai dibangun, kemudian akan dilanjutkan dengan menghubungkannya pada sebuah *cloud computing* yang telah dikonfigurasi dan siap sebagai web server agar sistem dapat diakses melalui internet. Pada penelitian ini akan menggunakan AWS EC2 sebagai layanan *cloud computing* dan Ubuntu *server* sebagai sistem operasi yang akan menunjang berjalannya sistem.

3.5 Pengujian

Pengujian yang akan dilakukan akan menggunakan pengujian fungsional dan pengujian non-fungsional. Pengujian fungsional dilakukan menggunakan pengujian validasi dengan meminta calon pengguna untuk mencoba sistem sehingga dapat diketahui kekurangan yang mungkin tidak dapat ditemukan oleh pengembang. Sedangkan pengujian non-fungsional menggunakan pengujian kompatibilitas yang dilakukan untuk mengetahui apakah sistem telah

dikembangkan sudah sesuai apabila dijalankan pada penelusur web yang berbeda-beda.

3.6 Kesimpulan dan Saran

Mengambil beberapa poin penting untuk menarik kesimpulan dari seluruh penelitian yang telah dilakukan. Selanjutnya saran akan ditulis agar melakukan perbaikan kekurangan pada penelitian serta kedepannya dapat dijadikan pedoman untuk mengembangkan dan perbaikan kelemahan.



BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM

Tahap dalam melakukan pengumpulan data dan analisis persyaratan. Analisis data berdasarkan pemodelan proses bisnis yang berjalan di Bengkel Mas Pur Baturaja untuk mendapatkan data atau informasi untuk merancang Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing*. Setelah itu, dilakukan analisis kebutuhan dari data yang telah di dapatkan agar mengetahui kebutuhan yang ingin dicapai dalam merancang sistem.

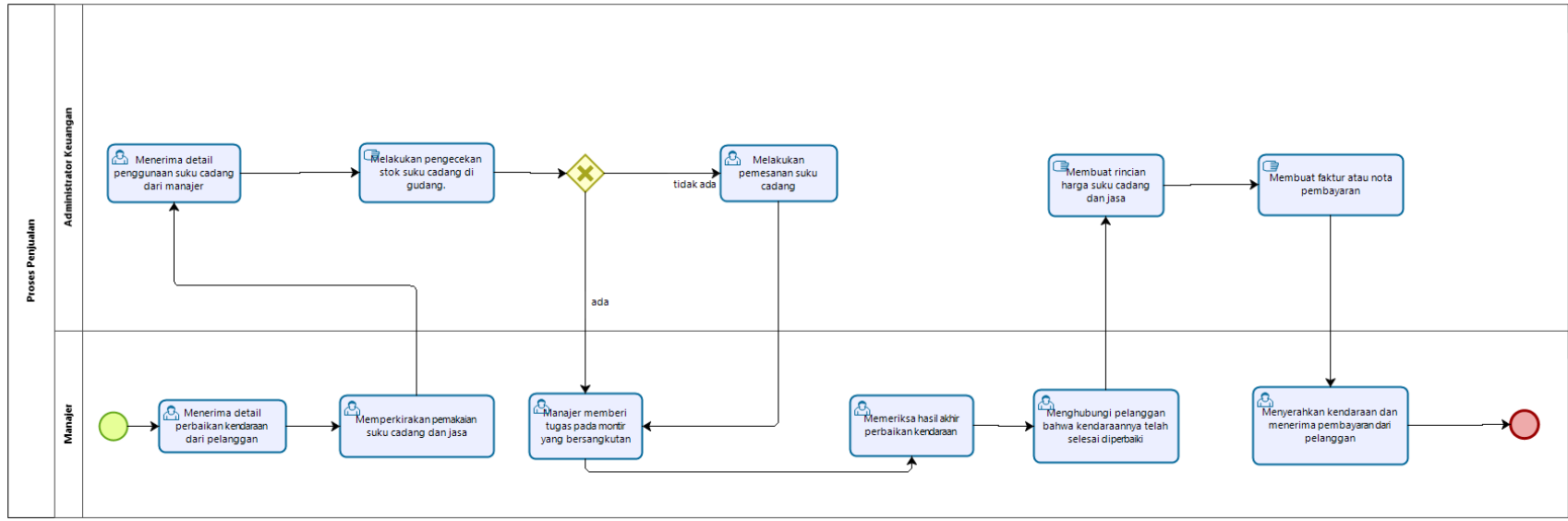
4.1 Pemodelan Proses Bisnis

Proses penggalian informasi dilaksanakan dengan tujuan agar mendapatkan data terkait dengan penerapan prosedur bisnis pada Bengkel Mas Pur Baturaja. Prosedur yang dilakukan merupakan proses dalam hal penjualan. Dengan mengilustrasikan jalannya bisnis melalui penggunaan *Business Process Model and Nation* (BPMN) agar urutan proses bisnis dapat lebih mudah dimengerti. Dalam pemodelan BPMN ini akan dimodelkan dalam dua kategori yakni pemodelan *As-Is* serta *To-Be*.

4.1.1 Pemodelan Proses Bisnis As-Is

Pemodelan ini adalah penggambaran prosedur bisnis yang sedang diterapkan dalam Bengkel Mas Pur Baturaja. Berikut merupakan BPMN dari proses penjualan yang dapat dilihat pada Gambar 4.1

1. Manajer menerima detail perbaikan dari pelanggan, bagian kendaraan mana yang akan diperbaiki.
2. Selanjutnya Manajer memperkirakan pemakaian suku cadang dan jasa yang akan digunakan dalam perbaikan kendaraan dan menyerahkan pada administrator keuangan untuk dicek stoknya.
3. Administrator Keuangan menerima detail penggunaan suku cadang dari Manajer.
4. Selanjutnya Administrator Keuangan melakukan pengecekan stok suku cadang di gudang perusahaan.
5. Jika tidak tersedia maka Administrator Keuangan akan melakukan pemesanan suku cadang, setelah tersedia maka Manajer akan menugaskan montir pada bidang jasa yang ditentukan diawal.
6. Setelah kendaraan selesai diperbaiki maka Manajer akan memeriksa hasil akhir perbaikan apakah sudah sesuai dan mencapai standar kualitas perusahaan.
7. Manajer menghubungi pelanggan bahwa kendaraannya telah diperbaiki.
8. Administrator Keuangan akan membuat rincian harga suku cadang dan jasa.
9. Administrator Keuangan membuat faktur atau nota pembayaran sebagai tanda pembayaran yang sah dari perusahaan.
10. Manajer menyerahkan kendaraan dan menerima pembayaran dari pelanggan.



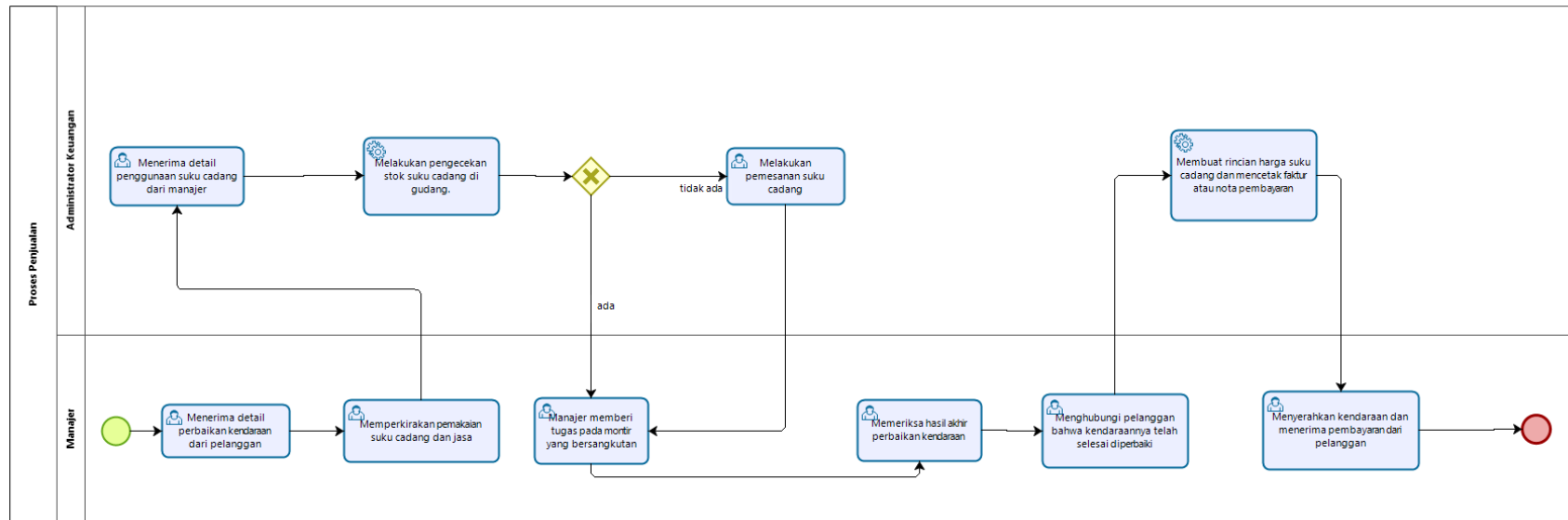
Gambar 4.1 Proses Bisnis Saat ini Penjualan
 Sumber: Penulis (2019).



4.1.2 Pemodelan Proses Bisnis To-Be

Proses bisnis *to-be* dibuat berdasarkan pemodelan proses bisnis yang berjalan saat ini pada perusahaan Bengkel Mas Pur yang masih terdapat beberapa aktivitas manual. Oleh karena itu, peneliti mengusulkan proses bisnis usulan atau *to-be*, peneliti akan memperbaiki beberapa aktivitas yang masih dilakukan secara manual tersebut. Berikut merupakan BPMN proses bisnis *to-be* dari proses penjualan yang dapat dilihat pada Gambar 4.2

1. Manajer menerima detail perbaikan dari pelanggan, bagian kendaraan mana yang akan diperbaiki.
2. Selanjutnya Manajer memperkirakan pemakaian suku cadang dan jasa yang akan digunakan dalam perbaikan kendaraan dan menyerahkan pada administrator keuangan untuk dicek stoknya.
3. Administrator Keuangan menerima detail penggunaan suku cadang dari Manajer.
4. Selanjutnya Administrator Keuangan melakukan pengecekan stok suku cadang di gudang perusahaan.
5. Jika tidak tersedia maka Administrator Keuangan akan melakukan pemesanan suku cadang, setelah tersedia maka Manajer akan menugaskan montir pada bidang jasa yang ditentukan diawal.
6. Setelah kendaraan selesai diperbaiki maka Manajer akan memeriksa hasil akhir perbaikan apakah sudah sesuai dan mencapai standar kualitas perusahaan.
7. Manajer akan menghubungi pelanggan bahwa kendaraannya telah selesai diperbaiki.
8. Administrator Keuangan akan membuat rincian harga suku cadang dan jasa dan mencetak faktur atau nota pembayaran sebagai tanda pembayaran yang sah dari perusahaan.
9. Manajer menyerahkan kendaraan dan menerima pembayaran dari pelanggan.



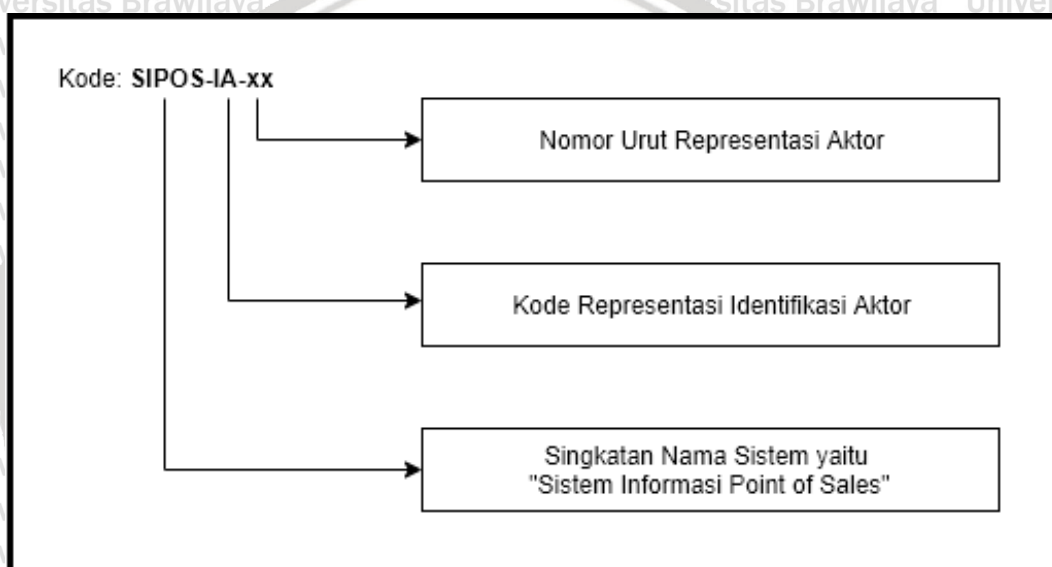
Gambar 4.2 Proses Bisnis Usulan Penjualan

Sumber: Penulis (2019).

4.2 Analisis Kebutuhan

4.2.1 Identifikasi Aktor

Pada bagian ini menjelaskan beberapa jenis aktor yang terlibat pada sistem. Setiap aktor akan memiliki fungsionalnya masing-masing dan diberikan aturan penomoran. Aturan penomoran akan dijelaskan pada Gambar 4.3. Aturan penomoran terdiri dari tiga kata, kata pertama merupakan singkatan dari nama sistem yaitu "Sistem Informasi *Point of Sales*". Kata kedua adalah representasi dari identifikasi aktor, dan kata ketiga merupakan representasi dari nomor urut identifikasi aktor.



Gambar 4.3 Aturan Penomoran Identifikasi Aktor

Sumber: Penulis (2019).

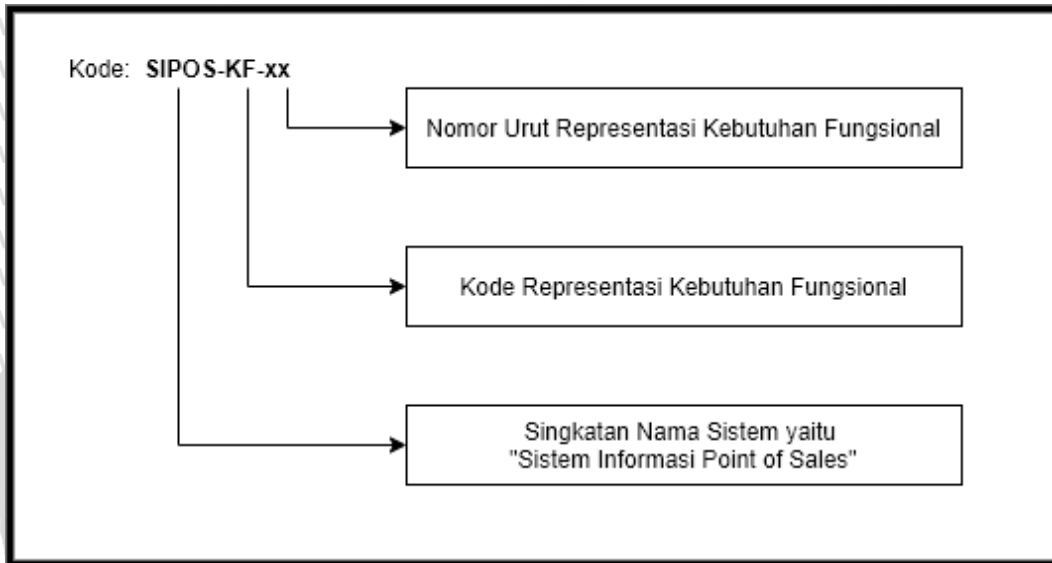
Identifikasi aktor dilakukan dengan menggunakan aturan penomoran yang dapat dijelaskan pada Tabel 4.1, sistem ini nantinya akan memiliki dua aktor yaitu Administrator Keuangan dan Manajer.

Tabel 4.1 Identifikasi Aktor

No.	Kode Aktor	Nama Aktor	Deskripsi
1	SIPOS-IA-01	Administrator Keuangan	Administrator keuangan merupakan seseorang yang memiliki tanggung jawab pada keuangan perusahaan
2	SIPOS-IA-02	Manajer	Manajer merupakan seseorang yang bertanggung jawab penuh terhadap jalannya perusahaan.

4.2.2 Kebutuhan Fungsional

Kebutuhan fungsional sistem akan dibagi berdasarkan jenis aktor yang terlibat. Pada masing – masing penjelasan kebutuhan fungsional diberikan aturan penomoran yang terdapat pada Gambar 4.4. Aturan penomoran terdiri dari tiga kata, kata pertama merupakan singkatan dari nama sistem yaitu “*Sistem Informasi Point of Sales*”. Kata kedua adalah representasi dari kebutuhan fungsional, dan kata ketiga merupakan representasi dari nomor urut kebutuhan fungsional.



Gambar 4.4 Aturan Penomoran Kebutuhan Fungsional

Sumber: Penulis (2019).

Kebutuhan fungsional pada Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* dapat dibagi menjadi dua yaitu kebutuhan fungsional aktor Administrator Keuangan ditunjukkan pada Tabel 4.2.

Tabel 4.2 Kebutuhan Fungsional Sistem Aktor Adminstrasi Keuangan

Kode Kebutuhan	[SIPOS-IA-01] Administrator Keuangan	
	Kebutuhan Fungsional	Deskripsi
SIPOS-KF-01	Melakukan <i>Login</i>	Sistem dapat mengidentifikasi pengguna sistem dan dapat memberikan batas akses pengguna terhadap menu sistem yang disediakan berdasarkan hak akses yang dimiliki pengguna.

Tabel 4.2 Kebutuhan Fungsional Sistem Aktor Adminstrasi Keuangan(Lanjutan)

Kode Kebutuhan	[SIPOS-IA-01] Administrator Keuangan	
	Kebutuhan Fungsional	Deskripsi
SIPOS-KF-02	Memasukan data Penjualan	Sistem dapat menambahkan data penjualan dan mencetak nota atau faktur pembayaran.
SIPOS-KF-03	Memasukan data Pembelian	Sistem dapat menambahkan data pembelian suku cadang.
SIPOS-KF-04	Memasukan data Gaji Karyawan	Sistem dapat menambahkan data penggajian karyawan perbulannya.
SIPOS-KF-05	Melihat Stok Suku Cadang	Sistem dapat menampilkan stok suku cadang yang ada di gudang berdasarkan data pembelian.
SIPOS-KF-06	Melihat laporan Pemasukan	Sistem dapat menampilkan laporan pemasukan perusahaan berdasarkan data penjualan.
SIPOS-KF-07	Melihat laporan Pengeluaran	Sistem dapat menampilkan laporan pengeluaran perusahaan berdasarkan pembelian suku cadang dan gaji karyawan.
SIPOS-KF-08	Melakukan Logout	Sistem dapat menghapus <i>session</i> dari pengguna yang telah <i>logout</i> .

Sumber: Penulis (2019).

Kebutuhan fungsional sistem aktor manajer dijelaskan Tabel 4.3, terdapat lima kebutuhan fungsional untuk aktor manajer.

Tabel 4.3 Kebutuhan Fungsional Sistem Aktor Manajer

Kode Kebutuhan	[SIPOS-IA-02] Manajer	
	Kebutuhan Fungsional	Deskripsi
SIPOS-KF-01	Melakukan <i>Login</i>	Sistem dapat mengidentifikasi pengguna sistem dan dapat memberikan batas akses pengguna terhadap menu sistem yang disediakan berdasarkan hak akses yang dimiliki pengguna.
SIPOS-KF-05	Melihat Stok Suku Cadang	Sistem dapat menampilkan stok suku cadang yang ada di gudang berdasarkan data pembelian.

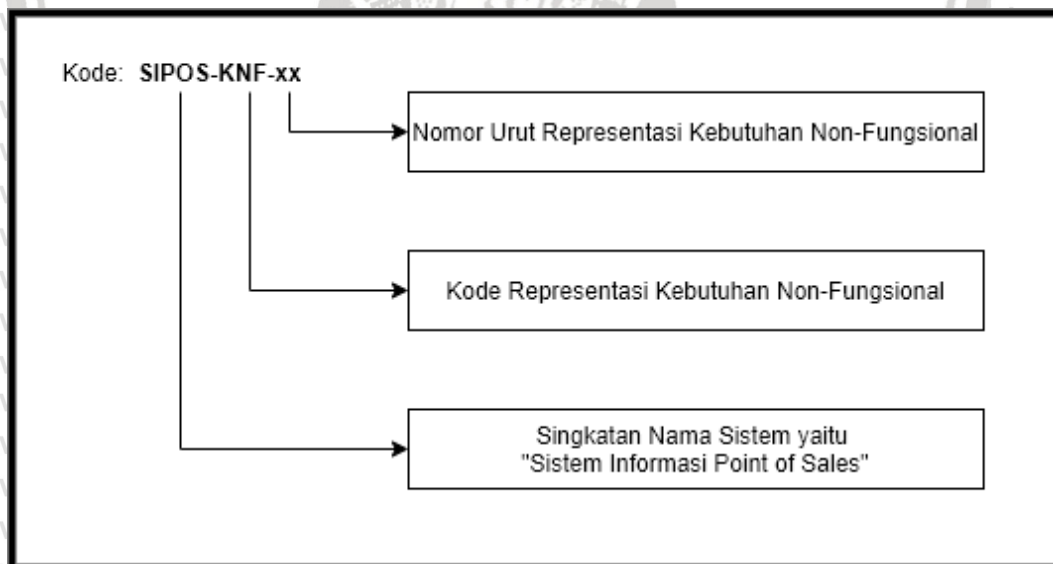
Tabel 4.3 Kebutuhan Fungsional Sistem Aktor Manajer (Lanjutan)

Kode Kebutuhan	[SIPOS-IA-02] Manajer	
	Kebutuhan Fungsional	Deskripsi
SIPOS-KF-06	Melihat laporan Pemasukan	Sistem dapat menampilkan laporan pemasukan perusahaan berdasarkan data penjualan.
SIPOS-KF-07	Melihat laporan Pengeluaran	Sistem dapat menampilkan laporan pengeluaran perusahaan berdasarkan pembelian suku cadang dan gaji karyawan.
SIPOS-KF-08	Melakukan <i>Logout</i>	Sistem dapat menghapus <i>session</i> dari pengguna yang telah <i>logout</i> .

Sumber: Penulis (2019).

4.2.3 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* akan menggunakan aturan penomoran seperti pada Gambar 4.5. Aturan penomoran terdiri dari tiga kata, kata pertama merupakan singkatan dari nama sistem yaitu "*Sistem Informasi Point of Sales*". Kata kedua adalah representasi dari kebutuhan non-fungsional, dan kata ketiga merupakan representasi dari nomor urut kebutuhan non-fungsional.



Gambar 4.5 Aturan Penomoran Kebutuhan Non-Fungsional

Sumber: Penulis (2019).



Kebutuhan non-fungsional pada sistem ini adalah *compatibility*, yang akan dijelaskan pada Tabel 4.4

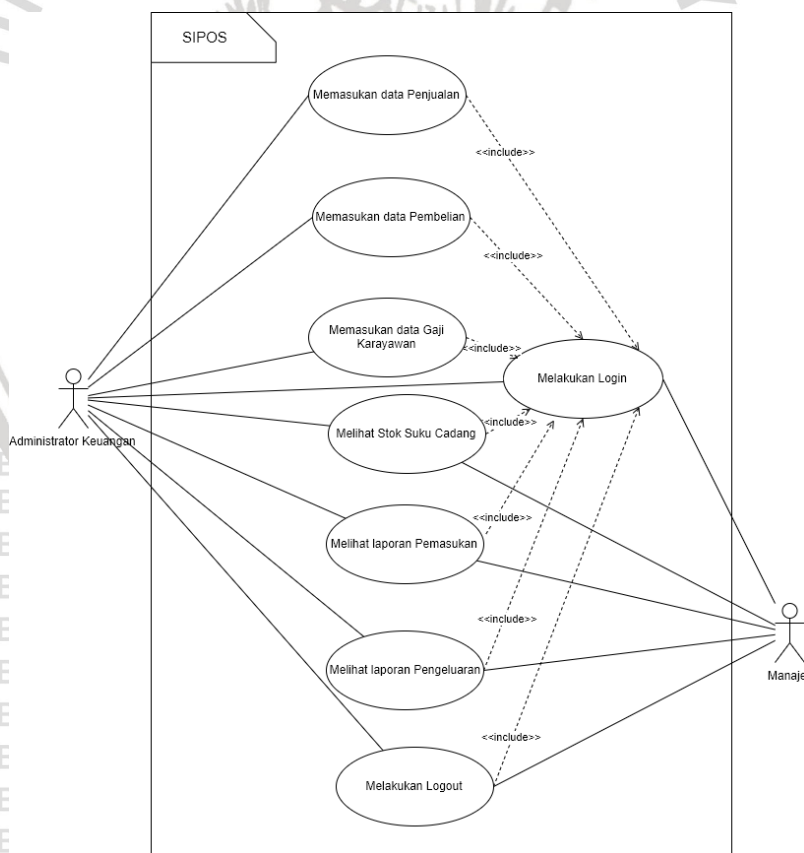
Tabel 4.4 Kebutuhan Non-Fungsional

No.	Kode Kebutuhan	Kebutuhan Non-Fungsional	Deskripsi
1.	SIPOS-KNF-01	<i>Compatibility</i>	Sistem dapat diakses pada penelusur web seperti Google Chrome, Mozilla Firefox, Internet Explore, Microsoft Edge dan Opera.

Sumber: Penulis (2019).

4.2.4 Pemodelan Use Case

Pada pemodelan *use case* akan diberikan gambaran mengenai siapa saja pengguna yang dapat menggunakan sistem dan apa saja yang dapat pengguna lakukan didalam sistem tersebut. Pada sistem ini terdapat delapan *use case*, dimana setiap *use case* memiliki hubungan *include* terhadap *use case* melakukan *login*. Berikut *use case* Sistem Informasi *Point of Sales* untuk Bengkel Berbasis *Cloud Computing* dijelaskan pada Gambar 4.6.



Gambar 4.6 Use Case Sistem Informasi Point of Sales untuk Bengkel

Sumber: Penulis (2019).

4.3 Spesifikasi Use Case

Spesifikasi *use case* dibuat untuk memperjelas mengenai alur dalam setiap *use case*. Tabel spesifikasi *use case* terdiri dari aktor yang menjalankan *use case* tersebut, deskripsi dari *use case*, tujuan dari *use case*, kondisi sebelum *use case*, alur utama dari *use case*, alur alternatif dari *use case*, dan kondisi setelah *use case* dijalankan.

4.3.1 Spesifikasi Use Case Melakukan Login

Spesifikasi *use case* melakukan *login* yang terdapat pada Tabel 4.5 menjelaskan terkait *use case* melakukan *login* yang dilakukan oleh semua aktor pada sistem informasi *point of sales*.

Tabel 4.5 Spesifikasi Use Case Melakukan Login

Spesifikasi Use Case Melakukan Login	
<i>Actor</i>	Administrator Keuangan, Manajer.
<i>Description</i>	Skenario untuk melakukan <i>login</i> .
<i>Use case Goal</i>	Aktor dapat melakukan <i>login</i> .
<i>Preconditions</i>	Aktor mengakses sistem pada penelusur web.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengakses sistem pada penelusur web. 2. Sistem menampilkan halaman <i>login</i>. 3. Aktor memasukan <i>username</i> dan <i>password</i>. 4. Sistem mengidentifikasi akun aktor. 5. Sistem menampilkan <i>dashboard</i>.
<i>Alternative Flow</i>	<p>Mengatasi kegagalan nomor 3.</p> <ol style="list-style-type: none"> 1. Jika <i>username</i> atau <i>password</i> salah maka sistem akan menampilkan pemberitahuan.
<i>PostCondition</i>	Aktor berhasil <i>login</i> kedalam sistem.

Sumber: Penulis (2019).

Pada Tabel 4.5 menjelaskan mengenai *use case* Melakukan *Login*, dimana aktor harus memasukan *username* dan *password* yang benar agar dapat masuk kedalam sistem.

4.3.2 Spesifikasi Use Case Memasukan Data Penjualan

Spesifikasi *use case* Memasukan Data Penjualan akan dijelaskan pada Tabel 4.6.

Tabel 4.6 Spesifikasi Use Case Memasukan Data Penjualan

Spesifikasi <i>Use Case</i> Memasukan Data Penjualan	
<i>Actor</i>	Administrator Keuangan.
<i>Description</i>	Skenario untuk memasukan data penjualan.
<i>Use case Goal</i>	Aktor dapat memasukan data penjualan.
<i>Preconditions</i>	Aktor telah <i>login</i> kedalam sistem.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu penjualan. 2. Sistem menampilkan halaman penjualan. 3. Aktor memasukan data penjualan, input suku cadang dan jasa lalu klik tombol tambah. 4. Aktor selesai memasukan barang lalu memasukkan jumlah uang yang diterima dari pelanggan, lalu klik tombol simpan. 5. Sistem menampilkan modal konfirmasi kemudian klik tombol konfirmasi. 6. Sistem menyimpan data. 7. Aktor klik tombol cetak. 8. Sistem menampilkan modal cetak. 9. Aktor memilih metode pencetakan bukti pembayaran, nota atau faktur. 10. Sistem menampilkan <i>print preview</i> dari bukti pembayaran. 11. Aktor mencetak bukti pembayaran.
<i>Alternative Flow</i>	-
<i>PostCondition</i>	Aktor mencetak bukti pembayaran untuk pelanggan.

Sumber: Penulis (2019).

Pada Tabel 4.6 dijelaskan mengenai *use case* Memasukan Penjualan, actor memasukkan setiap data penjualan yang terdiri dari data dari penjualan, suku cadang yang terjual dan jasa yang dipakai. Setelah data disimpan, aktor dapat mencetak bukti pembayaran.

4.3.3 Spesifikasi Use Case Memasukan Data Pembelian

Spesifikasi *use case* memasukan data pembelian akan dijelaskan pada Tabel 4.7.

Tabel 4.7 Spesifikasi Use Case Memasukan Data Pembelian

Spesifikasi <i>Use Case</i> Memasukan Data Pembelian	
<i>Actor</i>	Administrator Keuangan.
<i>Description</i>	Skenario untuk memasukan data pembelian.
<i>Use case Goal</i>	Aktor dapat memasukan data pembelian.
<i>Preconditions</i>	Aktor telah login kedalam sistem.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu pembelian. 2. Sistem menampilkan halaman pembelian. 3. Aktor memasukan data pembelian, input suku cadang lalu klik tombol tambah. 4. Aktor memasukan barang, lalu klik tombol simpan. 5. Sistem menampilkan modal konfirmasi kemudian klik tombol konfirmasi, sistem menyimpan data.
<i>Alternative Flow</i>	-
<i>PostCondition</i>	Data pembelian tersimpan pada basis data.

Sumber: Penulis (2019).

Pada Tabel 4.7 dijelaskan mengenai *use case* memasukan data pembelian, aktor memasukan data pembelian suku cadang, dimana nantinya data dari pembelian akan digunakan untuk menambahkan stok suku cadang.

4.3.4 Spesifikasi Use Case Memasukan Data Gaji Karyawan

Spesifikasi *use case* memasukan data gaji karyawan akan dijelaskan pada Tabel 4.8

Tabel 4.8 Spesifikasi Use Case Memasukan Data Gaji Karyawan

Spesifikasi <i>Use Case</i> Memasukan Data Gaji Karyawan	
<i>Actor</i>	Administrator Keuangan.
<i>Description</i>	Skenario untuk memasukan data gaji karyawan.
<i>Use case Goal</i>	Aktor dapat memasukan data gaji karyawan.
<i>Preconditions</i>	Aktor telah login kedalam sistem.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu gaji karyawan. 2. Sistem menampilkan halaman gaji karyawan. 3. Aktor memasukan data karyawan lalu klik tambah. 4. Sistem menyimpan data, dan menampilkan data gaji karyawan.

Tabel 4.8 Spesifikasi Use Case Memasukan Data Gaji Karyawan (Lanjutan)

Spesifikasi Use Case Memasukan Data Gaji Karyawan	
<i>Alternative Flow</i>	Mengatasi kegagalan nomor 3. 1. Aktor telah memasukan data karyawan pada bulan yang sama, sistem menampilkan pemberitahuan gaji bulan tersebut telah diinputkan.
<i>PostCondition</i>	Data gaji karyawan tersimpan pada basis data.

Sumber: Penulis (2019).

Pada Tabel 4.8 dijelaskan mengenai *use case* memasukan data gaji karyawan, aktor memasukan data gaji karyawan ke dalam sistem, sebagai salah satu data pengeluaran dari perusahaan.

4.3.5 Spesifikasi Use Case Melihat Stok Suku Cadang

Spesifikasi *use case* melihat stok suku cadang akan dijelaskan pada Tabel 4.9.

Tabel 4.9 Spesifikasi Use Case Melihat Stok Suku Cadang

Spesifikasi Use Case Melihat Stok Suku Cadang	
<i>Actor</i>	Administrator Keuangan, Manajer.
<i>Description</i>	Skenario untuk melihat stok suku cadang.
<i>Use case Goal</i>	Aktor dapat melihat stok suku cadang.
<i>Preconditions</i>	Aktor telah login kedalam sistem.
<i>Main Flow</i>	1. Aktor memilih menu stok suku cadang. 2. Sistem menampilkan halaman stok suku cadang. 3. Aktor mencari suku cadang. 4. Sistem menampilkan data suku cadang.
<i>Alternative Flow</i>	Mengatasi kegagalan nomor 3. 1. Jika data tidak tersedia, sistem tidak menampilkan data.
<i>PostCondition</i>	Aktor berhasil melihat stok suku cadang.

Sumber: Penulis (2019).

Pada Tabel 4.9 dijelaskan mengenai *use case* melihat stok suku cadang dimana aktor dapat melihat stok suku cadang yang masih tersedia di gudang perusahaan.

4.3.6 Spesifikasi Use Case Melihat Laporan Pemasukan

Spesifikasi *use case* melihat laporan pemasukan akan dijelaskan pada Tabel

4.10.

Tabel 4.10 Spesifikasi Use Case Melihat Laporan Pemasukan

Spesifikasi <i>Use Case</i> Melihat Laporan Pemasukan	
<i>Actor</i>	Administrator Keuangan, Manajer.
<i>Description</i>	Skenario untuk melihat laporan pemasukan.
<i>Use case Goal</i>	Aktor dapat melihat laporan pemasukan.
<i>Preconditions</i>	Aktor telah login kedalam sistem.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu pemasukan. 2. Sistem menampilkan <i>dropdown</i> bulan dan tahun . 3. Aktor me<input/>putkan bulan dan tahun lalu mengklik tombol cari. 4. Sistem menampilkan data.
<i>Alternative Flow</i>	Mengatasi kegagalan nomor 3. <ol style="list-style-type: none"> 1. Jika data tidak tersedia, maka sistem tidak menampilkan data.
<i>PostCondition</i>	Aktor berhasil melihat laporan pemasukan.

Sumber: Penulis (2019).

Pada Tabel 4.10 dijelaskan mengenai *use case* melihat laporan pemasukan berdasarkan bulan dan tahun yang dimasukkan oleh aktor.

4.3.7 Spesifikasi Use Case Melihat Laporan Pengeluaran

Spesifikasi *use case* Melihat Laporan Pengeluaran akan dijelaskan pada Tabel 4.11.

Tabel 4.11 Spesifikasi Use Case Melihat Laporan Pengeluaran

Spesifikasi <i>Use Case</i> Melihat Laporan Pengeluaran	
<i>Actor</i>	Administrator Keuangan, Manajer.
<i>Description</i>	Skenario untuk melihat laporan pengeluaran.
<i>Use case Goal</i>	Aktor dapat melihat laporan Pengeluaran.
<i>Preconditions</i>	Aktor telah login kedalam sistem.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu Pengeluaran. 2. Sistem menampilkan <i>dropdown</i> bulan dan tahun . 3. Aktor me<input/>putkan bulan dan tahun lalu klik cari. 4. Sistem menampilkan data.



Tabel 4.11 Spesifikasi Use Case Melihat Laporan Pengeluaran (Lanjutan)

Spesifikasi Use Case Melihat Laporan Pengeluaran	
<i>Alternative Flow</i>	Mengatasi kegagalan nomor 3. 1. Jika data tidak tersedia, maka sistem tidak menampilkan data.
<i>PostCondition</i>	Aktor berhasil melihat laporan Pengeluaran.

Sumber: Penulis (2019).

Pada Tabel 4.11 dijelaskan mengenai *use case* melihat laporan pengeluaran berdasarkan bulan dan tahun yang dimasukan oleh aktor, kemudian sistem akan menampilkan data pembelian suku cadang dan gaji karyawan.

4.3.8 Spesifikasi Use Case Melakukan Logout

Berikut spesifikasi *use case* melakukan *logout* akan dijelaskan pada Tabel 4.12.

Tabel 4.12 Spesifikasi Use Case Melakukan Logout

Spesifikasi Use Case Melakukan Logout	
<i>Actor</i>	Administrator Keuangan, Manajer
<i>Description</i>	Skenario untuk melakukan <i>logout</i> .
<i>Use case Goal</i>	Aktor dapat <i>logout</i> dari sistem.
<i>Preconditions</i>	Aktor telah <i>login</i> kedalam sistem.
<i>Main Flow</i>	1. Aktor memilih menu <i>logout</i> . 2. Sistem menghapus <i>session</i> pengguna. 3. Sistem menampilkan halaman <i>login</i> .
<i>Alternative Flow</i>	-
<i>PostCondition</i>	Aktor berhasil melakukan <i>logout</i> .

Sumber: Penulis (2019).

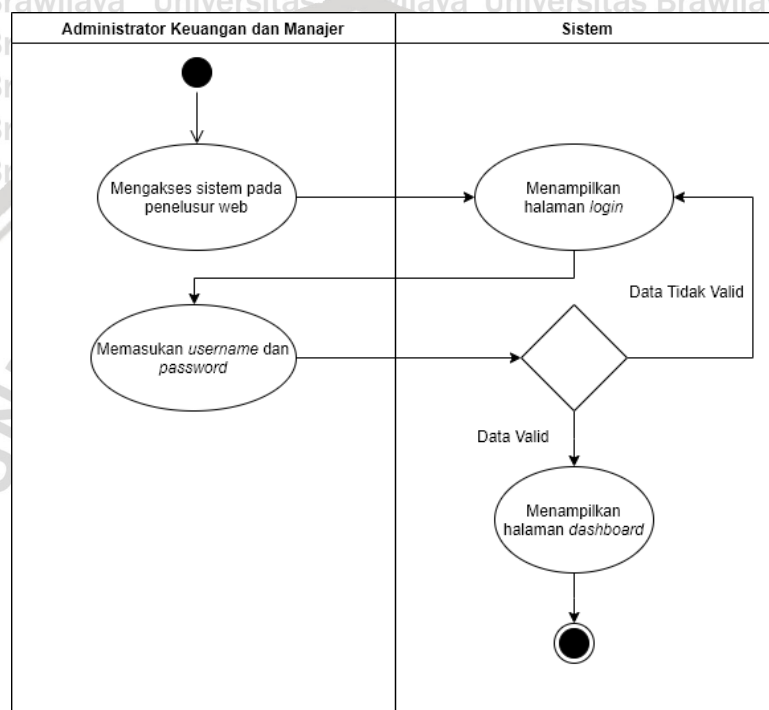
Pada Tabel 4.12 dijelaskan mengenai *use case* melakukan *logout* dimana aktor keluar dari sistem, dan sistem akan menghapus *session* pengguna.

4.4 Activity Diagram

Activity diagram digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity diagram* digunakan untuk memodelkan berbagai aksi yang dilakukan saat sebuah operasi dijalankan dan memodelkan hasil dari aksi tersebut. *Activity diagram* dapat menjelaskan bagaimana aktivitas dalam suatu sistem dari mulai hingga berakhir. Berikut ini beberapa *activity diagram* yang digunakan untuk merancang sistem yang akan dibangun.

4.4.1 Activity Diagram Melakukan Login

Activity diagram melakukan login dijelaskan pada Gambar 4.7. Pada activity diagram melakukan login dimulai dengan pengguna mengakses sistem pada penelusur web. Pengguna memasukan data login yang terdiri dari *username* dan *password*, kemudian setelah itu menekan tombol login, sistem akan mencocokkan data login pengguna dengan basis data, apabila data login salah akan dikembalikan ke halaman login dan akan ditampilkan pemberitahuan, dan jika benar maka pengguna berhasil masuk kedalam sistem dan akan diarahkan ke halaman dashboard.



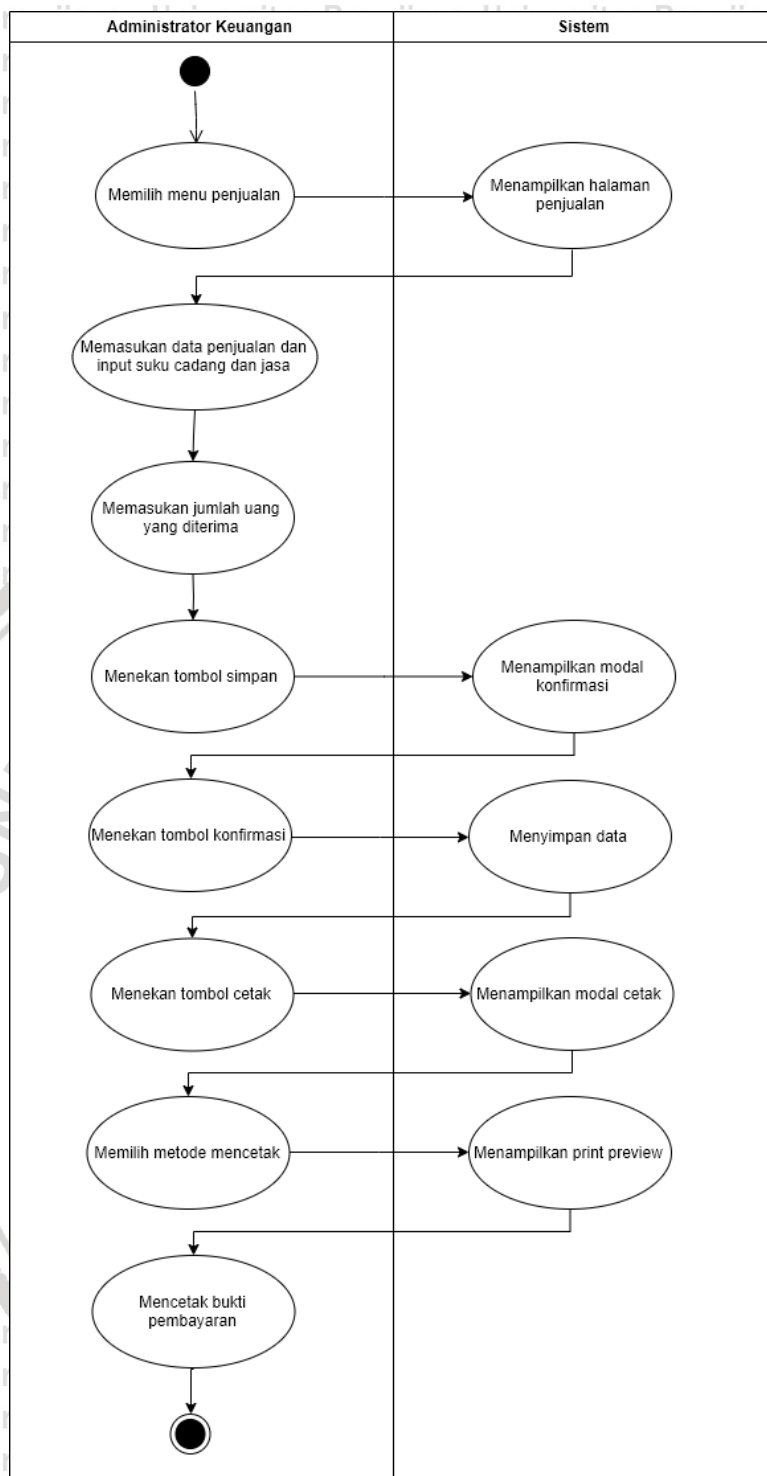
Gambar 4.7 Activity Diagram Melakukan Login

Sumber: Penulis (2019).

4.4.2 Activity Diagram Memasukan Data Penjualan

Activity diagram memasukan data penjualan dimulai dengan administrator keuangan memasukan data penjualan berupa data kendaraan dan tanggal penjualan. Selanjutnya administrator keuangan memasukan data *input* suku cadang dan jasa yang digunakan dalam perbaikan kendaraan. Administrator keuangan akan memasukan nilai uang yang diterima kedalam sistem, kemudian selanjutnya adalah menekan tombol simpan dan mengkonfirmasi semua suku cadang dan jasa yang dipakai.

Terakhir administrator keuangan dapat mencetak bukti pembayaran yang dilakukan oleh pelanggan. Terdapat dua metode cetak yang dapat dipilih oleh administrator keuangan yakni cetak *invoice* atau cetak faktur. Activity diagram memasukan data penjualan dapat dilihat pada Gambar 4.8.



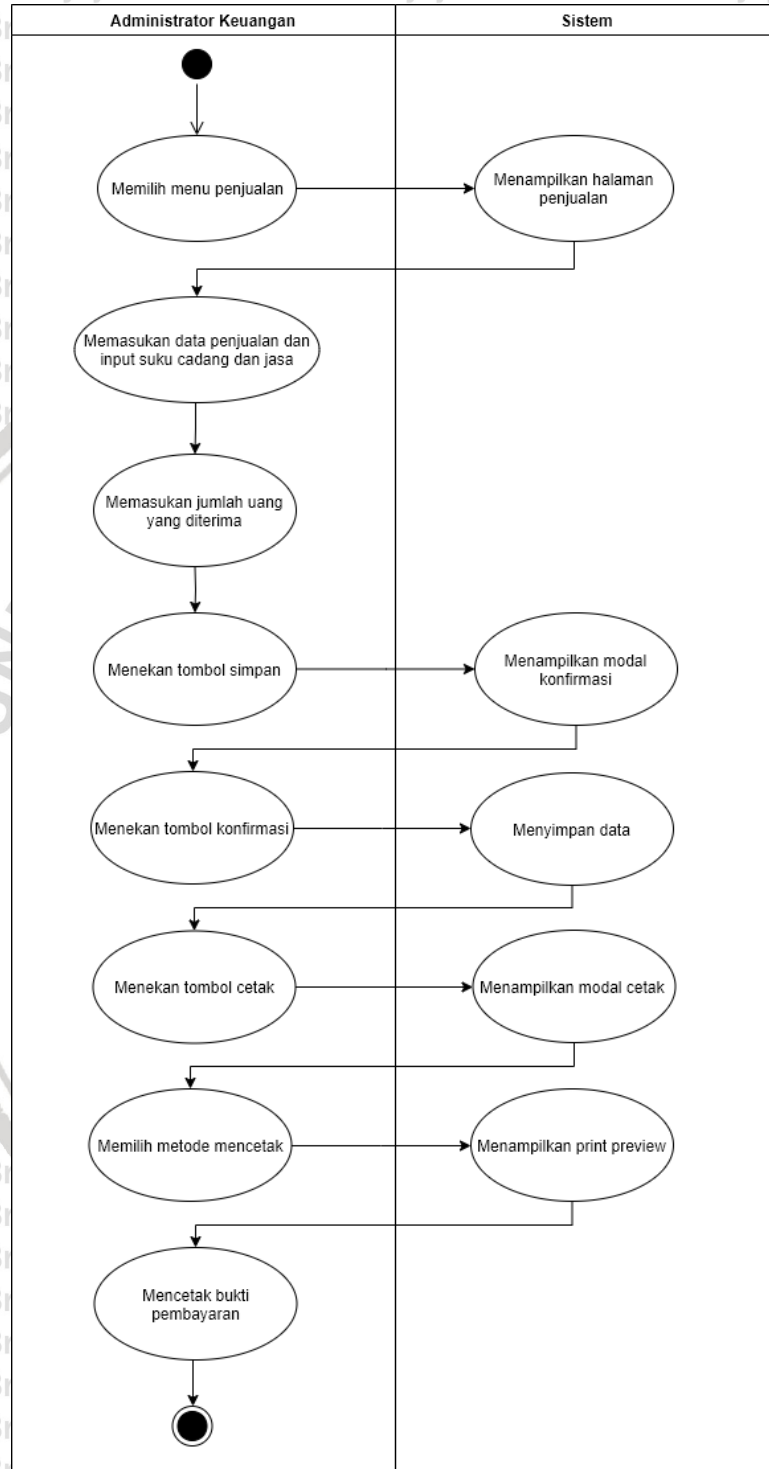
Gambar 4.8 Activity Diagram Memasukan Data Penjualan

Sumber: Penulis (2019).

4.4.3 Activity Diagram Memasukan Data Pembelian

Activity diagram memasukan data pembelian dapat dilihat pada Gambar

4.9.



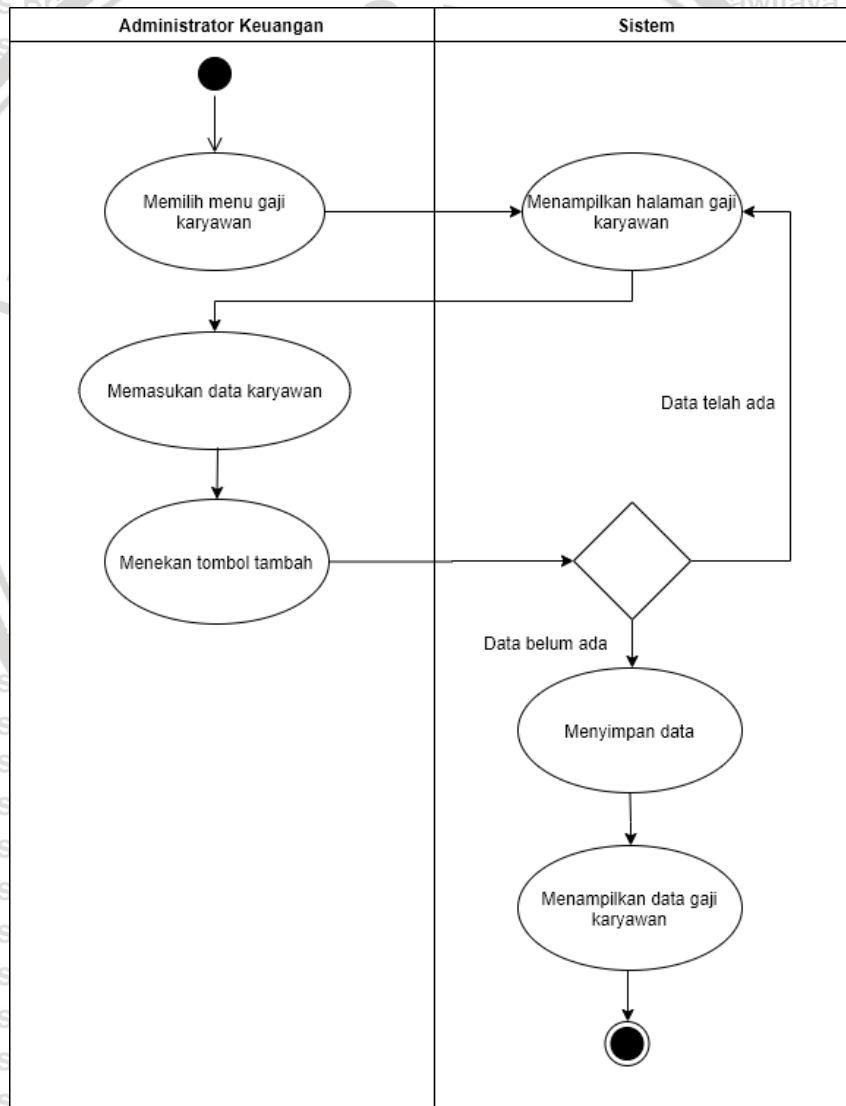
Gambar 4.9 Activity Diagram Memasukan Data Pembelian

Sumber: Penulis (2019).

Activity diagram memasukkan data pembelian dimulai dengan administrator keuangan memasukkan data pembelian yang terdiri dari data pemasok dan tanggal pembelian. Kemudian selanjutnya administrator keuangan memasukkan data suku cadang yang dibeli, harga jual, harga beli serta jumlah unit yang dibeli. Setelah selesai memasukkan data suku cadang administrator keuangan akan menekan tombol simpan untuk menyimpan data kedalam basis data.

4.4.4 Activity Diagram Memasukkan Data Gaji Karyawan

Activity diagram memasukkan data gaji karyawan dimulai dengan administrator keuangan memasukkan data karyawan yang terdiri dari nama karyawan dan jumlah absen, setelah itu administrator keuangan menekan tombol tambah untuk menyimpan data. Kemudian sistem akan menghitung gaji yang didapat oleh setiap karyawan dan menyimpan data kedalam basis data. *Activity diagram* memasukkan data gaji karyawan dapat dilihat pada Gambar 4.10.

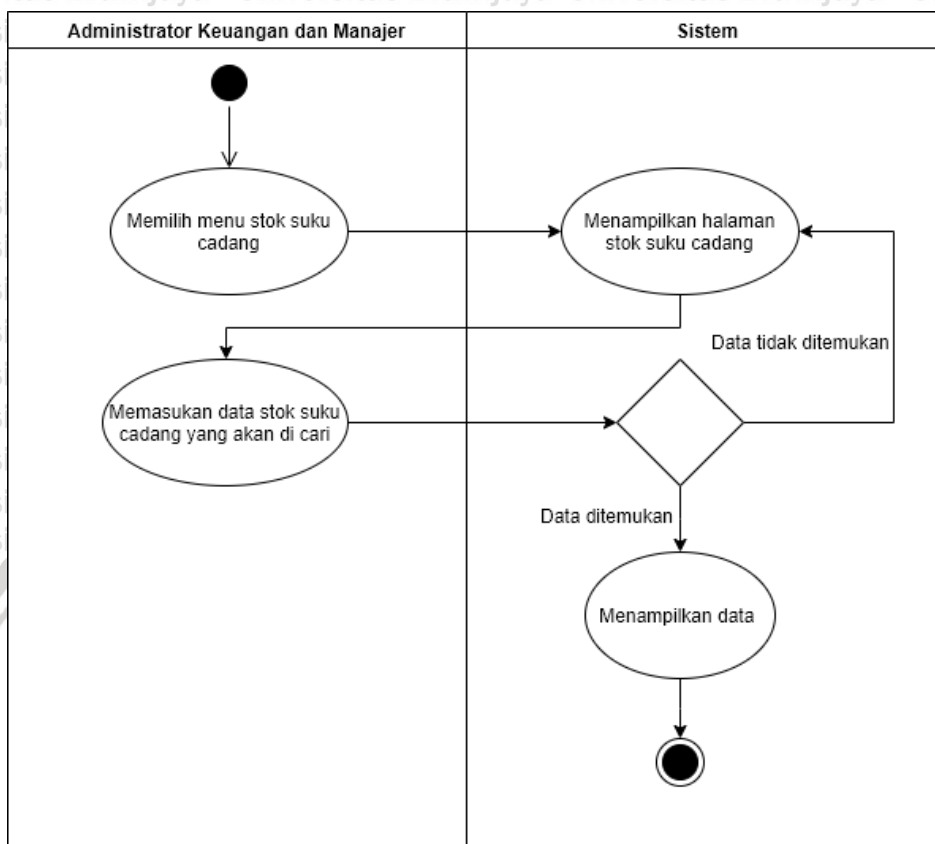


Gambar 4.10 Activity Diagram Memasukkan Gaji Karyawan

Sumber: Penulis (2019).

4.4.5 Activity Diagram Melihat Stok Suku Cadang

Activity diagram melihat stok suku cadang dapat dilihat pada Gambar 4.11.



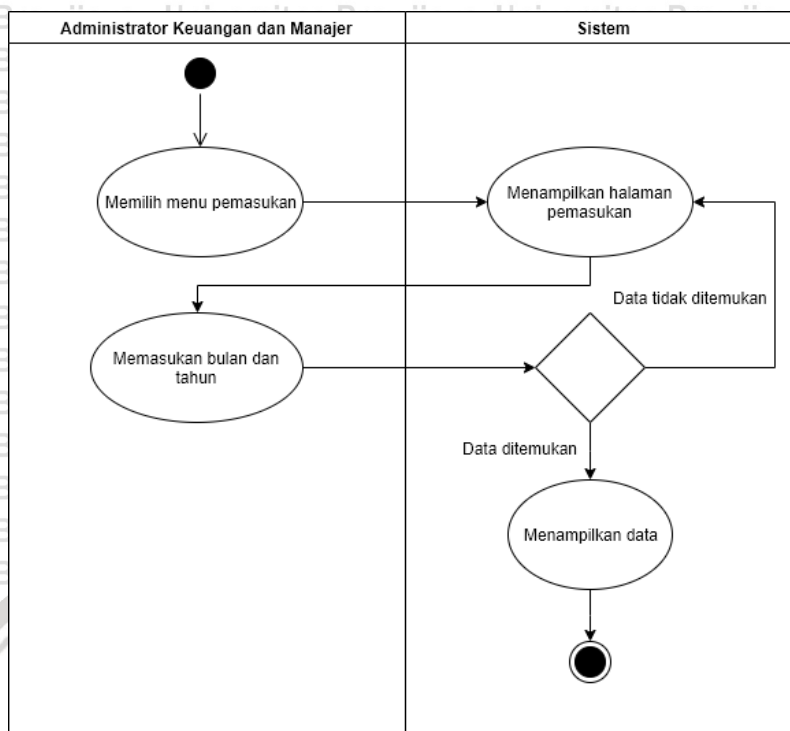
Gambar 4.11 Activity Diagram Melihat Stok Suku Cadang

Sumber: Penulis (2019).

Pada Gambar 4.11 dijelaskan *activity diagram* melihat stok suku cadang dimulai dengan memilih menu stok suku cadang, kemudian sistem akan menampilkan semua data stok suku cadang. Jika pengguna ingin mencari data secara spesifik dapat dengan melakukan pencarian menggunakan kolom cari.

4.4.6 Activity Diagram Melihat Laporan Pemasukan

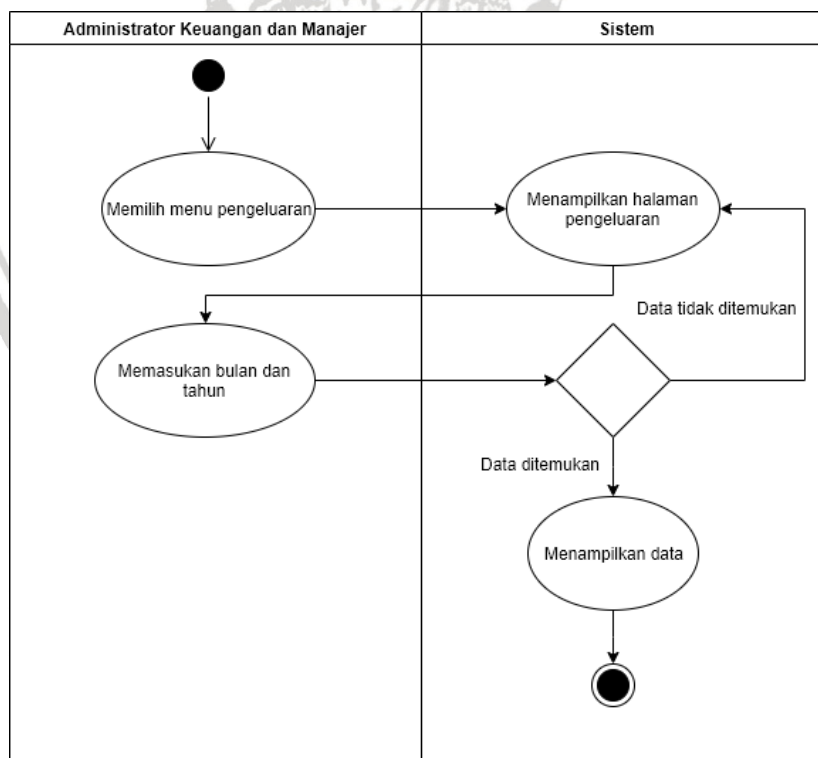
Activity diagram melihat laporan pemasukan dapat dilihat pada Gambar 4.12, dimana *activity diagram* ini dimulai dengan pengguna memasukan data filter yang terdiri dari bulan dan tahun, kemudian menekan tombol cari. Jika data tersedia maka sistem akan menampilkan data. Namun jika data tidak tersedia maka sistem tidak akan menampilkan data.



Gambar 4.12 Activity Diagram Melihat Laporan Pemasukan

Sumber: Penulis (2019).

4.4.7 Activity Diagram Melihat Laporan Pengeluaran



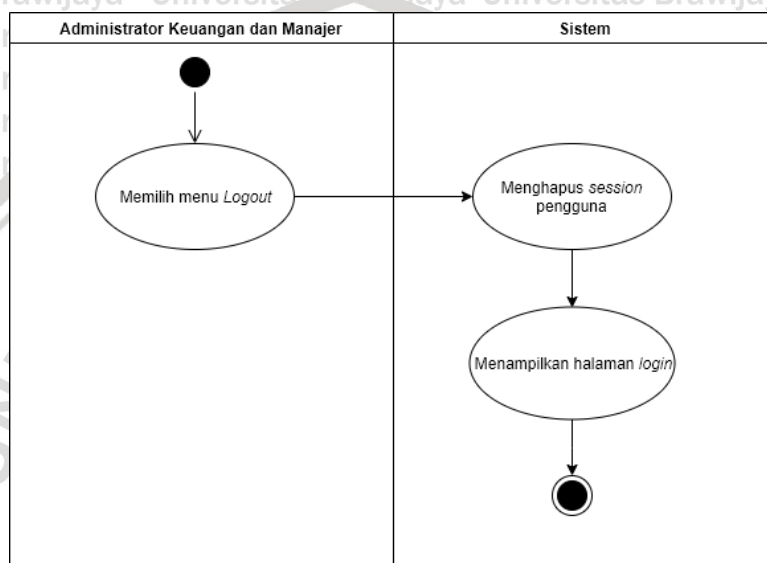
Gambar 4.13 Activity Diagram Melihat Laporan Pengeluaran

Sumber: Penulis (2019).

Activity diagram melihat laporan pengeluaran dimulai dengan pengguna memasukan data filter yang terdiri dari bulan dan tahun, kemudian menekan tombol cari. Jika data tersedia maka sistem akan menampilkan data. Namun jika data tidak tersedia maka sistem tidak akan menampilkan data. *Activity diagram* melihat laporan pengeluaran dapat dilihat pada Gambar 4.13.

4.4.8 Activity Diagram Melakukan Logout

Activity diagram melakukan *logout* dapat dilihat pada Gambar 4.14, dengan pengguna menekan tombol *logout* atau memilih menu *logout* maka sistem akan menghapus *session* pengguna tersebut dan akan menampilkan halaman *login*.



Gambar 4.14 Activity Diagram Melakukan Logout

Sumber: Penulis (2019).

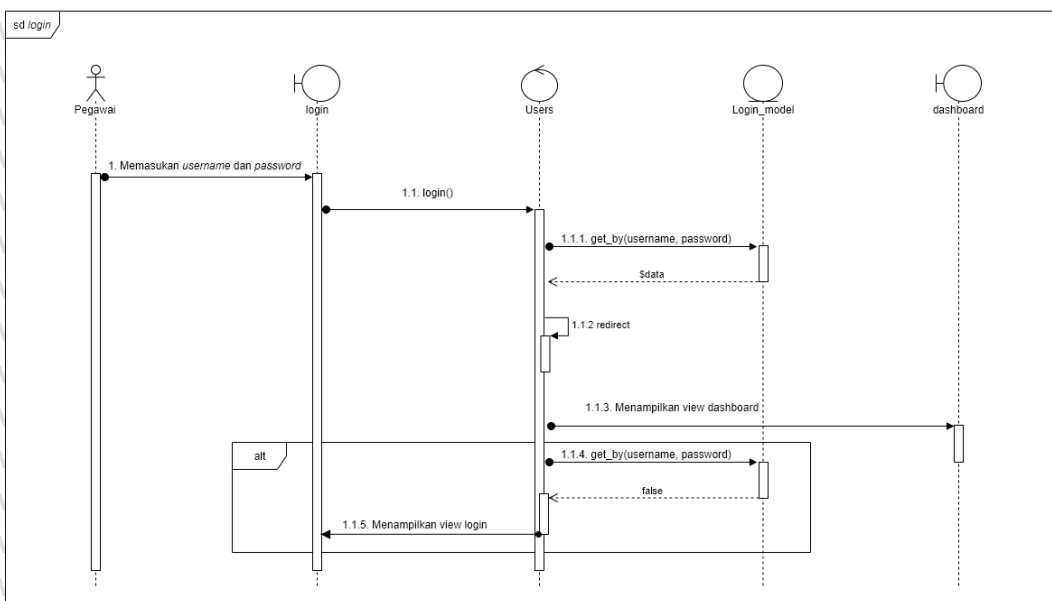
4.5 Sequence Diagram

Sequence diagram menotasikan alur algoritma atau pesan untuk menjalankan sistem. Pesan dapat berupa sinyal atau panggilan terhadap suatu operasi. Proses alur diagram ini melalui aktor, *boundary*, *control*, *entity*, kembali ke *control*, *boundary*, aktor.

4.5.1 Sequence Diagram Melakukan Login

Sequence diagram melakukan *login* dapat dilihat pada Gambar 4.15. Pada *sequence diagram* melakukan *login* terdapat aktor pegawai sebagai pengguna yang menjalankan sistem, *boundary object* yang digunakan pada *sequence diagram* melakukan login terdiri dari *login* dan *dashboard*, *control object* yang berperan disini adalah *controller Users*, dan *entity object* yang digunakan adalah *Login_model*.

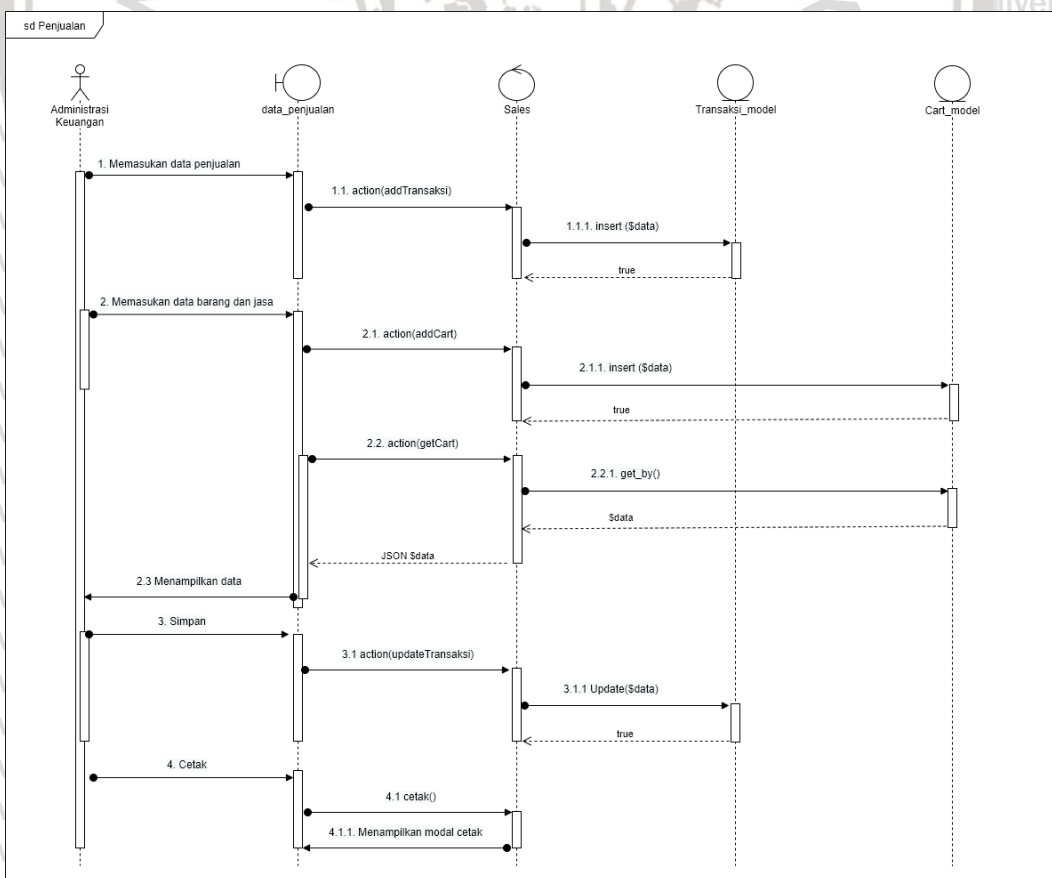




Gambar 4.15 Sequence Diagram Melakukan Login

Sumber : Penulis (2019)

4.5.2 Sequence Diagram Memasukan Data Penjualan



Gambar 4.16 Sequence Diagram Memasukan Data Penjualan

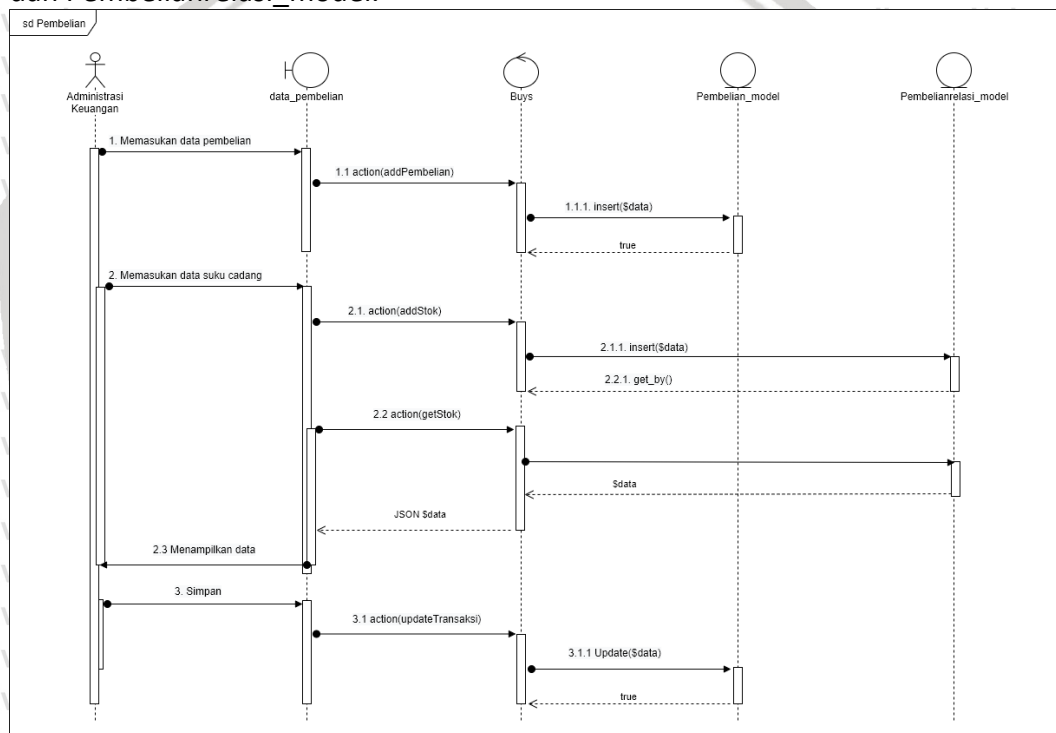
Sumber: Penulis (2019).



Pada Gambar 4.16 menggambarkan mengenai *sequence diagram* memasukan data penjualan. *Sequence diagram* memasukan data penjualan terdapat aktor administrator keuangan sebagai pengguna yang menjalankan sistem, *boundary object* yang digunakan adalah data_penjualan, *control object* yang berperan disini adalah *controller Sales*, dan *entity object* yang digunakan terdiri atas Transaksi_model dan Cart_model.

4.5.3 Sequence Diagram Memasukan Data Pembelian

Sequence diagram memasukan data pembelian dapat dilihat pada Gambar 4.17. Pada *sequence diagram* memasukan data pembelian terdapat aktor administrator keuangan sebagai pengguna yang menjalankan sistem, *boundary object* yang digunakan adalah data_pembelian, *control object* yang berperan disini adalah *controller Buys*, dan *entity object* yang digunakan adalah Pembelian_model dan Pembelianrelasi_model.

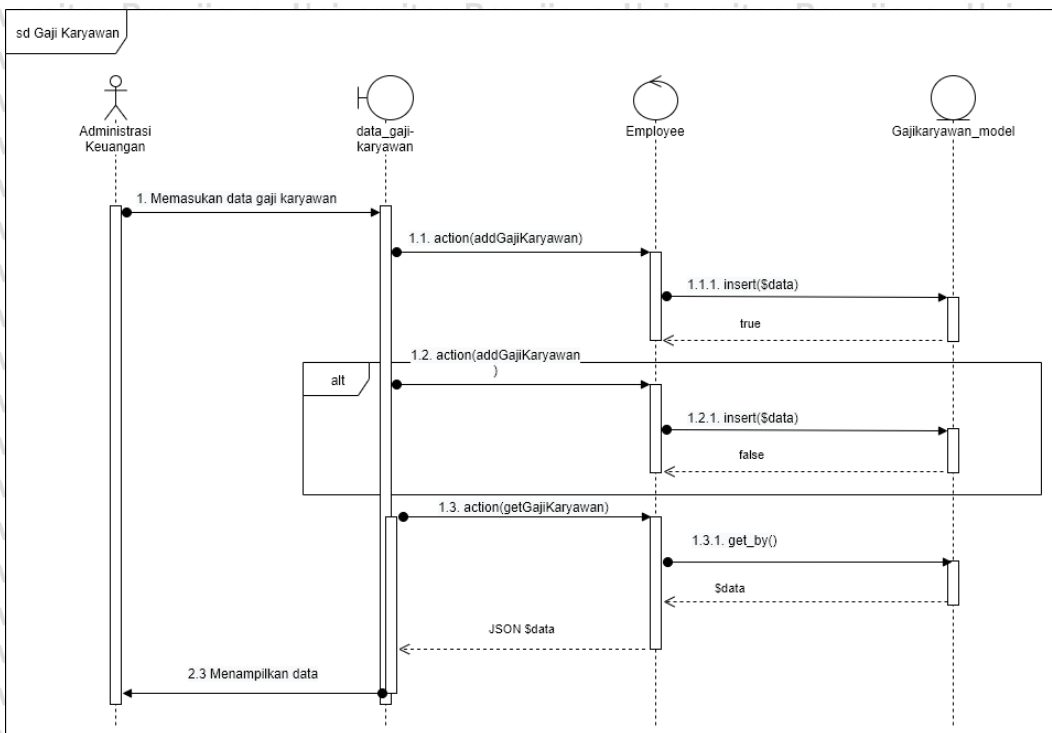


Gambar 4.17 Sequence Diagram Memasukan Data Pembelian

Sumber: Penulis (2019).

4.5.4 Sequence Diagram Memasukan Data Gaji Karyawan

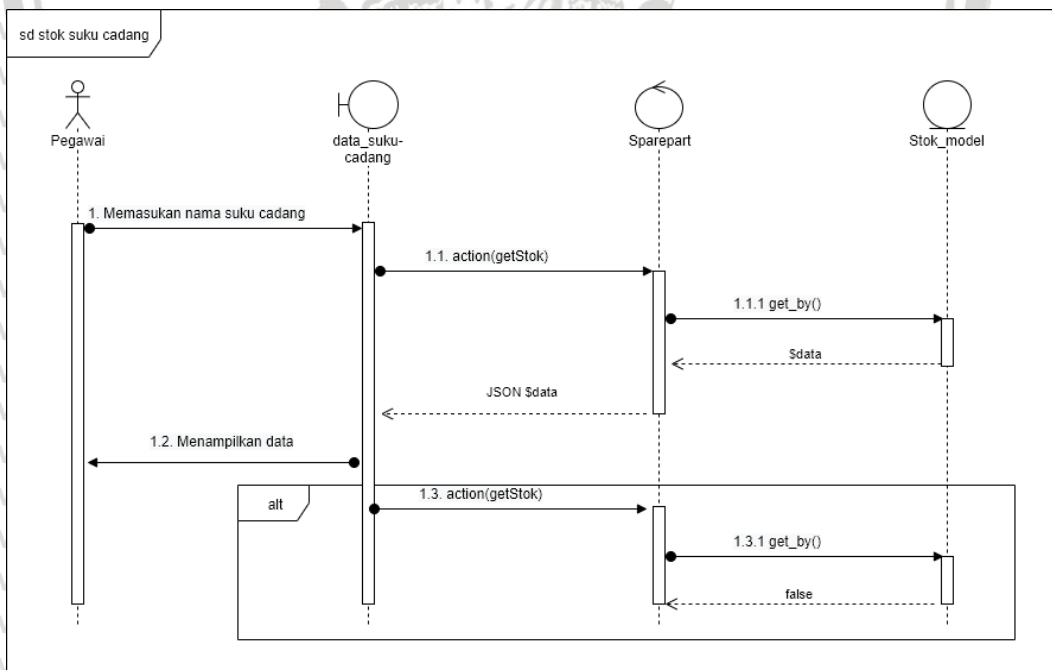
Pada Gambar 4.18 menggambarkan mengenai *sequence diagram* memasukan data gaji karyawan. *Sequence diagram* memasukan data gaji karyawan terdapat aktor administrator keuangan sebagai pengguna yang menjalankan sistem, *boundary object* yang digunakan adalah data_gaji-karyawan, *control object* yang berperan disini adalah *controller Employee*, dan *entity object* yang digunakan adalah Gajikaryawan_model.



Gambar 4.18 Sequence Diagram Memasukan Data Gaji Karyawan

Sumber: Penulis (2019).

4.5.5 Sequence Diagram Melihat Stok Suku Cadang



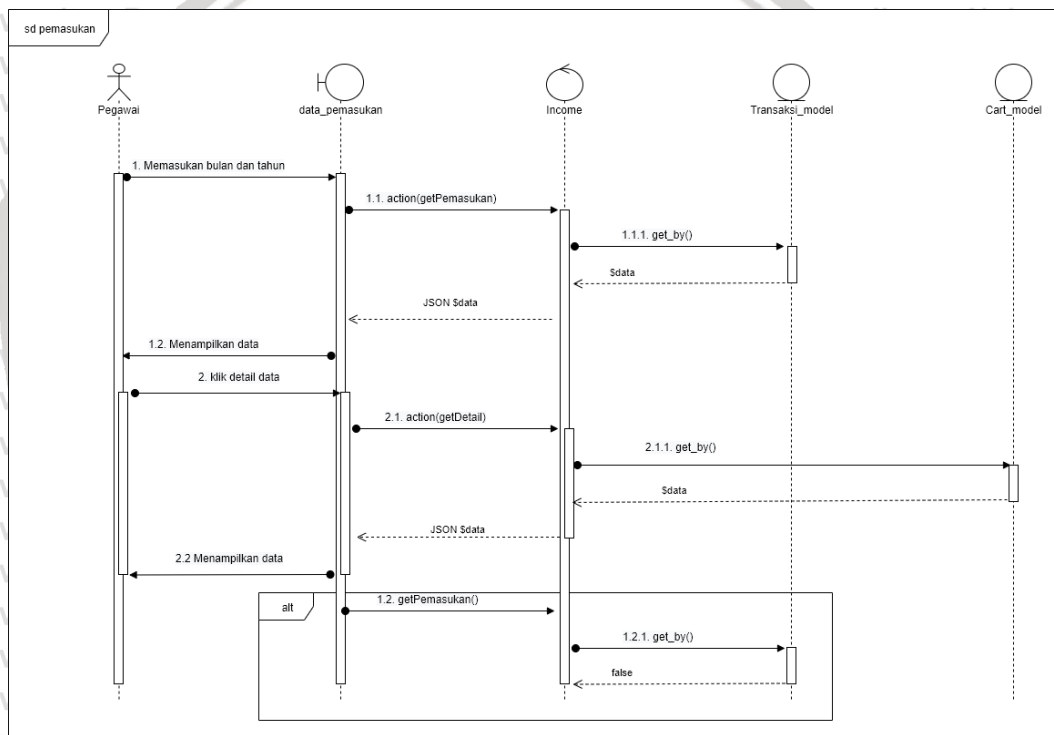
Gambar 4.19 Sequence Diagram Melihat Stok Suku Cadang

Sumber: Penulis (2019).

Sequence diagram melihat stok suku cadang dapat dilihat pada Gambar 4.19. Pada *sequence diagram* melihat stok suku cadang terdapat aktor pegawai sebagai pengguna yang menjalankan sistem, *boundary object* yang digunakan adalah *data_suku-cadang*, *control object* yang berperan disini adalah *controller Sparepart*, dan *entity object* yang digunakan adalah *Stok_model*.

4.5.6 Sequence Diagram Melihat Laporan Pemasukan

Pada Gambar 4.20 menggambarkan mengenai *sequence diagram* melihat laporan pemasukan. *Sequence diagram* melihat laporan pemasukan terdapat aktor Pegawai sebagai pengguna yang menjalankan sistem, *boundary object* yang digunakan adalah *data_pemasukan*, *control object* yang berperan disini adalah *controller Income*, dan *entity object* yang digunakan terdiri atas *Transaksi_model* dan *Cart_model*.



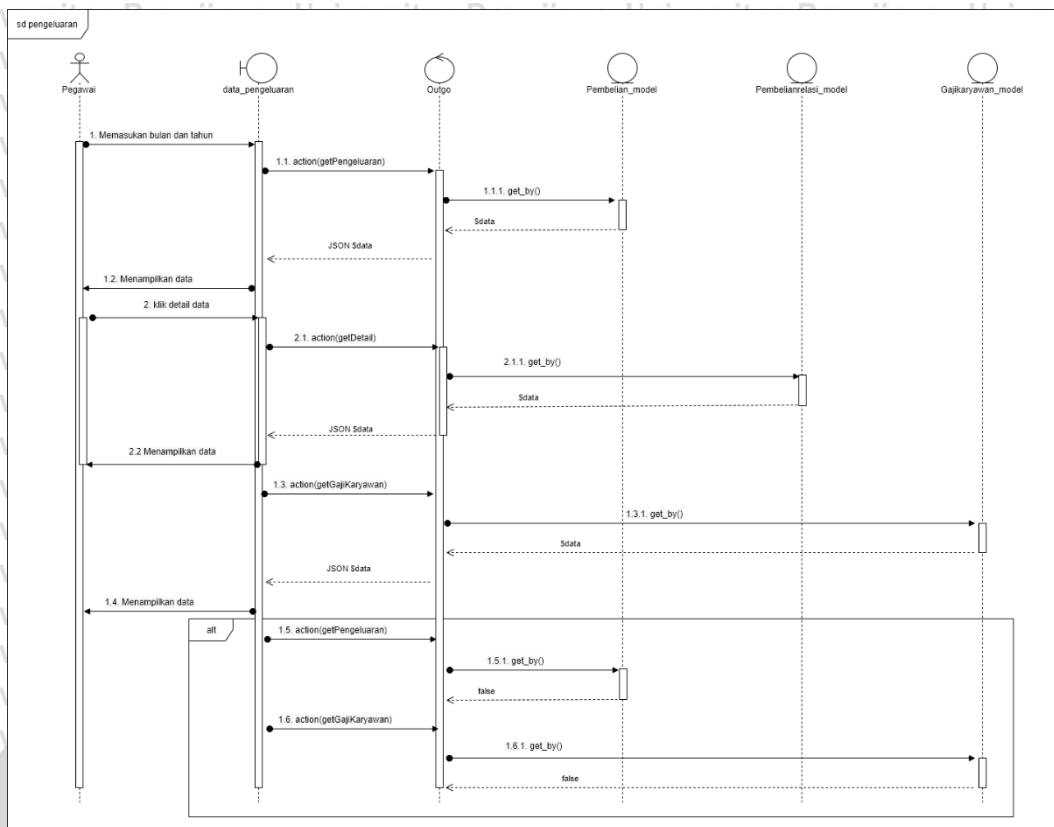
Gambar 4.20 Sequence Diagram Melihat Laporan Pemasukan

Sumber: Penulis (2019).

4.5.7 Sequence Diagram Melihat Laporan Pengeluaran

Sequence diagram melihat laporan pengeluaran dapat dilihat pada Gambar 4.21. Pada *sequence diagram* melihat laporan pengeluaran terdapat aktor pegawai sebagai pengguna yang menjalankan sistem, *boundary object* yang digunakan adalah *data_pengeluaran*, *control object* yang berperan disini adalah *controller Outgo*, dan *entity object* yang digunakan terdiri atas *Pembelian_model*, *Pembelianrelasi_model* dan *Gajikaryawan_model*.



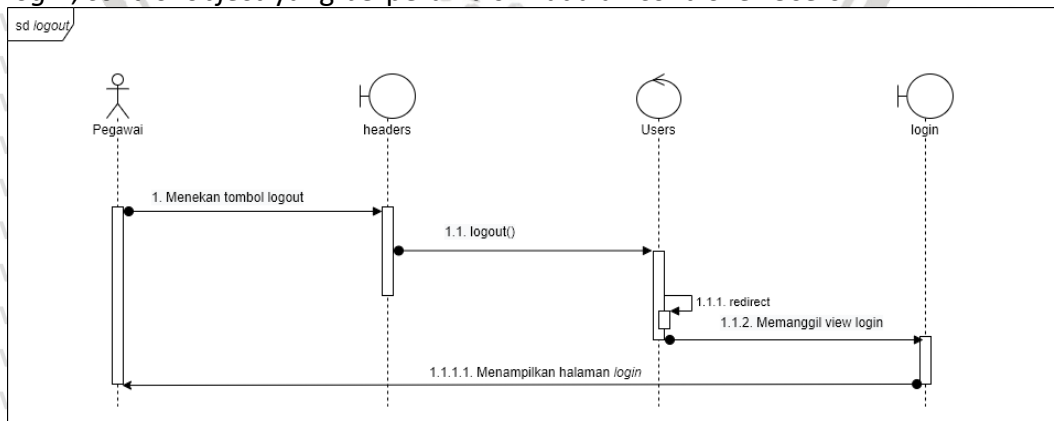


Gambar 4.21 Sequence Diagram Melihat Laporan Pengeluaran

Sumber: Penulis (2019).

4.5.8 Sequence Diagram Melakukan Logout

Sequence diagram melakukan logout dapat dilihat pada Gambar 4.22. Pada sequence diagram melakukan logout terdapat aktor pegawai sebagai pengguna yang menjalankan sistem, boundary object yang digunakan adalah headers dan login, control object yang berperan disini adalah controller Users.



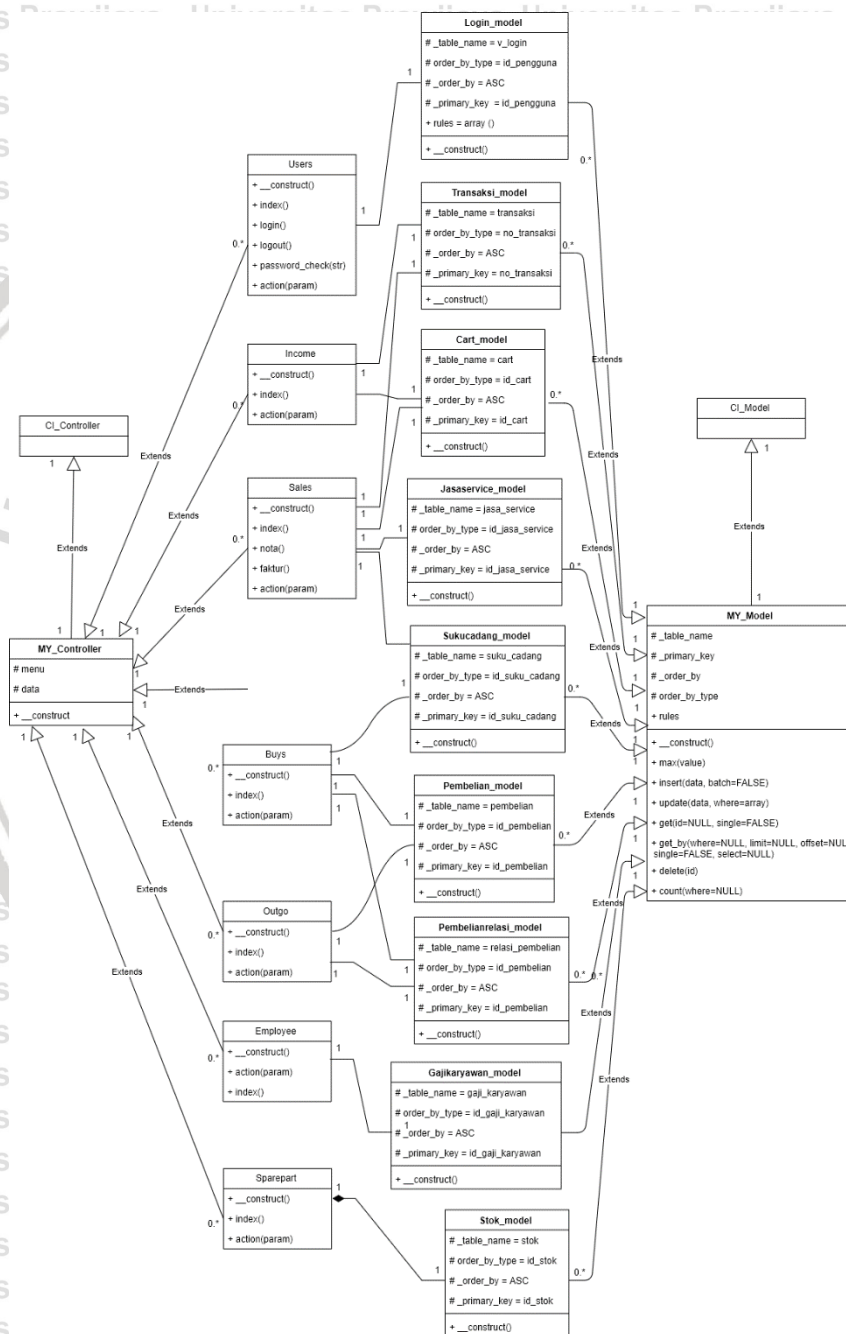
Gambar 4.22 Sequence Diagram Melakukan Logout

Sumber: Penulis (2019).



4.6 Class Diagram

Class diagram menggambarkan notasi hubungan setiap kelas rancangan sistem. Pada perancangan ini menggunakan *framework Codeigniter* sehingga menggunakan kelas controller dan model dalam menjalankan sistem. Pada Gambar 4.23 akan digambarkan *class diagram* dari sistem informasi *point of sales* untuk bengkel.



Gambar 4.23 Class Diagram Sistem Informasi Point of Sales untuk Bengkel

Sumber: Penulis (2019).

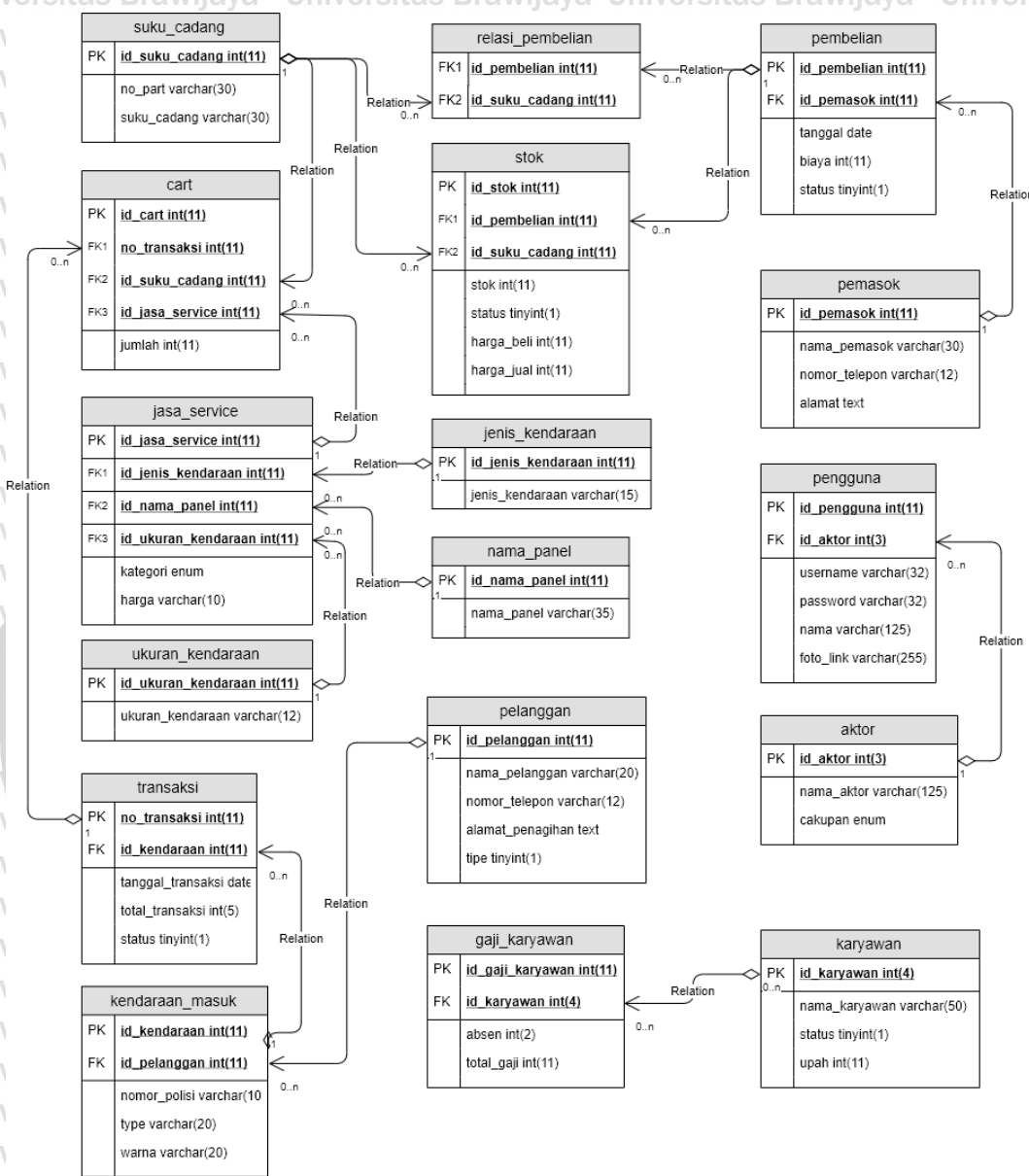


Pada Gambar 4.23 terdapat dua kelas yang merupakan kelas dari *framework Codeigniter* yaitu *CI_Model* dan *CI_Controller*. Pada penelitian ini peneliti menambahkan kelas inti yang nantinya akan menjadi kelas induk dari beberapa kelas yakni *MY_Model* dan *MY_Controller*. Untuk kelas *model* yang terdapat pada pengembangan sistem informasi point of sales untuk bengkel ini terdapat sembilan kelas terdiri dari *Login_model*, *Transaksi_model*, *Cart_model*, *Pembelian_model*, *Pembelianrelasi_model*, *Gajikaryawan_model*, *Sukucadang_model*, *Jasaservice_model* dan *Stok_model* yang setiap kelas tersebut *extend* pada kelas *MY_Model*. Untuk kelas *controller* terdapat tujuh kelas yang terdiri dari *Users*, *Sales*, *Buys*, *Employee*, *Sparepart*, *Income* dan *Outgo* yang kesemuanya *extend* pada kelas *MY_Controller*

4.7 Physical Data Model

Physical Data Model (PDM) adalah pemodelan yang memiliki konsep hampir mirip dengan pemodelan kelas. PDM memiliki tujuan untuk memodelkan skema dari internal basis data, menggambarkan tabel dengan kolomnya dan juga hubungan antar tabelnya (Ambysoft Inc., 2003).

Pada Gambar 4.24 dapat dilihat *Physical Data Model* yang didalamnya terdapat tujuhbelas tabel yang terdiri atas *suku_cadang*, *relasi_pembelian*, *pembelian*, *cart*, *stok*, *pemasok*, *jasa_service*, *jenis_kendaraan*, *pengguna*, *aktor*, *nama_panel*, *ukuran_kendaraan*, *transaksi*, *pelanggan*, *kendaraan_masuk*, *gaji_karyawan*, dan *karyawan*.



Gambar 4.24 Physical Data Model Sistem Informasi Point of Sales untuk Bengkulu

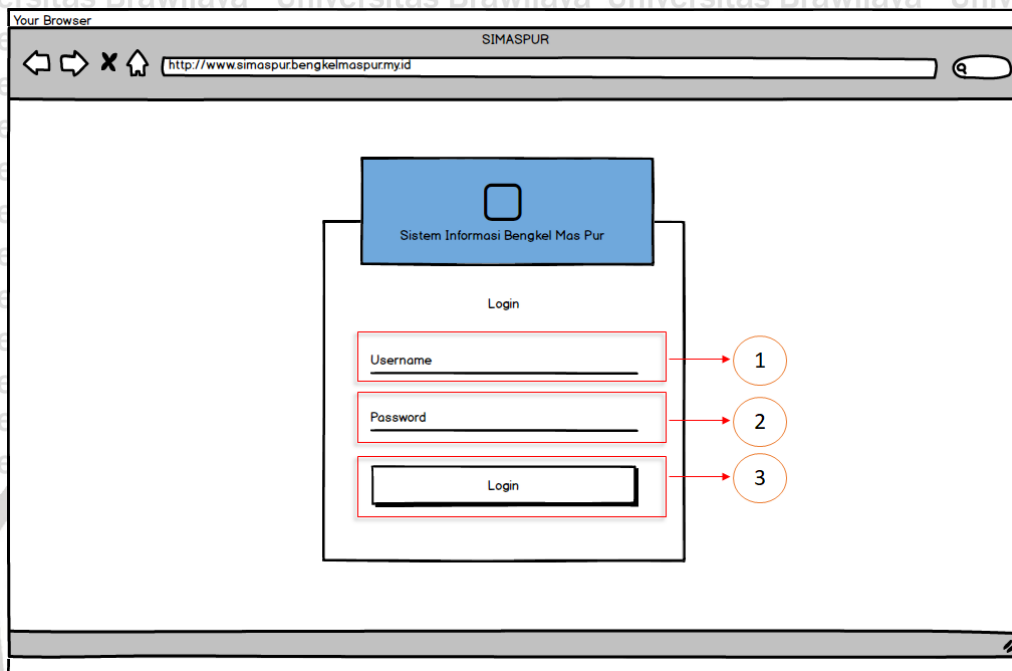
Sumber: Penulis (2019).

4.8 Perancangan Antarmuka

Perancangan antarmuka berfungsi untuk menggambarkan rancangan antarmuka bagi pengguna sistem. Dalam hal ini peneliti menggunakan perangkat lunak *Balsamiq Mockups* versi 3.5.17 untuk membuat setiap rancangan antarmuka yang akan menjadi panduan dalam pengembangan sistem informasi *point of sales* untuk bengkel.

4.8.1 Perancangan Antarmuka Halaman Login

Pada Gambar 4.25 terdapat rancangan antarmuka halaman *login* yang ditampilkan saat pengguna pertama kali mengakses sistem. Pengguna akan diarahkan secara otomatis oleh sistem ke halaman ini apabila belum melakukan *login*.



Gambar 4.25 Perancangan Antarmuka Halaman Login

Sumber: Penulis (2019).

Keterangan rancangan antarmuka halaman login pada Gambar 4.25.

1. Formulir untuk memasukkan *username*.
2. Formulir untuk memasukkan *password*.
3. Tombol untuk *Login* setelah mengisi semua formulir.

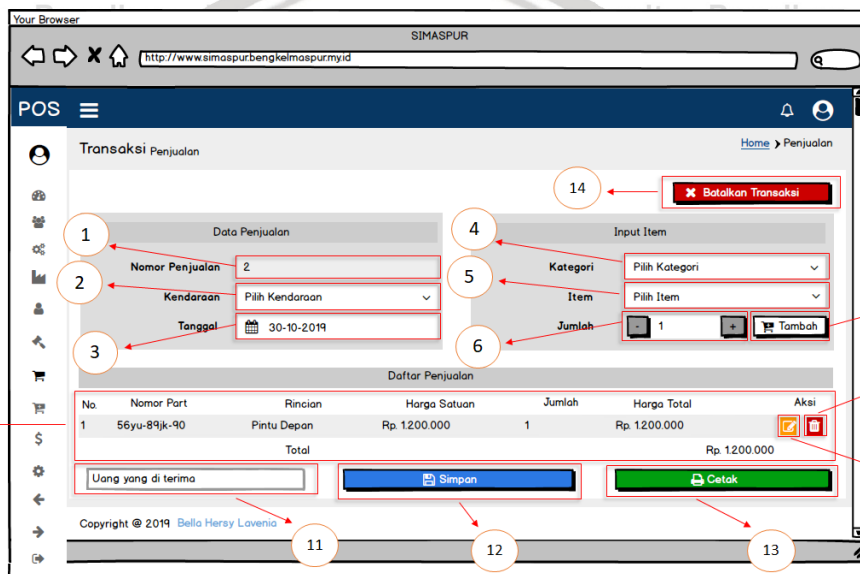
4.8.2 Perancangan Antarmuka Halaman Penjualan

Pada Gambar 4.26 terdapat rancangan antarmuka halaman penjualan yang ditampilkan pertama kali setelah administrator keuangan berhasil *login*. Pada halaman ini administrator keuangan dapat mengelola data penjualan berupa suku cadang dan jasa yang digunakan selama perbaikan kendaraan dari pelanggan. Halaman ini hanya bisa diakses oleh administrator keuangan, aktor manajer tidak bisa mengakses halaman ini, namun dapat menerima laporan pada halaman laporan pemasukan.

Keterangan rancangan antarmuka halaman penjualan pada Gambar 4.26.

1. Formulir untuk menampilkan nomor penjualan.
2. Formulir untuk memasukkan data kendaraan.
3. Formulir untuk memasukkan tanggal penjualan.
4. Formulir untuk memasukkan kategori item yang akan dimasukkan.

5. Formulir untuk memasukkan item, formulir ini bergantung dengan formulir kategori.
6. Formulir untuk memasukkan jumlah *item*.
7. Tombol tambah untuk menambahkan *item* kedalam daftar penjualan.
8. Tabel informasi daftar *item* penjualan.
9. Tombol ubah untuk mengubah data *item* penjualan pada kolom tersebut.
10. Tombol hapus untuk menghapus data *item* penjualan pada kolom tersebut.
11. Formulir untuk memasukkan uang yang diterima dari pelanggan.
12. Tombol simpan untuk menyimpan semua data kedalam basis data.
13. Tombol cetak untuk mencetak data dalam bentuk *invoice* atau faktur.
14. Tombol batalkan transaksi untuk membatalkan transaksi yang dilakukan.



Gambar 4.26 Perancangan Antarmuka Halaman Penjualan

Sumber: Penulis (2019).

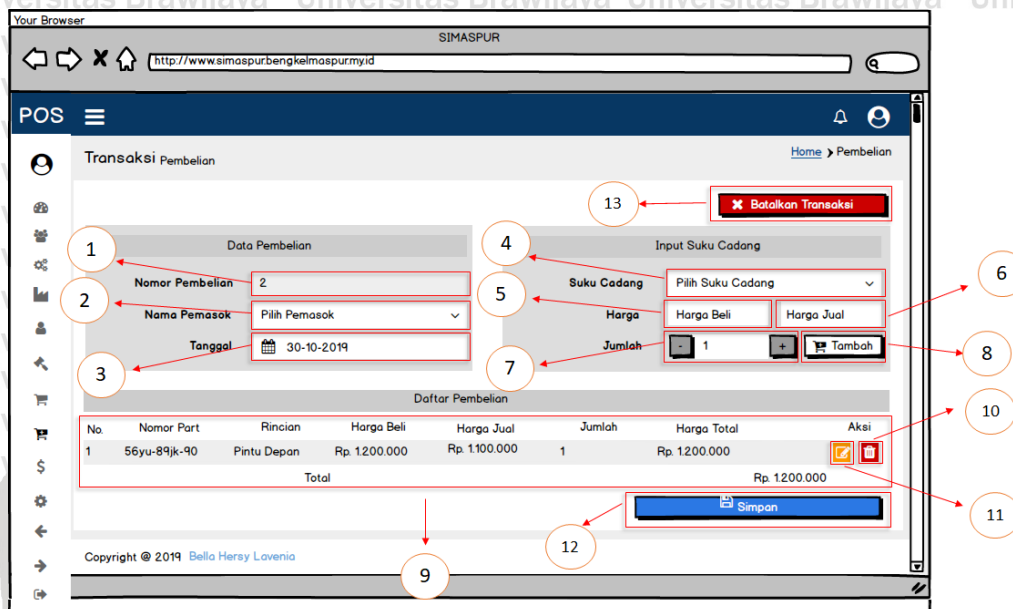
4.8.3 Perancangan Antarmuka Halaman Pembelian

Halaman Pembelian merupakan halaman yang digunakan oleh administrator keuangan untuk memasukkan data dari pembelian suku cadang. Data dari halaman ini nantinya akan digunakan juga pada halaman stok suku cadang, agar semua pengguna dapat melakukan pengecekan stok suku cadang sebelum dijual ke pelanggan.

Keterangan rancangan antarmuka halaman pembelian pada Gambar 4.27.

1. Formulir untuk menampilkan nomor pembelian.
2. Formulir untuk memasukkan data pemasok.
3. Formulir untuk memasukkan tanggal pembelian.
4. Formulir untuk memasukkan data suku cadang yang dibeli.
5. Formulir untuk memasukkan harga beli suku cadang.
6. Formulir untuk memasukkan harga jual suku cadang.
7. Formulir untuk memasukkan jumlah suku cadang yang dibeli.

8. Tombol tambah untuk menambahkan suku cadang yang dibeli.
9. Tabel informasi suku cadang yang dibeli.
10. Tombol ubah untuk mengubah data suku cadang pada kolom tersebut.
11. Tombol hapus untuk menghapus data suku cadang pada kolom tersebut.
12. Tombol simpan untuk menyimpan transaksi pembelian.
13. Tombol batalkan transaksi untuk membatalkan transaksi yang dilakukan.



Gambar 4.27 Perancangan Antarmuka Halaman Pembelian

Sumber: Penulis (2019).

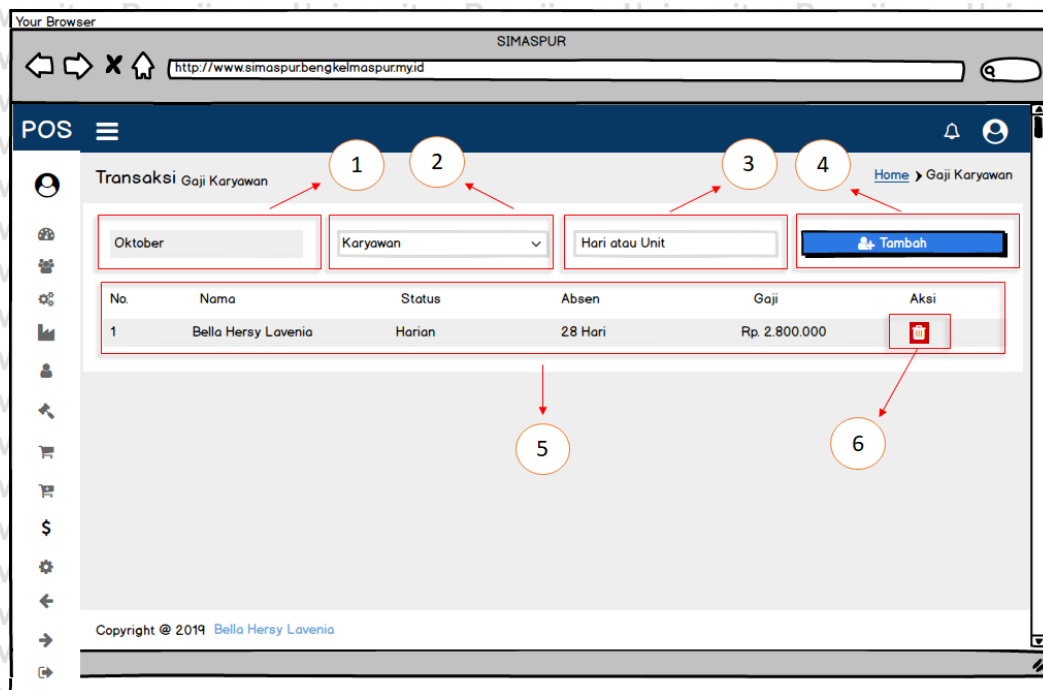
4.8.4 Perancangan Antarmuka Halaman Gaji Karyawan

Halaman gaji karyawan merupakan halaman yang digunakan oleh administrator keuangan untuk memasukkan data gaji karyawan. Halaman ini hanya aktif pada waktu tertentu saja, yaitu pada tanggal 27 sampai tanggal 30 setiap bulannya. Seperti dilihat pada Gambar 4.28 yang merupakan rancangan antarmuka gaji karyawan pada saat aktif, administrator keuangan dapat memasukkan data gaji karyawan berupa data karyawan yang bersangkutan serta jumlah absen atau jumlah unit yang dikerjakan oleh karyawan tersebut.

Keterangan rancangan antarmuka halaman gaji karyawan pada Gambar

4.28.

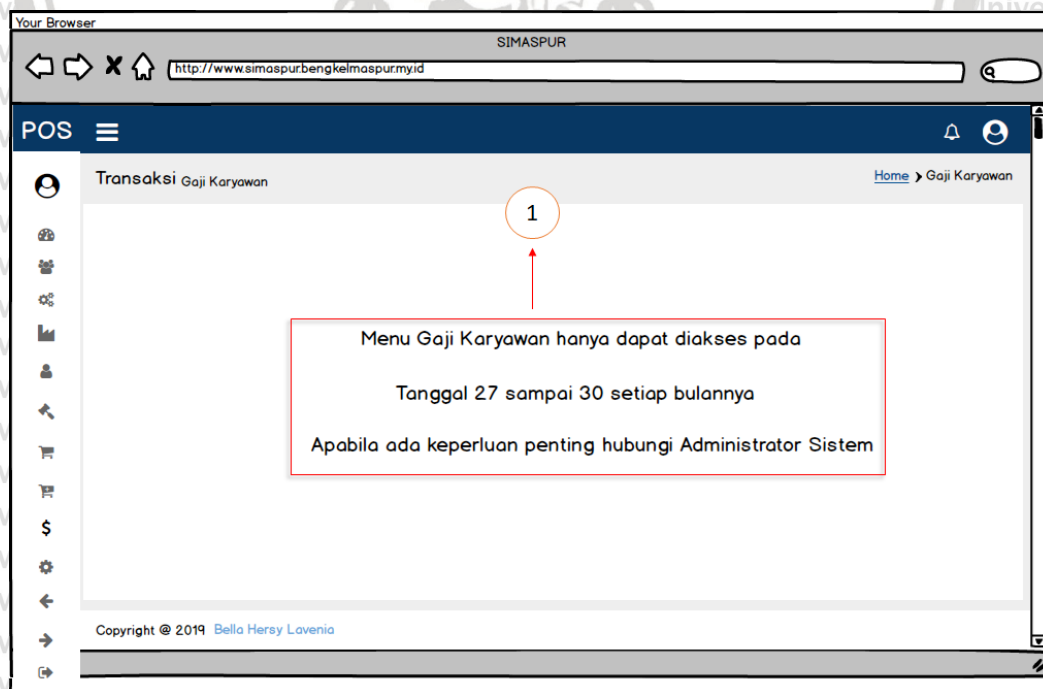
1. Formulir untuk menampilkan bulan ini.
2. Formulir untuk memasukkan karyawan.
3. Formulir untuk memasukkan absen atau unit.
4. Tombol tambah untuk menambahkan data gaji karyawan.
5. Tabel informasi gaji karyawan.
6. Tombol hapus untuk menghapus data gaji karyawan pada kolom tersebut.



Gambar 4.28 Perancangan Antarmuka Halaman Gaji Karyawan – Aktif

Sumber: Penulis (2019).

Pada Gambar 4.29 dapat dilihat rancangan antarmuka halaman gaji karyawan saat tidak aktif. Halaman ini dapat diaktifkan kembali apabila administrator keuangan mengajukan permintaan pengaktifan halaman pada administrator sistem.

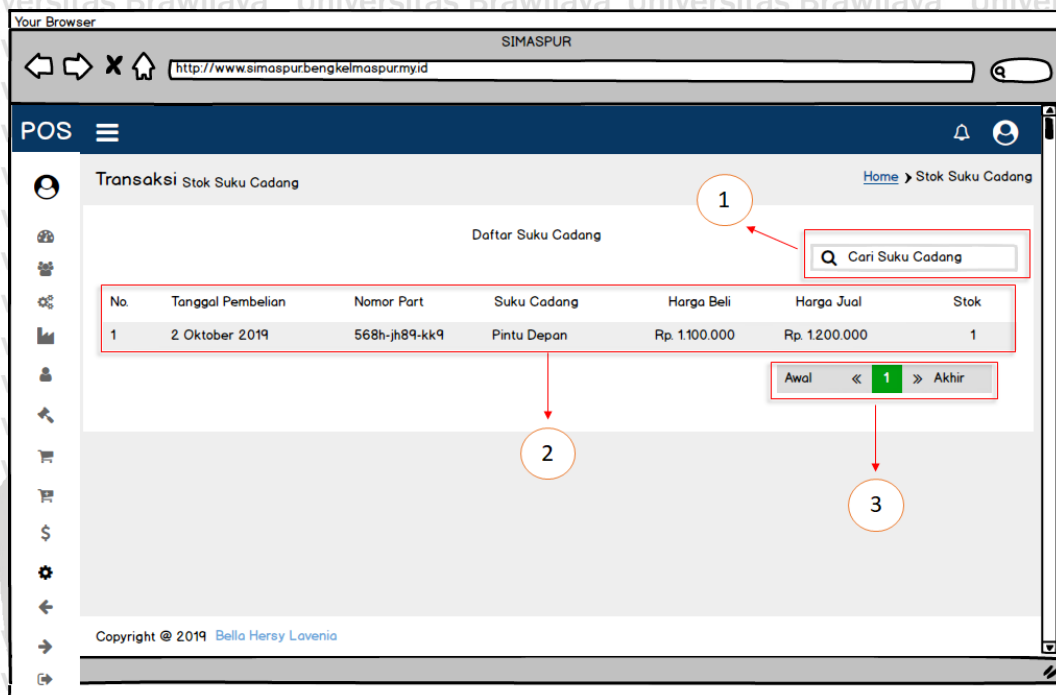


Gambar 4.29 Perancangan Antarmuka Halaman Gaji Karyawan – Tidak Aktif

Sumber: Penulis (2019).

4.8.5 Perancangan Antarmuka Halaman Stok Suku Cadang

Pada Gambar 4.30 terdapat rancangan antarmuka halaman stok suku cadang yang bisa diakses oleh semua pengguna. Halaman ini menampilkan informasi mengenai stok suku cadang yang dimiliki oleh perusahaan. Dengan adanya halaman ini pengguna tidak perlu mengecek langsung ke gudang apakah suku cadang yang akan digunakan tersedia atau tidak.



Gambar 4.30 Perancangan Antarmuka Halaman Stok Suku Cadang

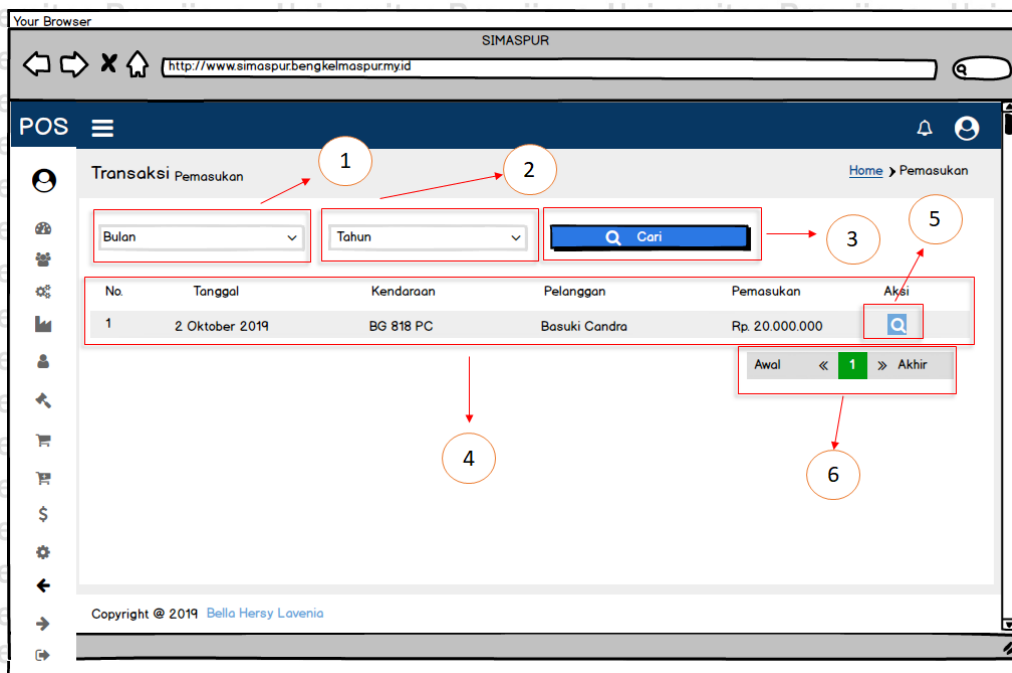
Sumber: Penulis (2019).

Keterangan rancangan antarmuka halaman gaji karyawan pada Gambar 4.28.

1. Formulir memasukan pencarian suku cadang berdasarkan nomor *part* atau nama suku cadang.
2. Tabel informasi mengenai informasi stok suku cadang.
3. Penomoran halaman tabel, pengguna dapat mengganti halaman sesuai dengan yang diinginkan.

4.8.6 Perancangan Antarmuka Halaman Pemasukan

Rancangan halaman pemasukan dapat dilihat pada Gambar 4.31, halaman ini akan menampilkan informasi mengenai pemasukan keuangan perusahaan. Untuk mendapatkan informasi tersebut, pengguna diharuskan memasukan bulan dan tahun untuk membatasi informasi yang akan ditampilkan sistem agar mudah dibaca oleh pengguna. Informasi yang akan ditampilkan berupa tanggal pemasukan, kendaraan yang diperbaiki, nama pelanggan serta total pemasukan dari transaksi tersebut. Untuk melihat lebih detail informasi setiap transaksinya pengguna dapat menekan tombol detail pada bagian aksi.



Gambar 4.31 Perancangan Antarmuka Halaman Pemasukan

Sumber: Penulis (2019).

Keterangan rancangan antarmuka halaman gaji karyawan pada Gambar

4.31.

1. Formulir memasukan bulan.
2. Formulir memasukan tahun.
3. Tombol cari untuk mencari informasi pemasukan berdasarkan bulan dan tahun yang telah dimasukan.
4. Tabel informasi pemasukan.
5. Tombol detail untuk menampilkan detail informasi transaksi pemasukan.
6. Penomoran halaman tabel informasi pemasukan yang ditampilkan.

4.8.7 Perancangan Antarmuka Halaman Pengeluaran

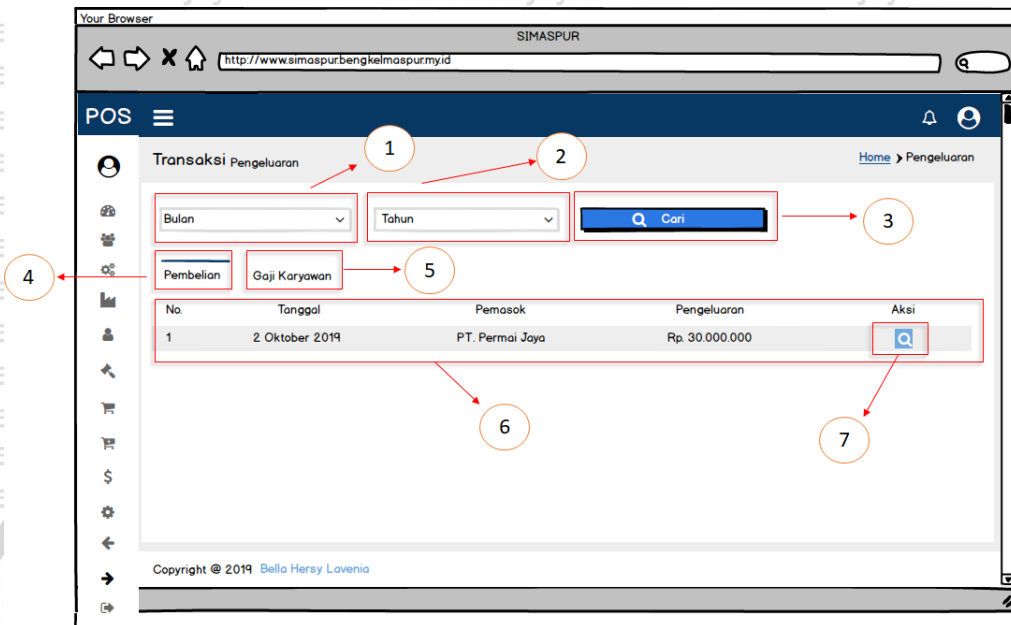
Halaman pengeluaran merupakan halaman yang akan menampilkan informasi mengenai pengeluaran keuangan perusahaan. Informasi yang akan ditampilkan pada halaman ini terdapat dua jenis yaitu informasi pengeluaran pembelian suku cadang dan informasi gaji karyawan. Untuk rancangan antarmuka halaman pengeluaran tab pembelian dapat dilihat pada Gambar 4.32 sedangkan rancangan antarmuka halaman pengeluaran tab gaji karyawan dapat dilihat pada Gambar 4.33

Keterangan rancangan antarmuka halaman gaji karyawan pada Gambar

4.32.

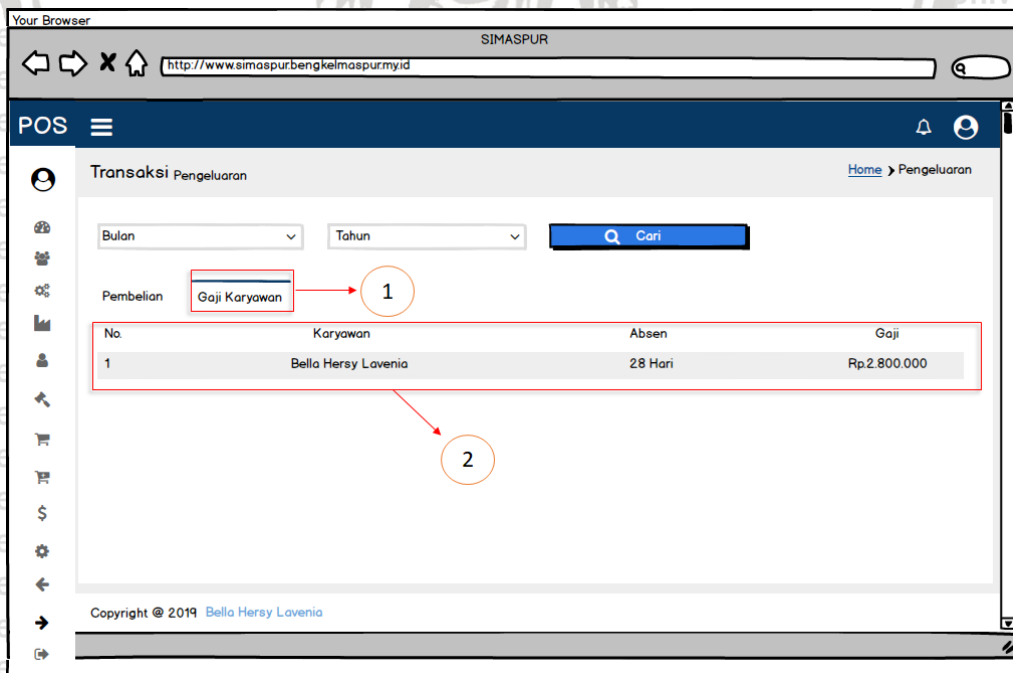
1. Formulir memasukan bulan.
2. Formulir memasukan tahun.
3. Tombol cari untuk mencari informasi pengeluaran berdasarkan bulan dan tahun yang telah dimasukan.

4. Tab pembelian.
5. Tab gaji karyawan.
6. Tabel informasi pengeluaran dari pembelian suku cadang.
7. Tombol detail, untuk mengetahui detail dari informasi transaksi pembelian.



Gambar 4.32 Perancangan Antarmuka Halaman Pengeluaran – Pembelian

Sumber: Penulis (2019).



Gambar 4.33 Perancangan Antarmuka Halaman Pengeluaran – Gaji Karyawan

Sumber: Penulis (2019).

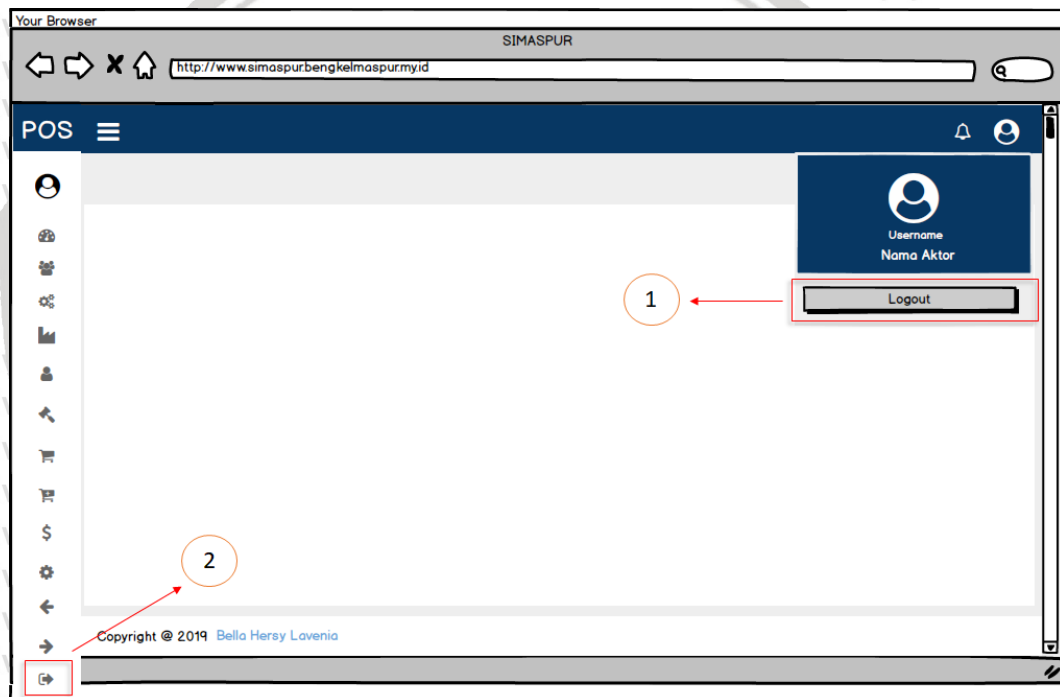
Keterangan rancangan antarmuka halaman pengeluaran tab gaji karyawan pada Gambar 4.33.

1. Tab untuk memilih informasi gaji karyawan.
2. Informasi mengenai gaji karyawan yang telah dimasukkan.

4.8.8 Perancangan Antarmuka Logout

Untuk keluar dari sistem, pengguna harus melakukan *logout* terlebih dahulu, terdapat dua cara untuk bisa keluar dari sistem. Cara tersebut dapat dilihat pada Gambar 4.34 dengan keterangan sebagai berikut:

1. Menekan foto pengguna pada kanan atas halaman, kemudian pilih tombol *Logout*.
2. Memilih menu Keluar pada *menu bar*.



Gambar 4.34 Perancangan Antarmuka Logout

Sumber: Penulis (2019).

BAB 5 IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini akan menjelaskan mengenai implementasi dan pengujian sistem yang akan dilakukan pada penelitian ini. Bahasan pada sub bab terbagi menjadi implementasi sistem dan pengujian sistem. Implementasi sistem akan membahas mengenai tahap pengembangan sistem informasi *point of sales* untuk bengkel. Setelah tahap pembangunan selesai maka akan dilanjutkan dengan pengujian sistem, dimana sistem akan di uji untuk melihat kelayakan sistem yang sedang dibangun.

5.1 Implementasi Sistem

Pada sub bab ini akan dijelaskan mengenai implementasi sistem yang akan dibagi menjadi beberapa bahasan yaitu spesifikasi lingkungan implementasi, implementasi kode program, implementasi antarmuka dan implementasi *cloud computing*.

5.1.1 Spesifikasi Lingkungan Implementasi

Sub bab ini akan menjelaskan mengenai spesifikasi dari lingkungan implementasi sistem berupa spesifikasi perangkat keras dan perangkat lunak yang akan digunakan dalam pengembangan sistem.

5.1.1.1 Spesifikasi Perangkat Keras

Lingkungan perangkat keras yang digunakan dalam membangun sistem informasi *point of sales* untuk bengkel dapat dilihat pada Tabel 5.1

Tabel 5.1 Spesifikasi Lingkungan Perangkat Keras

Processor	AMD A8-7410 APU (4CPUs) 2.2GHz
Memory (RAM)	4096 MB
Graphic Card	AMD Radeon (TM) R5 Graphics

Sumber: Penulis (2019).

5.1.1.2 Spesifikasi Perangkat Lunak

Lingkungan perangkat lunak yang digunakan dalam membangun sistem informasi *point of sales* untuk bengkel dapat dilihat pada Tabel 5.2

Tabel 5.2 Spesifikasi Lingkungan Perangkat Lunak

Sistem Operasi	Windows 10 Pro 64 bit
Text Editor	Visual Studio Code <i>Version</i> 1.39.2
Web Server	Apache <i>Version</i> 2.4.37
Penelusur Web	Firefox Developer Edition <i>Version</i> 71.0

Sumber: Penulis (2019).

5.1.2 Implementasi Kode Program

Sub bab implementasi kode program akan membahas mengenai kode program sebagai bentuk implementasi penelitian. Implementasi kode program sistem informasi *point of sales* untuk bengkel menggunakan bahasa pemrograman PHP dan *framework CodeIgniter*. Kode program yang akan dijelaskan pada sub bab ini meliputi kode program penjualan, gaji karyawan dan pemasukan.

5.1.2.1 Kode Program Penjualan

Kode program penjualan dibuat berdasarkan kebutuhan fungsional memasukan data penjualan. Kode program penjualan akan dijelaskan pada Tabel 5.3.

Tabel 5.3 Kode Program Penjualan

Kode Program Penjualan	
1	public function action(\$param)
2	{
3	if (!empty(\$_SERVER['HTTP_X_REQUESTED_WITH']) && strtolower
4	(\$_SERVER['HTTP_X_REQUESTED_WITH']) == 'xmlhttprequest') {
5	\$post = \$this->input->post(NULL, TRUE);
6	switch (\$param) {
7	case "addTransaksi":
8	if (!empty(\$post['kendaraan'])) {
9	\$data = array(
10	'no_transaksi' => \$post['no_penjualan'],
11	'id_kendaraan' => \$post['kendaraan'],
12	'tanggal_transaksi' => date('Y-m-
13	'd', strtotime(\$post['tanggal']))
14);
15	} else {
16	\$data = array(
17	'no_transaksi' => \$post['no_penjualan'],
18	'tanggal_transaksi' => date('Y-m-
19	'd', strtotime(\$post['tanggal']))
20);
21	if (\$this->transaksi_model->insert(\$data)) {
22	\$result = array('status' => 'success', 'pes
23	an' => "Berhasil menambahkan Data Transaksi");
24	} else {
25	\$result = array('status' => 'failed', 'pesa
26	n' => 'Gagal menambahkan Data Transaksi');
27	}
28	echo json_encode(\$result);
29	break;
30	case "updateTransaksi":
31	\$data = array(
32	'no_transaksi' => \$post['no_penjualan'],
33	'id_kendaraan' => \$post['kendaraan'],
34	'tanggal_transaksi' => date('Y-m-
35	'd', strtotime(\$post['tanggal'])),
36	'total_transaksi' => \$post['total_transaksi
37	'],



Tabel 5.3 Kode Program Penjualan (Lanjutan)

```

32         'status' => $post['status']
33     );
34     $kunci = array('no_transaksi' => $post['no_penj
35     ualan']);
36     if ($this->transaksi_model-
37     >update($data, $kunci)) {
38         $result = array('status' => 'success', 'pes
39         an' => 'Berhasil mengubah Data Transaksi');
40     } else {
41         $result = array('status' => 'failed', 'pesa
42         n' => 'Gagal mengubah Data Transaksi');
43     }
44     echo json_encode($result);
45     break;
46     case "getCart":
47         if (!empty($post['idjs'])) {
48             $data = $this->cartjasa_model-
49             >get_by(array('id_cart' => $post['idjs']), 1, NULL, TRUE, array('id
50             _cart', 'jenis_kendaraan', 'ukuran_kendaraan', 'nama_panel', 'jumlah'
51             ));
52             echo json_encode($data);
53         } else if (!empty($post['idsc'])) {
54             $data = $this->cartsc_model-
55             >get_by(array('id_cart' => $post['idsc']), 1, NULL, TRUE, array('id
56             _cart', 'suku_cadang', 'jumlah',));
57             echo json_encode($data);
58         } else {
59             $jasa_service = $this->cartjasa_model-
60             >get_by(array('no_transaksi' => $post['no_penjualan']));
61             $suku_cadang = $this->cartsc_model-
62             >get_by(array('no_transaksi' => $post['no_penjualan']));
63             echo json_encode(
64                 array(
65                     'jasa_service' => $jasa_service,
66                     'suku_cadang' => $suku_cadang
67                 )
68             );
69         }
70     break;
71     case "addCart":
72         $data = array(
73             'id_jasa_service' => $post['jasa_service'],
74             'id_suku_cadang' => $post['suku_cadang'],
75             'jumlah' => $post['jumlah'],
76             'no_transaksi' => $post['no_penjualan']
77         );
78         if ($this->cart_model->insert($data)) {
79             $result = array('status' => 'success', 'pes
80             an' => "Berhasil menambahkan Data Cart");
81         } else {
82             $result = array('status' => 'failed', 'pesa
83             n' => 'Gagal menambahkan Data Cart');
84         }
85         echo json_encode($result);
86         break;

```

Tabel 5.3 Kode Program Penjualan (Lanjutan)

```

74     case "updateCart":
75         $data = array(
76             'id_cart' => $post['id_cart'],
77             'jumlah' => $post['jumlah'],
78         );
79         $kunci = array('id_cart' => $post['id_cart']);
80         if ($this->cart_model->
81             >update($data, $kunci)) {
82             $result = array('status' => 'success', 'pe
83             san' => 'Berhasil mengubah Data Cart');
84         } else {
85             $result = array('status' => 'failed', 'pes
86             an' => 'Gagal mengubah Data Cart');
87         }
88         echo json_encode($result);
89         break;
90     case "delCart":
91         $kunci = $post['id_cart'];
92         if ($this->cart_model->delete($kunci)) {
93             $result = array('status' => 'success', 'pe
94             san' => 'Berhasil menghapus Data Cart');
95         } else {
96             $result = array('status' => 'failed', 'pes
97             an' => 'Gagal menghapus Data Cart');
98         }
99         echo json_encode($result);
100        break;
101    default:
102        echo json_encode(array('pesan' => "Request Tid
103        ak Valid"));
104    }
105 }

```

Sumber: Penulis (2019).

Penjelasan kode program pada Tabel 5.3 adalah sebagai berikut:

1. Baris 3 merupakan baris kode yang digunakan untuk mendeteksi adanya *request ajax*.
2. Baris 6 – 25 merupakan baris kode yang menangani permintaan untuk menambahkan data transaksi baru. *Array* data akan menyimpan *no_transaksi*, *id_kendaraan*, dan *tanggal_transaksi* sebagai atribut yang akan disimpan ke dalam tabel transaksi pada basis data. Jika data transaksi berhasil ditambahkan maka akan mengirim *result* berupa status *success* dan pesan “Berhasil menambahkan Data Transaksi”, namun jika gagal maka *result* akan berisi status *failed* dan pesan “Gagal menambahkan Data Transaksi”.
3. Baris 26 – 41 merupakan baris kode yang menangani permintaan untuk mengubah data transaksi. *Array* data akan menyimpan *no_transaksi*, *id_kendaraan*, *tanggal_transaksi*, *total_transaksi* dan status. Variabel kunci akan menyimpan *no_transaksi* yang menjadi *primary key* yang akan digunakan untuk menjalankan *query update* pada tabel transaksi. Jika data transaksi

berhasil diubah maka akan mengirim *result* berupa status *success* dan pesan “Berhasil mengubah Data Transaksi”, namun jika gagal maka *result* akan berisi status *failed* dan pesan “Gagal mengubah Data Transaksi”.

4. Baris 42 – 59 merupakan baris kode yang menangani permintaan untuk menampilkan data *cart*. Jika *idjs* tidak kosong maka akan dijalankan seleksi kondisi yang pertama yakni menampilkan data *cart* untuk jasa *service*, jika *idsc* tidak kosong maka akan dijalankan seleksi kondisi yang kedua yakni menampilkan data *cart* untuk suku cadang. Namun jika *idjs* dan *idsc* kosong maka akan dijalankan seleksi kondisi yang terakhir yakni menampilkan data *cart* berdasarkan *no_transaksi*.
5. Baris 60 – 73 merupakan baris kode yang menangani permintaan untuk menambah data *cart* baru. *Array* data akan menyimpan *id_jasa_service*, *id_suku_cadang*, jumlah dan *no_transaksi* sebagai atribut yang akan disimpan ke dalam tabel *cart* pada basis data. Jika data *cart* berhasil ditambahkan maka akan mengirim *result* berupa status *success* dan pesan “Berhasil menambahkan Data Cart”, namun jika gagal maka *result* akan berisi status *failed* dan pesan “Gagal menambahkan Data Cart”.
6. Baris 74 – 86 merupakan baris kode yang menangani permintaan untuk mengubah data *cart*. *Array* data akan menyimpan *id_cart* dan jumlah sebagai atribut yang akan terpengaruh saat *query update* dijalankan. Variabel kunci akan menyimpan *id_cart* yang menjadi *primary key* yang akan digunakan untuk menjalankan *query update* pada tabel *cart*. Jika data *cart* berhasil diubah maka akan mengirim *result* berupa status *success* dan pesan “Berhasil mengubah Data Cart”, namun jika gagal maka *result* akan berisi status *failed* dan pesan “Gagal mengubah Data Cart”.
7. Baris 87 – 95 merupakan baris kode yang menangani permintaan untuk menghapus data *cart*. Variabel kunci akan menyimpan *id_cart* yang menjadi *primary key* yang akan digunakan untuk menjalankan *query delete* pada tabel *cart*. Jika data *cart* berhasil dihapus maka akan mengirim *result* berupa status *success* dan pesan “Berhasil menghapus Data Cart”, namun jika gagal maka *result* akan berisi status *failed* dan pesan “Gagal menghapus Data Cart”.

5.1.2.2 Kode Program Gaji Karyawan

Kode program gaji karyawan merupakan hasil implementasi dari kebutuhan fungsional memasukan data gaji karyawan. Kode program gaji karyawan akan dijelaskan pada tabel 5.4.

Tabel 5.4 Kode Program Gaji Karyawan

Kode Program Gaji Karyawan	
1	<code>public function action(\$param)</code>
2	<code>{</code>
3	<code>if (!empty(\$_SERVER['HTTP_X_REQUESTED_WITH']) && strtolower(\$_SERVER['HTTP_X_REQUESTED_WITH']) == 'xmlhttprequest') {</code>
4	<code> \$post = \$this->input->post(NULL, TRUE);</code>



Tabel 5.4 Kode Program Gaji Karyawan (Lanjutan)

```

5      switch ($param) {
6          case "addGajiKaryawan":
7              $data = array('id_gaji_karyawan' => $post['id_gaji_karyaw
8                  an'],
9                  'id_karyawan' => $post['karyawan'],
10                 'absen' => $post['absen'],
11                 'total_gaji' => $post['upah']
12            );
13            if ($this->gajikaryawan_model->insert($data)) {
14                $result = array('status' => 'success', 'pes
15                    an' => "Berhasil menambahkan Data Gaji Karyawan");
16            } else {
17                $result = array('status' => 'failed', 'pesa
18                    n' => 'Gagal menambahkan Data Gaji Karyawan');
19            }
20            echo json_encode($result);
21            break;
22            case "getGajiKaryawan":
23                if (!empty($post['id_gaji_karyawan'])) {
24                    $data = $this->gajijoin_model-
25                    >get_by(array('id_gaji_karyawan' => $post['id_gaji_karyawan']), 1,
26                    NULL, TRUE, array('id_gaji_karyawan', 'nama_karyawan'));
27                    echo json_encode($data);
28                } else {
29                    $id = $post['id'];
30                    $data = $this->gajijoin_model-
31                    >get_by("id_gaji_karyawan LIKE '%$id%'", NULL, NULL, FALSE, array('
32                    id_gaji_karyawan', 'nama_karyawan', 'status', 'absen', 'total_gaji'
33                    ));
34                    echo json_encode($data);
35                }
36            break;
37            case "delGajiKaryawan":
38                $kunci = $post['id_gaji_karyawan'];
39                if ($this->gajikaryawan_model-
40                >delete($kunci)) {
41                    $result = array('status' => 'success', 'pes
42                    an' => 'Berhasil menghapus Data Gaji Karyawan');
43                } else {
44                    $result = array('status' => 'failed', 'pesa
45                    n' => 'Gagal menghapus Data Gaji Karyawan');
46                }
47            echo json_encode($result);
48            break;
49            default:
50                echo json_encode(array('pesan' => "Request Tida
51                k Valid"));
52            }
53        }

```

Sumber: Penulis (2019).

Penjelasan kode program gaji karyawan pada Tabel 5.4 sebagai berikut:

1. Baris 3 merupakan baris kode yang digunakan untuk mendeteksi adanya *request ajax*.
2. Baris 6 – 19 merupakan baris kode yang menangani permintaan menambahkan data gaji karyawan. Array data akan menyimpan *id_gaji_karyawan*, *id_karyawan*, absen dan *total_gaji* sebagai atribut yang akan disimpan ke dalam tabel *gaji_karyawan* pada basis data. Jika data gaji karyawan berhasil ditambahkan maka akan mengirim *result* berupa status *success* dan pesan “Berhasil menambahkan Data Gaji Karyawan”, namun jika gagal maka *result* akan berisi status *failed* dan pesan “Gagal menambahkan Data Gaji Karyawan”.
3. Baris 20 – 29 merupakan baris kode yang menangani permintaan menampilkan data gaji karyawan. Jika *id_gaji_karyawan* tidak kosong maka akan menjalankan seleksi kondisi yang pertama yaitu menampilkan data gaji karyawan berdasarkan *id_gaji_karyawan*. Namun jika *id_gaji_karyawan* kosong maka akan menjalankan seleksi kondisi terakhir yaitu menampilkan data gaji karyawan berdasarkan bulan dan tahun saat menu diakses.
4. Baris 30 – 38 merupakan baris kode yang menangani permintaan menghapus data gaji karyawan. Variabel kunci akan menyimpan *id_gaji_karyawan* yang menjadi *primary key* yang akan digunakan untuk menjalankan *query delete* pada tabel *gaji_karyawan*. Jika data gaji karyawan berhasil dihapus maka akan mengirim *result* berupa status *success* dan pesan “Berhasil menghapus Data Gaji Karyawan”, namun jika gagal maka *result* akan berisi status *failed* dan pesan “Gagal menghapus Data Gaji Karyawan”.

5.1.2.3 Kode Program Pemasukan

Kode program pemasukan dibuat berdasarkan adanya kebutuhan fungsional melihat data pemasukan. Kode program pemasukan akan dijelaskan pada Tabel 5.5.

Tabel 5.5 Kode Program Pemasukan

Kode Program Pemasukan	
1	public function action(\$param)
2	{
3	if (!empty(\$_SERVER['HTTP_X_REQUESTED_WITH']) && strtolower(\$_SERVER['HTTP_X_REQUESTED_WITH']) == 'xmlhttprequest'){
4	\$post = \$this->input->post(NULL, TRUE);
5	switch (\$param) {
6	case "getTransaksi":
7	\$bulan = \$post['bulan'];
8	\$tahun = \$post['tahun'];
9	\$data = \$this->transaksirelasi_model->get_by("MONTH(tanggal)='\$bulan' AND YEAR(tanggal)='\$tahun'", NULL, NULL, FALSE, array('no_transaksi', 'tanggal_transaksi', 'nomor_polisi', 'total_transaksi', 'nama_pelanggan'));
10	echo json_encode(\$data);
11	break;



Tabel 5.5 Kode Program Pemasukan (Lanjutan)

```

12     case "getCart":
13         $jasa_service = $this->cartjasa_model-
>get_by(array('no_transaksi' => $post['no_penjualan']));
14         $suku_cadang = $this->cartsc_model-
>get_by(array('no_transaksi' => $post['no_penjualan']));
15         echo json_encode(
16             array(
17                 'jasa_service' => $jasa_service,
18                 'suku_cadang' => $suku_cadang
19             )
20         );
21         break;
22     default:
23         echo json_encode(array('pesan' => "Request Tida
k Valid"));
24     }
25 }

```

Sumber: Penulis (2019).

Penjelasan kode program pemasukan pada Tabel 5.5 sebagai berikut:

1. Baris 3 merupakan baris kode yang digunakan untuk mendeteksi adanya *request ajax*.
2. Baris 6 – 11 merupakan baris kode yang menangani permintaan menampilkan data transaksi. Baris kode ini akan menjalankan *query select* dengan menampilkan data berdasarkan variabel bulan dan tahun.
3. Baris 12 – 21 merupakan baris kode yang menangani permintaan menampilkan data *cart*. Baris kode ini akan menjalankan *query select* pada tabel *cart* berdasarkan *no_transaksi*.

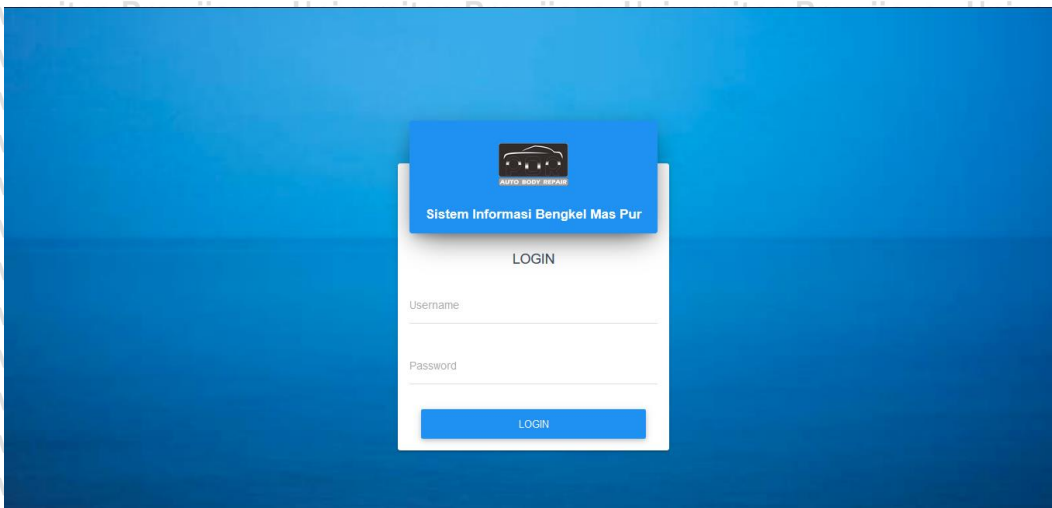
5.1.3 Implementasi Antarmuka

Implementasi antarmuka merupakan tahapan implementasi pembuatan antarmuka yang dibuat menggunakan bahasa HTML dan *Javascript* berdasarkan perancangan antarmuka yang telah dibahas pada bab sebelumnya.

5.1.3.1 Implementasi Antarmuka Halaman Login

Antarmuka halaman *login* merupakan halaman yang digunakan oleh pengguna untuk bisa masuk ke dalam sistem informasi *point of sales* untuk bengkel. Pengguna dapat masuk ke dalam sistem apabila memiliki akun yang telah diberikan oleh administrator sistem. Antarmuka halaman *login* dapat dilihat pada Gambar 5.1.



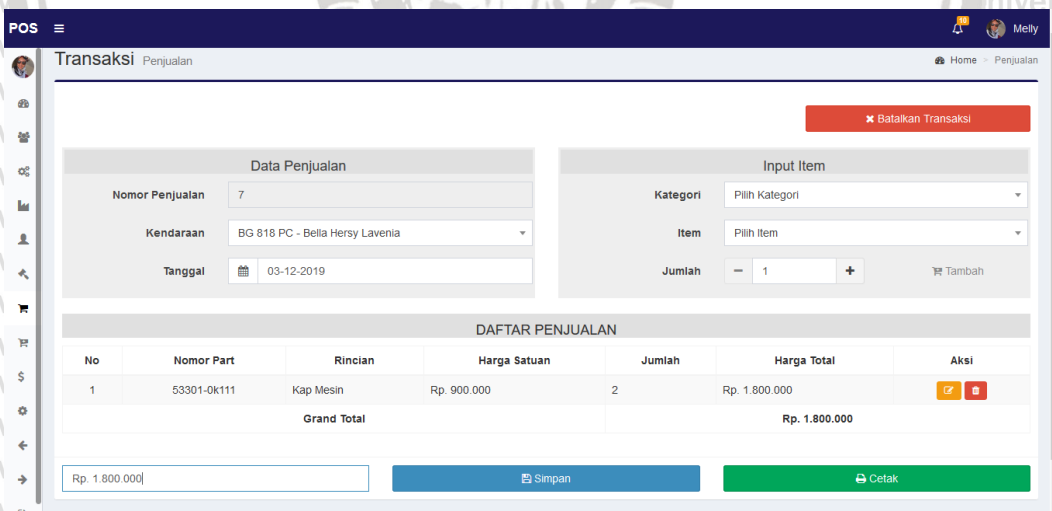


Gambar 5.1 Antarmuka Halaman Login

Sumber: Penulis (2019).

5.1.3.2 Implementasi Antarmuka Halaman Penjualan

Antarmuka halaman penjualan merupakan halaman yang digunakan oleh administrator keuangan untuk memasukkan data penjualan. Data penjualan yang dimasukan oleh administrator keuangan akan disimpan dalam sistem dan sistem juga akan melakukan pencetakan bukti pembayaran apabila pelanggan memintanya pada administrator keuangan. Antarmuka halaman penjualan dapat dilihat pada Gambar 5.2.



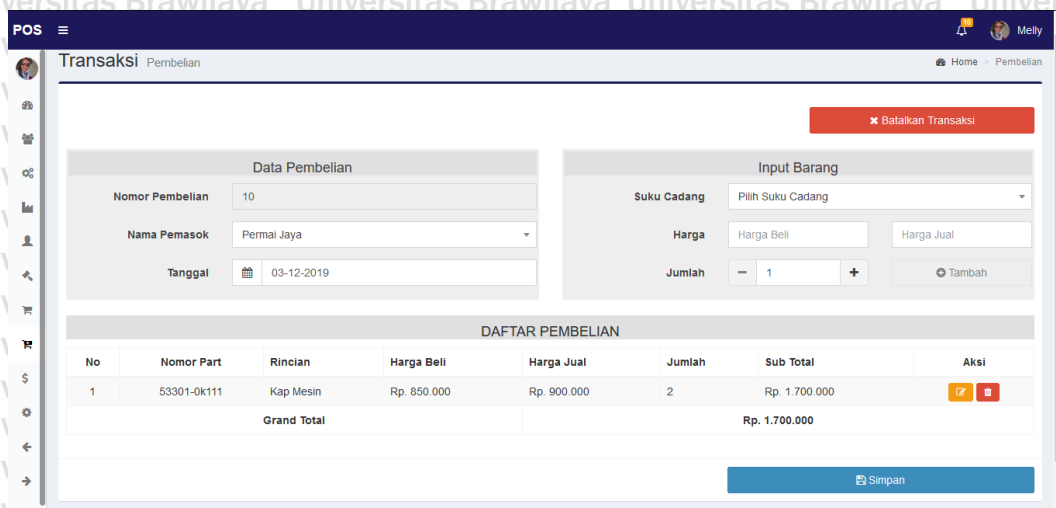
Gambar 5.2 Antarmuka Halaman Penjualan

Sumber: Penulis (2019).



5.1.3.3 Implementasi Antarmuka Halaman Pembelian

Antarmuka halaman pembelian merupakan halaman yang digunakan oleh administrator keuangan untuk memasukkan data pembelian suku cadang. Sistem akan menyimpan data tersebut untuk digunakan sebagai data keuangan perusahaan. Antarmuka halaman pembelian dapat dilihat pada Gambar 5.3.

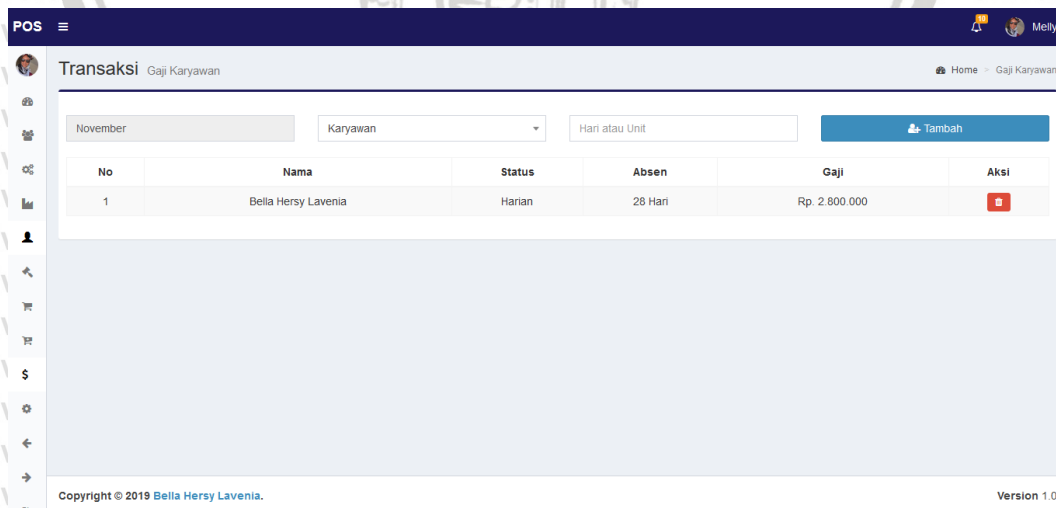


Gambar 5.3 Antarmuka Halaman Pembelian

Sumber: Penulis (2019).

5.1.3.4 Implementasi Antarmuka Halaman Gaji Karyawan

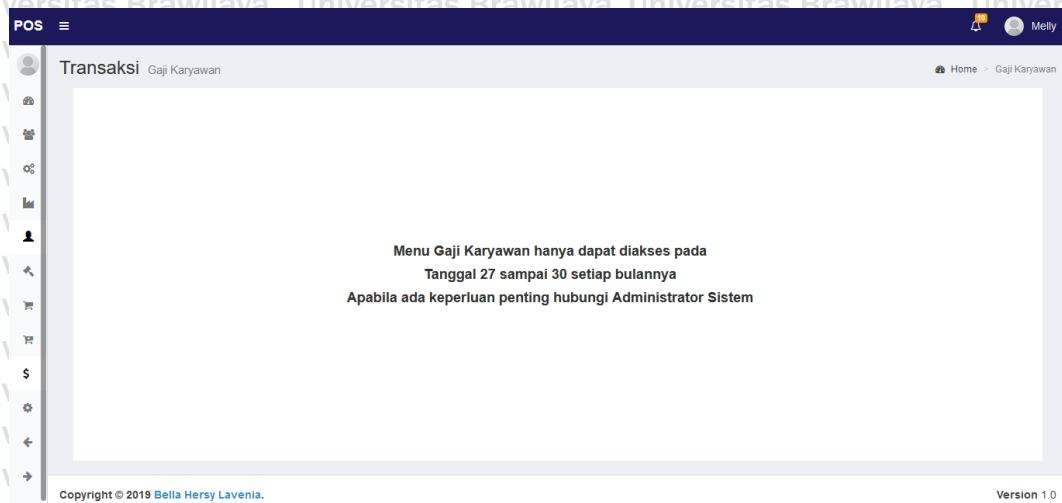
Antarmuka halaman gaji karyawan merupakan halaman yang digunakan untuk memasukkan dan menyimpan data gaji karyawan perusahaan. Antarmuka halaman gaji karyawan hanya dapat diakses pada tanggal 27 sampai 30 disetiap bulannya. Antarmuka halaman gaji karyawan saat aktif dapat dilihat pada Gambar 5.4.



Gambar 5.4 Antarmuka Halaman Gaji Karyawan – Aktif

Sumber: Penulis (2019).

Antarmuka halaman gaji karyawan saat tidak dapat diakses akan menampilkan informasi seperti pada Gambar 5.5.

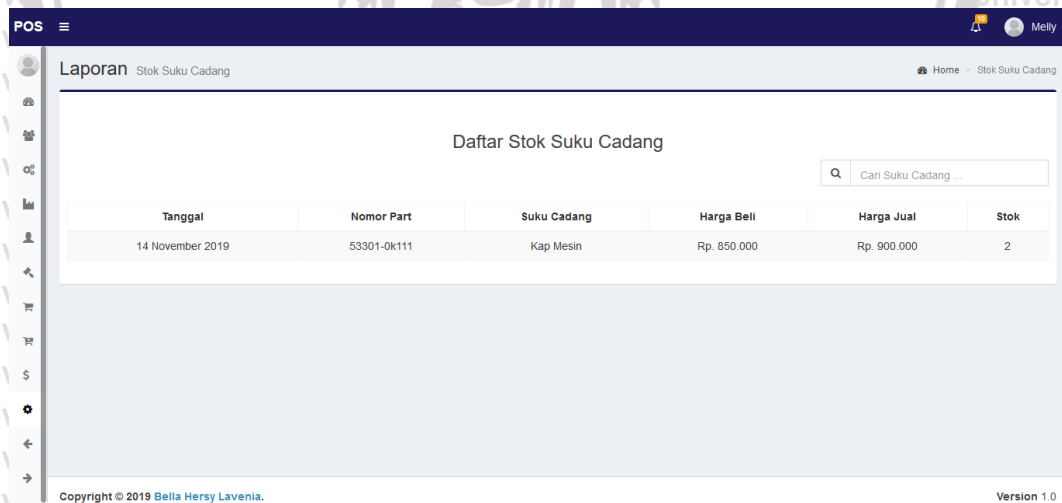


Gambar 5.5 Antarmuka Halaman Gaji Karyawan – Tidak Aktif

Sumber: Penulis (2019).

5.1.3.5 Implementasi Antarmuka Halaman Stok Suku Cadang

Antarmuka halaman stok suku cadang merupakan halaman yang digunakan oleh pengguna untuk melihat stok suku cadang yang terdapat pada gudang perusahaan. Informasi yang ditampilkan pada halaman ini berasal dari halaman pembelian. Antarmuka halaman stok suku cadang dapat dilihat pada Gambar 5.6.



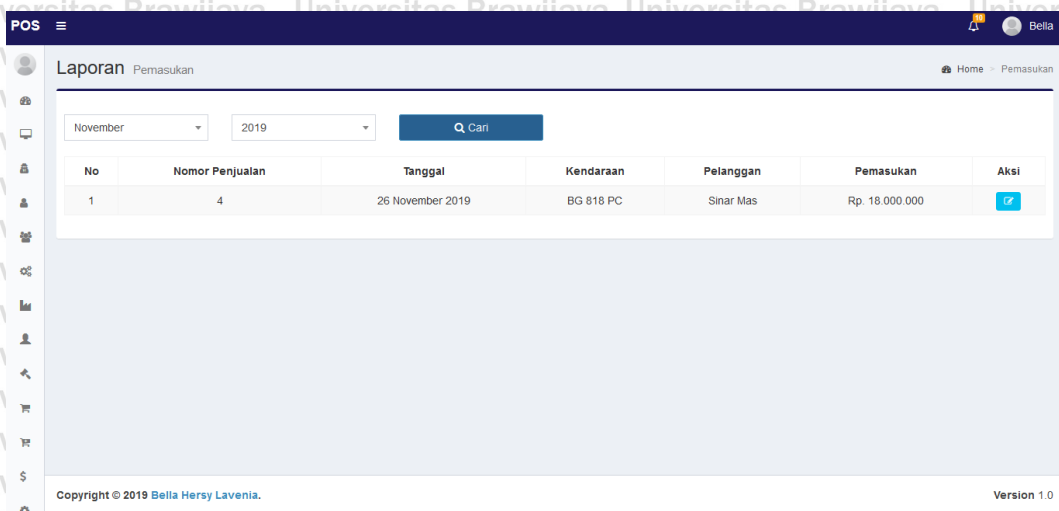
Gambar 5.6 Antarmuka Halaman Stok Suku Cadang

Sumber: Penulis (2019).

5.1.3.6 Implementasi Antarmuka Halaman Pemasukan

Antarmuka halaman pemasukan adalah halaman yang menampilkan informasi mengenai pemasukan keuangan perusahaan pada bulan dan tahun yang

telah dimasukan oleh pengguna. Informasi yang ditampilkan dalam halaman ini berasal dari halaman penjualan. Antarmuka halaman pemasukan dapat dilihat pada Gambar 5.7.

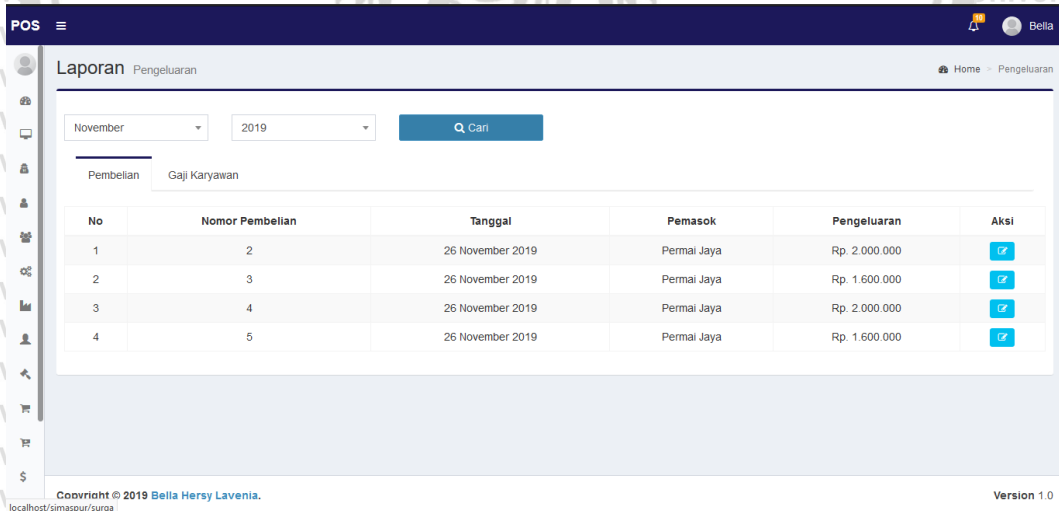


Gambar 5.7 Antarmuka Halaman Pemasukan

Sumber: Penulis (2019).

5.1.3.7 Implementasi Antarmuka Halaman Pengeluaran

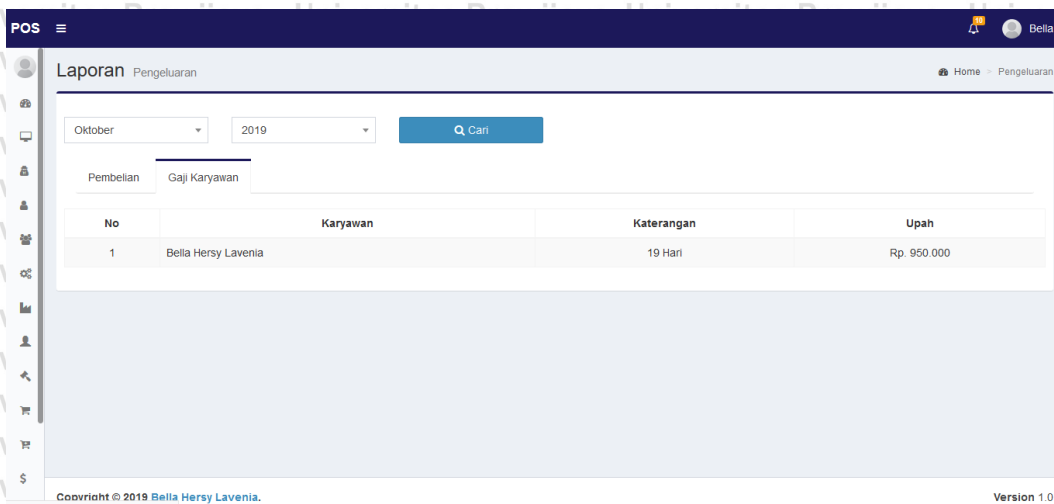
Antarmuka halaman pengeluaran merupakan halaman yang menyediakan informasi mengenai pengeluaran keuangan perusahaan. Informasi pada halaman ini didapat dari halaman pembelian dan juga halaman gaji karyawan. Antarmuka halaman pengeluaran untuk *tab* pembelian dapat dilihat pada Gambar 5.8.



Gambar 5.8 Antarmuka Halaman Pengeluaran – Pembelian

Sumber: Penulis (2019).

Untuk antarmuka halaman pengeluaran *tab* gaji karyawan dapat dilihat pada Gambar 5.9.

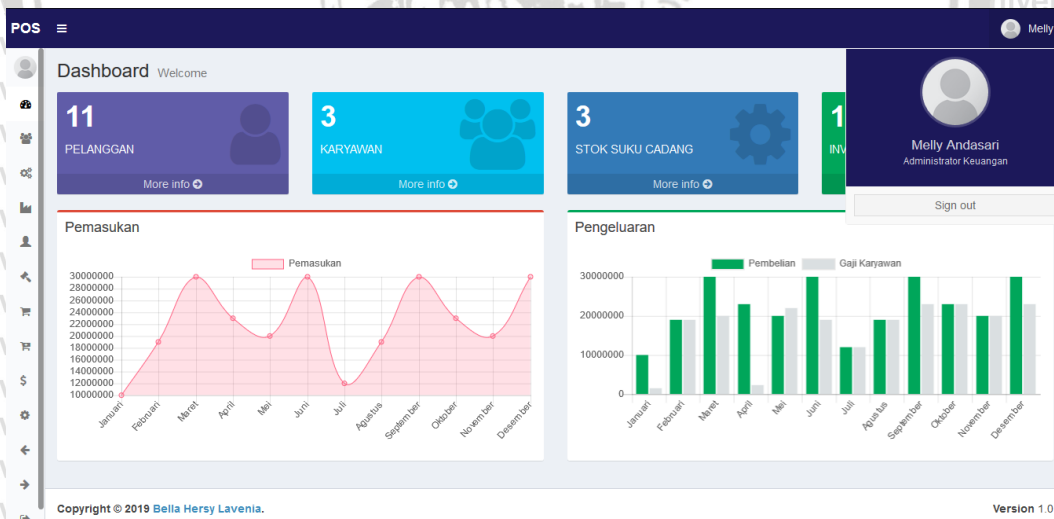


Gambar 5.9 Antarmuka Halaman Pengeluaran – Gaji Karyawan

Sumber: Penulis (2019).

5.1.3.8 Implementasi Antarmuka Logout

Pengguna dapat melakukan *logout* dari sistem dengan dua cara. Pertama, pengguna dapat menekan tombol pada ikon pengguna pada kanan atas pada halaman manapun, kemudian tekan tombol *Log Out*. Kedua, pengguna dapat memilih menu Keluar yang terdapat pada *menu bar*. Antarmuka *logout* dapat dilihat pada Gambar 5.10.



Gambar 5.10 Antarmuka Logout

Sumber: Penulis (2019).

5.1.4 Implementasi Cloud Computing

Implementasi *cloud computing* pada sistem informasi *point of sales* untuk bengkel memanfaatkan layanan yang disediakan oleh *Amazon Web Service* (AWS).

5.1.4.1 Spesifikasi AWS EC2

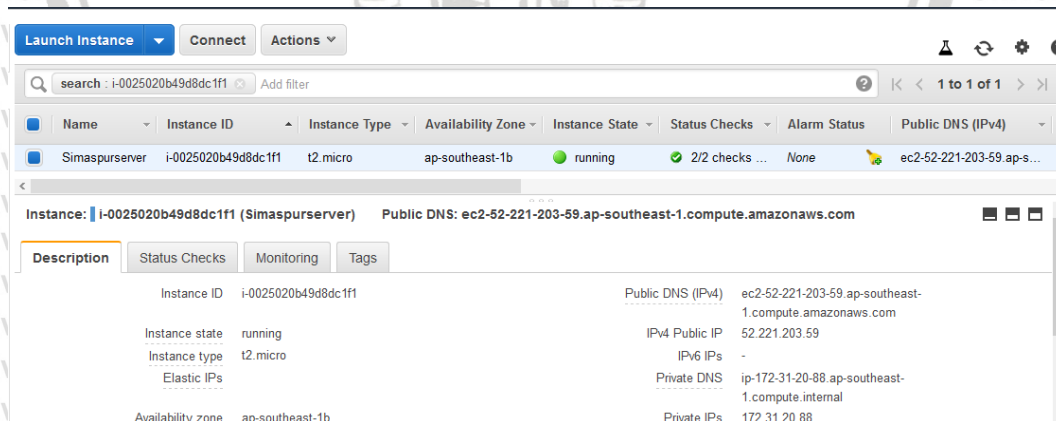
Amazon Web Service Elastic Compute Cloud (AWS EC2) adalah layanan web yang memberikan kapasitas komputasi yang aman dan berukuran fleksibel. Pada penelitian ini sistem akan menggunakan AWS EC2 sebagai *web server* dengan spesifikasi *instance* yang harus didefinisikan diawal saat konfigurasi *instance* tersebut seperti pada Tabel 5.6.

Tabel 5.6 Spesifikasi Instance AWS EC2

Sistem Operasi	<i>Ubuntu Server</i> 16.04 LTS
Instance Type	t2.micro
Memory (RAM)	8 GiB <i>EBS General Purpose</i> (SSD)
Security Group	SSH port 22
	HTTP port 80
	MYSQL port 443

Sumber: Penulis (2019).

Setelah konfigurasi selesai maka pada halaman *Instances* akan menampilkan detail informasi *instance* yang telah dibuat. Dapat dilihat pada Gambar 5.11 informasi yang ditampilkan diantaranya berupa IP versi 4 yang dapat diakses secara publik serta beberapa informasi lain mengenai status dari *instance* tersebut. Publik IP dari *instance* yang digunakan adalah 52.221.203.59.



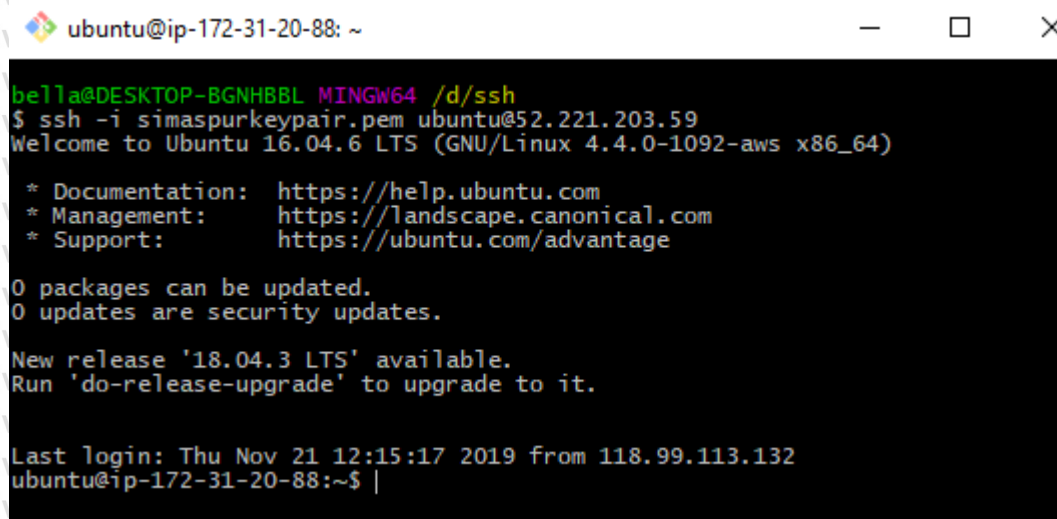
Gambar 5.11 Instance AWS EC2

Sumber: Penulis (2019).

AWS EC2 menyediakan keamanan pada setiap *instance*, keamanan tersebut berupa *key pair* yang hanya diberikan pada pemilik *instance* tersebut. *Key pair* tersebut nantinya digunakan sebagai kata sandi saat akan mengakses



instance dengan protokol SSH. Gambar 5.12 merupakan percobaan mengakses *instance* menggunakan *key pair* dan protokol SSH.



Gambar 5.12 Login pada Instance Menggunakan Protokol SSH

Sumber: Penulis (2019).

5.1.4.2 Komponen Perangkat Lunak

Setelah menyiapkan AWS EC2 sebagai perangkat keras di dalam *cloud*, maka selanjutnya adalah menyiapkan komponen perangkat lunak yang mendukung berjalannya sistem agar dapat diakses oleh pengguna. Berikut komponen perangkat lunak penunjang sistem yang perlu dipasang dalam *instance*, dapat dilihat Tabel 5.7.

Tabel 5.7 Komponen Perangkat Lunak

Web Server	<i>Apache Version 2.4.41</i>
Bahasa Pemrograman	<i>PHP Version 7.2.24</i>
Basis Data	<i>MySQL Version 14.14</i>

Sumber: Penulis (2019).

Apache versi 2.4.41 atau yang lebih dikenal dengan *Apache2* merupakan salah satu *web server* yang sering digunakan untuk menjalankan web pada sisi *server*. Konfigurasi *Apache2* relatif mudah sehingga akan sangat membantu administrasi sistem dalam membangun dan memelihara *web server* pada *cloud*. Gambar 5.13 membahas mengenai pemasangan *apache2* pada *Ubuntu 16.04*.

```

ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libbrotli1 libcurl3
  libjansson4 liblua5.2-0 libnghttp2-14 libssl1.1 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
  openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libbrotli1 libcurl3
  libjansson4 liblua5.2-0 libnghttp2-14 libssl1.1 ssl-cert
0 upgraded, 15 newly installed, 0 to remove and 51 not upgraded.
Need to get 4,130 kB of archives.
After this operation, 13.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libcurl3 amd64 7.47.0-1ubuntu2.14 [186 kB]
Get:2 http://ppa.launchpad.net/ondrej/apache2/ubuntu xenial/main amd64 libapr1 amd64 1.6.2-1+ubuntu16.04.1+deb.sury.org-2 [96.3 kB]
Get:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libjansson4 amd64 2.7-3ubuntu0.1 [27.1 kB]

```

Gambar 5.13 Pemasangan Apache2 pada Ubuntu 16.04

Sumber: Penulis (2019).

Setelah pemasangan *web server* perlu adanya pemasangan bahasa pemrograman yang digunakan yaitu PHP7 untuk menunjang berjalannya *web server*. Pada Gambar 5.14 ditunjukkan pemasangan PHP versi 7.2.24 pada *Ubuntu 16.04*.

```

ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo apt install php7.2 libapache2-mod-php7.2 php
7.2-common php7.2-mbstring php7.2-xmlrpc php7.2-gd php7.2-xml php7.2-mysql
php7.2-cli php7.2-zip php7.2-curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libargon2-0 libfontconfig1 libgd3
  libjbig0 libjpeg-turbo8 libjpeg8 libpcre3 libsodium23 libtiff5 libwebp6
  libxmlrpc-epi0 libxpm4 libzip4 php-common php7.2-json php7.2-opcache
  php7.2-readline
Suggested packages:
  php-pear libgd-tools
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libapache2-mod-php7.2 libargon2-0
  libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8 libsodium23
  libtiff5 libwebp6 libxmlrpc-epi0 libxpm4 libzip4 php-common php7.2
  php7.2-cli php7.2-common php7.2-curl php7.2-gd php7.2-json
  php7.2-mbstring php7.2-mysql php7.2-opcache php7.2-readline php7.2-xml
  php7.2-xmlrpc php7.2-zip
The following packages will be upgraded:
  libpcre3
1 upgraded, 29 newly installed, 0 to remove and 52 not upgraded.
Need to get 7,038 kB of archives.
After this operation, 26.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libjpeg-turbo8 amd64 1.4.2-0ubuntu3.3 [111 kB]
Get:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial/main amd64 fonts-dejavu-core all 2.35-1 [1,039 kB]
Get:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 fontconfig-config all 2.11.84-0ubuntu1.1 [48.9 kB]

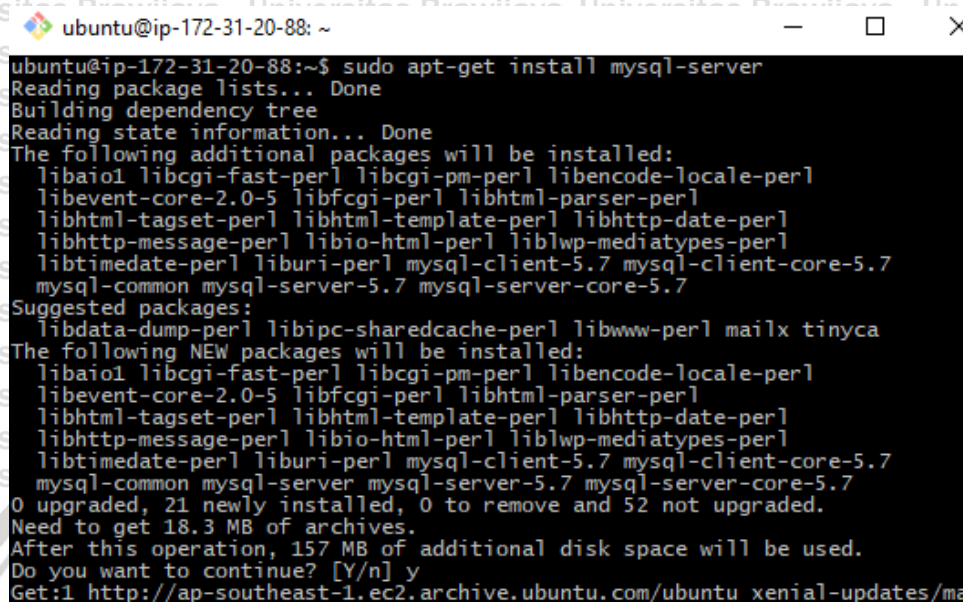
```

Gambar 5.14 Pemasangan PHP7 pada Ubuntu 16.04

Sumber: Penulis (2019).



Basis data yang digunakan dalam penelitian ini adalah MySQL versi 14.14. MySQL akan menyimpan semua data-data yang ada pada sistem dan membantu bahasa pemrograman dalam mengolah dan menampilkan data. Pada Gambar 5.15 ditunjukkan mengenai pemasangan MySQL pada *Ubuntu 16.04*.



```

ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl
  libevent-core-2.0-5 libfcgi-perl libhtml-parser-perl
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
  libhttp-message-perl libio-html-perl liblwp-mediatypes-perl
  libtimedate-perl liburi-perl mysql-client-5.7 mysql-client-core-5.7
  mysql-common mysql-server-5.7 mysql-server-core-5.7
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinycsa
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl
  libevent-core-2.0-5 libfcgi-perl libhtml-parser-perl
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
  libhttp-message-perl libio-html-perl liblwp-mediatypes-perl
  libtimedate-perl liburi-perl mysql-client-5.7 mysql-client-core-5.7
  mysql-common mysql-server mysql-server-5.7 mysql-server-core-5.7
0 upgraded, 21 newly installed, 0 to remove and 52 not upgraded.
Need to get 18.3 MB of archives.
After this operation, 157 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/ma

```

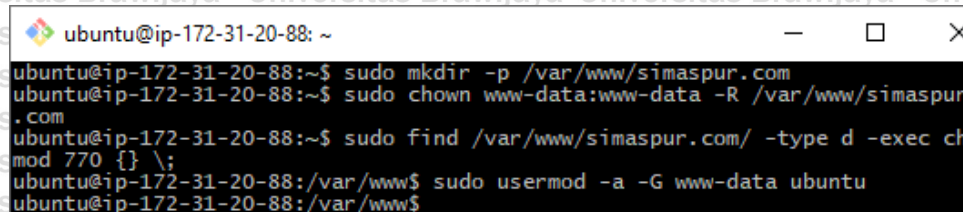
Gambar 5.15 Pemasangan MySQL pada Ubuntu 16.04

Sumber: Penulis (2019).

5.1.4.3 Konfigurasi Ubuntu

Pada sub bab ini akan membahas mengenai konfigurasi tambahan yang bertujuan untuk memudahkan administrator sistem dalam mengakses folder kode program dan basis data sistem informasi *point of sales* untuk bengkel. Sistem operasi yang digunakan adalah sistem operasi yang menggunakan antarmuka CLI, sehingga akan sedikit menyulitkan administrator sistem saat akan mengubah atau menambahkan file kode program. Untuk itu diperlukan sebuah perangkat lunak pihak ketiga yang berfungsi untuk mengakses instance dari komputer lokal administrator sistem, salah satu perangkat lunak yang akan mendukung penggunaan protokol SFTP adalah FileZilla.

FileZilla akan mengakses instance dengan menggunakan akses pengguna ubuntu. Sedangkan folder kode program hanya dapat diakses secara penuh oleh pengguna root, maka perlu dilakukan konfigurasi seperti pada Gambar 5.16.



```

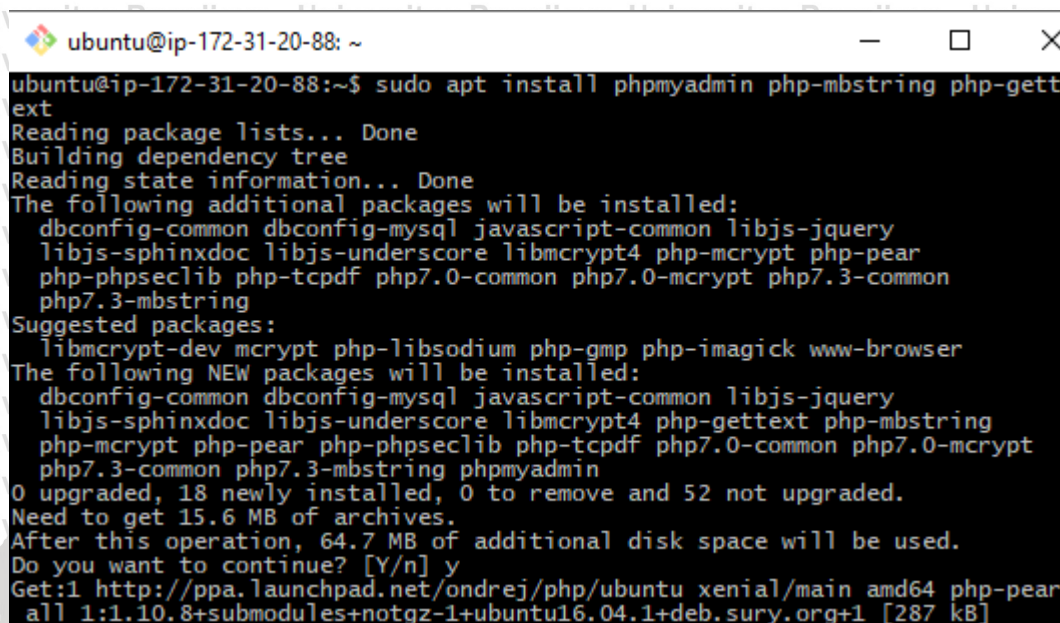
ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo mkdir -p /var/www/simaspur.com
ubuntu@ip-172-31-20-88:~$ sudo chown www-data:www-data -R /var/www/simaspur.com
ubuntu@ip-172-31-20-88:~$ sudo find /var/www/simaspur.com/ -type d -exec ch
mod 770 {} \;
ubuntu@ip-172-31-20-88:/var/www$ sudo usermod -a -G www-data ubuntu
ubuntu@ip-172-31-20-88:/var/www$

```

Gambar 5.16 Konfigurasi Akses Folder Kode Program

Sumber: Penulis (2019).

Agar basis data dapat mudah diakses maka perlu dilakukan pemasangan phpMyAdmin. PhpMyAdmin akan membantu administrator sistem dalam mengakses dan mengatur basis data dengan tampilan web, sehingga dapat membantu dalam meminimalisir kesalahan. Pemasangan phpMyAdmin dapat dilihat pada Gambar 5.17.



```

ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo apt install phpmyadmin php-mbstring php-gettext
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  dbconfig-common dbconfig-mysql javascript-common libjs-jquery
  libjs-sphinxdoc libjs-underscore libmcrypt4 php-mcrypt php-pear
  php-phpseclib php-tcpdf php7.0-common php7.0-mcrypt php7.3-common
  php7.3-mbstring
Suggested packages:
  libmcrypt-dev mcrypt php-lib sodium php-gmp php-imagick www-browser
The following NEW packages will be installed:
  dbconfig-common dbconfig-mysql javascript-common libjs-jquery
  libjs-sphinxdoc libjs-underscore libmcrypt4 php-gettext php-mbstring
  php-mcrypt php-pear php-phpseclib php-tcpdf php7.0-common php7.0-mcrypt
  php7.3-common php7.3-mbstring phpmyadmin
0 upgraded, 18 newly installed, 0 to remove and 52 not upgraded.
Need to get 15.6 MB of archives.
After this operation, 64.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ppa.launchpad.net/ondrej/php/ubuntu xenial/main amd64 php-pear
all 1:1.10.8+submodules+notgz-1+ubuntu16.04.1+deb.sury.org+1 [287 kB]

```

Gambar 5.17 Pemasangan PhpMyAdmin pada Ubuntu 16.04

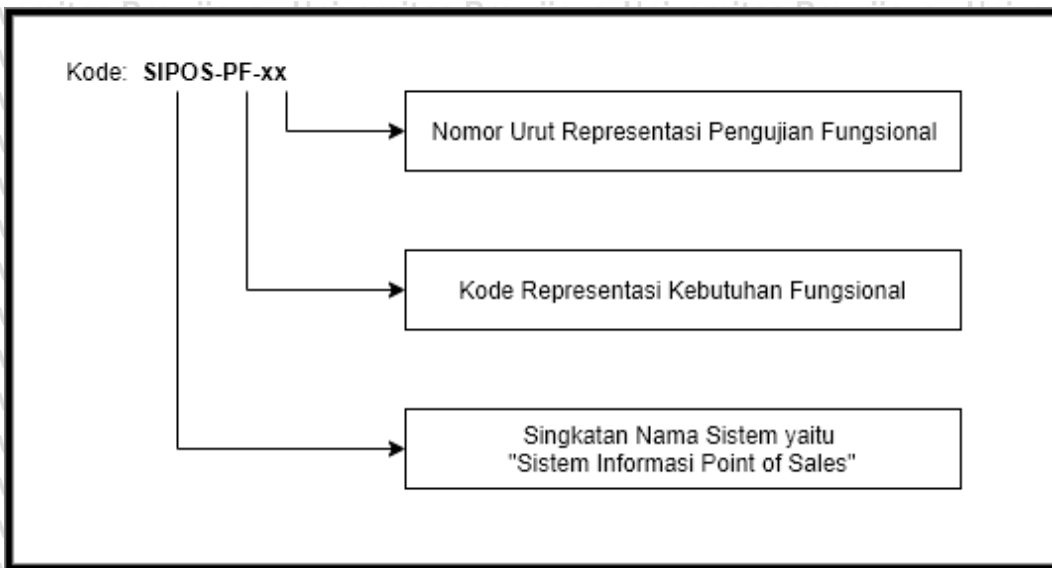
Sumber: Penulis (2019).

5.2 Pengujian Sistem

Pada sub bab ini akan membahas mengenai tahap pengujian yang akan dilakukan pada sistem informasi *point of sales* untuk bengkel. Pengujian akan dilakukan dengan menguji kebutuhan fungsional dan kebutuhan non-fungsional dari sistem informasi *point of sales* untuk bengkel. Kebutuhan fungsional akan di uji dengan menggunakan pengujian fungsional atau pengujian validasi pada setiap kebutuhan fungsional sistem. Untuk kebutuhan non-fungsional akan di uji kompatibilitas sistem terhadap beberapa penelusur web, pengujian non-fungsional ini akan menggunakan perangkat lunak *Sortsite*.

5.2.1 Pengujian Fungsional

Pengujian fungsional akan menguji sistem informasi *point of sales* untuk bengkel menggunakan uji validasi untuk memastikan sistem dapat berjalan sesuai dengan kebutuhan fungsional yang telah didefinisikan pada bab sebelumnya. Setiap pengujian fungsional akan diberikan penomoran dengan format pada Gambar 5.18. Aturan penomoran terdiri dari tiga kata, kata pertama merupakan singkatan dari nama sistem yaitu "*Sistem Informasi Point of Sales*". Kata kedua adalah representasi dari pengujian fungsional, dan kata ketiga merupakan representasi dari nomor urut pengujian fungsional.



Gambar 5.18 Aturan Penomoran Pengujian Fungsional

Sumber: Penulis (2019).

Pengujian fungsional sistem informasi *point of sales* untuk bengkel akan dijelaskan pada Tabel 5.8 sampai Tabel 5.15. Pada Tabel 5.8 dijelaskan mengenai langkah-langkah pengujian validasi melakukan *login* dengan memasukkan *username* dan *password* yang benar. Kasus SIPOS-PF-01 bertujuan untuk memastikan bahwa sistem dapat digunakan untuk melakukan *login* ke dalam sistem. Pengujian dilakukan terhadap penguji yang teridentifikasi sebagai administrator keuangan.

Tabel 5.8 Uji Validasi Melakukan Login

Kode Pengujian	SIPOS-PF-01.
Nama Kasus Uji	Kasus uji melakukan login.
Kode Kebutuhan Fungsional	SIPOS-KF-01.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melakukan <i>login</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengakses sistem. 2. Penguji memasukkan “melly” pada formulir <i>username</i>. 3. Penguji memasukkan “passwordku” pada formulir <i>password</i>. 4. Penguji menekan tombol <i>Login</i>.
Hasil yang diharapkan	Sistem menampilkan halaman penjualan.
Keluaran	Sistem menampilkan halaman penjualan.
Status Uji	Valid.

Sumber: Penulis (2019).

Pada Tabel 5.9 dilakukan uji validasi memasukan data penjualan. Pengujian ini dilakukan terhadap penguji yang teridentifikasi sebagai administrator keuangan. Dalam kasus SIPOS-PF-02 penguji akan memastikan apakah sistem dapat menyimpan data penjualan serta mencetak bukti pembayaran.

Tabel 5.9 Uji Validasi Memasukan Data Penjualan

Kode Pengujian	SIPOS-PF-02.
Nama Kasus Uji	Kasus uji memasukan data penjualan.
Kode Kebutuhan Fungsional	SIPOS-KF-02.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji melakukan <i>login</i>. 2. Penguji memilih menu penjualan. 3. Penguji memilih “BG 818 – Bella Hersy Lavenia” pada formulir kendaraan. 4. Penguji memilih “14-11-2019” pada formulir tanggal. 5. Penguji memilih “Suku Cadang” pada formulir kategori. 6. Penguji memilih “53301-0k111 – Kap Mesin” pada formulir item. 7. Penguji memasukan “2” pada formulir jumlah. 8. Penguji menekan tombol tambah. 9. Penguji memasukan “900000” pada formulir uang yang diterima. 10. Penguji menekan tombol simpan. 11. Penguji menekan tombol konfirmasi. 12. Penguji menekan tombol cetak. 13. Penguji menekan tombol faktur.
Hasil yang diharapkan	Sistem menyimpan data penjualan dan mencetak faktur pembayaran.
Keluaran	Sistem menyimpan data penjualan dan mencetak faktur pembayaran.
Status Uji	Valid.

Sumber: Penulis (2019).

Pada Tabel 5.10 dilakukan uji validasi memasukan data pembelian, pengujian ini akan melibatkan penguji yang teridentifikasi sebagai administrator keuangan. Pada kasus SIPOS-PF-03 penguji akan memastikan apakah sistem dapat menyimpan data pembelian ke dalam sistem.

Tabel 5.10 Uji Validasi Memasukan Data Pembelian

Kode Pengujian	SIPOS-PF-03.
Nama Kasus Uji	Kasus uji memasukan data pembelian.
Kode Kebutuhan Fungsional	SIPOS-KF-03.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk memasukan data pembelian.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji melakukan <i>login</i>. 2. Penguji memilih menu pembelian. 3. Penguji memilih “Permai Jaya” pada formulir nama pemasok. 4. Penguji memilih “14-11-2019” pada formulir tanggal. 5. Penguji memilih “53301-0k111 – Kap Mesin” pada formulir item. 6. Penguji memasukan “850000” pada formulir harga beli. 7. Penguji memasukan “900000” pada formulir harga jual. 8. Penguji memasukan “2” pada formulir jumlah. 9. Penguji menekan tombol tambah. 10. Penguji menekan tombol simpan. 11. Penguji menekan tombol konfirmasi.
Hasil yang diharapkan	Sistem menyimpan data pembelian.
Keluaran	Sistem menyimpan data pembelian.
Status Uji	Valid.

Sumber: Penulis (2019).

Pada Tabel 5.11 dilakukan uji validasi memasukan data gaji karyawan dengan penguji yang teridentifikasi sebagai administrator keuangan. Pada kasus uji SIPOS-PF-04 penguji akan memasukan data gaji karyawan dan memastikan sistem dapat menyimpan data tersebut.

Tabel 5.11 Uji Validasi Memasukan Data Gaji Karyawan

Kode Pengujian	SIPOS-PF-04.
Nama Kasus Uji	Kasus uji memasukan data gaji karyawan.
Kode Kebutuhan Fungsional	SIPOS-KF-04.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk memasukan data gaji karyawan.

Tabel 5.11 Uji Validasi Memasukan Data Gaji Karyawan (Lanjutan)

Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji melakukan <i>login</i>. 2. Penguji memilih menu gaji karyawan. 3. Penguji memilih “Bella Hersy Lavenia” pada formulir karyawan. 4. Penguji memasukkan “28” pada formulir absen. 5. Penguji menekan tombol tambah.
Hasil yang diharapkan	Sistem menyimpan data gaji karyawan dan menampilkannya.
Keluaran	Sistem menyimpan data gaji karyawan dan menampilkannya.
Status Uji	Valid.

Sumber: Penulis (2019).

Dari hasil pengujian validasi dengan melibatkan dua penguji sebagai calon pengguna, 14 kasus uji dinyatakan valid dari total 15 kasus uji maka dapat ditarik kesimpulan bahwa pengujian fungsional mendapatkan hasil 93% valid . Dengan hasil tersebut dapat dinyatakan bahwa kebutuhan fungsional sistem sebagian besar terpenuhi.

5.2.2 Pengujian Non-Fungsional

Pengujian non-fungsional akan menguji kompatibilitas sistem terhadap penelusur web yang mungkin akan digunakan oleh pengguna sistem. Pengujian ini dilakukan untuk mengetahui sudah sesuainkah sistem yang telah dikembangkan apabila sistem dijalankan pada penelusur web yang berbeda-beda. Untuk melihat hasil uji kompatibilitas sistem informasi *point of sales* untuk bengkel terdapat pada Gambar 5.19.

Browser	IE	Edge	Firefox	Safari	Opera	Chrome	iOS	Android				
Version	11	18	69	≤ 12 13	60	77	≤ 11 12 13	≤ 3 4*				
Critical Issues	✓	✓	●	✓	✓	✓	●	✓	✓	✓	✓	✓
Major Issues	●	✓	●	●	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	●	●	●	●	●	●	●	●	✓	✓	●	●

Gambar 5.19 Hasil Uji Kompatibilitas menggunakan Perangkat Lunak Sortsite.

Sumber: Penulis (2019).

Pengujian dilakukan menggunakan perangkat lunak *Sortsite* dengan memasukkan alamat URL sistem, pada penelitian ini akan menggunakan alamat IP dari *instance* AWS EC2 sebagai *web server* sistem informasi *point of sales* untuk bengkel. Pada Gambar 5.12. dapat dilihat ada beberapa masalah yang terjadi saat sistem diakses menggunakan penelusur web tertentu.



Gambar 5.20 Petunjuk Kehilangan Konten atau Fungsionalitas

Sumber: Penulis (2019).

Gambar 5.20 merupakan gambar petunjuk saat sistem mengalami kehilangan konten atau fungsionalitas tertentu saat diakses menggunakan penelusur web tertentu. Sistem informasi *point of sales* untuk bengkel terdapat dua *critical issues* pada *firefox* versi 69 dan *chrome* versi 77. *Critical issues* terjadi pada halaman *login* dikarenakan *firefox* dan *chrome* merupakan penelusur web yang akan menampilkan beberapa peringatan keamanan untuk bidang kata sandi yang tidak menggunakan protokol HTTPS, ini disebabkan karena sistem informasi *point of sales* untuk bengkel masih menggunakan protokol HTTP.

Gambar 5.21 Petunjuk Masalah Berat pada Tata Letak atau Kinerja

Sumber: Penulis (2019).

Gambar 5.21 merupakan gambar petunjuk masalah berat pada tata letak atau kinerja sistem tertentu saat diakses menggunakan penelusur web tertentu. Pada sistem informasi *point of sales* untuk bengkel terdapat dua *major issues* saat diakses menggunakan *internet explorer* versi 11 dan *safari* versi kurang dari sama dengan 12 dan 13. Masalah ini dikarenakan *internet explorer* dan *safari* yang tidak mendukung *bootstrap*.

Gambar 5.22 Petunjuk Masalah Ringan pada Tata Letak atau Kinerja

Sumber: Penulis (2019).

Gambar 5.22 merupakan gambar petunjuk saat sistem mengalami masalah ringan pada tata letak atau kinerja saat diakses menggunakan penelusur web tertentu. Terdapat 10 *minor issues* saat diakses oleh beberapa penelusur web, hal ini terjadi karena penelusur web tersebut tidak mendukung beberapa properti dari CSS.

BAB 6 PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan maka dapat kesimpulan sebagai berikut.

1. Model proses bisnis *to-be* dihasilkan dari kegiatan pemodelan proses bisnis yang telah dilakukan, dengan adanya model proses bisnis *to-be* akan memberikan kemudahan pada proses penjualan Bengkel Mas Pur Baturaja. Hal tersebut dikarena terdapat sistem informasi yang dibangun akan berfungsi untuk membantu pekerjaan administrator keuangan serta meminimalisir terjadinya kesalahan dan kerusakan data. Selanjutnya analisis kebutuhan dengan mengumpulkan kebutuhan pengguna kemudian dapat dirumuskan menjadi kebutuhan sistem. Kebutuhan sistem yang dihasilkan adalah delapan kebutuhan fungsional dan satu kebutuhan non-fungsional sistem informasi *point of sales* untuk bengkel yang menggambarkan sebagian besar kemampuan pengelolaan data keuangan Bengkel Mas Pur Baturaja. Urutan aktifitas yang dihasilkan dari analisis kebutuhan akan disajikan kedalam *use case diagram* untuk menggambarkan apa saja yang dapat dilakukan oleh pengguna terhadap sistem. Perancangan sistem yang dilakukan menggunakan model UML dengan menggunakan *activity diagram*, *sequence diagram*, *class diagram*, *physical data model* serta perancangan antarmuka.
2. Setelah tahap analisis kebutuhan dan perancangan sistem selesai selanjutnya sistem akan mulai dibangun. Hasil dari tahap implementasi ini menghasilkan sistem informasi *point of sales* untuk bengkel berbasis *cloud computing*. Sistem ini dikembangkan dengan *framework CodeIgniter* serta menggunakan bahasa pemrograman PHP, HTML, CSS dan *Javascript* serta dengan menggunakan basis data MySQL. Sistem informasi *point of sales* untuk bengkel ini menghasilkan fitur *login*, penjualan, pembelian, gaji karyawan, stok suku cadang, pemasukan, pengeluaran dan *logout* yang semuanya disesuaikan dengan kebutuhan dari Bengkel Mas Pur. Sistem informasi *point of sales* untuk bengkel akan berjalan diatas *cloud*, dengan menggunakan layanan Amazon Web Service. Layanan AWS yang akan digunakan dalam pengembangan sistem adalah AWS EC2 yang merupakan layanan penyedia komputasi awan yang ukuran dan kapasitasnya dapat disesuaikan oleh pengguna. Di dalam AWS EC2 telah dipasang perangkat lunak seperti *Apache*, PHP, dan MySQL untuk mendukung berjalannya sistem agar lebih mudah diakses oleh pengguna.
3. Pengujian fungsional menggunakan uji validasi pada fungsi yang ada pada sistem, kemudian pengujian akan menjalankan prosedur uji dan menilai apakah hasil dari pengujian yang dilakukan valid atau tidak. Pengujian fungsional dilakukan oleh administrator keuangan dan manajer, hasil yang didapat dari pengujian fungsional menyatakan bahwa sistem telah 93% memenuhi sebagian besar kebutuhan fungsional. Pada tahap pengujian non-fungsional kompatibilitas menggunakan perangkat lunak *sortsite*, dengan hasil sistem

dapat berjalan sesuai dengan harapan pada beberapa penelusur web. Namun saat sistem diakses menggunakan penelusur web *chrome* versi 77 dan *firefox* versi 69 terjadi *critical issues* yaitu pada halaman login dikarenakan *chrome* dan *firefox* merupakan penelusur web yang akan memberikan peringatan keamanan apabila terdapat aktivitas *login* tanpa menggunakan protokol HTTPS, sedangkan sistem berjalan menggunakan protokol HTTP. Selain itu terdapat dua *major issues* dan sepuluh *minor issues* yang disebabkan oleh penggunaan *bootstrap* dan CSS yang tidak didukung oleh beberapa penelusur web yang menjadi kasus uji.

6.2 Saran

Berikut saran yang dapat digunakan pada penelitian lebih lanjut mengenai sistem informasi *point of sales* untuk bengkel adalah sebagai berikut.

1. Perlu dilakukan penambahan fitur baru yaitu fitur yang digunakan untuk mencetak bukti pembayaran transaksi penjualan yang telah disimpan sebelumnya. Fitur ini akan sangat membantu apabila pelanggan meminta bukti pembayaran dikemudian hari. Fitur gaji karyawan juga perlu diperbarui apabila kedepannya perusahaan menerapkan *fingerprnt* sebagai verifikasi absen setiap karyawan yang terdapat pada perusahaan.
2. Evaluasi antamuka sistem informasi *point of sales* untuk bengkel perlu dilakukan agar dapat mengetahui kekurangan dan perbaikan yang mungkin perlu dilakukan. Selain itu juga perlu adanya evaluasi kebutuhan pengguna, apakah sistem dapat memenuhi setiap kebutuhan dari pengguna sistem informasi *point of sales* untuk bengkel.

DAFTAR REFERENSI

Amazon Web Services, 2019. *Amazon Web Services*. s.l.:Tersedia melalui: <https://aws.amazon.com> [Diakses 19 Agustus 2019].

Ambyssoft Inc., 2003. *Physical Data Model(PDM)s: An Agile Introduction*.

Anhar, 2010. *Panduan Menguasai PHP dan MySQL secara Otodidak*. [e-book] Jakarta: Mediakita. Tersedia di: Google Books <http://books.google.com> [Diakses 20 Agustus 2019].

Berner, S., 2003. *About the Development of a Point of Sale System: an Experience Report*. s.l., IEEE Computer Society.

Booch, G., Rumbaugh, J. & Jacobson, I., 1998. *The Unified Modelling Language User Guide*. Boston: Addison Valley.

Dyson, J., 2015. *Paperless Receipt Solution (PRS) System*. [Online] Available at: <http://www.jamesdysonaward.org> [Diakses 8 Januari 2020].

Hall, J., 2008. *Accounting Information System*. Mason: Cengage Learning.

Hewitt, C., 2008. ORGs for Scalable, Robust, Privacy-Friendly Client Cloud Computing. *Massachusetts Institute of Technology (emeritus)*, Volume 12, pp. 96-99.

Hidayatullah, P. & Kawistara, J., 2014. *Pemrograman Web*. Bandung: Informatika.

Kendall, K. & Kendall, J., 2011. *System Analysis and Design*. New Jersey: Person Education.

Laudon, K. C. A. J. P., 2014. *Management Information Systems: Managing the Digital Firm.13th ed..* United States of America: Pearson Education.

Namasudra, S., Roy, P. & Balusamy, B., 2017. *Cloud Computing: Fundamentals and Research Issues*. s.l., 2017 Second International Conference on Recent Trends and Challenges in Computational Models.

Object Management Group, Inc., 2011. *Business Process Model and Notation(BPMN)*. [Online] Available at: http://www.oatsolutions.com.br/artigos/SpecBPMN_v2.pdf [Diakses 20 Agustus 2019].

O'Brien, J. G. M. a. M., 2014. *Sistem Informasi Manajemen Edisi 9 Buku 1*. Jakarta: Salemba Empat.

Pressman, R. S., 2010. *Software Engineering: A Practitioner,s Approach*. s.l.:McGraw-Hill.

Raharjo, B., 2015. *Belajar Otodidak Framework CodeIgniter*. Bandung: Informatika Bandung.

Rosa, A. & Shalahudin, M., 2016. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.

Solichin, A., 2016. *Pemrograman Web dengan PHP dan MySQL*. [e-book] Jakarta: Budi Luhur. Tersedia di: Google Books <http://books.google.com> [Diakses 20 Agustus 2019].

Sommerville, I., 2011. *Software Engineering: Ninth Edition*. New York: Addison Wesley.

Supono, V. P., 2018. *Pemrograman Web dengan Menggunakan PHP dan Framework Codeigniter*. [e-book] Yogyakarta: Deepublish. Tersedia di: Google Books <http://books.google.com> [Diakses 20 Agustus 2019].

Whitten, J. L. & Bentley, L. D., 2007. *Systems Analysis & Design Methods*. West Lafayette: McGraw-Hill.

Wieclaw, L. et al., 2017. *Cloud Computing Technologies in "Smart City" Projects*. Bucharest, The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications .



LAMPIRAN A HASIL WAWANCARA



A.1 Hasil Wawancara Dengan Administrasi Keuangan

Hasil Wawancara

Tanggal : 30 Agustus 2019, 13.00 – Selesai.
 Tempat : Bengkel Mas Pur Baturaja
 Nama : Melly Andasari
 Jabatan : Administrasi

Pertanyaan	Jawaban
Kendala yang dihadapi dengan metode yang digunakan saat ini.	Pembukuan transaksi masih menggunakan buku dan sulit melakukan perekapan transaksi perbulan.
Pelaporan transaksi pada pemilik bengkel mas pur	Untuk pelaporan biasanya pemilik atau manajer akan ikut langsung dalam proses transaksi dan pembayaran tagihan pelanggan, sehingga langsung menerima uang pembayaran dari pelanggan tanpa adanya pelaporan.
Fitur sistem yang diinginkan	Terdapat menu penjualan yang dapat mencetak bukti pembayaran, pembelian stok gudang, ada laporan perbulan keuangan dari transaksi bengkel.
Penjabaran laporan keuangan	Laporan keuangan terdiri dari pemasukan yang direkap dari data penjualan, kemudian pengeluaran itu hasil rekap dari data pembelian stok gudang dan data penggajian karyawan.
Penjabaran fitur penggajian karyawan	Setiap akhir bulan ada rekap gaji karyawan yang dihitung, berdasarkan hari atau unit yang dikerjakan oleh karyawan atau montir.
Jenis karyawan yang ada di bengkel mas pur.	2, karyawan harian dihitung per hari setiap masuk kerja dan karyawan borongan dihitung per unit kendaraan yang diselesaikan.

Baturaja, 11 November 2019



Melly Andasari
 Administrasi

A.2 Hasil Wawancara Dengan Manajer

Hasil Wawancara

Tanggal : 30 Agustus 2019, 09.00 – Selesai.
 Tempat : Bengkel Mas Pur Baturaja
 Nama : Purwanto
 Jabatan : Pemilik Bengkel

Pertanyaan	Jawaban
Kendala yang dihadapi dengan metode yang digunakan saat ini.	Laporan keuangan tidak ada, pembukuan transaksi tidak lengkap atau ada yang hilang.
Jumlah rata-rata kendaraan yang melakukan perbaikan dalam satu bulan.	20 sampai 25 kendaraan.
Jumlah karyawan saat ini	10 orang
Mekanisme transaksi di bengkel mas pur baturaja	Pemilik kendaraan menghubungi manajer atau pemilik bengkel untuk mendapatkan kepastian tanggal kendaraan bisa dimasukan ke bengkel, lalu jika sudah dibengkel manajer bisa mengecek dan memeriksa kendaraan untuk di hitung harga perbaikan sementara. Jika pemilik kendaraan setuju maka manajer akan menghubungi administrator keuangan untuk pengecekan stok suku cadang yang akan digunakan. Setelah perbaikan selesai administrator keuangan akan membuat bukti pembayaran untuk pemilik kendaraan.
Jenis kendaraan yang melakukan perbaikan di bengkel mas pur baturaja.	Mobil pribadi sampai alat berat proyek.
Jenis pelanggan dari bengkel mas pur baturaja.	Individu, perusahaan dan asuransi
Jenis layanan jasa yang ditawarkan oleh bengkel mas pur baturaja.	Ketok, poles dan cat, semua yang berhubungan dengan body, karena kita tidak melayani jasa untuk perbaikan mesin.
Sistem informasi sebelumnya.	Ada, hanya media promosi namun tidak dilanjutkan lagi sewa hostingnya karena minim pengunjung dan tidak efektif, sekarang orang lebih sering buka instagram dan media sosial.
Lebih memilih membangun infrastruktur TI sendiri atau menyewa.	Sewa, karena keterbatasan lahan dan dana jika harus membangun sendiri.
Sistem yang akan dikembangkan tergetnya akan digunakan berapa pengguna.	Untuk sementara 3 orang, namun rencananya beberapa tahun kedepan akan dibuka cabang baru bengkel mas pur baturaja dan sistem akan dipakai disana juga.

Baturaja, 11 November 2019


PURWANTO
 AUTO BODY REPAIR
 Pemilik Bengkel

LAMPIRAN B KONFIGURASI AWS EC2

B.1 Konfigurasi Awal AWS EC2

B.1.1 Memilih Sistem Operasi

Step 1: Choose an Amazon Machine Image (AMI)
 An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search: ubuntu server 16.04

Quick Start (2)

- My AMIs (0)**
- AWS Marketplace (21)**
 - Free tier eligible** **Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type.** Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
 - Root device type: ebs
 - Virtualization type: hvm
 - ENA Enabled: Yes
- Community AMIs (1367)**

Free tier only (1)

Ubuntu Server 16.04 LTS (HVM) with SQL Server 2017 Standard - ami-089951c15775770aa
 Microsoft SQL Server 2017 Standard edition on Ubuntu Server 16.04 LTS. The AMI also comes pre-installed with NET Core 2.0 and PowerShell 6.0.
 Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

B.1.2 Memilih Jenis Instance

Step 2: Choose an Instance Type
 Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All Instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Buttons: Cancel Previous **Review and Launch** Next: Configure Instance Details

B.1.3 Konfigurasi Detail Instance

Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

- Number of instances: 1 [Launch into Auto Scaling Group](#)
- Purchasing option: Request Spot instances
- Network: vpc-44545823 (default) [Create new VPC](#)
- Subnet: No preference (default subnet in any Availability Zone) [Create new subnet](#)
- Auto-assign Public IP: Use subnet setting (Enable)
- Placement group: Add instance to placement group
- Capacity Reservation: Open [Create new Capacity Reservation](#)
- IAM role: None [Create new IAM role](#)

Buttons: Cancel, Previous, **Review and Launch**, Next: Add Storage

B.1.4 Konfigurasi Memori

Step 4: Add Storage
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GIB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-008fa23fc82732b52	10	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Buttons: Cancel, Previous, **Review and Launch**, Next: Add Tags

B.1.5 Konfigurasi Tags

Step 5: Add Tags
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webservier. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
Name	Simaspurserver	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Buttons: Cancel, Previous, **Review and Launch**, Next: Configure Security Group



B.1.6 Konfigurasi Security Group

Step 6: Configure Security Group

Assign a security group: Create a new security group Select an existing security group

Security Group ID	Name	Description	Actions
sg-559fe82a	default	default VPC security group	Copy to new
sg-06c4bef326ce92a8f	launch-wizard-1	launch-wizard-1 created 2019-10-17T16:07:03.380+07:00	Copy to new

Inbound rules for sg-06c4bef326ce92a8f (Selected security groups: sg-06c4bef326ce92a8f)

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	:::0	
SSH	TCP	22	0.0.0.0/0	
SSH	TCP	22	:::0	
HTTPS	TCP	443	0.0.0.0/0	
HTTPS	TCP	443	:::0	

Buttons: Cancel, Previous, Review and Launch

B.1.7 Membuat Key Pair

Step 7: Review Instance Launch

Please review your instance launch details. You can't edit these details until the launch process is complete.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair:

Select a key pair:

I acknowledge that I have access to the selected private key file (simaspurkeypair.pem), and that without this file, I won't be able to log into my instance.

Buttons: Cancel, Launch Instances

Buttons: Cancel, Previous, Launch



B.2 Konfigurasi Perangkat Lunak Pendukung

B.2.1 Pemasangan Apache2

B.2.1.1 Perintah Update

```

ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo apt update
Hit:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates In
Release [109 kB]
Get:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-backports
InRelease [107 kB]
Get:4 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial/universe a
md64 Packages [7,532 kB]
Get:5 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial/universe T
ranslation-en [4,354 kB]
Get:6 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Get:7 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial/multiverse
amd64 Packages [144 kB]
Get:8 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial/multiverse
Translation-en [106 kB]
Get:9 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/ma
in amd64 Packages [1,069 kB]
Get:10 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/m
ain Translation-en [413 kB]
Get:11 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/u
niverse amd64 Packages [769 kB]
Get:12 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/u
niverse Translation-en [323 kB]
Get:13 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/m
ultiverse amd64 Packages [16.8 kB]
Get:14 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/m
ultiverse Translation-en [8,468 B]
Get:15 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-backports
/main amd64 Packages [7,280 B]
Get:16 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-backports
/main Translation-en [4,456 B]

```

B.2.1.2 Perintah Add Repository Apache2

```

ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo add-apt-repository ppa:ondrej/apache2
This branch follows latest Apache2 packages as maintained by the Debian Ap
ache2 team with couple of compatibility patches on top.

It also includes some widely used Apache 2 modules (if you need some other
feel free to send me a request).

BUGS&FEATURES: This PPA now has a issue tracker: https://deb.sury.org/#bug-
reporting

PLEASE READ: If you like my work and want to give me a little motivation, p
lease consider donating: https://deb.sury.org/#donate
More info: https://launchpad.net/~ondrej/+archive/ubuntu/apache2
Press [ENTER] to continue or ctrl-c to cancel adding it

gpg: keyring `/tmp/tmp1v0pgd22/secring.gpg' created
gpg: keyring `/tmp/tmp1v0pgd22/pubring.gpg' created
gpg: requesting key E5267A6C from hkp server keyserver.ubuntu.com
gpg: /tmp/tmp1v0pgd22/trustdb.gpg: trustdb created
gpg: key E5267A6C: public key "Launchpad PPA for Ondřej Surý" imported
gpg: Total number processed: 1
gpg:      imported: 1 (RSA: 1)
OK
ubuntu@ip-172-31-20-88:~$ |

```

B.2.1.3 Perintah Pemasangan Apache2

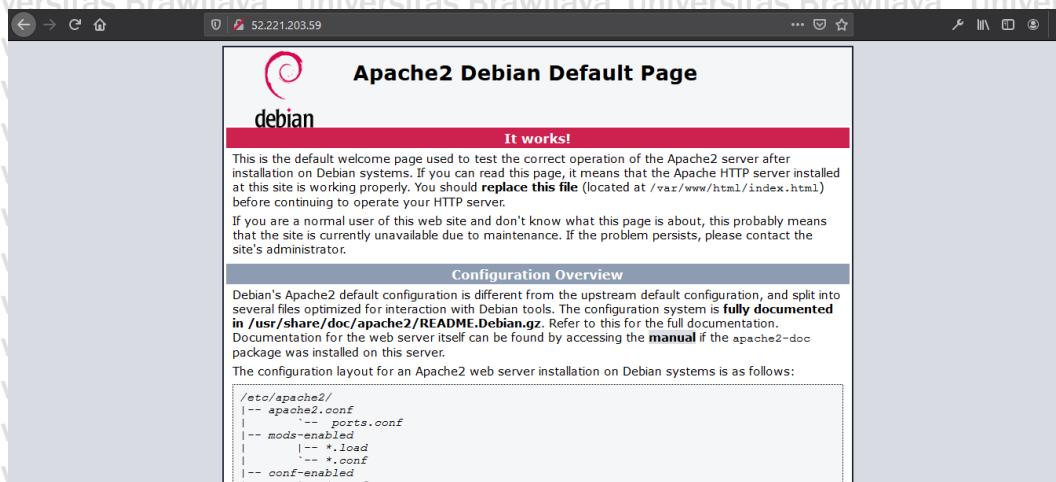
```

ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libbrotli1 libcurl3
  libjansson4 liblua5.2-0 libnghttp2-14 libssl1.1 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
  openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libbrotli1 libcurl3
  libjansson4 liblua5.2-0 libnghttp2-14 libssl1.1 ssl-cert
0 upgraded, 15 newly installed, 0 to remove and 51 not upgraded.
Need to get 4,130 kB of archives.
After this operation, 13.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libcurl3 amd64 7.47.0-1ubuntu2.14 [186 kB]
Get:2 http://ppa.launchpad.net/ondrej/apache2/ubuntu xenial/main amd64 libapr1 amd64 1.6.2-1+ubuntu16.04.1+deb.sury.org+2 [96.3 kB]
Get:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libjansson4 amd64 2.7-3ubuntu0.1 [27.1 kB]

```

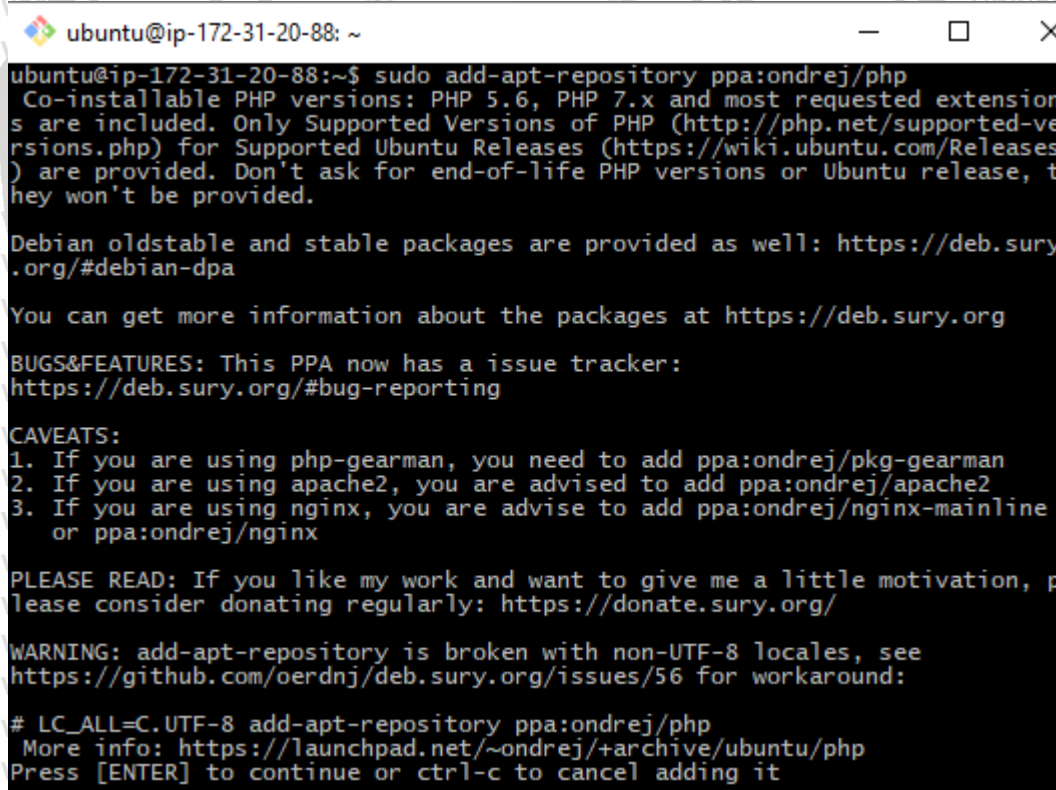


B.2.1.4 Percobaan Apache2



B.2.2 Pemasangan PHP7

B.2.2.1 Perintah Add Repository PHP



B.2.2.2 Perintah Pemasangan PHP7

```

ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo apt install php7.2 libapache2-mod-php7.2 php7.2-common php7.2-mbstring php7.2-xmlrpc php7.2-gd php7.2-xml php7.2-mysql php7.2-cli php7.2-zip php7.2-curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libargon2-0 libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8 libpcre3 libsodium23 libtiff5 libwebp6 libxmlrpc-epi0 libxpm4 libzip4 php-common php7.2-json php7.2-opcache php7.2-readline
Suggested packages:
  php-pear libgd-tools
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libapache2-mod-php7.2 libargon2-0 libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8 libsodium23 libtiff5 libwebp6 libxmlrpc-epi0 libxpm4 libzip4 php-common php7.2 php7.2-cli php7.2-common php7.2-curl php7.2-gd php7.2-json php7.2-mbstring php7.2-mysql php7.2-opcache php7.2-readline php7.2-xml php7.2-xmlrpc php7.2-zip
The following packages will be upgraded:
  libpcre3
1 upgraded, 29 newly installed, 0 to remove and 52 not upgraded.
Need to get 7,038 kB of archives.
After this operation, 26.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libjpeg-turbo8 amd64 1.4.2-0ubuntu3.3 [111 kB]
Get:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial/main amd64 fonts-dejavu-core all 2.35-1 [1,039 kB]
Get:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 fontconfig-config all 2.11.0-4ubuntu1.1 [49.9 kB]

```

B.2.3 Pemasangan MySQL

B.2.3.1 Perintah Pemasangan MySQL Server

```

ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.0-5 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client-5.7 mysql-client-core-5.7 mysql-common mysql-server-5.7 mysql-server-core-5.7
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinyc
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.0-5 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client-5.7 mysql-client-core-5.7 mysql-common mysql-server mysql-server-5.7 mysql-server-core-5.7
0 upgraded, 21 newly installed, 0 to remove and 52 not upgraded.
Need to get 18.3 MB of archives.
After this operation, 157 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 mysql-common all 5.7.28-0ubuntu0.16.04.2 [14.7 kB]
1% [Working]

```



B.2.3.2 Percobaan masuk kedalam Console MySQL

```

ubuntu@ip-172-31-20-88: ~
mysql> sudo mysql -p -u root
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.28-0ubuntu0.16.04.2 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
    
```

B.2.4 Pemasangan PhpMyAdmin

B.2.4.1 Perintah Pemasangan PhpMyAdmin

```

ubuntu@ip-172-31-20-88: ~
sudo apt install phpmyadmin php-mbstring php-gettext
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  dbconfig-common dbconfig-mysql javascript-common libjs-jquery
  libjs-sphinxdoc libjs-underscore libmcrypt4 php-mcrypt php-pear
  php-phpseclib php-tcpdf php7.0-common php7.0-mcrypt php7.3-common
  php7.3-mbstring
Suggested packages:
  libmcrypt-dev mcrypt php-libsodium php-gmp php-imagick www-browser
The following NEW packages will be installed:
  dbconfig-common dbconfig-mysql javascript-common libjs-jquery
  libjs-sphinxdoc libjs-underscore libmcrypt4 php-gettext php-mbstring
  php-mcrypt php-pear php-phpseclib php-tcpdf php7.0-common php7.0-mcrypt
  php7.3-common php7.3-mbstring phpmyadmin
0 upgraded, 18 newly installed, 0 to remove and 52 not upgraded.
Need to get 15.6 MB of archives.
After this operation, 64.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ppa.launchpad.net/ondrej/php/ubuntu xenial/main amd64 php-pear
all 1:1.10.8+submodules+notgz-1+ubuntu16.04.1+deb.sury.org+1 [287 kB]
ubuntu@ip-172-31-20-88: ~
sudo phpenmod mbstring
ubuntu@ip-172-31-20-88: ~
sudo systemctl restart apache2
ubuntu@ip-172-31-20-88: ~
    
```



B.2.4.2 Konfigurasi Keamanan Instance PhpMyAdmin

```

ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo nano /etc/apache2/conf-available/phpmyadmin.conf
ubuntu@ip-172-31-20-88:~$ |

ubuntu@ip-172-31-20-88: ~
GNU nano 2.5.3 File: /usr/share/phpmyadmin/.htaccess Modified
AuthType Basic
AuthName "Restricted Files"
AuthUserFile /etc/phpmyadmin/.htpasswd
Require valid-user

ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo nano /usr/share/phpmyadmin/.htaccess
ubuntu@ip-172-31-20-88:~$

ubuntu@ip-172-31-20-88: ~
GNU nano 2.5.3 File: ...apache2/conf-available/phpmyadmin.conf Modified
# phpMyAdmin default Apache configuration
Alias /phpmyadmin /usr/share/phpmyadmin

<Directory /usr/share/phpmyadmin>
Options FollowSymLinks
DirectoryIndex index.php
AllowOverride All

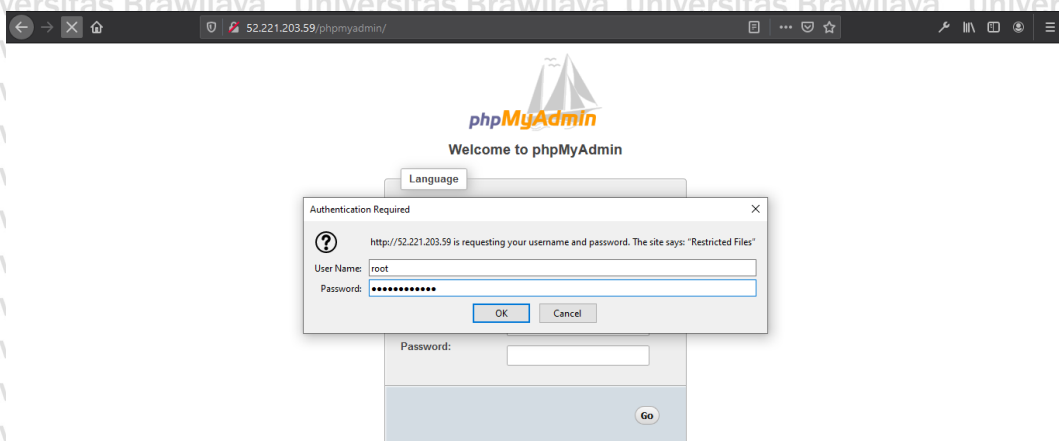
<IfModule mod_php.c>
<IfModule mod_mime.c>
AddType application/x-httpd-php .php
</IfModule>
<FilesMatch ".+\.php$">
SetHandler application/x-httpd-php
</FilesMatch>

php_flag magic_quotes_gpc Off
php_flag track_vars On
php_flag register_globals Off
php_admin_flag allow_url_fopen On
php_value include_path .
php_admin_value upload_tmp_dir /var/lib/phpmyadmin/tmp
php_admin_value open_basedir /usr/share/phpmyadmin:/etc/phpmyadmin
</IfModule>
</Directory>

ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo htpasswd -c /etc/phpmyadmin/.htpasswd root
New password:
Re-type new password:
Adding password for user root
ubuntu@ip-172-31-20-88:~$ |
    
```



B.2.4.3 Percobaan Akses MySQL melalui PhpMyAdmin



B.3 Konfigurasi simaspur.com

B.3.1 Pembuatan Direktori

```

ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo mkdir -p /var/www/simaspur.com
ubuntu@ip-172-31-20-88:~$ sudo chown www-data:www-data -R /var/www/simaspur.com
ubuntu@ip-172-31-20-88:~$ sudo find /var/www/simaspur.com/ -type d -exec chown www-data:www-data {} \;
ubuntu@ip-172-31-20-88:~$ sudo find /var/www/simaspur.com/ -type f -exec chown www-data:www-data {} \;
ubuntu@ip-172-31-20-88:~$ |
    
```

B.3.2 Konfigurasi simaspur.conf

```

ubuntu@ip-172-31-20-88: ~
ubuntu@ip-172-31-20-88:~$ sudo vim /etc/apache2/sites-available/simaspur.conf
ubuntu@ip-172-31-20-88:~$ |
    
```

```

root@ip-172-31-20-88: /home/ubuntu
<VirtualHost *:80>
  ServerAdmin bella.hersylavenia2905@gmail.com
  DocumentRoot /var/www/simaspur.com/public_html
  ServerName simaspur.com
  ServerAlias www.simaspur.com

  <Directory /var/www/simaspur.com/>
    Options +FollowSymLinks
    AllowOverride All
    Require all granted
  </Directory>

  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

"/etc/apache2/sites-available/simaspur.conf" 15L, 445C 1,1 All
    
```



```
ubuntu@ip-172-31-20-88: ~  
ubuntu@ip-172-31-20-88:~$ sudo a2ensite simaspur.conf  
Enabling site simaspur.  
To activate the new configuration, you need to run:  
systemctl reload apache2  
ubuntu@ip-172-31-20-88:~$ sudo a2dissite 000-default.conf  
Site 000-default disabled.  
To activate the new configuration, you need to run:  
systemctl reload apache2  
ubuntu@ip-172-31-20-88:~$ sudo apache2ctl configtest  
Syntax OK  
ubuntu@ip-172-31-20-88:~$ sudo service apache2 restart  
ubuntu@ip-172-31-20-88:~$ |
```

B.3.3 Konfigurasi User pada Group www-data

```
ubuntu@ip-172-31-20-88: /var/www  
ubuntu@ip-172-31-20-88:/var/www$ sudo usermod -a -G www-data ubuntu  
ubuntu@ip-172-31-20-88:/var/www$
```



LAMPIRAN C HASIL PENGUJIAN VALIDASI

C.1 Hasil Pengujian Validasi pada Administrator Keuangan

Kuesioner Pengujian Validasi kepada Administrator Keuangan

Pengguna : Administrator Keuangan
 Nama : Melly Andasari
 Umur : 25

Petunjuk Pengisian

1. Penguji adalah administrator keuangan Bengkel Mas Pur Baturaja
2. Penguji menjalankan sistem dengan langkah-langkah seperti pada prosedur uji.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	SIPOS-PF-01.
Nama Kasus Uji	Kasus uji melakukan login.
Kode Kebutuhan Fungsional	SIPOS-KF-01.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melakukan <i>login</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengakses sistem. 2. Penguji memasukan "melly" pada formulir <i>username</i>. 3. Penguji memasukan "passwordku" pada formulir <i>password</i>. 4. Penguji menekan tombol <i>Login</i>.
Hasil yang diharapkan	Sistem menampilkan halaman penjualan.
Keluaran	<ol style="list-style-type: none"> 1. Keluaran sesuai dengan yang diharapkan. 2. Keluaran tidak sesuai dengan yang diharapkan.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-02.
Nama Kasus Uji	Kasus uji memasukan data penjualan.
Kode Kebutuhan Fungsional	SIPOS-KF-02.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji telah melakukan <i>login</i>. 2. Penguji memilih menu penjualan. 3. Penguji memilih "BG B18 – Bella Hersy Lavenia" pada formulir kendaraan. 4. Penguji memilih "14-11-2019" pada formulir tanggal. 5. Penguji memilih "Suku Cadang" pada formulir kategori. 6. Penguji memilih "53301-0k111 – Kap Mesin" pada formulir item. 7. Penguji memasukan "2" pada formulir jumlah. 8. Penguji menekan tombol tambah. 9. Penguji memasukan "900000" pada formulir uang yang diterima. 10. Penguji menekan tombol simpan. 11. Penguji menekan tombol konfirmasi. 12. Penguji menekan tombol cetak. 13. Penguji menekan tombol faktur.
Hasil yang diharapkan	Sistem menyimpan data penjualan dan mencetak faktur pembayaran.
Keluaran	Sistem menyimpan data penjualan dan mencetak faktur pembayaran.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-03.
Nama Kasus Uji	Kasus uji memasukan data pembelian.
Kode Kebutuhan Fungsional	SIPOS-KF-03.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk memasukan data pembelian.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji telah melakukan <i>login</i>. 2. Penguji memilih menu pembelian. 3. Penguji memilih "Permai Jaya" pada formulir nama pemasok. 4. Penguji memilih "14-11-2019" pada formulir tanggal. 5. Penguji memilih "53301-0k111 - Kap Mesin" pada formulir item. 6. Penguji memasukan "850000" pada formulir harga beli. 7. Penguji memasukan "900000" pada formulir harga jual. 8. Penguji memasukan "2" pada formulir jumlah. 9. Penguji menekan tombol tambah. 10. Penguji menekan tombol simpan. 11. Penguji menekan tombol konfirmasi.
Hasil yang diharapkan	Sistem menyimpan data pembelian.
Keluaran	Sistem menyimpan data pembelian.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-04.
Nama Kasus Uji	Kasus uji memasukan data gaji karyawan.
Kode Kebutuhan Fungsional	SIPOS-KF-04.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk memasukan data gaji karyawan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji telah melakukan <i>login</i>. 2. Penguji memilih menu gaji karyawan. 3. Penguji memilih "Bella Hersy Lavenia" pada formulir karyawan. 4. Penguji memasukan "28" pada formulir absen. 5. Penguji menekan tombol tambah.
Hasil yang diharapkan	Sistem menyimpan data gaji karyawan dan menampilkannya.
Keluaran	Sistem menyimpan data gaji karyawan dan menampilkannya.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-05.
Nama Kasus Uji	Kasus uji memasukan data gaji karyawan.
Kode Kebutuhan Fungsional	SIPOS-KF-04.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk memasukan data gaji karyawan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji telah melakukan <i>login</i>. 2. Penguji memilih menu gaji karyawan. 3. Penguji memilih "Bella Hersy Lavenia" pada formulir karyawan. 4. Penguji memasukan "28" pada formulir absen. 5. Penguji menekan tombol tambah.
Hasil yang diharapkan	Sistem tidak menyimpan data gaji karyawan dan menampilkannya.
Keluaran	<ol style="list-style-type: none"> 1. Keluaran sesuai dengan yang diharapkan. 2. Keluaran tidak sesuai dengan yang diharapkan.
Status Uji	Invalid



Kode Pengujian	SIPOS-PF-06.
Nama Kasus Uji	Kasus uji melakukan login.
Kode Kebutuhan Fungsional	SIPOS-KF-01.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melakukan <i>login</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengakses sistem. 2. Penguji memasukan "melly" pada formulir <i>username</i>. 3. Penguji memasukan "passwordsalah" pada formulir <i>password</i>. 4. Penguji menekan tombol <i>Login</i>.
Hasil yang diharapkan	Sistem menampilkan pemberitahuan <i>password salah</i> .
Keluaran	<ol style="list-style-type: none"> 3. Keluaran sesuai dengan yang diharapkan. 4. Keluaran tidak sesuai dengan yang diharapkan.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-07.
Nama Kasus Uji	Kasus uji melihat stok suku cadang.
Kode Kebutuhan Fungsional	SIPOS-KF-05.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melihat stok suku cadang.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji telah melakukan <i>login</i>. 2. Penguji memilih menu suku cadang. 3. Penguji memasukan "kap mesin" pada formulir cari.
Hasil yang diharapkan	Sistem menampilkan data stok suku cadang.
Keluaran	<ol style="list-style-type: none"> 1. Keluaran sesuai dengan yang diharapkan. 2. Keluaran tidak sesuai dengan yang diharapkan.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-08.
Nama Kasus Uji	Kasus uji melihat stok suku cadang.
Kode Kebutuhan Fungsional	SIPOS-KF-05.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melihat stok suku cadang.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji telah melakukan <i>login</i>. 2. Penguji memilih menu suku cadang. 3. Penguji memasukan "roti" pada formulir cari.
Hasil yang diharapkan	Sistem tidak menampilkan data stok suku cadang.
Keluaran	<ol style="list-style-type: none"> 1. Keluaran sesuai dengan yang diharapkan. 2. Keluaran tidak sesuai dengan yang diharapkan.
Status Uji	Valid



Kode Pengujian	SIPOS-PF-09.
Nama Kasus Uji	Kasus uji melihat laporan pemasukan.
Kode Kebutuhan Fungsional	SIPOS-KF-06.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melihat laporan pemasukan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji telah melakukan <i>login</i>. 2. Penguji memilih menu penjualan. 3. Penguji memilih "Oktober" pada formulir bulan. 4. Penguji memilih "2019" pada formulir tahun. 5. Penguji menekan tombol Cari.
Hasil yang diharapkan	Sistem menampilkan data penjualan.
Keluaran	Sistem menampilkan data penjualan.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-10.
Nama Kasus Uji	Kasus uji melihat laporan pemasukan.
Kode Kebutuhan Fungsional	SIPOS-KF-06.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melihat laporan pemasukan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji telah melakukan <i>login</i>. 2. Penguji memilih menu penjualan. 3. Penguji memilih "Januari" pada formulir bulan. 4. Penguji memilih "2019" pada formulir tahun. 5. Penguji menekan tombol Cari.
Hasil yang diharapkan	Sistem tidak menampilkan data penjualan.
Keluaran	<ol style="list-style-type: none"> 1. Keluaran sesuai dengan yang diharapkan. 2. Keluaran tidak sesuai dengan yang diharapkan.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-11.
Nama Kasus Uji	Kasus uji melihat laporan pengeluaran.
Kode Kebutuhan Fungsional	SIPOS-KF-07.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melihat laporan pengeluaran.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji telah melakukan <i>login</i>. 2. Penguji memilih menu pengeluaran. 3. Penguji memilih "Oktober" pada formulir bulan. 4. Penguji memilih "2019" pada formulir tahun. 5. Penguji menekan tombol Cari.
Hasil yang diharapkan	Sistem menampilkan data pembelian dan data gaji karyawan.
Keluaran	Sistem menampilkan data pembelian dan data gaji karyawan.
Status Uji	Valid



Kode Pengujian	SIPOS-PF-12.
Nama Kasus Uji	Kasus uji melihat laporan pengeluaran.
Kode Kebutuhan Fungsional	SIPOS-KF-07.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melihat laporan pengeluaran.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji telah melakukan <i>login</i>. 2. Penguji memilih menu pengeluaran. 3. Penguji memilih "Januari" pada formulir bulan. 4. Penguji memilih "2019" pada formulir tahun. 5. Penguji menekan tombol Cari.
Hasil yang diharapkan	Sistem tidak menampilkan data pembelian dan data gaji karyawan.
Keluaran	<ol style="list-style-type: none"> 1. Keluaran sesuai dengan yang diharapkan. 2. Keluaran tidak sesuai dengan yang diharapkan.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-13.
Nama Kasus Uji	Kasus uji melakukan <i>logout</i> .
Kode Kebutuhan Fungsional	SIPOS-KF-08.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melakukan <i>logout</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji telah melakukan <i>login</i>. 2. Penguji memilih menu <i>logout</i>.
Hasil yang diharapkan	Sistem menampilkan halaman <i>login</i> .
Keluaran	Sistem menampilkan halaman <i>login</i> .
Status Uji	Valid

Baturaja, 11 November 2019



Meliy Andasari
Administrasi



C.2 Hasil Pengujian Validasi pada Manajer

Kuesioner Pengujian Validasi kepada Manajer (Pemilik Bengkel)

Pengguna : Manajer
 Nama : Purwanto
 Umur : 53
 Petunjuk Pengisian

1. Penguji adalah manajer atau pemilik Bengkel Mas Pur Baturaja
2. Penguji menjalankan sistem dengan langkah-langkah seperti pada prosedur uji.
3. Untuk jawaban hasil pengujian diisi dengan "valid" apabila keluaran sesuai dengan harapan dan diisi dengan "tidak valid" apabila keluaran tidak sesuai harapan.

Kode Pengujian	SIPOS-PF-14.
Nama Kasus Uji	Kasus uji melakukan login.
Kode Kebutuhan Fungsional	SIPOS-KF-01.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melakukan <i>login</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengakses sistem. 2. Penguji memasukkan "purwanto" pada formulir <i>username</i>. 3. Penguji memasukkan "bismillah" pada formulir <i>password</i>. 4. Penguji menekan tombol <i>Login</i>.
Hasil yang diharapkan	Sistem menampilkan halaman <i>dashboard</i> .
Keluaran	<ol style="list-style-type: none"> 1. Keluaran sesuai dengan yang diharapkan. 2. Keluaran tidak sesuai dengan yang diharapkan.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-15.
Nama Kasus Uji	Kasus uji melakukan login.
Kode Kebutuhan Fungsional	SIPOS-KF-01.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melakukan <i>login</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengakses sistem. 2. Penguji memasukkan "manajer" pada formulir <i>username</i>. 3. Penguji memasukkan "bismillah" pada formulir <i>password</i>. 4. Penguji menekan tombol <i>Login</i>.
Hasil yang diharapkan	Sistem menampilkan pemberitahuan <i>username</i> tidak ditemukan.
Keluaran	<ol style="list-style-type: none"> 5. Keluaran sesuai dengan yang diharapkan. 6. Keluaran tidak sesuai dengan yang diharapkan.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-07.
Nama Kasus Uji	Kasus uji melihat stok suku cadang.
Kode Kebutuhan Fungsional	SIPOS-KF-05.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melihat stok suku cadang.
Prosedur Uji	4. Penguji telah melakukan <i>login</i> . 5. Penguji memilih menu suku cadang. 6. Penguji memasukkan "kap mesin" pada formulir cari.
Hasil yang diharapkan	Sistem menampilkan data stok suku cadang.
Keluaran	3. Keluaran sesuai dengan yang diharapkan. 4. Keluaran tidak sesuai dengan yang diharapkan.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-08.
Nama Kasus Uji	Kasus uji melihat stok suku cadang.
Kode Kebutuhan Fungsional	SIPOS-KF-05.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melihat stok suku cadang.
Prosedur Uji	4. Penguji telah melakukan <i>login</i> . 5. Penguji memilih menu suku cadang. 6. Penguji memasukkan "roti" pada formulir cari.
Hasil yang diharapkan	Sistem tidak menampilkan data stok suku cadang.
Keluaran	3. Keluaran sesuai dengan yang diharapkan. 4. Keluaran tidak sesuai dengan yang diharapkan.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-09.
Nama Kasus Uji	Kasus uji melihat laporan pemasukan.
Kode Kebutuhan Fungsional	SIPOS-KF-06.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melihat laporan pemasukan.
Prosedur Uji	6. Penguji telah melakukan <i>login</i> . 7. Penguji memilih menu penjualan. 8. Penguji memilih "Oktober" pada formulir bulan. 9. Penguji memilih "2019" pada formulir tahun. 10. Penguji menekan tombol Cari.
Hasil yang diharapkan	Sistem menampilkan data penjualan.
Keluaran	Sistem menampilkan data penjualan.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-10.
Nama Kasus Uji	Kasus uji melihat laporan pemasukan.
Kode Kebutuhan Fungsional	SIPOS-KF-06.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melihat laporan pemasukan.
Prosedur Uji	6. Penguji telah melakukan login. 7. Penguji memilih menu penjualan. 8. Penguji memilih "Januari" pada formulir bulan. 9. Penguji memilih "2019" pada formulir tahun. 10. Penguji menekan tombol Cari.
Hasil yang diharapkan	Sistem tidak menampilkan data penjualan.
Keluaran	3. Keluaran sesuai dengan yang diharapkan. 4. Keluaran tidak sesuai dengan yang diharapkan.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-11.
Nama Kasus Uji	Kasus uji melihat laporan pengeluaran.
Kode Kebutuhan Fungsional	SIPOS-KF-07.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melihat laporan pengeluaran.
Prosedur Uji	6. Penguji telah melakukan login. 7. Penguji memilih menu pengeluaran. 8. Penguji memilih "Oktober" pada formulir bulan. 9. Penguji memilih "2019" pada formulir tahun. 10. Penguji menekan tombol Cari.
Hasil yang diharapkan	Sistem menampilkan data pembelian dan data gaji karyawan.
Keluaran	Sistem menampilkan data pembelian dan data gaji karyawan.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-12.
Nama Kasus Uji	Kasus uji melihat laporan pengeluaran.
Kode Kebutuhan Fungsional	SIPOS-KF-07.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melihat laporan pengeluaran.
Prosedur Uji	6. Penguji telah melakukan login. 7. Penguji memilih menu pengeluaran. 8. Penguji memilih "Januari" pada formulir bulan. 9. Penguji memilih "2019" pada formulir tahun. 10. Penguji menekan tombol Cari.
Hasil yang diharapkan	Sistem tidak menampilkan data pembelian dan data gaji karyawan.
Keluaran	3. Keluaran sesuai dengan yang diharapkan. 4. Keluaran tidak sesuai dengan yang diharapkan.
Status Uji	Valid

Kode Pengujian	SIPOS-PF-13.
Nama Kasus Uji	Kasus uji melakukan <i>logout</i> .
Kode Kebutuhan Fungsional	SIPOS-KF-08.
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat digunakan untuk melakukan <i>logout</i> .
Prosedur Uji	3. Penguji telah melakukan <i>login</i> . 4. Penguji memilih menu <i>logout</i> .
Hasil yang diharapkan	Sistem menampilkan halaman <i>login</i> .
Keluaran	Sistem menampilkan halaman <i>login</i> .
Status Uji	Valid

Baturaja, 11 November 2019


PURWANTO
AUTO BODY REPAIR
Purwanto
Pemilik Bengkel

