

**PEMBANGUNAN SISTEM PEMELIHARAAN SARANA DAN
PEMANTAUAN PEMASARAN PRODUK BERBASIS WEB
MENGUNAKAN KERANGKA KERJA LARAVEL**

Studi Kasus : PT. Pancamas Elite

SKRIPSI

Diajukan untuk memenuhi salah satu persyaratan untuk mencapai
gelar Sarjana Komputer

Disusun oleh:

Kanthen Raraswati

NIM: 115060801111062

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA
PROGRAM STUDI INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

PEMBANGUNAN SISTEM PEMELIHARAAN SARANA DAN PEMANTAUAN PEMASARAN PRODUK BERBASIS WEB MENGUNAKAN KERANGKA KERJA LARAVEL

SKRIPSI

Diajukan untuk memenuhi salah satu persyaratan untuk mencapai gelar Sarjana
Komputer

Disusun Oleh :

Kanthi Raraswati

NIM: 115060801111062

Skrripsi ini telah diuji dan dinyatakan lulus pada
18 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Fajar Pradana S.ST., M.Eng.
NIP: 19871121 201504 1004

Agi Putra Kharisma, ST, MT.
NIK: 201304 860430 1001

Mengetahui
Ketua Program Studi Informatika

Drs. Marji, MT.

NIP: 19670801 199203 1 001

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 pustaka.

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, 18 Januari 2016



Kanthi Raraswati

NIM: 115060801111062

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, yang atas rahmat-Nya maka penulis dapat menyelesaikan penyusunan tugas akhir yang berjudul “Rancang Bangun Sistem Pemeliharaan Sarana dan Pemantauan Produk Berbasis Web dengan Kerangka kerja Laravel”. Shalawat serta salam atas junjungan besar Nabi kita Muhammad S.A.W. beserta keluarga dan para sahabat sekalian.

Tugas Akhir ini disusun sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Komputer di Program Studi Informatika/Illmu Komputer, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.

Dalam penulisan skripsi ini penulis merasa masih banyak kekurangan baik pada teknis penulisan maupun materi. Mengingat akan kemampuan yang dimiliki penulis, untuk itu penulis menyampaikan ucapan terimakasih dan penghargaan kepada :

1. Orang tua penulis dan seluruh keluarga besar tercinta yang senantiasa tiada hentinya memberikan doa, dorongan semangat, dan segala pengorbanan baik materil maupun moril demi terselesainya skripsi ini.
2. Bapak Fajar Pradana S.ST., M.Eng selaku dosen pembimbing I dan Bapak Agi Putra Kharisma, ST, MT selaku dosen pembimbing II yang telah sabar memberikan bimbingan, arahan, ilmu dan saran untuk laporan skripsi ini.
3. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Bapak Drs. Marji, MT dan Bapak Issa Arwani, S.Kom., M.Sc selaku Ketua dan Sekretaris Program Studi Informatika serta segenap Bapak/Ibu Dosen, Staff Administrasi dan Perpustakaan Program Studi Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
5. Seluruh dosen Program Studi Informatika atas kesediaan membagi ilmunya kepada penulis.
6. Ibu Indah, Bapak Deden, Mbak Krisna dan Mbak Rury selaku coordinator dan karyawan PT.Pancamas Elite Malang yang memberikan kelancaran dalam skripsi ini.
7. Chasandra Puspitasari, Rizki Sepasthika, Aghnia Farda, Anisah, Rizki Anggarsasi, Ken Savilla Leonita selaku sahabat yang tidak pernah lelah memberikan motivasi dan saran yang selalu membangun dalam penulisan skripsi ini serta teman-teman seperjuangan Informatika/Illmu Komputer angkatan 2011 yang tidak bisa disebutkan namanya satu persatu atas dukungannya.
8. Seluruh Keluarga Besar Program Teknologi Informasi dan Ilmu Komputer yang telah memberikan tempat untuk kuliah dan mencari pengalaman di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya
9. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesainya skripsi ini.

Semoga jasa dan amal baik mendapatkan balasan dari Allah SWT. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna dan tidak terlepas dari kekurangan dan kesalahan karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Oleh karena itu, segala kritik dan saran yang bersifat membangun penulis harapkan untuk penyempurnaan skripsi ini. Penulis juga berharap mudah-mudahan skripsi ini dapat bermanfaat bagi semua pihak yang memerlukan.

Malang, 18 Januari 2016

Penulis

kantyraras@gmail.com



ABSTRAK

Sistem Pemeliharaan Sarana dan Pemantauan Pemasaran Produk Berbasis Web menggunakan Kerangka Kerja Laravel merupakan sistem pemeliharaan sarana prasarana dan pemantauan produk hasil produksi yang ada di dalam perusahaan yang bertujuan untuk mengelola dan mengatur proses bisnis pemeliharaan sarana prasarana dan pemantauan produk hasil produksi dengan menerapkan teknologi kerangka kerja terbaru. Peran utama kerangka kerja *Laravel* adalah sebagai alat bantu dalam menyelesaikan implementasi sistem pemeliharaan sarana dan pemantauan produk. Kerangka kerja laravel memiliki fitur ORM (*Object Relationship Mapping*) khusus untuk menangani Model, Manajemen *Routing* dan *controller* serta *Blade Templating* yang powerful dan ekspresif dari segi kode pemrograman. Di sinilah kerangka kerja *Laravel* dapat membantu proses pengembangan sistem pemeliharaan sarana dan pemantauan produk menjadi lebih cepat.

Pada penelitian ini akan dilakukan perancangan dan implementasi Sistem Pemeliharaan Sarana dan Pemantauan Produk dengan menggunakan kerangka kerja *Laravel* yang bertujuan untuk membantu menyelesaikan kebutuhan pengguna. Penelitian akan dilakukan di PT.Pancamas Elite di dalam divisi pemeliharaan yang mengelola sarana prasarana dan divisi pemasaran yang mengelola keluar masuk produk hasil produksi. Hasil Pengujian menunjukkan bahwa sistem yang telah diimplementasikan telah sesuai dengan kebutuhan fungsional yang ada. Dari 33 kebutuhan fungsional yang diuji menunjukkan bahwa 100% kebutuhan fungsional dari sistem telah valid ketika diuji dengan kasus tertentu.

Kata Kunci : *Sistem Pemeliharaan, Sistem Pemantauan, Kerangka Kerja Laravel*

ABSTRACT

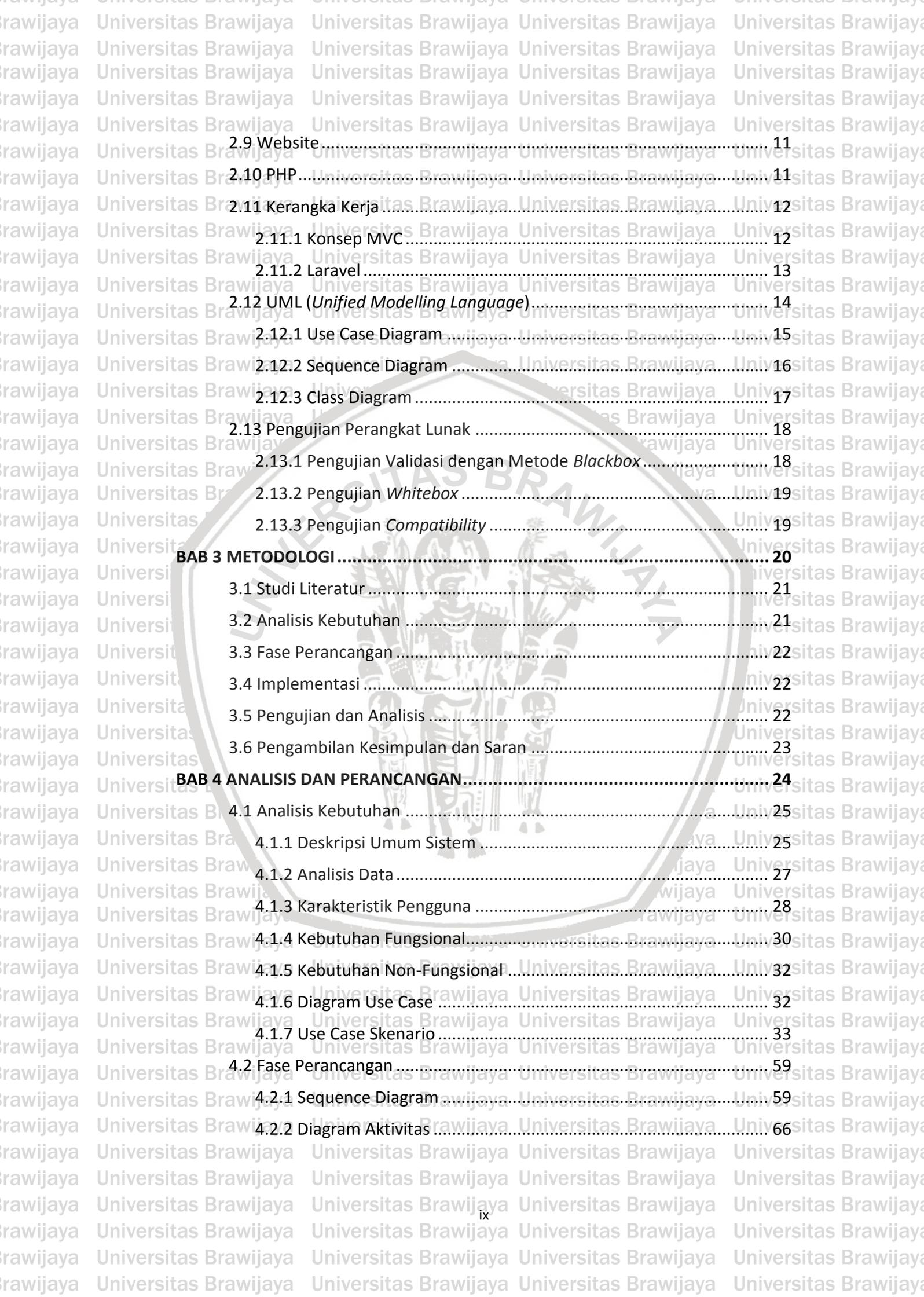
System Maintenance Facility and Monitoring Product Marketing Web Based using the Framework Laravel is a system of maintenance of infrastructure and monitoring products production results in the firms that aim to organize and manage the business processes of maintenance of infrastructure and monitoring products produks by applying the technology framework for the latest , The main role Laravel framework is as a tool in completing the implementation of the system of maintenance of facilities and monitoring products. Laravel framework features the ORM (Object Relationship Mapping) model for handling, management controller and Blade Routing and Templating powerful and expressive in terms of programming code. This is where Laravel framework can assist in the development of maintenance of facilities and monitoring systems become faster products.

In this research will be the design and implementation of the Facilities and Maintenance System Monitoring Products using Laravel framework that aims to help solve the needs of users. Research will be conducted in PT.Pancamas Elite in the maintenance division that manages the infrastructure and marketing division that manages out of the product production. Testing results indicate that the system has been implemented in accordance with existing functional needs. Of the 33 functional requirements were tested showed that 100% functional requirements of the system have been valid when tested with a particular case.

Keywords: *Maintenance System, Monitoring System, Laravel Framework*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
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 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka.....	5
2.2 Dasar Teori.....	5
2.2.1 PT. Pancamas Elite.....	5
2.2.2 Visi dan Misi PT.Pancamas Elite.....	6
2.3 Struktur Organisasi.....	6
2.4 Pemeliharaan.....	7
2.4.1 Jenis-Jenis Pemeliharaan.....	7
2.5 Pemantauan.....	8
2.6 Sistem Informasi.....	8
2.7 Rekayasa Perangkat Lunak (<i>Software Engineering</i>).....	9
2.7.1 <i>User Requirements</i>	9
2.7.2 <i>System Requirements</i>	9
2.8 <i>System Development Life Cycle (SDLC)</i>	9
2.8.1 <i>Waterfall Model</i>	10



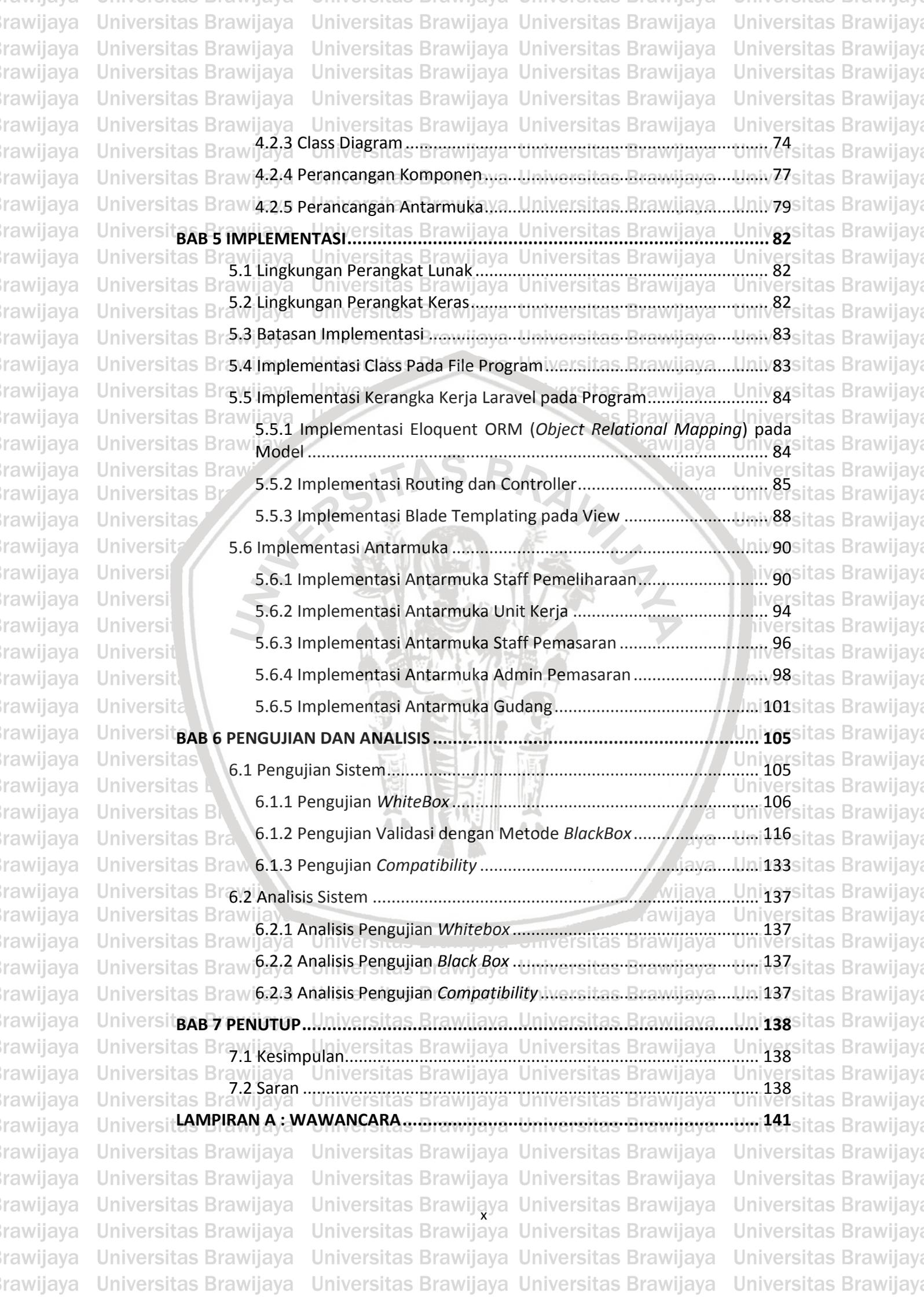
2.9 Website.....	11
2.10 PHP.....	11
2.11 Kerangka Kerja.....	12
2.11.1 Konsep MVC.....	12
2.11.2 Laravel.....	13
2.12 UML (<i>Unified Modelling Language</i>).....	14
2.12.1 Use Case Diagram.....	15
2.12.2 Sequence Diagram.....	16
2.12.3 Class Diagram.....	17
2.13 Pengujian Perangkat Lunak.....	18
2.13.1 Pengujian Validasi dengan Metode <i>Blackbox</i>	18
2.13.2 Pengujian <i>Whitebox</i>	19
2.13.3 Pengujian <i>Compatibility</i>	19

BAB 3 METODOLOGI..... 20

3.1 Studi Literatur.....	21
3.2 Analisis Kebutuhan.....	21
3.3 Fase Perancangan.....	22
3.4 Implementasi.....	22
3.5 Pengujian dan Analisis.....	22
3.6 Pengambilan Kesimpulan dan Saran.....	23

BAB 4 ANALISIS DAN PERANCANGAN..... 24

4.1 Analisis Kebutuhan.....	25
4.1.1 Deskripsi Umum Sistem.....	25
4.1.2 Analisis Data.....	27
4.1.3 Karakteristik Pengguna.....	28
4.1.4 Kebutuhan Fungsional.....	30
4.1.5 Kebutuhan Non-Fungsional.....	32
4.1.6 Diagram Use Case.....	32
4.1.7 Use Case Skenario.....	33
4.2 Fase Perancangan.....	59
4.2.1 Sequence Diagram.....	59
4.2.2 Diagram Aktivitas.....	66



4.2.3 Class Diagram.....	74
4.2.4 Perancangan Komponen.....	77
4.2.5 Perancangan Antarmuka.....	79
BAB 5 IMPLEMENTASI.....	82
5.1 Lingkungan Perangkat Lunak.....	82
5.2 Lingkungan Perangkat Keras.....	82
5.3 Batasan Implementasi.....	83
5.4 Implementasi Class Pada File Program.....	83
5.5 Implementasi Kerangka Kerja Laravel pada Program.....	84
5.5.1 Implementasi Eloquent ORM (<i>Object Relational Mapping</i>) pada Model.....	84
5.5.2 Implementasi Routing dan Controller.....	85
5.5.3 Implementasi Blade Templating pada View.....	88
5.6 Implementasi Antarmuka.....	90
5.6.1 Implementasi Antarmuka Staff Pemeliharaan.....	90
5.6.2 Implementasi Antarmuka Unit Kerja.....	94
5.6.3 Implementasi Antarmuka Staff Pemasaran.....	96
5.6.4 Implementasi Antarmuka Admin Pemasaran.....	98
5.6.5 Implementasi Antarmuka Gudang.....	101
BAB 6 PENGUJIAN DAN ANALISIS.....	105
6.1 Pengujian Sistem.....	105
6.1.1 Pengujian <i>WhiteBox</i>	106
6.1.2 Pengujian Validasi dengan Metode <i>BlackBox</i>	116
6.1.3 Pengujian <i>Compatibility</i>	133
6.2 Analisis Sistem.....	137
6.2.1 Analisis Pengujian <i>Whitebox</i>	137
6.2.2 Analisis Pengujian <i>Black Box</i>	137
6.2.3 Analisis Pengujian <i>Compatibility</i>	137
BAB 7 PENUTUP.....	138
7.1 Kesimpulan.....	138
7.2 Saran.....	138
LAMPIRAN A : WAWANCARA.....	141

DAFTAR TABEL

Tabel 2.1 Notasi Use Case Diagram	15
Tabel 2.2 Notasi Sequence Diagram	16
Tabel 4.1 Karakteristik Pengguna	29
Tabel 4.2 Spesifikasi Kebutuhan Fungsional (KF)	30
Tabel 4.3 Kebutuhan non fungsional	32
Tabel 4.4 Skenario Login	34
Tabel 4.5 Skenario Mengubah <i>profile</i> pengguna	34
Tabel 4.6 Skenario Melaporkan kerusakan	35
Tabel 4.7 Skenario Pemberitahuan Kerusakan	36
Tabel 4.8 Skenario Melihat Riwayat Pengajuan Kerusakan	36
Tabel 4.9 Skenario Melihat Daftar Sarana	38
Tabel 4.10 Skenario Menambah Sarana dan Ruang	38
Tabel 4.11 Skenario Mengubah Informasi Sarana	39
Tabel 4.12 Skenario Menghapus Sarana	40
Tabel 4.13 Skenario Melihat Ruangan	41
Tabel 4.14 Skenario Mengubah Informasi Ruangan	42
Tabel 4.15 Skenario Menghapus Ruangan	42
Tabel 4.16 Skenario Melihat Status Terakhir Sarana	43
Tabel 4.17 Skenario Konfirmasi Perbaikan	44
Tabel 4.18 Skenario Menerima status perbaikan	45
Tabel 4.19 Skenario Melihat Laporan Perbaikan dan Perawatan	45
Tabel 4.20 Skenario Melihat Pemeliharaan Rutin	46
Tabel 4.21 Skenario Konfirmasi Perawatan Rutin	47
Tabel 4.22 Skenario Pemberitahuan Jadwal Pemeliharaan	48
Tabel 4.23 Skenario Menampilkan daftar produk	49
Tabel 4.24 Skenario Menambah produk	49
Tabel 4.25 Skenario Mengubah Informasi Produk	50
Tabel 4.26 Skenario Menghapus Produk	51
Tabel 4.27 Skenario Menambah Pemesanan Produk	51
Tabel 4.28 Skenario Validasi Pemesanan Produk	52
Tabel 4.29 Skenario Melihat Pesanan Sementara	53

Tabel 4.30 Skenario Melihat riwayat Pemesanan.....	54
Tabel 4.31 Skenario Pemberitahuan status pengajuan pemesanan	55
Tabel 4.32 Skenario Pemberitahuan Pemesanan baru.....	55
Tabel 4.33 Skenario Melihat Status stok produk	56
Tabel 4.34 Skenario Melihat informasi produk.....	57
Tabel 4.35 Skenario Konfirmasi Stok Produk.....	57
Tabel 4.36 Skenario Mencetak Faktur penjualan (<i>Invoice</i>).....	58
Tabel 5.1 Spesifikasi perangkat keras komputer.....	82
Tabel 5.2 Struktur Class dan lokasi file	83
Tabel 5.3 Tabel Eloquent.....	84
Tabel 5.4 Implementasi ORM pada Model	84
Tabel 5.5 Pattern Front Controller.....	85
Tabel 5.6 Implementasi Blade Templating.....	89
Tabel 6.1 Pseudocode Menampilkan Pemberitahuan.....	106
Tabel 6.2 Kasus Uji Halaman Pemberitahuan Pengguna	108
Tabel 6.3 Pseudo ode Menambah Pemesanan Produk.....	110
Tabel 6.4 Kasus Uji menambah pemesanan produk.....	112
Tabel 6.5 Pseudocode Menambah Sarana dan Ruang	113
Tabel 6.6 Kasus Uji menambah sarana dan ruang.....	115
Tabel 6.7 Pengujian Black Box Sistem Pemeliharaan Sarana dan Pemantauan Pemasaran Produk PT.Pancamas Elite.....	116
Tabel 6.8 Daftar kebutuhan pengujian linux ubuntu.....	133
Tabel 6.9 skenario pengujian linux ubuntu.....	133
Tabel 6.10 Daftar kebutuhan pengujian windows 8 pro	134
Tabel 6.11 skenario pengujian windows 8 pro	134

DAFTAR GAMBAR

Gambar 2.1 Struktur organisasi PT.Pancamas Elite	6
Gambar 2.2 SDLC Model	10
Gambar 2.3 Alur proses MVC.....	12
Gambar 2.4 Hasil Survey <i>Sitepoint.com</i>	13
Gambar 2.5 Notasi class	18
Gambar 3.1 Tahap-tahap penelitian	20
Gambar 4.1 Diagram Alir Perancangan Sistem	24
Gambar 4.2 Proses bisnis pemeliharaan sarana	26
Gambar 4.3 Proses bisnis pemesanan produk.....	27
Gambar 4.4 Diagram Use Case.....	33
Gambar 4.5 Model Use Case Mengelola Sarana.....	37
Gambar 4.6 Use Case Mengelola Ruang	41
Gambar 4.7 Use Case Mengelola Produk.....	48
Gambar 4.8 Diagram Sekuen Melaporkan Kerusakan Sarana	60
Gambar 4.9 Diagram Sekuen Konfirmasi Perbaikan	60
Gambar 4.10 Diagram Sekuen Melihat Laporan perbaikan dan perawatan	61
Gambar 4.11 Diagram Sekuen Melihat Daftar Sarana.....	61
Gambar 4.12 Diagram Sekuen Menambah Sarana dan Ruang.....	62
Gambar 4.13 Diagram Sekuen Melihat status terakhir sarana.....	62
Gambar 4.14 Diagram Sekuen Menghapus Sarana	63
Gambar 4.15 Diagram Sekuen Membuat Pemesanan Baru	63
Gambar 4.16 Diagram Sekuen Menampilkan Daftar Produk	64
Gambar 4.17 Diagram Sekuen Melihat Riwayat Pemesanan	64
Gambar 4.18 Diagram Sekuen Konfirmasi Stok Produk.....	65
Gambar 4.19 Diagram Sekuen Mencetak Faktur Penjualan.....	65
Gambar 4.20 Diagram Aktivitas Melaporkan kerusakan sarana	66
Gambar 4.21 Diagram Aktivitas Konfirmasi perbaikan.....	67
Gambar 4.22 Diagram Aktivitas Melihat Laporan Perbaikan dan Perawatan	67
Gambar 4.23 Diagram Aktivitas Melihat Perawatan Rutin.....	68
Gambar 4.24 Diagram Aktivitas Menambah Sarana dan Ruang.....	69
Gambar 4.25 Diagram Aktivitas Menambah pemesanan baru	70

Gambar 4.26 Diagram Aktivitas Validasi pemesanan	71
Gambar 4.27 Diagram Aktivitas Konfirmasi stok produk.....	72
Gambar 4.28 Diagram Aktivitas Mencetak Faktur (<i>Invoice</i>).....	72
Gambar 4.29 Diagram Aktivitas Menambah produk	73
Gambar 4.30 Class Diagram pada <i>Controller</i>	74
Gambar 4.31 Class Diagram pada <i>Model</i>	75
Gambar 4.32 Perancangan Class Diagram	76
Gambar 4.33 Perancangan Antarmuka <i>Login</i>	79
Gambar 4.34 Perancangan Antarmuka Staf Pemasaran.....	79
Gambar 4.35 Perancangan Antarmuka Admin Pemasaran	80
Gambar 4.36 Perancangan Antarmuka Staf Gudang	80
Gambar 4.37 Perancangan Antarmuka Staf Pemeliharaan	81
Gambar 4.38 Perancangan Antarmuka Unit Kerja.....	81
Gambar 5.1 Antarmuka Login	90
Gambar 5.2 Antarmuka Dashboard Staf Pemeliharaan.....	91
Gambar 5.3 Antarmuka Daftar Jenis Sarana.....	91
Gambar 5.4 Antarmuka Daftar Ruangan	92
Gambar 5.5 Antarmuka Sarana dan Ruangan.....	92
Gambar 5.6 Antarmuka Daftar Permintaan Perawatan	93
Gambar 5.7 Antarmuka Daftar Perawatan Rutin.....	93
Gambar 5.8 Antarmuka Daftar Laporan Perbaikan dan Perawatan	94
Gambar 5.9 Antarmuka Dashboard Unit Kerja	94
Gambar 5.10 Antarmuka Form Permintaan Perbaikan	95
Gambar 5.11 Antarmuka Daftar Permintaan Perbaikan.....	95
Gambar 5.12 Antarmuka Dashboard Staff Pemasaran.....	96
Gambar 5.13 Antarmuka Kategori Produk.....	96
Gambar 5.14 Antarmuka Kategori Produk.....	97
Gambar 5.15 Antarmuka Pesanan Sementara.....	97
Gambar 5.16 Antarmuka Tambah Identitas Pemesan.....	98
Gambar 5.17 Antarmuka Laporan Penjualan.....	98
Gambar 5.18 Antarmuka Dashboard Admin Pemasaran.....	99
Gambar 5.19 Antarmuka Pesanan Baru.....	99

Gambar 5.20 Antarmuka Pesanan Terproses	100
Gambar 5.21 Antarmuka Faktur Penjualan	100
Gambar 5.22 Antarmuka Dashboard Antarmuka Gudang.....	101
Gambar 5.23 Antarmuka Kategori Produk.....	101
Gambar 5.24 Antarmuka Daftar Produk.....	102
Gambar 5.25 Antarmuka Pesanan Terkonfirmasi.....	102
Gambar 5.26 Antarmuka Daftar Pesanan Tertunda	103
Gambar 6.1 Diagram Blok Bab Pengujian dan Analisis.....	105
Gambar 6.2 Flowgraph Menampilkan Pemberitahuan	107
Gambar 6.3 Flowgraph Menambah Pemesanan produk.....	111
Gambar 6.4 Flowgraph Menambah Sarana dan Ruang	114
Gambar 6.5 Halaman login di linux Ubuntu.....	134
Gambar 6.6 Halaman daftar perawatan rutin di browser google chrome.....	135
Gambar 6.7 Halaman kategori produk di browser mozilla firefork.....	135
Gambar 6.8 Halaman daftar produk di browser internet explorer.....	136

DAFTAR LAMPIRAN

LAMPIRAN A : WAWANCARA.....141

LAMPIRAN B : SURAT KETERANGAN.....143



BAB 1 PENDAHULUAN

1.1 Latar belakang

Dengan semakin berkembangnya teknologi maka persaingan antar perusahaan menjadi semakin ketat. Persaingan dapat dilaksanakan dengan maksimal apabila tujuan setiap perusahaan yang sejenis telah dapat dipenuhi (Utama, 2003). Setiap perusahaan atau organisasi selalu memiliki indikator maupun standar untuk mencapai tujuan sesuai dengan yang diharapkan. Diantaranya dengan melaksanakan pemeliharaan dan perawatan terhadap inventaris yang dimiliki oleh perusahaan yang bersangkutan agar kegiatan operasional perusahaan tersebut dapat berjalan dengan lancar. Disamping itu, pemasaran jasa atau produk harus dapat dipantau dengan ketat agar dapat diketahui dengan pasti hasil pemasaran jasa atau produk perusahaan yang bersangkutan.

PT Pancamas Elite adalah perusahaan manufaktur logam yang sedang berkembang di Kota Malang saat ini. Dengan perkembangannya yang semakin pesat maka organisir inventaris dan pemantauan pemasaran hasil produksi pun semakin perlu diperhatikan. Selama ini perawatan inventaris hanya dilakukan ketika ada kerusakan. Alur proses yang harus dilalui yaitu Unit Kerja menyerahkan Dokumen Pelaporan Kerusakan kepada Staff Pemeliharaan, setelah itu Staff Pemeliharaan akan mengecek satu persatu dokumen tersebut dan memperbaikinya, hal tersebut membuat proses perbaikan menjadi cukup lama. Kendala lain yang terjadi di lapangan adalah dokumen tersebut terkadang hilang sebelum inventaris yang rusak diperbaiki. Selain itu, selama ini Laporan Riwayat Kerusakan Inventaris belum ada sehingga sulit untuk diketahui kerusakan inventaris apa saja yang pernah ditangani oleh Staff Pemeliharaan.

Kegiatan kedua yang juga sangat penting di PT Pancamas Elite adalah pemantauan kegiatan pemasaran hasil produksi. Selama ini Admin Pemasaran akan merekap dan mengevaluasi pemasaran yang sudah dilakukan oleh Staff Pemasaran berdasarkan laporan tertulis yang telah dibuat oleh Staff Pemasaran. Kegiatan pemantauan menjadi terhambat ketika Admin Pemasaran tidak memiliki kesempatan untuk bertatap muka dengan Staff Pemasaran yang ditugaskan. Selain itu, kendala lainnya adalah seringnya ditemukan kecurangan dalam bentuk ketidaksesuaian informasi jumlah produk yang keluar dengan hasil yang diterima oleh perusahaan.

Dewasa ini semakin banyak kerangka kerja yang tersedia secara bebas untuk mengembangkan sistem berbasis web. Kerangka kerja berbasis web dapat mempermudah pembangunan sistem berbasis web karena memiliki banyak fitur khusus yang dirancang untuk efisiensi waktu dan kode pemrograman sehingga dapat mempercepat pembangunan sistem. Salah satu kerangka kerja yang dimaksud adalah kerangka kerja *Laravel*. *Laravel* memiliki fitur unik yang tidak dimiliki oleh kerangka kerja lain diantaranya yaitu *Eloquent ORM* dan *Blade Template* yang dapat mempercepat penulisan kode pemrograman sehingga proses pengembangan dapat dilakukan dengan mudah. *Laravel* dapat digunakan

untuk aplikasi tingkat perusahaan besar atau personal dan cocok untuk semua ukuran proyek. Selain itu, komunitas-komunitas pengguna *Laravel* telah banyak terbentuk sehingga memberikan kemudahan pemecahan masalah ketika pengguna *Laravel* mengalami kesulitan. Hal ini membuat *Laravel* menjadi kian diminati seperti yang dilansir oleh *sitepoint.com* bahwa pada tahun 2015 telah diadakan survey dengan jumlah responden 7730 orang dari berbagai jenjang usia dan negara, hasilnya didapatkan peringkat 3 besar Kerangka Kerja Berbasis Web (PHP) terbaik yang adalah *Laravel* (24.39%), *Symfony2* (10.26%), dan *CodeIgniter* (5.79%).

Dengan dasar gambaran permasalahan di atas maka dapat dikatakan bahwa kerangka kerja *Laravel* mampu membantu permasalahan dalam membangun sistem yang dapat menyediakan informasi pemeliharaan inventaris. *Laravel* menyediakan beberapa fitur yang dapat mengefisiensi kode pemrograman antara lain dengan adanya standar fungsi-fungsi untuk menambah, mengubah, dan menghapus data serta tersedianya fungsi-fungsi untuk pembelian *online*. Selain itu, *Laravel* juga menyediakan fasilitas ORM (*Object Relational Mapping*) untuk mengatur pemetaan relasi antar objek sehingga pengaturan relasi pada basisdata juga dapat dilakukan di dalam kerangka kerja.

Karena pertimbangan di atas, penulis menawarkan penelitian untuk mengimplementasikan kerangka kerja *Laravel* pada pembangunan sistem pemeliharaan dan pemantauan pemasaran produk. Diharapkan dengan penggunaan kerangka kerja tersebut dapat meningkatkan efektifitas dan efisiensi pembangunan sistem. Berdasarkan latar belakang yang telah dikemukakan maka penulis mengusulkan judul penelitian pada skripsi ini yaitu "Rancang Bangun Sistem Pemeliharaan Sarana dan Pemantauan Pemasaran Produk Berbasis Web Menggunakan Kerangka Kerja *Laravel*" dengan studi kasus di PT Pancamas Elite.

1.2 Rumusan masalah

Berdasar latar belakang yang dikemukakan diatas maka terdapat beberapa rumusan masalah di penelitian ini :

1. Bagaimana membangun aplikasi untuk pemeliharaan inventaris dan pemantauan pemasaran produk dengan menggunakan pendekatan berorientasi objek?
2. Bagaimana merancang sistem berdasarkan hasil analisis kebutuhan yang telah dilakukan dan membangun sistem berbasis web yang menggunakan kerangka kerja *Laravel* berdasarkan rancangan yang telah dibuat?
3. Bagaimana membangun sistem berbasis web yang dapat membantu meminimalisir kendala pada bagian pemeliharaan inventaris dan pemasaran produk?

1.3 Batasan Masalah

Ruang lingkup yang membatasi permasalahan yang akan dibahas pada penelitian ini antara lain :

1. Sistem Pemeliharaan sarana dan pemantauan pemasaran produk difokuskan pada sistem informasi untuk mengelola pemeliharaan rutin dan

pengajuan permintaan perbaikan pada divisi pemeliharaan serta memantau laporan penjualan dan validasi pemesanan pada divisi pemasaran di PT.Pancamas Elite.

2. Divisi marketing hanya menangani masalah pemantauan pemesanan produk, mekanisme pembayaran dikelola oleh divisi keuangan dan pengiriman dikelola oleh divisi mutu. Mekanisme perbaikan kerusakan sarana pada divisi pemeliharaan hanya menangani laporan permintaan perbaikan dan perawatan rutin, mekanisme penggantian suku cadang dilakukan oleh pihak ketiga.
3. Dalam fase menggunakan sistem pada tahapan *waterfall* tidak dilakukan tahapan pendukung (*support*) karena instalasi aplikasi belum dilakukan.

1.4 Tujuan

Adapun tujuan diadakannya penelitian ini adalah membangun sistem untuk PT Pancamas Elite bagian Pemeliharaan Inventaris dan Pemantauan Pemasaran Produk yang handal menggunakan kerangka kerja terbaru dan terbaik sehingga dapat meminimalisir kendala pada proses pemeliharaan sarana dan pemantauan pemasaran produk.

1.5 Manfaat Penelitian

Manfaat dari pembuatan skripsi ini adalah :

1. Bagi Penulis, meningkatkan wawasan mengenai perancangan dan pembangunan sistem berbasis web menggunakan kerangka kerja *Laravel*.
2. Bagi PT Pancamas Elite, dapat meningkatkan kinerja bagian pemeliharaan dan pemasaran produk dan meminimalisir kendala pada bagian pemeliharaan dan kecurangan pada bagian pemasaran.
3. Bagi Masyarakat Umum, penelitian ini diharapkan dapat menjadi sumber wawasan mengenai efektifitas sistem berbasis web menggunakan kerangka kerja *Laravel*.
4. Bagi Ilmu Pengetahuan, penelitian ini diharapkan dapat dijadikan sumber referensi bagi pengembangan ilmu terkait.

1.6 Sistematika Penulisan

Untuk mencapai tujuan yang diharapkan maka sistematika penulisan yang disusun dalam tugas akhir ini adalah:

Bab I : Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta sistematika penulisan.

Bab II : Kajian Pustaka dan Dasar Teori

Bab ini berisi tentang tinjauan pustaka dan dasar teori yang berhubungan dengan sistem pemeliharaan sarana dan pemantauan produk yang akan diteliti.

Bab III : Metodologi Penelitian

Bab ini membahas tentang metode yang digunakan dalam menyelesaikan masalah yang dibahas dalam penelitian membangun sistem informasi pemeliharaan dan pemantauan produk.

Bab IV : Perancangan

Pada bab ini akan dijelaskan mengenai perancangan sistem yang digunakan dalam penyelesaian masalah sistem pemeliharaan sarana dan pemantauan produk.

Bab V : Implementasi

Bab ini berisi tentang gambaran sistem dan deskripsi sistem hasil analisa kebutuhan dan perancangan yang diimplementasikan kedalam program komputer.

Bab VI : Pengujian dan Analisis

Memuat tentang analisa dan hasil pengujian dari penelitian yang telah dilakukan.

Bab VII : Penutup

Memuat tentang kesimpulan dan saran-saran berdasarkan hasil pembahasan



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian mengenai pembangunan sistem informasi dan pemeliharaan sarana dan pemantauan pemasaran produk telah dilakukan oleh peneliti sebelumnya. Beberapa penelitian menunjukkan kemajuan perkembangan teknologi telah mengubah cara hidup masyarakat di dunia dalam menjalankan aktivitas sehari-hari (Fadmara, 2010). Sistem pencatatan, penjadwalan dan penyimpanan data pemeliharaan secara manual yaitu dicatat pada buku dan kertas dapat menimbulkan hal negatif, salah satunya ketidakefisienan waktu, tempat penyimpanan hasil pencatatan, kehilangan dokumen, sulit dalam pencarian data lampau dan pembuatan laporan menjadi sulit karena data yang diperlukan untuk membuatnya karena tidak tersusun dengan baik (Aminuddin, 2009).

Berbagai penelitian telah dilakukan untuk menangani masalah pemeliharaan sarana. Salah satunya Penelitian yang berjudul "*Sistem Informasi Pemeliharaan Sarana di Program Teknologi Informasi dan Ilmu Komputer (PTIIK)*". Pada penelitian ini membahas proses pemeliharaan inventaris yang meliputi proses pengajuan kerusakan sarana dan prasarana berbasis web, penelitian ini hanya memfokuskan pada pengajuan kerusakan mendadak (Endah, 2014).

Produk selain harus bersaing dengan para pesaing dengan strategi pemasaran yang jitu juga harus meningkatkan kreatifitas dan efisiensi dalam memantau proses dari pemasaran dan penjualan yang digunakan untuk pengambilan keputusan strategi pemasaran, dikarenakan pemasaran dan penjualan produk tersebar ke beberapa daerah (Fadmara, 2010). Penelitian selanjutnya yang membahas tentang "*Rancang Bangun Sistem Informasi Penjualan Produk menggunakan kerangka kerja Laravel*" (Wulandari, 2015). Penelitian tersebut membahas tentang pengelolaan pemasaran produk berbasis web. Tujuan dari penelitian adalah untuk memenuhi kebutuhan masyarakat dan lingkungan gaya hidup masyarakat yang semakin *modern* serta tuntutan masyarakat terhadap kemudahan dan efisiensi dalam pemenuhan kebutuhan mereka yang semakin tinggi.

2.2 Dasar Teori

Pada bagian ini menjelaskan tentang dasar teori yang digunakan dalam pembangunan sistem informasi pemeliharaan sarana dan pemantauan produk menggunakan kerangka kerja laravel. Adapun dasar teori yang diuraikan adalah dasar teori tempat studi kasus, sistem informasi, model rekayasa perangkat lunak dan kerangka kerja pembangunan aplikasi berbasis web.

2.2.1 PT. Pancamas Elite

PT. Pancamas Elite adalah perusahaan yang bergerak dalam bidang manufaktur logam seperti *handle* pintu, aksesoris jendela dan lain-lain.

PT. Pancamas Elite memiliki sejumlah divisi dimana setiap divisi memiliki tugas dan kewajiban yang berbeda satu dengan yang lainnya.

PT. Pancamas Elite didirikan tahun 1987 di Malang, Jawa Timur. Selama tahun 1987-1997 merupakan perusahaan patungan dengan Taiwan yang memberikan *support* teknologi terhadap seluruh proses produksi. Sejak awal tahun 1998 hingga saat ini, PT. Pancamas Elite menjadi perusahaan lokal dengan tetap mengadopsi teknologi Taiwan. Di Indonesia, PT. Pancamas Elite merupakan salah satu perusahaan yang menggunakan proses *Fully Automatic Die Casting (Hot & Cold Chamber)* dalam memproduksi produk-produk unggulannya.

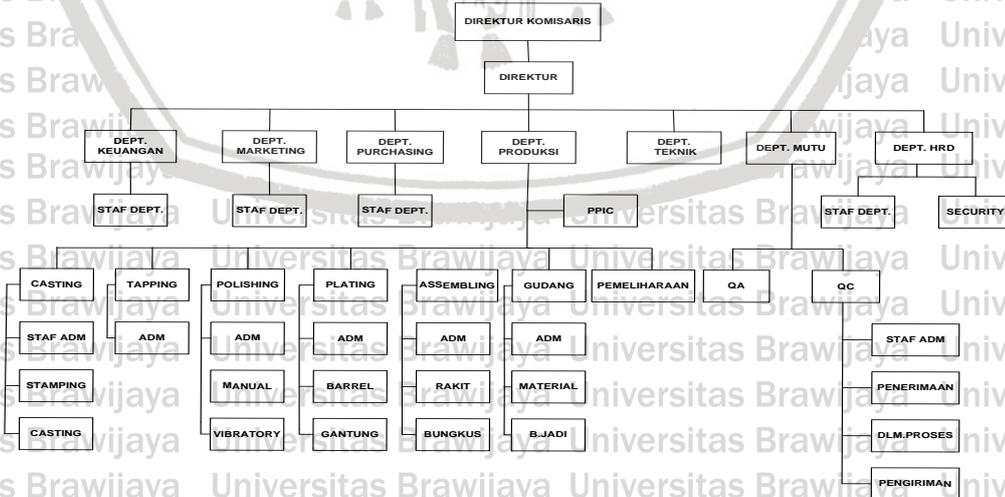
Untuk menjamin konsistensi kerja mesin, sehingga mutu produk yang dihasilkan selalu konsisten. Manajemen melakukan penerapan TPM (*Total Production Maintenance*). Dengan berbagai kombinasi diatas, produk PT. Pancamas Elite memiliki daya saing yang tinggi dan konsisten dalam mutu. Dalam menghasilkan produk yang memenuhi standar dan konsisten.

2.2.2 Visi dan Misi PT. Pancamas Elite

Visi dari PT. Pancamas Elite adalah menjadi perusahaan yang terbaik diantara industri *Die Casting* dan menguasai dan mengembangkan teknologi *Die Casting* dengan baik, menjadi perusahaan yang terbaik di antara industri *Die Casting*. Untuk mencapai perusahaan manufaktur yang terbaik diantara industry *Die Casting* ditempuh melalui misi antara lain dengan meningkatkan kualitas sumber daya manusia untuk mencapai efektifitas dan efisiensi kerja pada setiap bagian perusahaan, Meningkatkan mutu proses pada setiap bagian untuk menjamin konsistensi mutu produk sesuai permintaan pelanggan dan menghasilkan produk yang memenuhi standar dan konsisten sesuai penerapan manajemen mutu yang sesuai dengan ISO 9001:2008.

2.3 Struktur Organisasi

Struktur Organisasi pada PT. Pancamas Elite sebagai berikut :



Gambar 2.1 Struktur organisasi PT. Pancamas Elite

Divisi Pemeliharaan adalah bagian yang menangani masalah pemeliharaan di PT.Pancamas Elite, Divisi ini dibawah oleh Departemen Produksi yang juga mengatur masalah produksi. Divisi pemeliharaan menangani pemeliharaan alat-alat produksi seperti mesin *CNC*, *Pneumatic Marking Machine* dan lain lain beserta pemeliharaan alat-alat kantor seperti personal komputer (PC), printer, scanner, mesin fotokopi, Faximile, *Air Conditioner* (AC) dan lain-lain. Selanjutnya untuk pemesanan produk hasil produksi dikelola oleh Departemen Pemasaran yang terdiri dari beberapa staf departemen yang menangani masalah keluar masuk produk yang ada di PT.Pancamas Elite. Dalam pemesanan produk, divisi pemasaran bertanggung jawab terhadap kuantitas produk sebelum didistribusikan kepada konsumen.

2.4 Pemeliharaan

Pada umumnya sebuah produk yang dihasilkan oleh manusia, tidak ada yang tidak mungkin rusak, tetapi usia penggunaannya dapat diperpanjang dengan melakukan perbaikan yang dikenal dengan pemeliharaan. Fungsi pemeliharaan adalah agar dapat memperpanjang umur ekonomis dari mesin dan peralatan produksi yang ada serta mengusahakan agar mesin dan peralatan produksi tersebut selalu dalam keadaan optimal dan siap pakai untuk pelaksanaan proses produksi (Aminuddin, 2009). Oleh karena itu, sangat dibutuhkan kegiatan pemeliharaan yang meliputi kegiatan pemeliharaan dan perawatan sarana prasarana yang digunakan dalam proses produksi.

2.4.1 Jenis-Jenis Pemeliharaan

Pemeliharaan adalah kombinasi dari berbagai kegiatan yang dilakukan untuk memelihara fasilitas produksi termasuk mesin dan alat-alat produksi lainnya atau untuk memperbaikinya sampai pada suatu kondisi yang dapat diterima. Pemeliharaan terdiri dari 2 jenis diantaranya adalah sebagai berikut.

a. Pemeliharaan Pencegahan

Pemeliharaan Pencegahan adalah kegiatan pemeliharaan dan perawatan untuk mencegah kerusakan yang tak terduga dan menemukan kondisi atau keadaan yang menyebabkan fasilitas tidak berjalan dengan baik.

Alternatif dalam Pemeliharaan Pencegahan adalah (Goyal, 2009):

1. Berdasar waktu, yaitu melakukan pemeliharaan pada periode secara teratur.
2. Berdasar pekerjaan, yaitu pemeliharaan setelah sejumlah jam operasi atau *volume* produksi tertentu.
3. Berdasar kesempatan, yaitu pemeliharaan yang dilakukan apabila ada kesempatan. Berdasar kondisi terencana, yaitu tergantung pada hasil pemantauan kondisi fasilitas produksi.

Pemeliharaan pencegahan sangat tepat dilakukan, karena kegunaannya sangat efektif dalam menghadapi fasilitas-fasilitas produksi yang termasuk dalam *critical unit*, yaitu peralatan atau fasilitas yang membahayakan

kehatan dan keselamatan kerja, mempengaruhi produk yang dihasilkan dan jika tidak dilakukan perawatan dapat menyebabkan kemacetan seluruh proses produksi.

b. Pemeliharaan *Corrective*

Disebut juga *break down* pemeliharaan, yaitu kegiatan pemeliharaan dan perawatan yang dilakukan setelah terjadi kerusakan, kegagalan, atau kelainan fasilitas produksi sehingga tidak dapat berfungsi dengan baik.

2.5 Pemantauan

Pemantauan adalah suatu proses mengukur, mencatat, mengumpulkan, memproses dan mengkomunikasikan informasi untuk membantu pengambilan keputusan manajemen program atau proyek. Pemantauan adalah suatu proses untuk mengumpulkan informasi dari produk yang keluar dan masuk agar dapat dianalisis untuk tujuan selanjutnya. Pemantauan adalah bagian dari kegiatan pengawasan, dalam pengawasan ada aktivitas memantau kegiatan pemasaran. Pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa apakah unit yang telah berjalan sesuai dengan sasaran atau sesuai dengan tujuan dari program. Secara lebih terperinci pemantauan bertujuan antara lain untuk (Rubin, 1995):

1. Memberikan informasi tentang metode yang tepat untuk melaksanakan proses kerja.
2. Membantu para pengambil keputusan untuk merencanakan strategi selanjutnya dengan baik.
3. Memberikan informasi sirkulasi suatu proses kerja, sehingga proses kerja tersebut dapat dipantau dengan baik untuk meminimalisir kesalahan.

2.6 Sistem Informasi

Sistem informasi adalah proses yang menjalankan fungsi mengumpulkan, memproses, menyimpan, menganalisis dan menyebarkan informasi untuk tujuan tertentu. Komponen-komponen sistem informasi adalah (Sabharwai, 2009):

1. Perangkat keras (*hardware*) adalah serangkaian peralatan seperti prosesor, monitor, keyboard dan printer. Berbagai peralatan tersebut menerima data serta informasi, memprosesnya dan menampilkannya.
2. Perangkat lunak (*software*) adalah sekumpulan program yang memungkinkan peranti keras untuk memproses data.
3. Basis data (*database*) adalah sekumpulan file, tabel, relasi dan lain-lainnya yang saling berkaitan dan menyimpan data serta berbagai hubungan di antaranya.
4. Jaringan (*network*) adalah sistem koneksi dengan kabel atau nirkabel yang memungkinkan adanya berbagai sumber daya antar berbagai komputer yang berbeda.
5. Orang adalah berbagai individu yang bekerja dengan sistem informasi, berinteraksi dengannya atau menggunakan hasilnya.

Berdasarkan uraian diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan komponen yang saling berhubungan dan bekerja sama untuk mengumpulkan, memproses dan menyediakan informasi yang berguna untuk pengambilan keputusan saat ini ataupun di masa yang akan datang.

2.7 Rekayasa Perangkat Lunak (Software Engineering)

Rekayasa Perangkat Lunak adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal *requirement capturing* (analisa kebutuhan pengguna), *specification* (menentukan spesifikasi dari kebutuhan pengguna), *design, coding, testing* sampai pemeliharaan sistem setelah digunakan (Sommerville, 2007). *Requirement* untuk sistem merupakan deskripsi dari layanan yang diberikan oleh sistem dan menjadi dasar dalam melakukan perancangan terhadap sistem yang akan dibangun. Persyaratan ini mencerminkan kebutuhan pelanggan untuk sistem yang membantu memecahkan beberapa masalah seperti pencarian informasi. Perbedaan yang digunakan, pertama dengan menggunakan *user requirements* untuk jangka yang tertentu dengan *high level abstract requirement* dan yang kedua yaitu *system requirements* yaitu penjelasan rinci mengenai sistem yang harus dilakukan. *User requirements* dan *system requirements* dapat didefinisikan sebagai berikut.

2.7.1 User Requirements

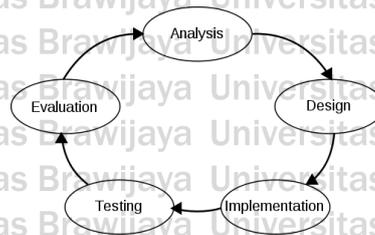
User Requirement merupakan tahap analisis kebutuhan yang berupa pernyataan dengan bahasa yang dapat dipahami oleh pengguna dan digambarkan dalam diagram agar mempermudah penalaran, serta melihat apakah layanan bisa berjalan sesuai harapan atau tidak. *User requirement* juga menggambarkan *functional* dan *non-functional requirement* yang dapat dipahami oleh pengguna (*user*) yang tidak memiliki latar belakang teknis yang cukup. *User requirement* menjelaskan perilaku luar dari sistem, tidak secara teknis, karena itu perlu menggunakan bahasa yang sederhana (Dharwiyanti, 2003).

2.7.2 System Requirements

System requirements berhubungan erat dengan *user requirements*. *System requirements* juga melakukan pendokumentasian yang wajib dilakukan oleh setiap developer. Proses ini menterjemahkan definisi kebutuhan yang diperoleh dari proses *user requirement* kedalam spesifikasi kebutuhan perangkat lunak.

2.8 System Development Life Cycle (SDLC)

System Development Life Cycle (SDLC) adalah suatu proses pembuatan dan perubahan sistem beserta model dan metodologi yang digunakan sistem dalam mengembangkan sistem (Roebuck, 2012).



Gambar 2.2 SDLC Model

SDLC mengembangkan pola perangkat lunak melalui beberapa tahapan antara lain *analysis, design, implementation, testing dan evaluation*. SDLC adalah pengembangan perangkat lunak yang bertujuan agar sistem yang dibangun dapat berjalan dengan kerangka kerja yang tepat sehingga kebutuhan pengguna dapat terpenuhi dengan baik.

2.8.1 Waterfall Model

Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak secara berurutan atau linier di mana proses selanjutnya dipengaruhi oleh proses sebelumnya. Prosesnya mengalir ke bawah melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian. Dalam pengembangannya metode waterfall memiliki beberapa tahapan yang runtut antara lain requirement (analisis kebutuhan), design sistem (*system design*), coding dan testing, penerapan program dan pemeliharaan. Dalam pengembangannya metode waterfall memiliki beberapa tahapan antara lain (Petersen, 2008):

1. Analisis kebutuhan (*Requirement analysis*).

Dalam langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Seseorang system analisis akan menggali informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan system analisis untuk menterjemahkan kedalam bahasa pemrograman.

2. Desain Sistem (*Design sistem*).

Proses design akan menterjemahkan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail algoritma prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

3. Penulisan kode program (*Implementation*)

Coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menerjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan computer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap system tersebut dan kemudian bisa diperbaiki.

4. Pengujian Program (*Integration & Testing*)

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, design dan pengkodean maka sistem yang sudah jadi akan digunakan oleh user.

5. Pemeliharaan (*Operation & Maintenance*)

Perangkat lunak yang susah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

2.9 Website

Website adalah kumpulan dari halaman - halaman *situs*, yang terangkum dalam sebuah *domain* atau *subdomain*. Sebuah halaman *web* biasanya berupa dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*) yang diakses melalui HTTP (Malatesta,2014). HTTP adalah sebuah protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*. *Website* atau situs juga diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

2.10 PHP

PHP adalah bahasa pemrograman script *server-side* yang didesain untuk pengembangan *web*. PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer *server*. Selanjutnya hasil pengolahan *script* atau program tersebut akan dikirim ke *client* sebagai *output*. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti JavaScript yang diproses pada *web browser (client)* (Malatesta,2014). PHP digunakan untuk merancang, membuat dan juga memprogram sebuah *website*. PHP juga sangat sering digunakan untuk membuat sebuah atau beberapa CMS. CMS ialah sebuah *software* atau perangkat lunak yang mempunyai kegunaan untuk

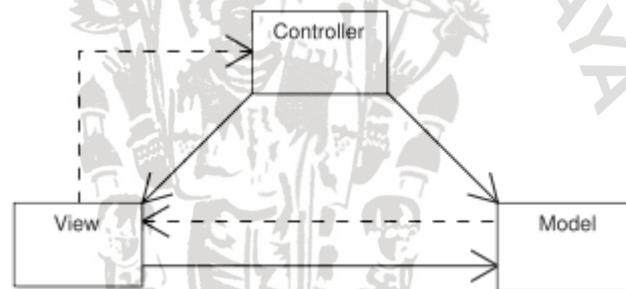
memanipulasi semua atau beberapa isi dari sebuah halaman *website*. PHP digunakan dan dijalankan di sebuah halaman *website* untuk mengolah isi data dari *website* tersebut yang akan dilihat oleh para pengunjung dari *website* tersebut.

2.11 Kerangka Kerja

Kerangka kerja atau Framework adalah sekumpulan perintah atau fungsi dasar yang dapat membantu dalam menyelesaikan proses-proses yang lebih kompleks, menangani berbagai masalah dalam pemrograman seperti koneksi basis data, pemanggilan *variable*, dan lain lain. Kerangka kerja memungkinkan *developer* lebih fokus dan lebih cepat dalam membangun aplikasi (Rosa, 2011).

2.11.1 Konsep MVC

Konsep MVC merupakan konsep yang digunakan didalam kerangka kerja untuk membangun aplikasi , pada konsep MVC dilakukan pemisahan antara *logic* dengan tampilan dan basis data. Manfaat konsep MVC adalah membuat *coding logic* lebih sederhana Konsep MVC ditampilkan dalam gambar 2.3 dibawah ini.



Gambar 2.3 Alur proses MVC

1. Model

Merupakan struktur data. Model berisi fungsi didalam pengolahan basis data, *script SQL* dikelola didalam fungsi ini.

2. View

Merupakan kode untuk tampilan suatu program. Tampilan dapat berupa *web page*, *header*, *footer*, *content* dan apa saja yang berjenis tampilan.

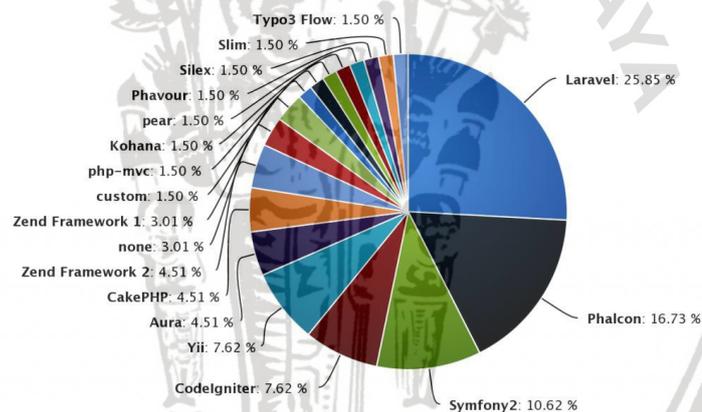
3. Controller

Merupakan kode untuk *logic*, algoritma dan sebagai penghubung antara *model*, *view* dan sumber lain yang diperlukan untuk mengolah *HTTP request* dan *generate web page*.

2.11.2 Laravel

Kerangka kerja Laravel dibuat oleh Taylor Otwell, proyek Laravel dimulai pada April 2011. Laravel merupakan kerangka kerja PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya (Rohman, 2014). Laravel menyediakan teknologi baru untuk memudahkan developer membangun aplikasi web berbasis bahasa pemrograman PHP. Laravel dibangun dengan basis MVC (*Model-View-Controller*). Laravel dilengkapi *command line tool* yang bernama “*Artisan*” yang bisa digunakan untuk *packaging bundle* dan *instalasi bundel*.

Laravel merupakan framework aplikasi web yang ekspresif dan sintaks yang elegan. Laravel berusaha mencoba menjadi framework yang mudah digunakan dengan mengurangi tugas-tugas umum yang sering digunakan dalam sebagian besar proyek-proyek web dengan menyediakan fungsi seperti *otentikasi*, *routing*, *session*, dan *caching* (Matula, 2013). Laravel menggabungkan yang terbaik dari apa ada dalam kerangka kerja web lain, termasuk kerangka kerja yang menggunakan bahasa lain, seperti *Ruby on Rails*, *ASP.NET MVC*, dan *Sinatra* (Rohman, 2014). Menurut survey yang dilakukan oleh *Sitepoint.com* pada Desember 2013 dalam popularitas kerangka kerja PHP, Laravel menduduki urutan teratas. Sehingga menjadikan Laravel sebagai *kerangka kerja PHP* terbaik.



Gambar 2.4 Hasil Survey Sitepoint.com

Laravel merupakan kerangka kerja yang mudah diakses, *powerful* dan menyediakan *tools* yang diperlukan untuk skala aplikasi besar maupun kecil. Laravel juga merupakan sebuah aplikasi luar biasa dari sebuah kumpulan program kontrol, sistem migrasi yang *ekspresif* dan dukungan *tools* yang anda butuhkan dalam menguji aplikasi anda yang terintegrasi dengan beberapa aplikasi lainnya (Sanjib, 2014).

Beberapa fitur yang dimiliki kerangka kerja Laravel :

- a. *Bundles* yaitu sebuah *fitur* dengan *system* pengemasan *modular* dan berbagai *bundle* telah tersedia untuk di gunakan dalam aplikasi.

- b. *Eloquent ORM* merupakan penerapan PHP lanjutan dari pola “*active record*”, menyediakan metode internal untuk mengatasi kendala hubungan antara objek database. Pembangun *query Laravel’s Fluent* ini didukung oleh *Eloquent*.
- c. *Application logic* merupakan bagian dari aplikasi yang dikembangkan, baik menggunakan *Controllers* atau sebagai bagian dari deklarasi *Route*. Sintak yang digunakan untuk mendefinisikannya mirip dengan yang digunakan oleh kerangka kerja *Sinatra*.
- d. *Reverse routing* mendefinisikan hubungan antara *Link* dan *Route*, sehingga jika suatu saat ada perubahan pada *route* secara otomatis akan tersambung dengan *link* yang relevan. Ketika *Link* yang dibuat dengan menggunakan nama-nama dari *Route* yang ada, secara otomatis Laravel akan membuat URL yang sesuai.
- e. *Restful controllers* memberikan sebuah *option* (pilihan) untuk memisahkan logika dalam melayani *HTTP GET* dan permintaan *POST*.
- f. *Class auto loading* menyediakan otomatis *loading* untuk *class-class* PHP, tanpa membutuhkan pemeriksaan manual terhadap jalur masuknya. Fitur ini mencegah *loading* yang tidak perlu.
- g. *View composers* adalah kode *unit logical* yang dapat dieksekusi ketika sebuah *View* di-load.
- h. *IoC Container* memungkinkan untuk objek baru yang dihasilkan dengan mengikuti prinsip control pembalik, dengan pilhan contoh dan referensi dari objek baru sebagai *singletons*.
- i. *Migrations* menyediakan versi *sistem control* untuk skema database, sehingga memungkinkan untuk menghubungkan perubahan dalam basis kode aplikasi dan keperluan yang dibutuhkan dalam merubah tata letak database, mempermudah dalam penempatan dan memperbarui aplikasi.
- j. *Unit Testing* mempunyai peran penting dalam kerangka kerja Laravel, dimana unit testing ini mempunyai banyak tes untuk medeteksi dan mencegah *regresi*. Unit testing ini dapat dijalankan melalui utilitas “*artisan command-line*”
- k. *Automatic pagination* menyederhanakan tugas dari penerapan halaman, menggantikan penerapan yang manual dengan metode otomatis yang terintegrasi ke Laravel.

2.12 UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Prayitno, 2005). UML disajikan dalam berbagai macam jenis dan memiliki fungsi masing-masing. Pada penelitian kali ini penulis menggunakan 4 jenis UML yaitu *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*. Berikut adalah penjelasan lebih rinci mengenai keempat diagram-diagram tersebut.

2.12.1 Use Case Diagram

Diagram *use case* merupakan salah satu diagram untuk memodelkan aspek perilaku sistem atau digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem. *Use Case* juga memiliki simbol-simbol dalam perancangannya. Berikut adalah simbol-simbol *use case* yang akan dijelaskan pada tabel dibawah ini.

Tabel 2.1 Notasi Use Case Diagram

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Spesifikasikan himpunan peran yang menggunakan ketika berinteraksi dengan <i>use case</i>
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur dari data objek yang ada diatasnya objek induk (<i>ancestor</i>)
4		<i>Include</i>	Spesifikasikan bahwa <i>use case</i> sumber secara eksplisit
5		<i>Extend</i>	Spesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Spesifikasikan paket yang menampilkan sistem secara terbatas
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan

			sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber : (Rosa, 2013)

2.12.2 Sequence Diagram

Sequence diagram digunakan untuk memodelkan skenario penggunaan. Skenario penggunaan adalah urutan kejadian yang terjadi selama satu eksekusi sistem. *Sequence diagram* adalah teknik pemodelan yang dinamis. UML *sequence diagram* biasanya digunakan antara lain untuk:

- Validasi* dan menyempurnakan logika dan kelengkapan penggunaan suatu skenario. Menunjukkan deskripsi cara bahwa sistem anda bisa digunakan.
- Menjelajahi desain karena menggambarkan langkah visual melalui permintaan dari operasi yang didefinisikan dari kelas.
- Memberikan perkiraan kelas mana dalam aplikasi yang akan menjadi kompleks.
- Mendeteksi hambatan dalam desain berorientasi objek, melihat pesan apa yang sedang dikirim ke objek, melihat kira-kira berapa lama waktu yang dibutuhkan untuk menjalankan metode, serta memahami tentang di mana perlunya mengubah desain untuk mempermudah jalannya sistem.

Berikut adalah simbol-simbol *sequence diagram* yang akan dijelaskan pada tabel 2.3.

Tabel 2.2 Notasi Sequence Diagram

No	Notasi	Nama	Keterangan
1		<i>View</i>	<i>View</i> merupakan bagian yang menangani <i>presentation logic</i> . <i>View</i> berfungsi untuk menerima dan merepresentasikan data kepada user
2		<i>Controller</i>	<i>Controller</i> merupakan bagian yang mengatur hubungan antara bagian model dan bagian <i>view</i> , <i>controller</i> berfungsi untuk menerima <i>request</i>

			dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi
3		<i>Model</i>	Model berhubungan langsung dengan database untuk memanipulasi data namun tidak dapat berhubungan langsung dengan bagian <i>view</i> .
2		<i>Actor</i>	<i>Actor</i> dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom.
3		<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu.
4		<i>Activation</i>	<i>Activation</i> mengindikasikan sebuah obyek yang akan melakukan sebuah aksi.
5		<i>Message</i>	<i>Message</i> mengindikasikan komunikasi antara <i>object-object</i> .

Sumber : (Barbara, 2009)

2.12.3 Class Diagram

Class diagram merupakan diagram yang digunakan untuk dokumen, berupa tabel untuk sistem informasi yang menjelaskan hubungan dan atribut dari tabel tersebut. *Class diagram* menggambarkan struktur dan deskripsi *Class Package* dan objek beserta hubungan satu sama lain seperti pewarisan, *asosiasi*, dan lain-lain. *Class diagram* menggambarkan *class-class* yang perlu dibuat dalam perancangan sistem berorientasi objek (Barbara, 2009). Gambar 2.5 berikut adalah notasi *class diagram*.

Nama_class
- Attribute pertama : int
- Attribute kedua : int
- Attribute selanjutnya... : int
+ Operation pertama() : void
+ Operation kedua() : void
+ Operation selanjutnya...() : void

Gambar 2.5 Notasi class

2.13 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan suatu proses yang dilakukan untuk mendapatkan informasi mengenai kualitas dari produk atau layanan yang sedang diuji. Pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara obyektif dan independen, yang bermanfaat dalam operasional bisnis untuk memahami tingkat risiko pada implementasinya (Pressman, 2010).

Strategi untuk proses pengujian perangkat lunak yaitu mengintegrasikan metode desain kasus uji perangkat lunak dengan kasus uji perangkat lunak pada serangkaian langkah yang disusun dengan baik dan menghasilkan konstruksi perangkat lunak yang berhasil. Strategi pada pengujian dapat dilakukan melalui pengujian tingkat rendah yakni pengujian yang dilakukan pada *source code* program untuk memberikan bukti segmen kode sumber yang kecil telah diimplementasikan dengan tepat. Selain itu juga ada pengujian tingkat tinggi memvalidasi fungsi-fungsi sistem mayor.

2.13.1 Pengujian Validasi dengan Metode *Blackbox*

Pengujian black box digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang. Pada teknik ini, kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut. Dari keluaran yang dihasilkan, kemampuan program dalam memenuhi kebutuhan pemakai dapat diukur sekaligus dapat diketahui kesalahan-kesalahannya. Beberapa jenis kesalahan yang dapat diidentifikasi :

1. Fungsi tidak benar atau hilang
2. Kesalahan antarmuka
3. Kesalahan pada struktur data (pengaksesan basis data)
4. Kesalahan inisialisasi dan akhir program
5. Kesalahan performansi

Walaupun sulit untuk menelusuri kesalahan yang mungkin didapat, teknik pengujian black box lebih sering dipilih untuk menguji perangkat lunak karena kemudahan dalam pelaksanaannya (Simamarta, 2010).

2.13.2 Pengujian *Whitebox*

Pengujian *whitebox* adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian kedalam beberapa kasus pengujian. Penentuan kasus uji disesuaikan dengan struktur system, pengetahuan mengenai program digunakan untuk mengidentifikasi kasus uji tambahan (Whitten, 2004). Tujuan penggunaan *whitebox* untuk menguji semua statement program. Penggunaan metode pengujian *whitebox* dilakukan untuk:

1. Memberikan jaminan bahwa semua jalur independen suatu modul digunakan minimal satu kali.
2. Menggunakan semua keputusan logis untuk semua kondisi true/false
3. Mengeksekusi semua perulangan pada batasan nilai dan operasional pada setiap kondisi.
4. Menggunakan struktur data internal untuk menjamin validitas jalur keputusan

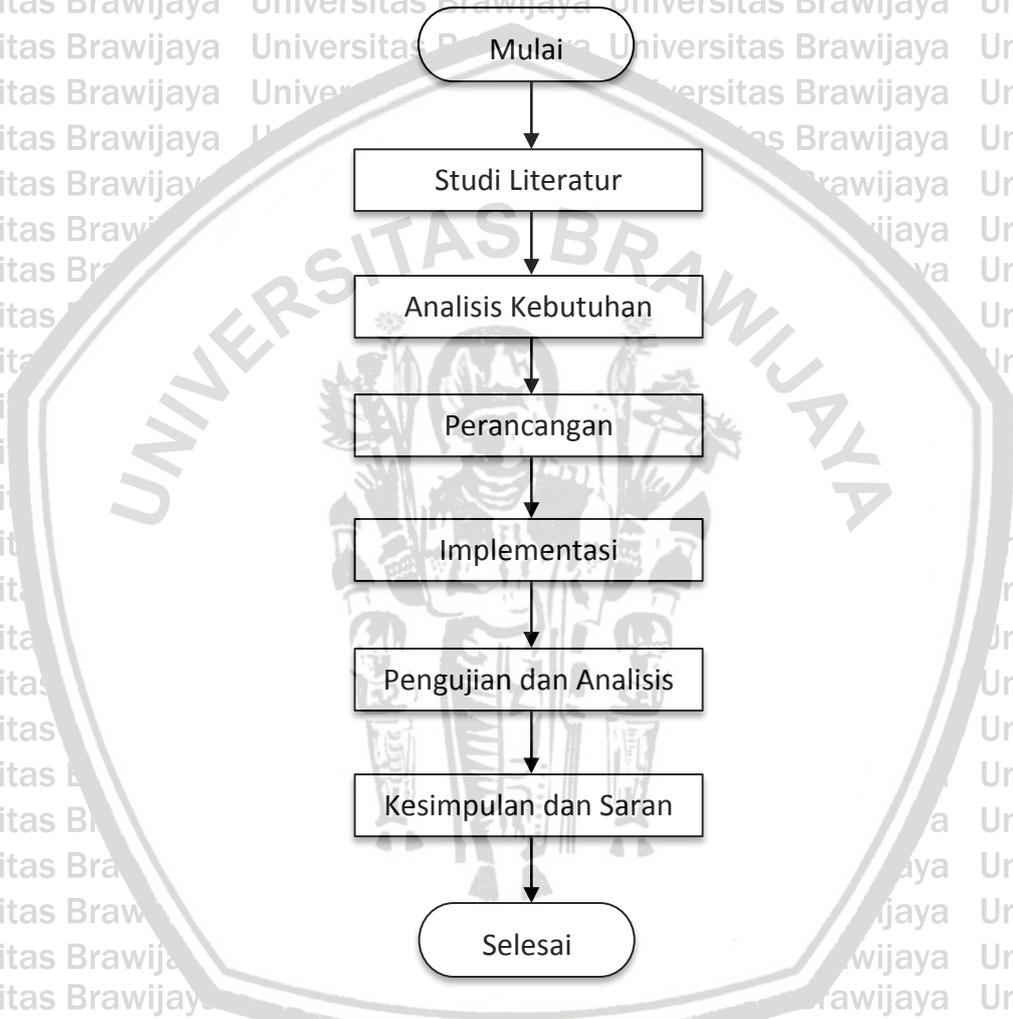
Dengan menggunakan metode pengujian *whitebox*, perancang sistem dapat melakukan kasus uji yang memberikan jaminan bahwa semua jalur independen telah digunakan paling tidak satu kali, menggunakan semua keputusan logis pada sisi *true* atau *false*, dan menggunakan struktur data internal untuk menjamin validitasnya. Jenis pengujian *whitebox* antara lain adalah *Basis Path Testing* dan *Control Structure Testing*.

2.13.3 Pengujian *Compatibility*

Pengujian *compatibility* adalah pengujian yang dilakukan untuk menguji bahwa fitur produk sistem bekerja konsisten dengan komponen infrastruktur yang berbeda. Pengujian berfungsi untuk menentukan lingkungan yang diharapkan dapat menjalankan aplikasi yang dikembangkan (Barbara, 2009). Semakin aplikasi dapat berjalan di banyak jenis perangkat yang berbeda, maka semakin baik aspek kompatibilitinya.

BAB 3 METODOLOGI

Dalam bab ini membahas metode dan tahapan pengerjaan yang dilakukan dalam penelitian. Penelitian ini menggunakan pendekatan *waterfall* pada pengembangan perangkat lunak. Penelitian ini dimulai dari studi literatur, fase analisa kebutuhan, fase perancangan, fase implementasi sistem, pengujian dan analisis serta pengambilan kesimpulan dan saran.



Gambar 3.1 Tahap-tahap penelitian

3.1 Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan sebagai sumber acuan untuk penunjang dan pendukung penulisan skripsi dengan pemahaman kepustakaan (*library research*). Sumber atau referensi yang digunakan antara lain buku, jurnal, laporan penelitian, dan bantuan mesin pencari (*search engine*) internet. Teori dan pustaka yang berkaitan dengan penelitian meliputi :

1. Sistem Pemeliharaan dan Pemantauan
2. Kerangka kerja Laravel
3. Konsep MVC
4. Website
5. PHP
6. *Unified Modelling Language* (UML)
7. *System Development Life Cycle* (SDLC)
8. *Waterfall Model*
9. Rekayasa Perangkat Lunak
10. Pengujian Perangkat Lunak

3.2 Analisis Kebutuhan

Analisis kebutuhan atau tahapan mendengarkan pelanggan merupakan tahap yang paling penting dalam suatu pengembangan perangkat lunak, karena tahapan yang pertama kali harus dilakukan sebelum melakukan perancangan dan pembuatan perangkat lunak. Metode analisis yang digunakan dalam penelitian ini adalah Object-Oriented Analysis (OOA) dengan menggunakan bahasa pemodelan Unified Modelling Language (UML). Diagram UML yang digunakan dalam proses analisis kebutuhan adalah diagram dan skenario use case. Diagram use case digunakan untuk memodelkan kebutuhan fungsionalitas sistem berdasarkan sudut pandang pengguna. Sedangkan skenario use case digunakan untuk mendeskripsikan bagaimana suatu use case berjalan. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan sistem. Adapun kegiatan yang dilakukan pada tahap ini adalah:

a. Observasi

Kegiatan ini merupakan suatu metode untuk mengumpulkan data dengan mengadakan pengamatan langsung terhadap bagian pemeliharaan dan pemasaran PT.Pancamas Elite. Data-data yang diamati berupa proses bisnis, struktur organisasi, serta pengelolaan sarana dan pemantauan produk.

b. Wawancara

Kegiatan ini merupakan suatu metode untuk mengumpulkan informasi yang dilakukan dengan cara bertanya langsung terhadap responden yang dianggap layak untuk memberikan informasi mengenai sistem pemeliharaan sarana dan pemantauan produk yang dibutuhkan pengguna. Dimana pada tahap ini pengumpulan kebutuhan dilakukan oleh penulis dengan melakukan wawancara dengan staff dan koordinator bagian pemeliharaan dan

pemasaran PT.Pancamas Elite mengenai kendala dan kebutuhan yang ingin diselesaikan. Proses wawancara juga membahas mengenai gambaran umum sistem dan identifikasi aktor-aktor yang terlibat di dalam sistem. Kegiatan ini bertujuan untuk melakukan analisis kebutuhan dan batasan kegiatan.

3.3 Fase Perancangan

Tahap perancangan dilakukan setelah tahap analisis kebutuhan. Perancangan merupakan proses untuk memperoleh desain sistem yang menunjukkan alur kerja sistem pemeliharaan sarana dan pemantauan pemasaran produk dengan menggunakan kerangka kerja Laravel yang menerapkan *Object Oriented Design* (OOD). Proses yang dilakukan dalam tahap perancangan adalah pembuatan desain basis data, desain antarmuka sistem, memodelkan objek-objek yang terlibat dengan sistem beserta struktur sistem dalam penyelesaian masalah beserta relasinya ke dalam use case serta class diagram, menggambarkan bagaimana beberapa objek-objek tersebut saling berinteraksi menggunakan sequence diagram, activity diagram menggambarkan aliran kerja atau aktivitas proses bisnis yang ada pada perangkat lunak dan perancangan antarmuka yang akan dibuat sesuai dengan kebutuhan pengguna.

3.4 Implementasi

Proses yang dilakukan pada tahapan implementasi adalah menterjemahkan proses analisis kebutuhan dan proses perancangan kedalam suatu bahasa pemrograman. Tahap ini, kode-kode program yang dihasilkan masih pada tahap modul-modul. Diakhir tahap ini, tiap modul di testing tanpa diintegrasikan. Tahap selanjutnya setelah implementasi adalah pengujian atau *testing*. Pada tahap ini unit program diintegrasikan dan diuji menjadi sistem yang lengkap untuk meyakinkan bahwa persyaratan perangkat lunak telah dipenuhi. Setelah uji coba, sistem disampaikan ke konsumen.

Implementasi perangkat lunak mengacu pada perancangan perangkat lunak. Implementasi perangkat lunak dimulai dengan implementasi basis data MySQL dengan software XAMPP 1.8.0. Selanjutnya implementasi kode program menggunakan Kerangka kerja *Laravel*. Implementasi dilakukan dengan menggunakan bahasa pemrograman PHP dan *JavaScript* dengan menggunakan arsitektur *Model-View-Controller* (MVC). Implementasi antarmuka berdasarkan perancangan yang telah dilakukan menggunakan HTML5 dan CSS3.

3.5 Pengujian dan Analisis

Pengujian sistem pada penelitian ini ditujukan untuk dapat menunjukkan bahwa sistem telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan sistem. Pengujian dilakukan dengan tiga metode yaitu pengujian *whitebox*, *blackbox* dan pengujian *compatibility*.

1. Pengujian *whitebox* dilakukan dengan menggunakan teknik *basis path testing*. Proses pengujian basis path yaitu dengan menentukan tiga unit yang akan diujikan sesuai dengan priorotas, kemudian mengeksekusi proses

pengujian *basis path*. Setelah itu, berdasarkan *flowgraph* yang telah dibuat akan ditentukan kompleksitas siklomatik serta himpunan basis dari jalur-jalur independen secara linier.

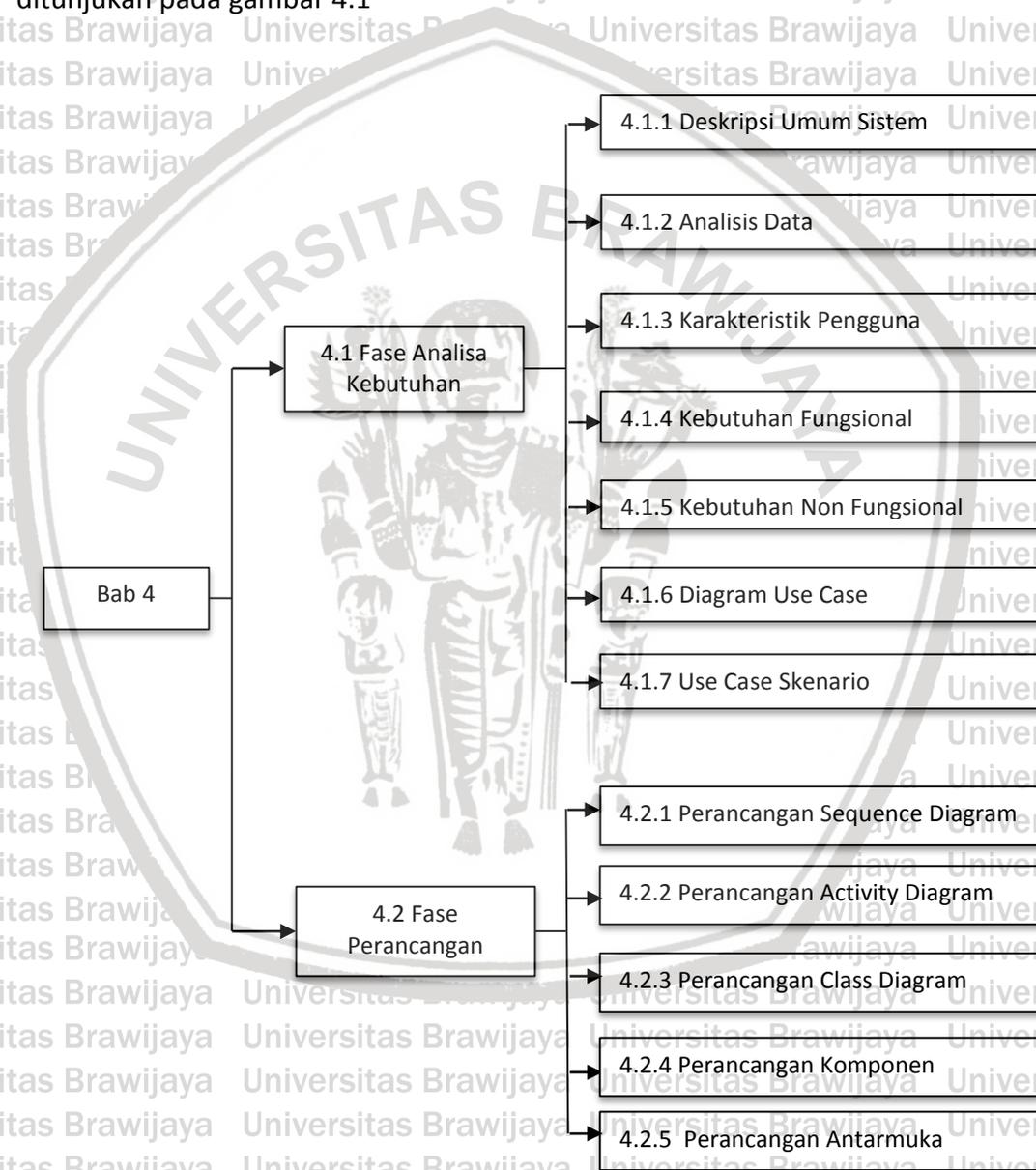
2. Pengujian *Blackbox* dilakukan melalui tahapan pengujian validasi untuk menguji fungsional kerja dari sistem. Skenario pada pengujian ini dilakukan dengan menjalankan sistem sesuai dengan fungsi-fungsi internal yang akan dijabarkan pada bab analisis dan perancangan sub bab use case scenario dengan mencoba fitur pada sistem minimal 3 kali percobaan. Jika dari minimal 3 kali percobaan sistem berjalan tanpa ada kesalahan maka pengujian dianggap valid. Sistem berjalan dengan baik sesuai dengan kebutuhan jika minimal 80% dari pengujian menunjukkan hasil yang valid. Setelah itu dilakukan pencatatan dalam bentuk tabel apakah hasil kebutuhan fungsional berhasil terpenuhi oleh sistem (valid) atau tidak (tidak valid).
3. Pengujian *compatibility* akan dilakukan pengujian untuk mengetahui kinerja sistem pada sistem operasi (OS) dan *browser* yang berbeda. Pengujian ini dilakukan dengan menggunakan daftar kebutuhan dan skenario pengujian. Setelah itu dilakukan hasil pengujian apakah sistem berjalan normal di *platform* atau *browser* lainnya.

3.6 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakannya penulisan serta memberikan pertimbangan atas pengembangan sistem informasi selanjutnya.

BAB 4 ANALISIS DAN PERANCANGAN

Bab ini membahas mengenai analisis kebutuhan dan perancangan sistem pemeliharaan sarana dan pemantauan pemasaran produk. Analisis dan perancangan aplikasi memiliki dua tahap. Tahap pertama adalah proses analisis kebutuhan perangkat lunak dilanjutkan dengan tahap kedua yaitu proses perancangan perangkat lunak. Diagram alir proses analisis dan perancangan ditunjukkan pada gambar 4.1



Gambar 4.1 Diagram Alir Perancangan Sistem

Terdapat dua tahapan besar dalam perancangan yaitu analisis kebutuhan dan perancangan sistem. Awal dari sub bab ini akan dilakukan fase analisa kebutuhan dengan cara menganalisis kebutuhan sistem yang terdiri dari delapan tahapan pengerjaan antara lain deskripsi umum sistem, analisis data, kebutuhan fungsional, kebutuhan non-fungsional, karakteristik pengguna serta pemodelan *usecase* dan *usecase* skenario. Sedangkan pada perancangan sistem terdapat empat tahap pengerjaan yaitu memodelkan *sequence diagram*, *activity diagram*, *class diagram*, perancangan komponen dan perancangan antarmuka.

4.1 Analisis Kebutuhan

Dalam analisis kebutuhan ini akan dipaparkan kebutuhan-kebutuhan dari aplikasi sistem pemeliharaan sarana dan pemantauan pemasaran produk di PT.Pancamas Elite. Untuk menganalisis kebutuhan-kebutuhan tersebut perlu dilakukan pembuatan deskripsi umum proses pemeliharaan sarana dan pemantauan pemasaran produk yang ada saat ini, identifikasi aktor, identifikasi kebutuhan serta pemodelan *use case*. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

4.1.1 Deskripsi Umum Sistem

Aplikasi pemeliharaan sarana dan pemantauan pemasaran produk adalah aplikasi yang dirancang dengan menggunakan teknologi kerangka kerja terbaru yaitu laravel. Aplikasi ini bertujuan untuk memberikan kemudahan dalam mengelola sarana prasarana beserta pemantauan pemasaran produk yang ada di PT. Pancamas Elite.

4.1.1.1 Pemeliharaan Sarana

Sistem pemeliharaan sarana terdiri dari dua bagian utama yaitu divisi pemeliharaan dan unit kerja.

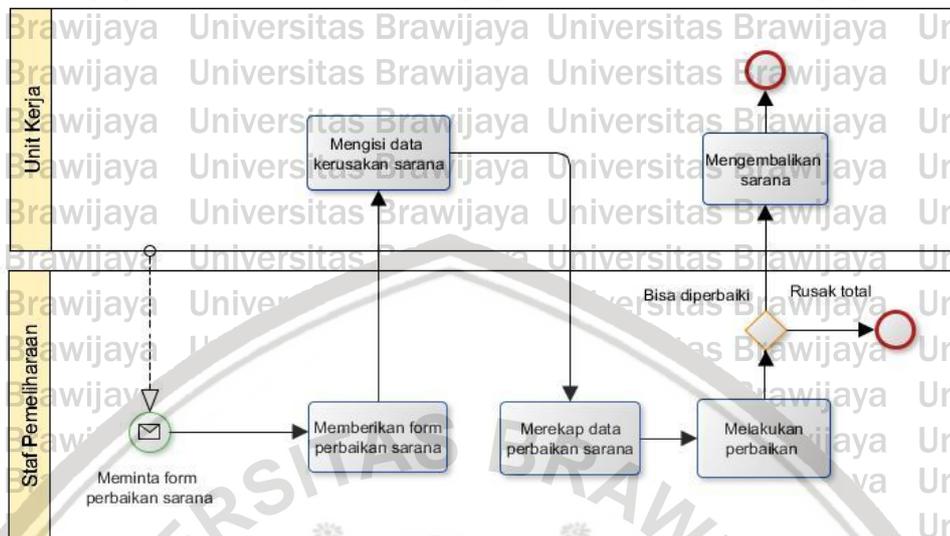
1. Subsistem divisi pemeliharaan

Divisi pemeliharaan adalah divisi PT.Pancamas Elite yang bertugas untuk melakukan proses pemeliharaan terhadap sarana produksi maupun inventaris kantor. Subsistem ini berfungsi dalam melakukan kegiatan perbaikan, perawatan dan pengelolaan sarana prasarana.

2. Subsistem unit kerja

Unit kerja adalah pegawai umum dari setiap divisi yang ada di PT.Pancamas Elite. Unit kerja bertugas untuk melakukan proses pengajuan kerusakan sarana jika ada sarana prasarana yang rusak. Subsistem ini berfungsi untuk melakukan kegiatan pelaporan sarana yang perlu dilakukan perbaikan.

Proses bisnis pemeliharaan sarana di PT.Pancamas Elite sebelum diterapkannya aplikasi penulis gambarkan dengan diagram *Business Process Model Notation* (BPMN) Gambar 4.2 berikut ini.



Gambar 4.2 Proses bisnis pemeliharaan sarana

Diagram BPMN pada Gambar 4.3 diatas menunjukkan proses bisnis sistem pemeliharaan sarana yang ada di PT.Pancamas Elite saat ini. Jika terdapat kerusakan sarana unit kerja meminta perbaikan dengan cara meminta form perbaikan kepada bagian pemeliharaan. Setelah itu staf pemeliharaan akan memberikan form yang kemudian diserahkan kepada unit kerja untuk diisi data sarana yang akan dilakukan perbaikan. Setelah itu form diserahkan kepada bagian staf pemeliharaan untuk direkap dan dilakukan perbaikan. Pihak staf pemeliharaan kemudian akan menentukan apakah sarana tersebut bisa diperbaiki atau tidak. Jika bisa diperbaiki, maka sarana akan dikembalikan setelah selesai diperbaiki kepada unit kerja yang mengajukan perbaikan.

4.1.1.2 Pemantauan Pemasaran Produk

Sistem pemasaran produk terdiri dari tiga bagian utama yaitu subsistem staf pemasaran, admin pemasaran dan gudang.

1. Subsistem divisi pemasaran

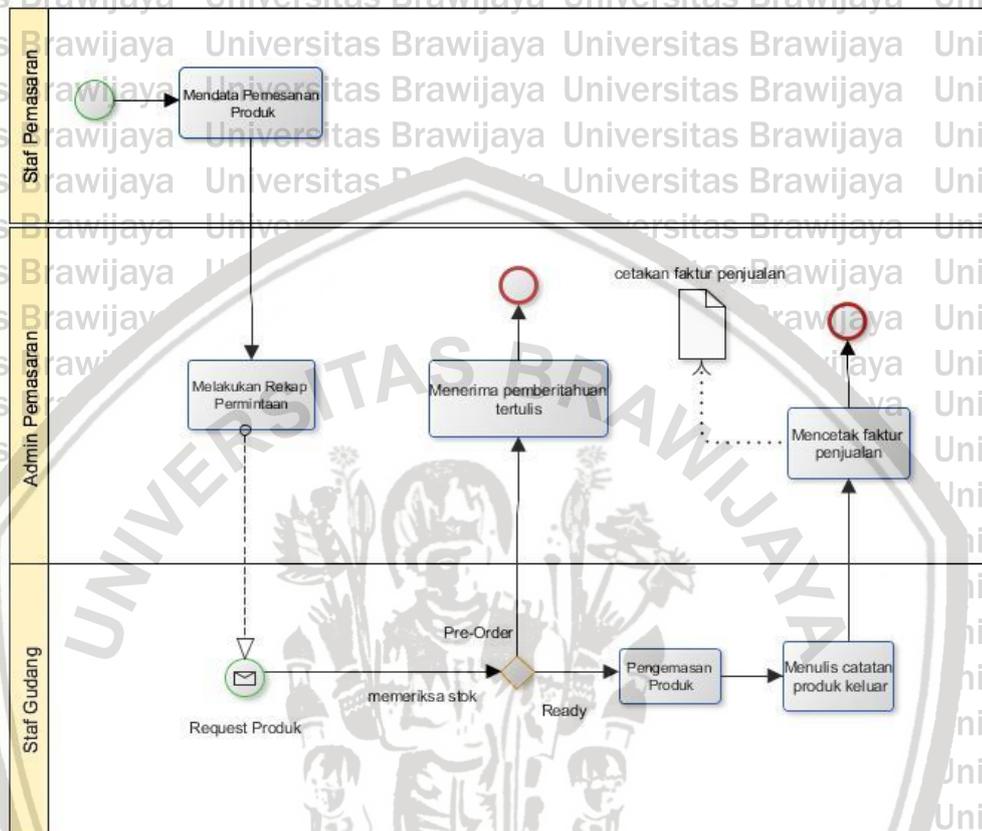
Divisi ini adalah divisi PT.Pancamas Elite yang bertugas dalam mengelola penjualan produk hasil produksi. Subsistem ini terdiri dari dua bagian yaitu subsistem untuk admin dan subsistem untuk bagian staf. Subsistem admin berfungsi untuk menerima dan menganalisis laporan penjualan. Subsistem staf pemasaran memiliki tugas untuk melakukan proses pengajuan pemasaran produk.

2. Subsistem divisi gudang

Divisi ini adalah divisi PT.Pancamas Elite yang bertugas dalam mengelola dan menyimpan produk sebelum disalurkan kepada divisi pemasaran. Subsistem

ini bertugas untuk melakukan konfirmasi pemesanan produk dan mengelola informasi data produk.

Selanjutnya, proses bisnis sistem pemesanan produk divisi pemasaran PT.Pancamas Elite sebelum diterapkannya aplikasi penulis gambarkan dengan diagram *Business Process Model Notation* (BPMN) Gambar 4.3 berikut ini.



Gambar 4.3 Proses bisnis pemesanan produk

Diagram BPMN pada Gambar 4.3 diatas menunjukkan proses bisnis sistem pemesanan produk yang ada di divisi pemasaran PT.Pancamas Elite saat ini. Staf Pemasaran mendata setiap permintaan produk oleh konsumen. Admin harus merekap setiap proses permintaan produk di catatan yang berbeda. Ketika ada penambahan produk atau perubahan informasi produk, admin harus merekap ulang setiap perubahan yang terjadi. Setelah proses rekap produk selesai maka admin akan melakukan *request* produk kepada staf gudang, kemudian staf gudang akan memeriksa stok produk. Jika stok produk *ready*, maka akan langsung dilakukan pengemasan dan faktur penjualan dikirim ke staf pemasaran.

4.1.2 Analisis Data

Tahap ini dilakukan untuk mengetahui masalah yang perlu dipecahkan dan mengenali calon pengguna perangkat lunak. Teknik yang digunakan dalam elisitasi kebutuhan adalah wawancara dan studi literatur. Studi literatur yang dilakukan mengacu kepada teori yang telah dipaparkan pada bagian kajian pustaka. Wawancara dilakukan untuk mengetahui masalah yang terjadi dalam sistem saat

ini. Berdasarkan hasil dari teknik wawancara dan studi literatur maka ditemukan kebutuhan yang harus ada dalam sistem yang akan dibangun antara lain:

1. Area Fungsional Pemeliharaan

- a. Area fungsional ini harus memiliki fungsi untuk mencatat setiap proses perbaikan yang dilakukan oleh staf pemeliharaan.
- b. Area fungsional ini harus memiliki fungsi yang dapat menerima setiap *request* pengajuan perbaikan sarana yang dilakukan oleh unit kerja serta memberikan notifikasi status perbaikan sarana.
- c. Area fungsional ini harus memiliki fungsi untuk memberikan notifikasi kepada staf pemeliharaan jika proses pemeliharaan atau perawatan rutin belum dilaksanakan sesuai jadwal.
- d. Area fungsional ini harus memiliki fungsi untuk mencari dan mencetak *history* perbaikan dan perawatan sesuai rentang waktu yang diinputkan user.

2. Area Fungsional Unit kerja

- a. Area fungsional ini harus memiliki fungsi untuk melakukan pengajuan permintaan perbaikan sarana yang meliputi nama sarana, ruangan, kode inventaris serta melihat status pengajuan perbaikan sarana dengan mudah.

3. Area Fungsional Pemasaran

- a. Area fungsional ini harus memiliki fungsi untuk melakukan pengajuan pemesanan produk serta melihat status konfirmasi pemesanan produk yang dilakukan oleh staf pemasaran dan admin pemasaran.
- b. Area fungsional ini harus menyediakan fungsi untuk mengelola dan mencetak *invoice* (faktur penjualan) setelah proses pemesanan produk selesai dilakukan oleh staf pemasaran.
- c. Area fungsional ini harus menyediakan fungsi untuk melihat laporan penjualan per-periode tertentu.

4. Area fungsional gudang

- a. Area fungsional ini harus memiliki fungsi untuk menampilkan daftar produk disertai jumlah total produk, nama serta harga produk.
- b. Area fungsional ini harus menyediakan fitur untuk memberikan notifikasi kepada admin pemasaran status pemesanan produk.
- c. Area fungsional ini harus memiliki fungsi untuk mengelola ketersediaan produk.

4.1.3 Karakteristik Pengguna

Pada tahap ini bertujuan untuk melakukan identifikasi terhadap pengguna yang berhubungan dengan sistem dan aplikasi yang dibuat. Penjelasan masing-masing pengguna dilihat pada tabel 4.3

Tabel 4.1 Karakteristik Pengguna

Aktor	Deskripsi	Hak Akses ke Aplikasi
Admin Pemasaran	Admin Pemasaran berperan sebagai pemantau keluar masuk penjualan produk dan memberikan konfirmasi dalam pengajuan pemesanan atau <i>order</i> . Mempunyai hak akses pada subsistem admin pemasaran.	KF_01, KF_02, KF_25, KF_27, KF_29, KF_30, KF_33
Staf Gudang	Staf Gudang merupakan pengguna yang bertugas untuk mengelola data produk, memantau ketersediaan produk yang akan disalurkan kepada bagian pemasaran. Mempunyai hak akses pada subsistem gudang.	KF_01, KF_02, KF_21, KF_22, KF_23, KF_31, KF_32
Staf Pemeliharaan	Staf Pemeliharaan merupakan pengguna yang melakukan proses kerja pemeliharaan sarana berupa perawatan rutin, mengelola pengajuan perbaikan dari <i>request</i> yg dikirim oleh unit kerja. Mempunyai hak akses pada subsistem staf pemeliharaan	KF_01, KF_02, KF_03, KF_04, KF_06, KF_07, KF_08, KF_09, KF_10, KF_11, KF_12, KF_13, KF_14, KF_15, KF_16, KF_17, KF_18, KF_19,
Staf Pemasaran	Staf Pemasaran merupakan pengguna yang melakukan proses kerja berupa penjualan produk kepada konsumen dan melakukan pemesanan produk kepada bagian gudang. Mempunyai hak akses pada subsistem staf pemasaran.	KF_01, KF_02, KF_24, KF_26, KF_27, KF_28, KF_33
Unit kerja	Unit kerja adalah karyawan umum yang dapat melakukan permintaan perbaikan sarana jika terdapat sarana prasarana yang membutuhkan perbaikan. Mempunyai hak akses pada subsistem unit kerja.	KF_01, KF_02, KF_03, KF_05

4.1.4 Kebutuhan Fungsional

Tahap ini merupakan kebutuhan fungsional yang didapat dari hasil wawancara dengan pengguna. Prioritas pada spesifikasi kebutuhan fungsional didapat dari hasil wawancara dengan mengetahui seberapa penting atau sering dari proses bisnis pemeliharaan dan pemesanan produk dan seberapa jarang proses bisnis ini digunakan. Prioritas kebutuhan fungsional digunakan untuk proses implementasi dalam sistem yang diprioritaskan dan sebagai acuan pengujian *whitebox* dengan *basis path* yang memiliki prioritas tinggi. Kebutuhan fungsional yang nantinya akan disediakan oleh sistem informasi ini yang dijelaskan pada tabel 4.1

Tabel 4.2 Spesifikasi Kebutuhan Fungsional (KF)

Kode	Fungsi	Deskripsi	Prioritas
KF_01	Login	Sistem mampu menyediakan fungsi login dan memberi batasan hak akses	high
KF_02	Mengubah <i>profile</i> pengguna	Sistem harus mampu mengubah informasi <i>profile</i> pengguna	low
KF_03	Melaporkan Kerusakan Sarana	Sistem harus mampu menyediakan fungsi untuk melaporkan kerusakan sarana	high
KF_04	Pemberitahuan Kerusakan	Sistem harus mampu menampilkan pemberitahuan kerusakan sarana	high
KF_05	Melihat Riwayat Pengajuan Kerusakan	Sistem harus mampu menampilkan riwayat kerusakan yang telah atau dalam proses perbaikan	medium
KF_06	Melihat Daftar Sarana	Sistem harus mampu menampilkan daftar Sarana Prasarana di PT.Pancamas Elite	medium
KF_07	Menambah Sarana dan Ruang	Sistem harus mampu menambah sarana dan ruang	medium
KF_08	Mengubah Informasi Sarana	Sistem harus mampu mengubah informasi sarana prasarana	medium
KF_09	Menghapus Sarana	Sistem harus mampu menghapus daftar sarana prasarana	low
KF_10	Melihat Ruangan	Sistem harus mampu melihat daftar ruangan di PT.Pancamas Elite	medium
KF_11	Mengubah Informasi Ruangan	Sistem harus mampu mengubah informasi ruangan di PT.Pancamas Elite	medium

KF_12	Menghapus Ruangan	Sistem harus mampu menghapus ruangan	low
KF_13	Melihat Status Terakhir Sarana	Sistem harus mampu menampilkan status terakhir sarana prasarana yang diperbaiki	medium
KF_14	Konfirmasi Perbaikan	Sistem harus mampu melakukan konfirmasi status sarana prasarana yang sedang ditangani	high
KF_15	Menerima Status Perbaikan	Sistem harus mampu menerima status sarana yang sedang dalam proses pengajuan perbaikan	high
KF_16	Melihat Laporan Perbaikan dan Perawatan	Sistem harus mampu menampilkan riwayat perbaikan sarana yang telah ditangani	medium
KF_17	Melihat Pemeliharaan Rutin	Sistem mampu menampilkan daftar sarana prasarana yang masuk perawatan rutin	high
KF_18	Konfirmasi Perawatan Rutin	Sistem harus mampu memberi konfirmasi status sarana prasarana yang telah dilakukan pengecekan rutin	high
KF_19	Pemberitahuan jadwal pemeliharaan	Sistem harus mampu memberi pemberitahuan daftar sarana prasarana yang belum dilakukan pengecekan rutin	high
KF_20	Menampilkan daftar produk	Sistem harus mampu menampilkan daftar produk yang diproduksi di PT.Pancamas Elite	medium
KF_21	Menambah produk	Sistem harus mampu menambah produk	medium
KF_22	Mengubah Informasi Produk	Sistem harus mampu mengubah informasi produk	medium
KF_23	Menghapus Produk	Sistem harus mampu menghapus produk	low
KF_24	Menambah Pemesanan Produk	Sistem harus mampu melakukan pemesanan produk	high
KF_25	Validasi Pemesanan	Sistem harus mampu memvalidasi pemesanan produk	high

KF_26	Melihat Pemesanan Sementara	Sistem harus mampu melihat pemesanan sementara	medium
KF_27	Melihat riwayat Pemesanan	Sistem harus mampu melihat riwayat pemesanan produk	medium
KF_28	Melihat status pengajuan pemesanan	Sistem harus mampu menampilkan status pengajuan pemesanan produk	high
KF_29	Pemberitahuan Pemesanan baru	Sistem harus mampu menampilkan pemberitahuan pemesanan produk baru	high
KF_30	Melihat Status stok produk	Sistem harus mampu menampilkan status ketersediaan produk	high
KF_31	Melihat informasi produk	Sistem harus mampu menampilkan informasi produk	high
KF_32	Konfirmasi Stok Produk	Sistem harus mampu melakukan konfirmasi stok produk sesuai jumlah ketersediaan di gudang	high
KF_33	Mencetak Faktur Penjualan (<i>Invoice</i>)	Sistem harus mampu mencetak faktur penjualan (<i>invoice</i>)	high

4.1.5 Kebutuhan Non-Fungsional

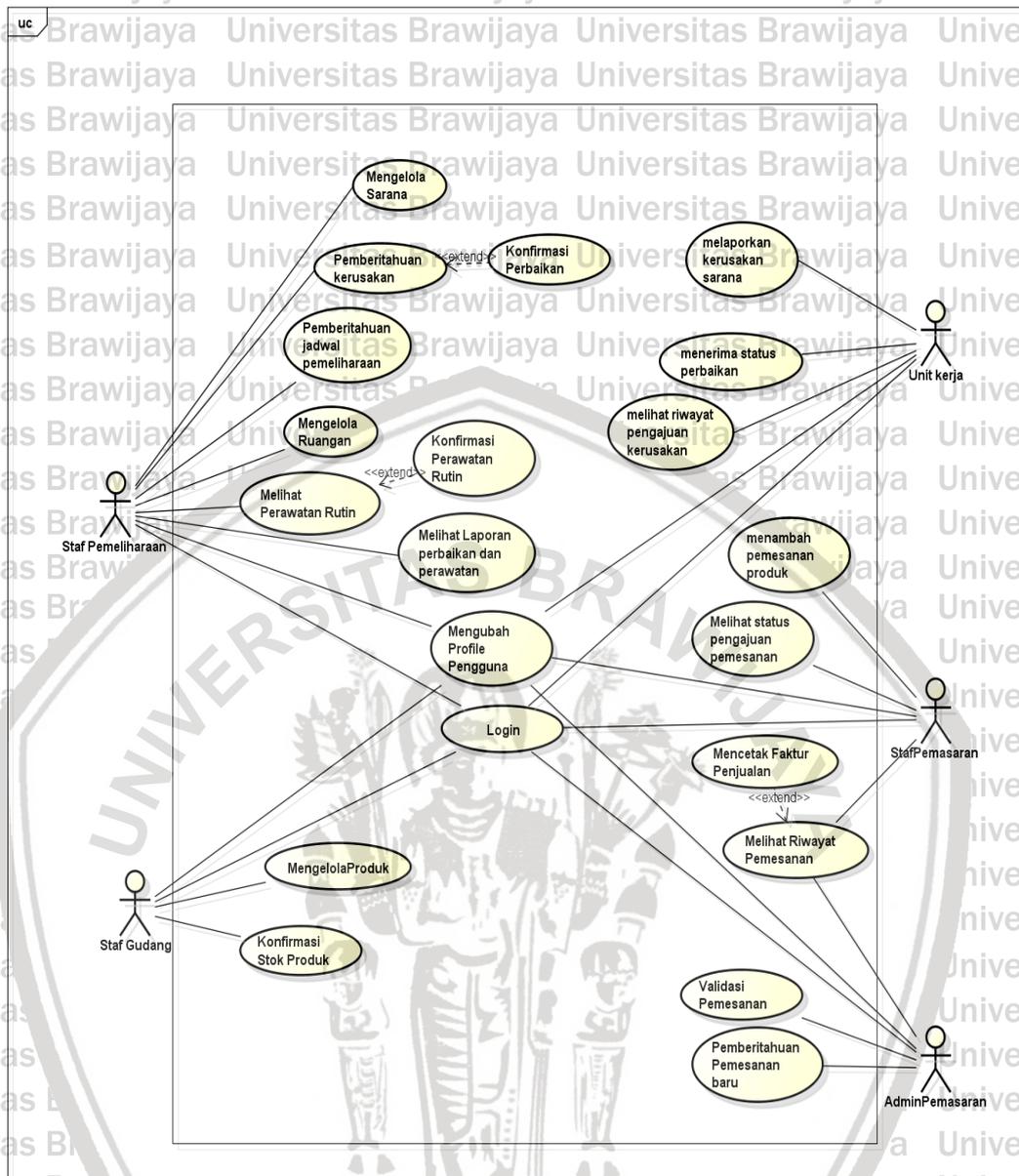
Kebutuhan *non fungsional* adalah kebutuhan diluar sistem yang menunjang perangkat lunak, berikut daftar kebutuhan non fungsional yang dijelaskan pada tabel 4.2

Tabel 4.3 Kebutuhan non fungsional

No	Parameter	Deskripsi Kebutuhan
1	<i>Compatibility</i>	Sistem dapat diakses melalui lingkungan yang berbeda seperti <i>browser</i> atau sistem operasi lain.

4.1.6 Diagram Use Case

Pemodelan diagram *use case* sistem diperoleh dari kebutuhan fungsional yang digunakan untuk menggambarkan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Diagram *use case* sistem pemeliharaan sarana dan pemantauan pemasaran produk ditunjukkan pada gambar 4.4.



Gambar 4.4 Diagram Use Case

4.1.7 Use Case Skenario

Setiap use case di atas akan dijelaskan lebih detail pada bagian skenario use case. Tabel skenario use case terdiri dari nomor Kebutuhan Fungsional (KF), nama use case tersebut, aktor yang menggunakannya, tujuan dan deskripsi use case, beserta skenario utama dan skenario alternatif yang dilakukan ketika menggunakan use case tersebut.

1. Skenario Use Case Login

Skenario use case login merupakan penjelasan lebih rinci dari use case login yang dilakukan oleh aktor. Tabel 4.5 menunjukkan skenario use case login.

Tabel 4.4 Skenario Login

KF_01 : Login	
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana sistem melakukan proses login ke dalam sistem
Aktor	Unit kerja, Staf Pemeliharaan, Staf Pemasaran, Admin Pemasaran, Gudang
Skenario Utama	
Kondisi Awal	Aktor telah masuk ke halaman login
Aksi Aktor	Reaksi Sistem
1. Aktor masuk ke halaman login	2. Sistem menampilkan form <i>username</i> dan <i>password</i>
3. Aktor mengisi form <i>username</i> dan <i>password</i> kemudian menekan tombol login	4. Sistem menampilkan halaman utama masing-masing user
Skenario Alternatif : Jika input ada yang tidak valid	
1. Aktor salah mengisi <i>username</i> atau <i>password</i>	3. Sistem menampilkan pesan " <i>These credentials do not match our records</i> "
2. Jika aktor tidak mengisi salah satu atau semua kolom input	4. Sistem menampilkan notifikasi " <i>please fill out this field</i> "
Kondisi Akhir	Aktor masuk ke halaman utama sistem

2. Skenario *Use Case* Mengubah *profile* pengguna

Skenario *use case* mengubah *profile* pengguna merupakan penjelasan lebih rinci dari *use case* mengubah *profile* pengguna yang dilakukan oleh aktor. Tabel 4.6 menunjukkan skenario *use case* mengubah *profile* pengguna.

Tabel 4.5 Skenario Mengubah *profile* pengguna

KF_02 : Mengubah <i>profile</i> pengguna	
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana aktor melakukan proses untuk mengubah <i>profile</i>
Aktor	Unit kerja, Staf Pemeliharaan, Staf Pemasaran, Admin Pemasaran, Gudang
Skenario Utama	

Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor menekan icon nama di layar dan memilih submenu <i>change profile</i>	2. Sistem menampilkan halaman <i>Edit profile</i>
3. Aktor mengisi kolom informasi yang akan diedit dan menekan tombol save	4. Sistem menyimpan perubahan informasi profil pengguna
Skenario Alternatif : Jika input ada yang tidak valid	
1. Aktor tidak mengisi salah satu <i>field</i> pada halaman form	2. Sistem menampilkan pesan ' <i>please fill out this field</i> '
Kondisi Akhir	Sistem mengubah informasi profil pengguna ke database

3. Skenario Use Case Melaporkan kerusakan sarana

Skenario use case melaporkan kerusakan sarana merupakan penjelasan lebih rinci dari use case melaporkan kerusakan sarana yang dilakukan oleh unit kerja. Tabel 4.7 menunjukkan skenario use case melaporkan kerusakan sarana.

Tabel 4.6 Skenario Melaporkan kerusakan

KF_03 : Melaporkan Kerusakan Sarana	
Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan proses permintaan perbaikan sarana prasarana
Aktor	Unit kerja
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor menekan menu Laporkan Kerusakan pada sistem	2. Sistem menampilkan halaman laporan kerusakan beserta formnya.
3. Aktor mengisi form untuk mengisi informasi kerusakan dan menekan tombol Submit	4. Sistem memproses data dan mengirimkan notifikasi ke staf bagian pemeliharaan
Skenario Alternatif : Jika ada input yang tidak valid	

1. Aktor tidak mengisi salah satu <i>field</i> pada form laporkan kerusakan	2. Sistem menampilkan pesan "kolom harus diisi"
Kondisi Akhir	Permintaan perbaikan dikirim ke subsistem staf pemeliharaan untuk kemudian dilakukan tindakan.

4. Skenario Use Case Pemberitahuan Kerusakan

Skenario use case pemberitahuan kerusakan merupakan penjelasan lebih rinci dari use case pemberitahuan kerusakan oleh staf pemeliharaan. Tabel 4.8 menunjukkan skenario use case pemberitahuan kerusakan.

Tabel 4.7 Skenario Pemberitahuan Kerusakan

KF_04 : Pemberitahuan Kerusakan	
Deskripsi	Use case ini menjelaskan bagaimana sistem memberikan notifikasi perbaikan
Aktor	Staf Pemeliharaan
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor masuk ke dashboard dan melihat menu notifikasi	2. Sistem menampilkan daftar permintaan perbaikan sarana
Skenario Alternatif : Jika Pengajuan Perbaikan Belum ada	
1. Aktor masuk ke halaman dashboard dan melihat menu notifikasi	2. Sistem menampilkan halaman pemberitahuan kosong
Kondisi Akhir	Sistem menampilkan pesan jika ada permintaan perbaikan sarana

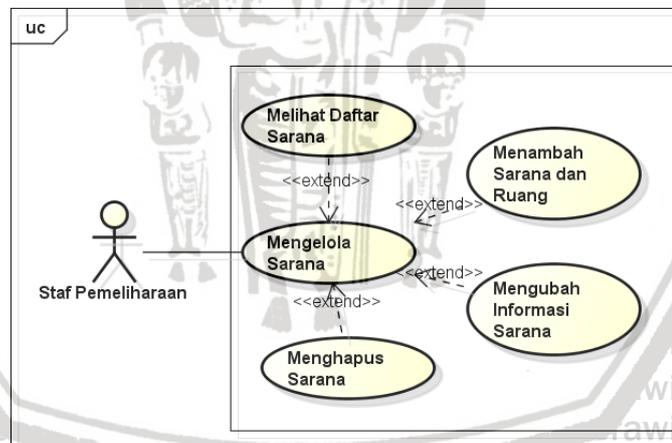
5. Skenario Use Case Melihat Riwayat Pengajuan Kerusakan

Skenario use case melihat riwayat pengajuan kerusakan merupakan penjelasan lebih rinci dari use case melihat riwayat pengajuan kerusakan yang dilakukan oleh unit kerja. Tabel 4.9 menunjukkan skenario melihat riwayat pengajuan kerusakan.

Tabel 4.8 Skenario Melihat Riwayat Pengajuan Kerusakan

KF_05 : Melihat Riwayat Pengajuan Kerusakan	
Deskripsi	Use case ini menjelaskan bagaimana sistem menampilkan halaman riwayat pengajuan kerusakan

Aktor	Unit kerja
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> Aktor menekan menu Riwayat Kerusakan pada sistem Aktor menekan memilih kolom tanggal riwayat pada sistem dan menekan tombol <i>filter</i> Aktor menekan memilih kolom tanggal riwayat pada sistem dan menekan tombol <i>clear filter</i> 	<ol style="list-style-type: none"> Sistem menampilkan halaman laporan riwayat kerusakan Sistem menampilkan riwayat kerusakan berdasarkan tanggal yang dipilih user Sistem menghapus riwayat kerusakan berdasarkan tanggal yang dipilih user
Skenario Alternatif 1: Jika aktor tidak mengisi tanggal	
1. Aktor menekan tombol <i>filter</i> tanpa memilih tanggal	2. Sistem menampilkan halaman riwayat kerusakan sesuai tanggal <i>default</i> .
Kondisi Akhir	Sistem menampilkan halaman riwayat kerusakan



Gambar 4.5 Model Use Case Mengelola Sarana

6. Skenario Use Case Melihat Daftar Sarana

Skenario use case melihat daftar sarana merupakan penjelasan lebih rinci dari use case melihat daftar sarana yang dilakukan oleh staf pemeliharaan. Tabel 4.10 menunjukkan skenario use case melihat daftar sarana.

Tabel 4.9 Skenario Melihat Daftar Sarana

KF_06 : Melihat Daftar Sarana	
Deskripsi	Use case ini menjelaskan bagaimana sistem menampilkan halaman daftar sarana
Aktor	Staf Pemeliharaan
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor menekan menu Daftar Jenis Sarana pada sistem	2. Sistem menampilkan halaman daftar sarana
Skenario Alternatif : Jika Daftar Sarana tidak tersedia	
1. Aktor menekan menu Daftar Jenis Sarana pada sistem	2. Sistem menampilkan pesan 'daftar sarana tidak tersedia'
Kondisi Akhir	Sistem menampilkan halaman daftar sarana

7. Skenario Use Case Menambah Sarana dan Ruang

Skenario use case menambah sarana dan ruang merupakan penjelasan lebih rinci dari use case menambah sarana dan ruang yang dilakukan oleh staf pemeliharaan.

Tabel 4.11 menunjukkan skenario use case menambah sarana dan ruang.

Tabel 4.10 Skenario Menambah Sarana dan Ruang

KF_07 : Menambah Sarana dan Ruang	
Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan proses menambah sarana dan ruang
Aktor	Staf Pemeliharaan
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem

<ol style="list-style-type: none"> Aktor menekan menu Sarana dan Ruang Aktor menekan tombol tambah sarana dan ruang Aktor mengisi form tambah sarana ruang kemudian menekan tombol create 	<ol style="list-style-type: none"> Sistem menampilkan halaman sarana dan ruang Sistem menampilkan halaman form tambah sarana dan ruang Sistem menyimpan data sarana dan ruang baru kedalam database
Skenario Alternatif 1: Jika input ada yang tidak valid	
<ol style="list-style-type: none"> Aktor tidak mengisi salah satu <i>field</i> pada halaman form 	<ol style="list-style-type: none"> Sistem menampilkan pesan "<i>please fill out this field</i>"
Kondisi Akhir	Sistem menyimpan data sarana dan ruang baru kedalam database

8. Skenario Use Case Mengubah Informasi Sarana

Skenario use case mengubah informasi sarana merupakan penjelasan lebih rinci dari use case mengubah informasi sarana yang dilakukan oleh Staf Pemeliharaan.

Tabel 4.12 menunjukkan skenario use case mengubah informasi sarana.

Tabel 4.11 Skenario Mengubah Informasi Sarana

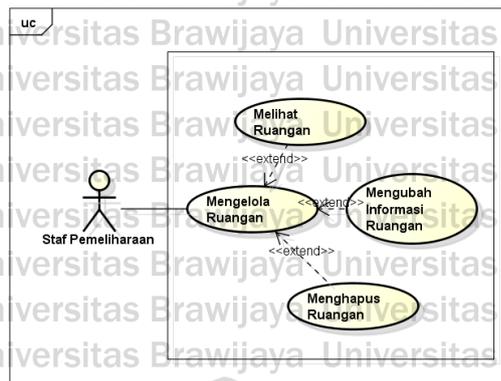
KF_08 : Mengubah Informasi Sarana	
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana sistem melakukan proses mengubah informasi sarana
Aktor	Staf Pemeliharaan
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> Aktor menekan menu daftar jenis sarana Aktor memilih menu <i>Edit</i> dan menekan tombol <i>create</i> 	<ol style="list-style-type: none"> Sistem menampilkan halaman daftar sarana Sistem menampilkan form untuk mengubah data
Skenario Alternatif : Jika input ada yang tidak valid	
<ol style="list-style-type: none"> Aktor tidak mengisi salah satu <i>field</i> pada halaman form 	<ol style="list-style-type: none"> Sistem menampilkan pesan "<i>please fill out this field</i>"
Kondisi Akhir	Sistem berhasil mengubah informasi sarana

9. Skenario Use Case Menghapus Sarana

Skenario use case menghapus sarana merupakan penjelasan lebih rinci dari use case menghapus sarana yang dilakukan oleh Staf Pemeliharaan. Tabel 4.13 menunjukkan skenario use case menghapus sarana

Tabel 4.12 Skenario Menghapus Sarana

KF_09 : Menghapus Sarana	
Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan proses menghapus sarana
Aktor	Staf Pemeliharaan
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Aktor menekan menu data sarana 2. Aktor memilih sarana yang akan dihapus. Klik <i>icon delete</i> 	<ol style="list-style-type: none"> 3. Sistem menampilkan halaman data sarana 4. Sistem menampilkan peringatan klik OK 5. Sistem menghapus data
Skenario Alternatif 1 : Jika aktor membatalkan aksi	
<ol style="list-style-type: none"> 1. Aktor menekan menu data sarana 2. Aktor memilih sarana yang akan dihapus. Klik <i>icon delete</i> 	<ol style="list-style-type: none"> 3. Sistem menampilkan halaman data sarana 4. Sistem menampilkan peringatan klik <i>Cancel</i> 5. Sistem tidak melakukan aksi apapun
Kondisi Akhir	Sistem berhasil menghapus data sarana dari database



Gambar 4.6 Use Case Mengelola Ruangan

10. Skenario Use Case Melihat Ruangan

Skenario use case melihat ruangan merupakan penjelasan lebih rinci dari use case melihat ruangan yang dilakukan oleh staf pemeliharaan. Tabel 4.14 menunjukkan skenario use case melihat ruangan.

Tabel 4.13 Skenario Melihat Ruangan

KF_10 : Melihat Ruangan	
Deskripsi	Use case ini menjelaskan bagaimana sistem menampilkan halaman daftar ruangan
Aktor	Staf Pemeliharaan
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor menekan menu Daftar Ruangan pada sistem	2. Sistem menampilkan halaman daftar ruangan
Skenario Alternatif 1 : Jika daftar ruangan tidak tersedia	
1. Aktor menekan menu Daftar Ruangan pada sistem	2. Sistem menampilkan pesan "daftar ruangan tidak tersedia"
Kondisi Akhir	Sistem menampilkan halaman daftar ruangan

11. Skenario Use Case Mengubah Informasi Ruangan

Skenario use case mengubah informasi ruangan merupakan penjelasan lebih rinci dari use case mengubah informasi ruangan yang dilakukan oleh staf pemeliharaan. Tabel 4.15 menunjukkan skenario use case mengubah informasi ruangan.

Tabel 4.14 Skenario Mengubah Informasi Ruangan

KF_11 : Mengubah Informasi Ruangan	
Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan proses mengubah informasi ruangan
Aktor	Staf Pemeliharaan
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor menekan menu daftar ruangan	2. Sistem menampilkan halaman halaman daftar ruangan
3. Aktor memilih menu <i>Edit</i> dan menekan tombol save	4. Sistem mengubah informasi ruangan
Skenario Alternatif 1 : Jika ada <i>input</i> yang tidak valid	
1. Aktor tidak mengisi salah satu <i>filed</i> pada halaman form	2. Sistem menampilkan pesan " <i>please fill out this field</i> "
Kondisi Akhir	Sistem mengubah informasi ruangan

12. Skenario Use Case Menghapus Ruangan

Skenario use case menghapus ruangan merupakan penjelasan lebih rinci dari use case menghapus ruangan yang dilakukan oleh staf pemeliharaan. Tabel 4.16 menunjukkan skenario use case menghapus ruangan.

Tabel 4.15 Skenario Menghapus Ruangan

KF_12 : Menghapus Ruangan	
Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan menghapus ruangan
Aktor	Staf Pemeliharaan
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem

1. Aktor menekan menu daftar ruangan 2. Aktor memilih ruangan yang akan dihapus. Klik <i>icon delete</i>	3. Sistem menampilkan halaman daftar ruangan 4. Sistem menampilkan peringatan klik OK 5. Sistem menghapus data
Skenario Alternatif 1: Jika aktor membatalkan aksi	
1. Aktor menekan menu daftar ruangan 2. Aktor memilih sarana yang akan dihapus. Klik <i>icon delete</i>	3. Sistem menampilkan halaman data ruangan 4. Sistem menampilkan peringatan klik <i>cancel</i> 5. Sistem tidak melakukan aksi apapun
Kondisi Akhir	Sistem menghapus data ruangan dari database

13. Skenario Use Case Melihat Status Terakhir Sarana

Skenario use case melihat status terakhir sarana merupakan penjelasan lebih rinci dari use case melihat status terakhir sarana yang dilakukan oleh staff pemeliharaan. Tabel 4.17 menunjukkan skenario use case melihat status terakhir sarana.

Tabel 4.16 Skenario Melihat Status Terakhir Sarana

KF_13 : Melihat Status Terakhir Sarana	
Deskripsi	Use case ini menjelaskan bagaimana sistem menampilkan status terakhir sarana
Aktor	Staf Pemeliharaan
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor menekan menu sarana dan ruang pada sistem	2. Sistem menampilkan halaman sarana dan ruang beserta tanggal status terakhir diperbaiki
Skenario Alternatif : Jika aktor menekan tombol delete	
1. Aktor menekan menu sarana dan ruang pada sistem	2. Sistem menampilkan halaman sarana dan ruang beserta tanggal status terakhir diperbaiki

3. Aktor memilih salah satu field kolom sarana dan ruang dan menekan tombol <i>delete</i>	4 Sistem menghapus daftar sarana dan ruang yang dipilih
Kondisi Akhir	Sistem menampilkan status terakhir sarana

14. Skenario Use Case Konfirmasi Perbaikan

Skenario use case konfirmasi perbaikan merupakan penjelasan lebih rinci dari use case melakukan konfirmasi perbaikan oleh staf pemeliharaan. Tabel 4.18 menunjukkan skenario use case melakukan konfirmasi perbaikan.

Tabel 4.17 Skenario Konfirmasi Perbaikan

KF_14 : Skenario Konfirmasi Perbaikan	
Deskripsi	Use case ini menjelaskan bagaimana proses konfirmasi perbaikan
Aktor	Staf Pemeliharaan
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Aktor masuk ke menu Daftar Permintaan Perawatan 2. Aktor menekan tombol perbaikan sekarang 3. Aktor menekan tombol perbaikan selesai 	<ol style="list-style-type: none"> 4. Sistem menampilkan halaman daftar permintaan perawatan 5. Sistem mengirim status kepada unit kerja status sarana sedang diperbaiki 6. Sistem mengirim status kepada unit kerja status sarana telah selesai diperbaiki.
Skenario Alternatif : Jika aktor mengisi menu <i>filter</i>	
<ol style="list-style-type: none"> 1. Aktor masuk ke menu Daftar Permintaan Perawatan dan mengisi filter riwayat 	<ol style="list-style-type: none"> 2. Sistem menampilkan daftar permintaan perawatan sesuai tanggal yang dipilih oleh aktor
Kondisi Akhir	Sistem mengubah status pengajuan perbaikan sarana

15. Skenario Use Case Menerima Status Perbaikan

Skenario use case menerima status perbaikan merupakan penjelasan lebih rinci dari use case menerima status perbaikan dengan aktor unit kerja. Tabel 4.19 menunjukkan skenario use case menerima status perbaikan.

Tabel 4.18 Skenario Menerima status perbaikan

KF_15 : Menerima Status Perbaikan	
Deskripsi	Use case ini menjelaskan bagaimana sistem memberikan notifikasi <i>progress</i> terhadap sarana yang telah diajukan perbaikan
Aktor	Unit Kerja
Skenario Utama	
Kondisi Awal	Aktor telah login ke dalam sistem
Aksi Aktor	Reaksi Sistem
1. Aktor masuk ke halaman utama	2. Sistem menampilkan notifikasi <i>progress</i> perbaikan sarana prasarana
Skenario Alternatif : Jika aktor menekan tombol <i>filter</i>	
1. Aktor masuk ke menu Riwayat kerusakan dan mengisi <i>filter</i> riwayat	2. Sistem menampilkan daftar permintaan perawatan sesuai tanggal yang dipilih oleh aktor
Kondisi Akhir	Sistem memberikan notifikasi <i>progress</i> perbaikan sarana prasarana di halaman utama aktor

16. Skenario Use Case Melihat Laporan Perbaikan dan Perawatan

Skenario use case melihat laporan perbaikan dan perawatan merupakan penjelasan lebih rinci dari use case melihat laporan perbaikan dan perawatan yang dilakukan oleh staf pemeliharaan. Tabel 4.20 menunjukkan skenario use case melihat laporan perbaikan dan perawatan.

Tabel 4.19 Skenario Melihat Laporan Perbaikan dan Perawatan

KF_16 : Melihat Laporan Perbaikan dan Perawatan	
Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan proses melihat laporan perbaikan dan perawatan
Aktor	Staf Pemeliharaan

Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor masuk ke menu Daftar Permintaan Perawatan 2. Aktor memilih tanggal pada menu <i>filter</i> riwayat kemudian menekan tombol <i>filter</i>	3. Sistem menampilkan halaman daftar permintaan perawatan 4. Sistem menampilkan daftar perbaikan dan perawatan sesuai tanggal yang dipilih oleh user
Skenario Alternatif 1: Aktor tidak memilih tanggal	
1. Aktor tidak memilih tanggal pada menu <i>filter</i> riwayat kemudian menekan tombol <i>filter</i>	2. Sistem menampilkan daftar perbaikan dan perawatan sesuai tanggal <i>default</i> sistem
Kondisi Akhir	Sistem menampilkan halaman riwayat perbaikan dan perawatan

17. Skenario Melihat Pemeliharaan Rutin

Skenario use case melihat pemeliharaan rutin merupakan penjelasan lebih rinci dari use case melihat pemeliharaan rutin. Tabel 4.21 menunjukkan skenario use case melihat pemeliharaan rutin.

Tabel 4.20 Skenario Melihat Pemeliharaan Rutin

KF_17 : Melihat Pemeliharaan Rutin	
Deskripsi	Use case ini menjelaskan bagaimana sistem menampilkan halaman perbaikan rutin
Aktor	Staf Pemeliharaan
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor menekan menu pemeliharaan rutin	2. Sistem menampilkan halaman pemeliharaan rutin
Skenario Alternatif : Jika ada sarana yang belum diperiksa	

1. Aktor masuk ke halaman dashboard utama	2. Sistem menampilkan pesan “terdapat sarana yang belum dilakukan pengecekan rutin”
Kondisi Akhir	Sistem menampilkan halaman pemeliharaan rutin

18. Skenario Use Case Konfirmasi Perawatan Rutin

Skenario use case konfirmasi perawatan rutin merupakan penjelasan lebih rinci dari use case konfirmasi perawatan rutin yang dilakukan oleh staf pemeliharaan.

Tabel 4.22 menunjukkan skenario use case konfirmasi perawatan rutin.

Tabel 4.21 Skenario Konfirmasi Perawatan Rutin

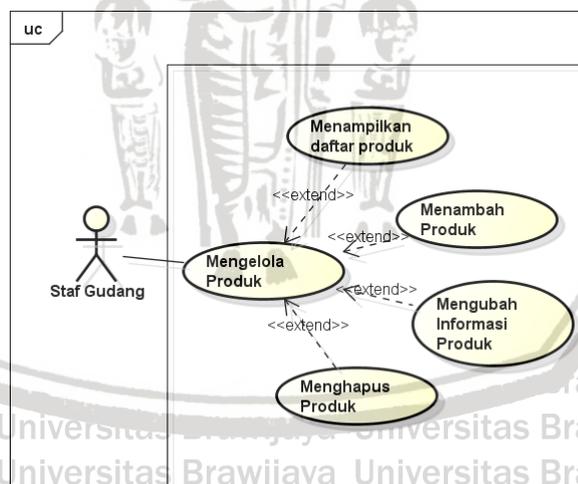
KF_18 : Konfirmasi Perawatan Rutin	
Deskripsi:	Use case ini menjelaskan bagaimana sistem melakukan proses konfirmasi perawatan rutin
Aktor	Staf Pemeliharaan
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> Aktor menekan menu perawatan rutin Aktor menekan tombol Baik Aktor menekan tombol perbaiki sekarang Aktor mengisi form kerusakan dan menekan tombol laporkan 	<ol style="list-style-type: none"> Sistem menampilkan halaman perawatan rutin Sistem menampilkan pesan bahwa sarana telah diperiksa Sistem menampilkan halaman form kerusakan Sistem menyimpan laporan kerusakan kedalam database
Skenario Alternatif : -	
Kondisi Akhir	Sistem menampilkan status perawatan rutin sarana

19. Skenario Use Case Pemberitahuan jadwal pemeliharaan

Skenario use case pemberitahuan jadwal pemeliharaan merupakan penjelasan lebih rinci dari use case pemberitahuan jadwal pemeliharaan. Tabel 4.23 menunjukkan skenario use case pemberitahuan jadwal pemeliharaan.

Tabel 4.22 Skenario Pemberitahuan Jadwal Pemeliharaan

KF_19 : Pemberitahuan jadwal pemeliharaan	
Deskripsi	Use case ini menjelaskan bagaimana sistem memberikan notifikasi pengingat jadwal kepada aktor
Aktor	Staf Pemeliharaan
Skenario Utama	
Kondisi Awal	Aktor telah login ke dalam sistem
Aksi Aktor	Reaksi Sistem
1. Aktor masuk ke halaman utama	2. Sistem menampilkan notifikasi jadwal rutin kegiatan pemeliharaan sarana prasarana
Skenario Alternatif : Jika semua sarana sudah dilakukan pengecekan	
1. Aktor masuk ke halaman utama	2. Sistem tidak menampilkan halaman notifikasi kosong
Kondisi Akhir	Sistem memberikan notifikasi jadwal rutin pemeliharaan sarana prasarana yang belum dilakukan pengecekan pada halaman utama aktor



Gambar 4.7 Use Case Mengelola Produk

20. Skenario Use Case Menampilkan daftar produk

Skenario use case menampilkan daftar produk merupakan penjelasan lebih rinci dari use case menampilkan daftar produk yang dilakukan oleh staf pemasaran.

Tabel 4.24 menunjukkan skenario use case menampilkan daftar produk.

Tabel 4.23 Skenario Menampilkan daftar produk

KF_20 : Menampilkan daftar produk	
Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan proses menampilkan daftar produk
Aktor	Staf Pemasaran
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Aktor menekan menu daftar produk 2. Aktor memilih kategori produk dan memilih produk yang dipesan 3. Aktor memilih jumlah dan menekan tombol beli 	<ol style="list-style-type: none"> 4. Sistem menampilkan halaman submenu kategori produk 5. Sistem menampilkan gambar produk dan menampilkan halaman untuk memilih jumlah 6. Data produk disimpan dalam menu pesanan sementara
Skenario Alternatif : Jika daftar produk tidak ada	
<ol style="list-style-type: none"> 1. Aktor menekan menu daftar produk 	<ol style="list-style-type: none"> 2. Sistem menampilkan pesan "daftar produk tidak tersedia"
Kondisi Akhir	Permintaan perbaikan dikirim ke subsistem staf pemeliharaan untuk kemudian dilakukan pengolahan data

21. Skenario Use Case Menambah produk

Skenario use case menambah produk merupakan penjelasan lebih rinci dari use case menambah produk yang dilakukan oleh gudang. Tabel 4.25 menunjukkan skenario use case menambah produk.

Tabel 4.24 Skenario Menambah produk

KF_21 : Menambah produk	
Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan proses menambah produk
Aktor	Gudang
Skenario Utama	

Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor menekan menu Data Produk 2. Aktor mengisi data dan menekan tombol <i>create</i>	3. Sistem menampilkan halaman data produk 4. Sistem menampilkan halaman form tambah produk dan menyimpan data ke dalam database
Skenario Alternatif : Jika <i>input</i> ada yang tidak valid	
1. Aktor tidak mengisi salah satu <i>field</i> pada halaman form	2. Sistem menampilkan pesan " <i>please fill out this field</i> "
Kondisi Akhir	sistem menambah data produk dan menyimpan kedalam database

22. Skenario *Use Case* Mengubah Informasi Produk

Skenario use case mengubah informasi produk merupakan penjelasan lebih rinci dari use case mengubah informasi produk yang dilakukan oleh gudang. Tabel 4.26 menunjukkan skenario use case mengubah informasi produk.

Tabel 4.25 Skenario Mengubah Informasi Produk

KF_22 : Mengubah Informasi Produk	
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana sistem melakukan proses mengubah informasi produk
Aktor	Gudang
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor menekan menu daftar produk 3. Aktor memilih menu <i>Edit</i> dan menekan tombol <i>save</i>	2. Sistem menampilkan halaman halaman daftar produk 4. Sistem mengubah informasi produk
Skenario Alternatif 1 : Jika <i>input</i> ada yang tidak valid	
1. Aktor mengisi menu <i>edit</i> dan tidak mengisi salah satu <i>field</i> pada halaman form	2. Sistem menampilkan pesan " <i>please fill out this field</i> "

Kondisi Akhir	Sistem mengubah informasi produk
----------------------	----------------------------------

23. Skenario *Use Case* Menghapus Produk

Skenario use case menghapus produk merupakan penjelasan lebih rinci dari use case menghapus produk yang dilakukan oleh gudang. Tabel 4.27 menunjukkan skenario use case menghapus produk.

Tabel 4.26 Skenario Menghapus Produk

KF_23 : Menghapus Produk	
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana sistem melakukan proses permintaan perbaikan sarana prasarana
Aktor	Gudang
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Aktor menekan menu daftar produk 2. Aktor memilih produk yang akan dihapus. Klik <i>icon delete</i> 	<ol style="list-style-type: none"> 3. Sistem menampilkan halaman daftar produk 4. Sistem menampilkan peringatan klik OK 5. Sistem menghapus data
Skenario Alternatif : Aktor membatalkan aksi	
<ol style="list-style-type: none"> 1. Aktor menekan menu daftar produk 2. Aktor memilih produk yang akan dihapus. Klik <i>icon delete</i> 	<ol style="list-style-type: none"> 3. Sistem menampilkan halaman daftar produk 4. Sistem menampilkan peringatan klik <i>cancel</i> 5. Sistem menghapus data
Kondisi Akhir	Sistem menghapus data produk

24. Skenario *Use Case* Menambah Pemesanan Produk

Skenario use case menambah pemesanan produk merupakan penjelasan lebih rinci dari use case menambah pemesanan produk. Tabel 4.28 menunjukkan skenario use case menambah pemesanan produk.

Tabel 4.27 Skenario Menambah Pemesanan Produk

KF_24 : Menambah Pemesanan Produk
--

Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan proses menambah pemesanan produk	
Aktor	Staf Pemasaran	
Skenario Utama		
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor	
Aksi Aktor	Reaksi Sistem	
<ol style="list-style-type: none"> 1. Aktor memilih menu daftar produk 4. Aktor memilih produk dalam daftar dan jumlah yang diinginkan dan menekan tombol 'Beli' 5. Aktor menekan tombol "Checkout" 7. Aktor menekan tombol "Create" 	<ol style="list-style-type: none"> 3. Sistem menampilkan halaman daftar produk 5. Sistem memproses data dan menampilkan halaman pesanan 6. Sistem menampilkan halaman form identitas pembeli 8. Sistem memproses pemesanan dan menyimpan kedalam database 	
Skenario Alternatif 1: Jika input ada yang tidak valid		
1. Aktor tidak mengisi salah satu <i>field</i> pada halaman form	2. Sistem menampilkan pesan "Kolom harus diisi"	
Kondisi Akhir	Pemesanan produk dikirim ke subsistem admin pemasaran untuk ditinjau	

25. Skenario Use Case Validasi Pemesanan Produk

Skenario use case validasi pemesanan produk merupakan penjelasan lebih rinci dari use case validasi pemesanan produk. Tabel 4.29 menunjukkan skenario use case validasi pemesanan produk.

Tabel 4.28 Skenario Validasi Pemesanan Produk

KF_25 : Validasi Pemesanan Produk		
Deskripsi	Use case ini menjelaskan bagaimana sistem memberikan validasi pemesanan produk	
Aktor	Admin Pemasaran	
Skenario Utama		
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor	

Aksi Aktor	Reaksi Sistem
1. Aktor masuk ke halaman pesanan baru dan menekan tombol <i>accept</i>	2. Sistem menampilkan halaman notifikasi pemesanan produk dari staf pemasaran 3. Sistem menampilkan halaman laporan penjualan dengan status produk diterima
Skenario Alternatif : Jika aktor menekan tombol <i>reject</i>	
1. Aktor masuk ke halaman pesanan baru dan menekan tombol <i>reject</i>	2. Sistem menampilkan halaman notifikasi pemesanan produk dari staf pemasaran 3. Sistem menampilkan halaman laporan penjualan dengan status produk ditolak
Kondisi Akhir	Sistem memberikan status konfirmasi permintaan produk kepada staf gudang

26. Skenario Use Case Melihat Pemesanan Sementara

Skenario Use case melihat pemesanan sementara merupakan penjelasan lebih rinci dari use case melihat pemesanan sementara yang dilakukan oleh staf pemasaran. Tabel 4.30 menunjukkan skenario use case melihat pemesanan sementara.

Tabel 4.29 Skenario Melihat Pesanan Sementara

KF_26 : Melihat Pemesanan Sementara	
Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan proses menampilkan pesanan sementara
Aktor	Staf Pemasaran
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor menekan menu Pesanan Sementara 2. Aktor menekan tombol <i>Checkout</i>	3. Sistem menampilkan halaman pesanan sementara 4. Sistem menampilkan halaman form identitas pembeli
Skenario Alternatif : Jika aktor menghapus daftar pemesanan sementara	

1. Aktor menekan menu Pesanan Sementara 3. Aktor menekan tombol <i>delete</i>	2. Sistem menampilkan halaman pesanan sementara 4. Sistem menghapus daftar produk dari halaman pesanan sementara
Kondisi Akhir	Sistem melihat pesanan sementara produk yang dipesan

27. Skenario Use Case Melihat riwayat Pemesanan

Skenario use case melihat riwayat pemesanan merupakan penjelasan lebih rinci dari use case melihat riwayat pemesanan yang dilakukan oleh staf pemasaran dan admin pemasaran. Tabel 4.31 menunjukkan skenario use case melihat riwayat pemesanan.

Tabel 4.30 Skenario Melihat riwayat Pemesanan

KF_27 : Melihat riwayat Pemesanan	
Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan proses melihat riwayat pemesanan
Aktor	Staf Pemasaran, Admin Pemasaran
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor menekan menu Laporan Penjualan	2. Sistem menampilkan halaman laporan penjualan
3. Aktor memilih tanggal pada menu <i>filter</i> riwayat dan menekan tombol <i>filter</i>	4. Sistem menampilkan halaman riwayat penjualan sesuai tanggal yang dipilih oleh user
Skenario Alternatif 1: Jika aktor tidak memilih tanggal	
1. Aktor tidak memilih tanggal pada menu <i>filter</i> riwayat dan menekan tombol <i>filter</i>	2. Sistem menampilkan halaman riwayat penjualan sesuai tanggal <i>default</i> dari sistem
Kondisi Akhir	Sistem menampilkan riwayat penjualan sesuai tanggal yang dipilih user

28. Skenario Use Case Pemberitahuan status pengajuan pemesanan

Skenario use case pemberitahuan status pengajuan pemesanan merupakan penjelasan lebih rinci dari use case pemberitahuan status pengajuan pemesanan yang dilakukan oleh staf pemasaran. Tabel 4.32 menunjukkan skenario use case pemberitahuan status pengajuan pemesanan.

Tabel 4.31 Skenario Pemberitahuan status pengajuan pemesanan

KF_28 : Pemberitahuan status pengajuan pemesanan	
Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan proses pemberitahuan status pengajuan pemesanan
Aktor	Staf Pemasaran
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor masuk ke halaman dashboard utama	2. Sistem menampilkan pesan notifikasi status pengajuan pemesanan
Skenario Alternatif : -	
Kondisi Akhir	Sistem menampilkan status pengajuan pemesanan pada halaman dashboard utama

29. Skenario Use Case Pemberitahuan Pemesanan baru

Skenario use case pemberitahuan pemesanan baru merupakan penjelasan lebih rinci dari use case pemberitahuan pemesanan baru yang dilakukan oleh admin pemasaran. Tabel 4.33 menunjukkan skenario use case pemberitahuan pemesanan baru.

Tabel 4.32 Skenario Pemberitahuan Pemesanan baru

KF_29 : Pemberitahuan Pemesanan baru	
Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan proses pemberitahuan pemesanan baru
Aktor	Admin Pemasaran
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor

Aksi Aktor	Reaksi Sistem
1. Aktor masuk ke halaman dashboard	2. Sistem menampilkan halaman notifikasi pesanan baru dari staf pemasaran.
Skenario Alternatif : Jika tidak ada pemesanan baru	
1. Aktor masuk ke halaman dashboard utama	2. Sistem menampilkan halaman notifikasi kosong
Kondisi Akhir	Sistem menampilkan halaman pemberitahuan pemesanan baru

30. Skenario Use Case Melihat Status stok produk

Skenario use case melihat status stok produk merupakan penjelasan lebih rinci dari use case melihat status stok produk yang dilakukan oleh admin pemasaran.

Tabel 4.34 menunjukkan skenario use case melihat status stok produk.

Tabel 4.33 Skenario Melihat Status stok produk

KF_30 : Melihat Status stok produk	
Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan proses melihat status stok produk
Aktor	Admin Pemasaran
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor menekan menu "Pesanan Terproses"	2. Sistem menampilkan halaman laporan penjualan beserta status ketersediaan produk dari gudang
Skenario Alternatif : -	
Kondisi Akhir	Sistem menampilkan halaman laporan penjualan beserta status ketersediaan produk.

31. Skenario Use Case Melihat Informasi Produk

Skenario use case melihat informasi produk merupakan penjelasan lebih rinci dari use case melihat informasi produk yang dilakukan oleh gudang. Tabel 4.35 menunjukkan skenario use case melihat informasi produk.

Tabel 4.34 Skenario Melihat informasi produk

KF_31 : Melihat informasi produk	
Deskripsi	Use case ini menjelaskan bagaimana sistem melakukan proses melihat informasi produk
Aktor	Gudang
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman utama aktor
Aksi Aktor	Reaksi Sistem
1. Aktor menekan menu "data produk"	2. Sistem menampilkan halaman data produk beserta status ketersediaan produk
Skenario Alternatif : Jika jumlah produk kosong	
1. Aktor menekan menu "data produk"	2. Sistem menampilkan pesan status produk "Belum Ready"
Kondisi Akhir	Sistem menampilkan status ketersediaan produk

32. Skenario Use Case Konfirmasi Stok Produk

Skenario use case konfirmasi stok produk merupakan penjelasan lebih rinci dari use case konfirmasi stok produk. Tabel 4.36 menunjukkan skenario use case konfirmasi stok produk.

Tabel 4.35 Skenario Konfirmasi Stok Produk

KF_32 : Konfirmasi Stok Produk	
Deskripsi	Use case ini menjelaskan bagaimana proses konfirmasi stok produk oleh sistem
Aktor	Gudang
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman Konfirmasi Permintaan
Aksi Aktor	Reaksi Sistem

1. Aktor masuk ke menu data produk	3. Sistem menampilkan halaman daftar produk beserta status ketersediaan produk yang dipesan
2. Aktor menekan tombol <i>ready</i>	4. Sistem memberikan pesan kepada admin pemasaran bahwa status pesanan <i>ready</i>
	5. Sistem mengirimkan faktur atau <i>invoice</i> pada staf pemasaran dan admin pemasaran
Skenario Alternatif 1 : Jika tombol <i>delay</i> pada halaman pesanan ditekan	
1. Aktor masuk ke halaman pesanan dan menekan tombol <i>delay</i>	2. Sistem akan mencatat pesanan produk di halaman pesanan tertunda.
	3. Sistem akan mengirimkan pesan kepada admin pemasaran bahwa pesanan produk dalam status <i>delay</i>
Kondisi Akhir	Sistem mengirimkan status ketersediaan produk kepada admin pemasaran

33. Skenario *Use Case* Mencetak Faktur Penjualan (*Invoice*)

Skenario *Use case* mencetak faktur penjualan (*invoice*) merupakan penjelasan lebih rinci dari *use case* mencetak faktur penjualan (*invoice*) yang dilakukan oleh admin dan staf pemasaran. Tabel 4.37 menunjukkan skenario *use case* mencetak faktur penjualan (*invoice*)

Tabel 4.36 Skenario Mencetak Faktur Penjualan (*Invoice*)

KF_33 : Mencetak Faktur Penjualan (<i>Invoice</i>)	
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana proses mencetak <i>invoice</i> (Faktur Penjualan)
Aktor	Admin Pemasaran, Staf Pemasaran
Skenario Utama	
Kondisi Awal	Aktor telah login dan sistem menampilkan halaman Daftar Pesanan
Aksi Aktor	Reaksi Sistem

1. Aktor menekan tombol <i>Invoice</i> pada sistem dan menekan tombol <i>Print</i>	2. Sistem menampilkan halaman pesanan produk yang akan dicetak. 3. Sistem menampilkan <i>pop-up</i> dalam bentuk <i>print preview</i>
Skenario Alternatif : Jika aktor akan menyimpan invoice	
1. Aktor menekan tombol <i>Invoice</i> pada sistem dan menekan tombol <i>Print</i> 4. Aktor menekan tombol "Save" pada <i>print preview</i>	2. Sistem menampilkan halaman pesanan produk yang akan dicetak 3. Sistem menampilkan <i>pop-up</i> dalam bentuk <i>print preview</i> . 5. Sistem menyimpan <i>invoice</i> pada <i>drive</i> aktor
Kondisi Akhir	Sistem mencetak faktur penjualan (<i>invoice</i>)

4.2 Fase Perancangan

Pada tahapan ini perancangan pada penelitian ini menggunakan pendekatan desain berorientasi objek yang direpresentasikan menggunakan UML (*Unified Modelling Language*)

4.2.1 Sequence Diagram

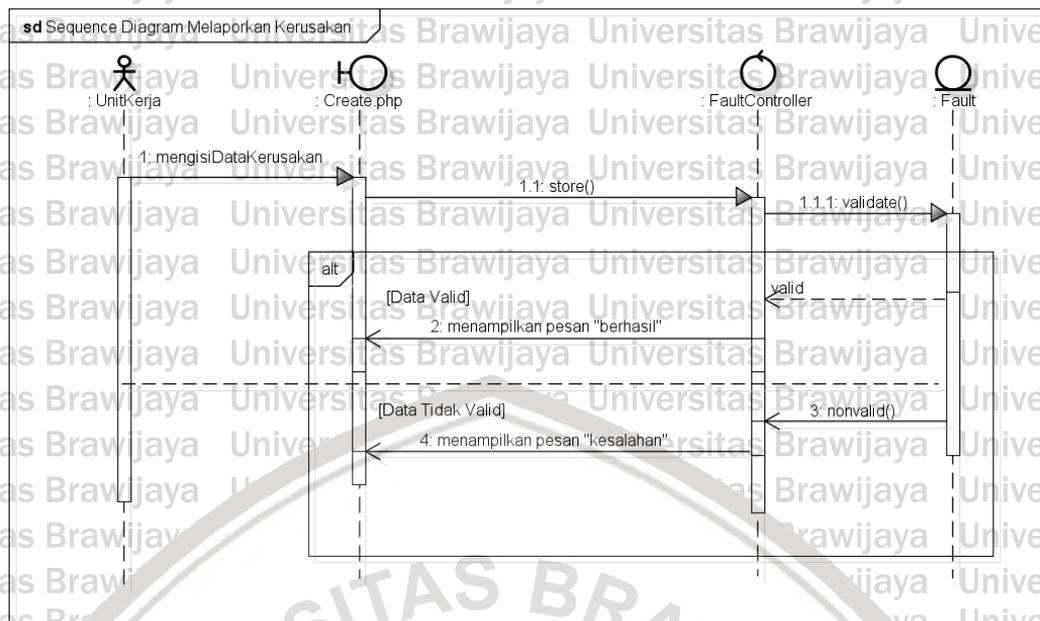
Diagram sekuen merepresentasikan model alir jalannya proses interaksi antar objek. Pemodelan diagram sekuen ini mengacu kepada beberapa diagram yang telah dibuat sebelumnya seperti operasi yang ada pada *use case* dan skenario *use case*.

a. Diagram Sekuen Melaporkan Kerusakan Sarana

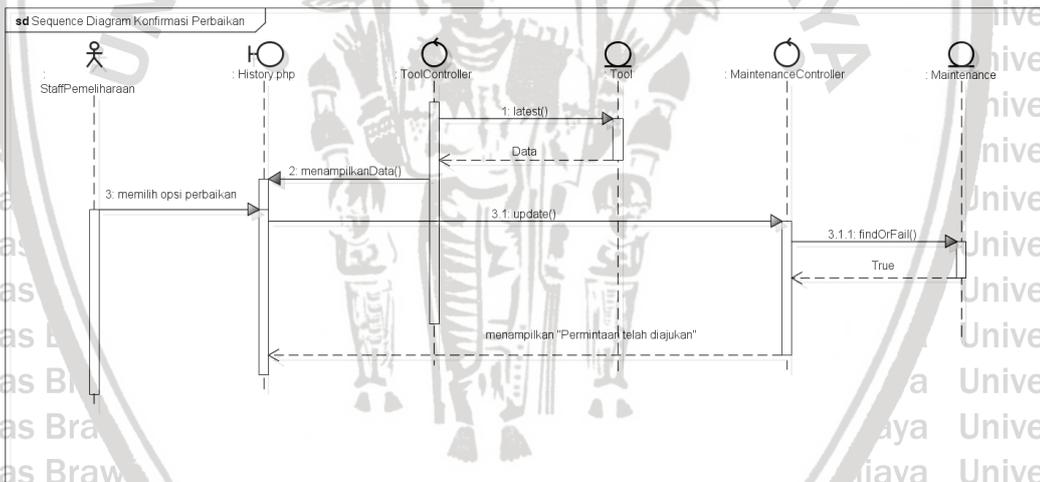
Diagram Sekuen melaporkan kerusakan sarana adalah event ketika bagian unit kerja ingin mengajukan perbaikan sarana kepada staf pemeliharaan. Gambar 4.8 menjelaskan urutan interaksi objek sistem untuk melaporkan kerusakan.

b. Diagram Sekuen Konfirmasi Perbaikan

Diagram sekuen fitur konfirmasi perbaikan menggambarkan urutan interaksi antar objek pada proses konfirmasi perbaikan. Aktor yang berinteraksi di sini adalah staf pemeliharaan. Gambar 4.9 menjelaskan urutan interaksi objek sistem untuk konfirmasi perbaikan.



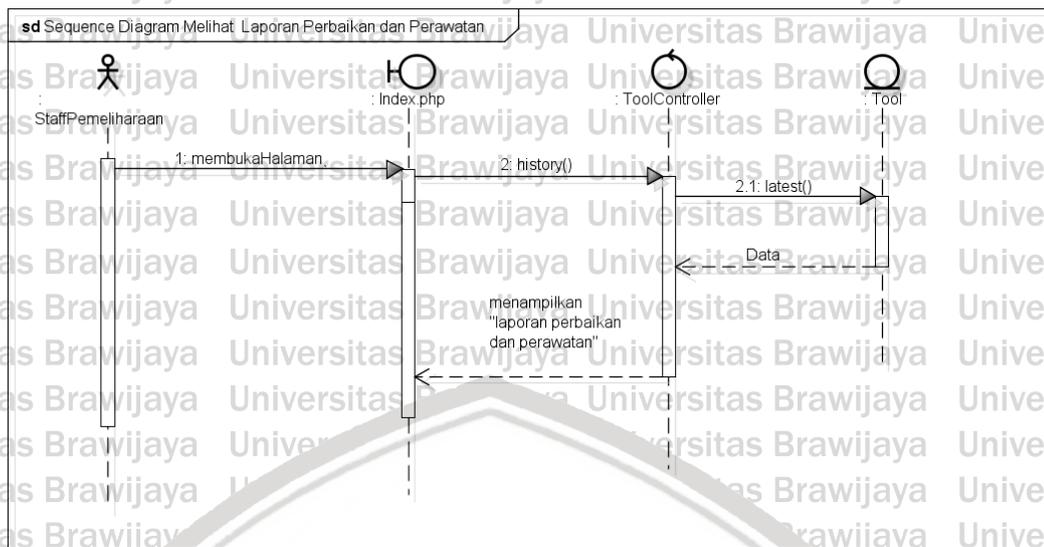
Gambar 4.8 Diagram Sekuen Melaporkan Kerusakan Sarana



Gambar 4.9 Diagram Sekuen Konfirmasi Perbaikan

c. Diagram Sekuen Melihat Laporan Perbaikan dan Perawatan

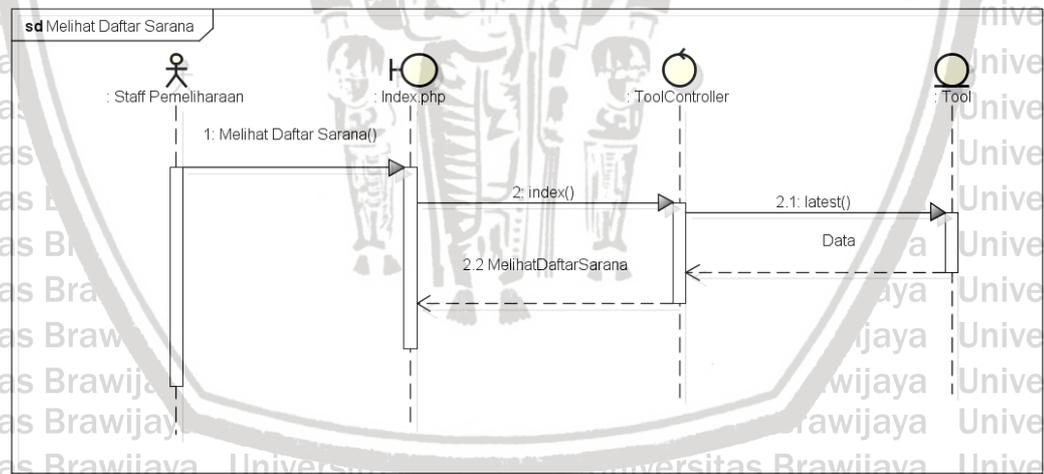
Diagram sekuen melihat laporan perbaikan dan perawatan adalah event ketika staf pemeliharaan ingin melihat daftar perbaikan dan perawatan yang dilakukan selama periode tertentu. Gambar 4.10 menjelaskan urutan interaksi objek sistem untuk melihat laporan perbaikan dan perawatan.



Gambar 4.10 Diagram Sekuen Melihat Laporan perbaikan dan perawatan

d. Diagram Sekuen Melihat Daftar Sarana

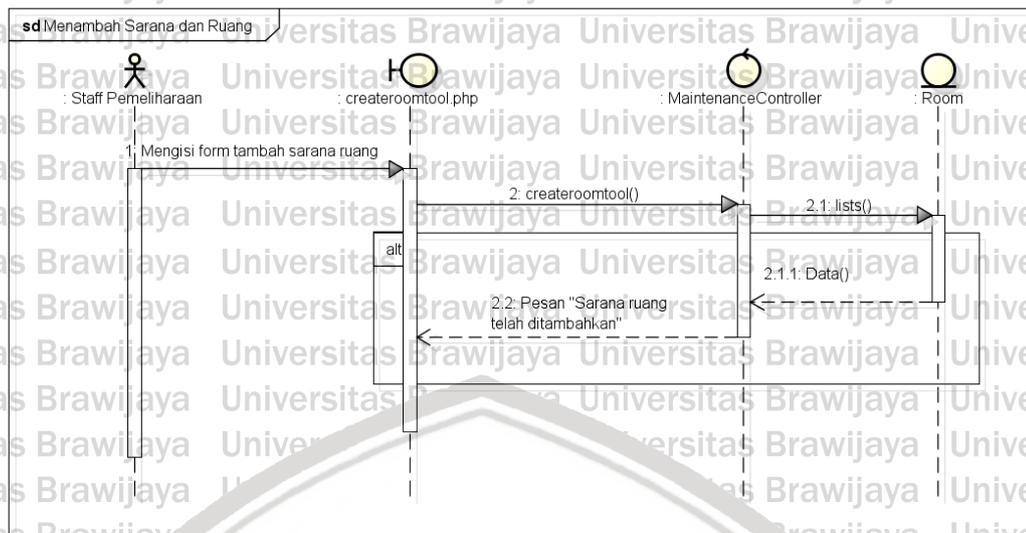
Diagram sekuen fitur melihat daftar sarana menggambarkan urutan interaksi antar objek pada proses melihat daftar sarana. Aktor yang berinteraksi di sini adalah staf pemeliharaan. Gambar 4.11 menjelaskan urutan interaksi objek sistem untuk melihat daftar sarana.



Gambar 4.11 Diagram Sekuen Melihat Daftar Sarana

e. Diagram Sekuen Menambah Sarana dan Ruang

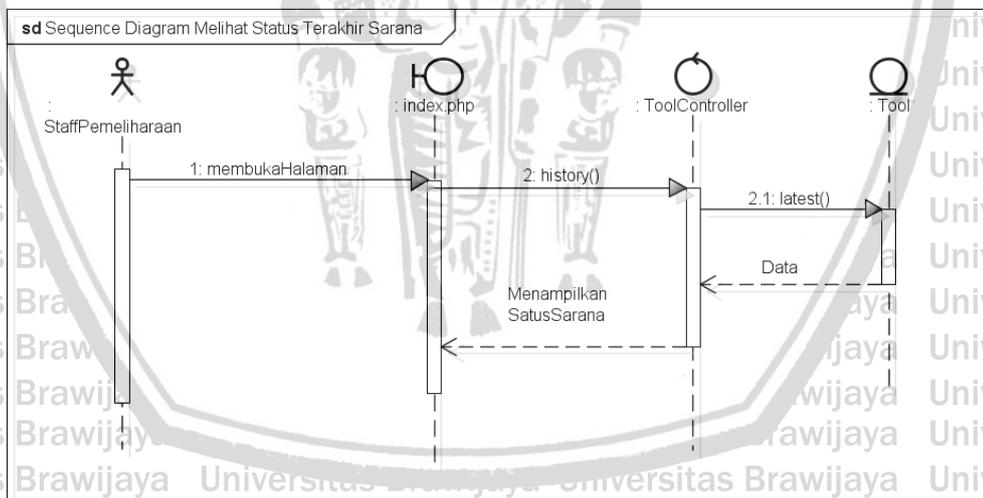
Diagram sekuen menambah sarana dan ruang adalah event ketika bagian staf pemeliharaan ingin menambah daftar sarana dan ruangan. Gambar 4.12 menjelaskan urutan interaksi objek sistem untuk menambah sarana dan ruang.



Gambar 4.12 Diagram Sekuen Menambah Sarana dan Ruang

f. Diagram Sekuen Melihat Status Terakhir Sarana

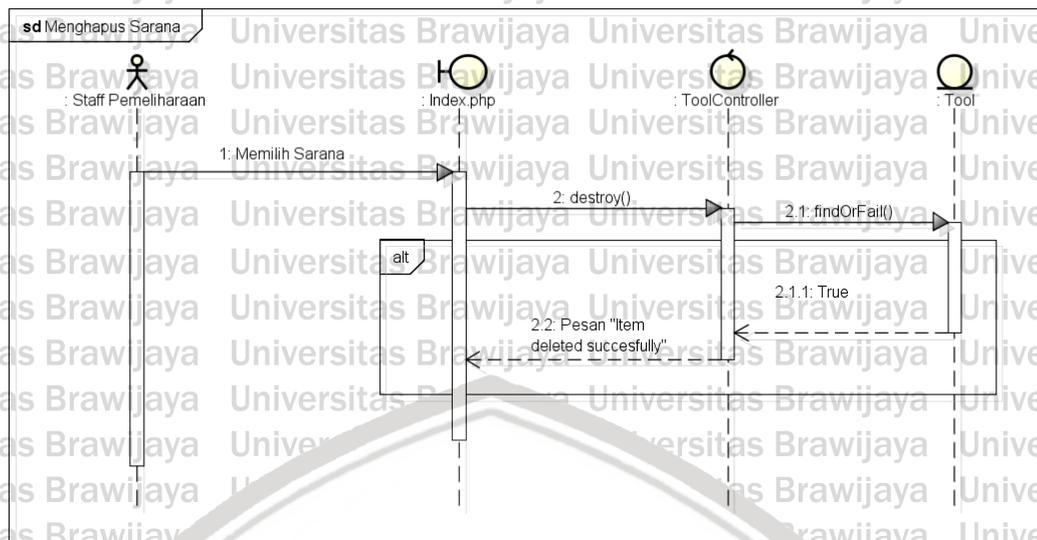
Diagram sekuen melihat status terakhir sarana menggambarkan urutan interaksi antar objek pada proses kerja fitur melihat status terakhir sarana. Aktor yang berinteraksi di sini adalah staf pemeliharaan. Gambar 4.13 menjelaskan urutan interaksi objek sistem untuk melihat status terakhir sarana.



Gambar 4.13 Diagram Sekuen Melihat status terakhir sarana

g. Diagram Sekuen Menghapus Sarana

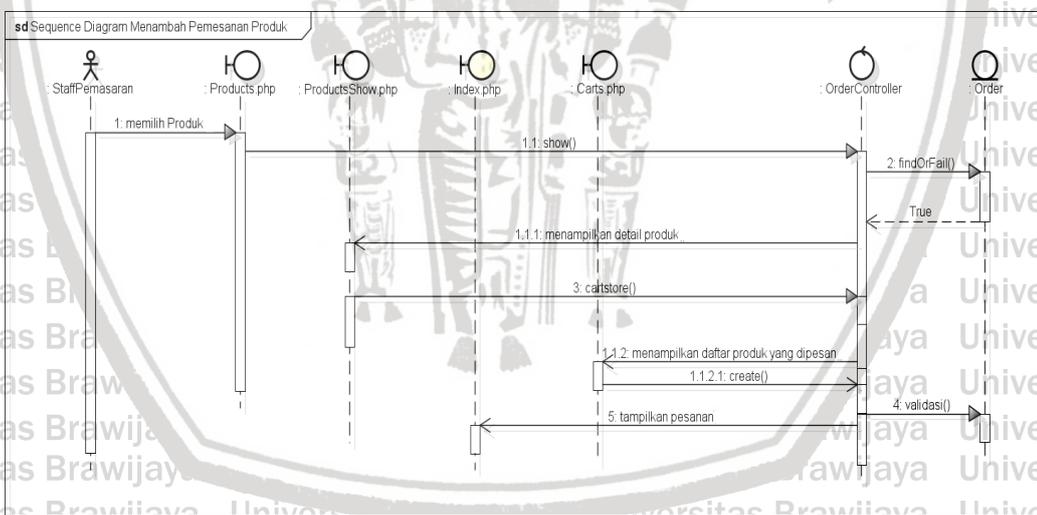
Diagram sekuen menghapus sarana adalah event ketika bagian staf pemeliharaan ingin menghapus ruangan. Gambar 4.14 menjelaskan urutan interaksi objek sistem untuk menghapus ruangan.



Gambar 4.14 Diagram Sekuen Menghapus Sarana

h. Diagram Sekuen Menambah Pemesanan Baru

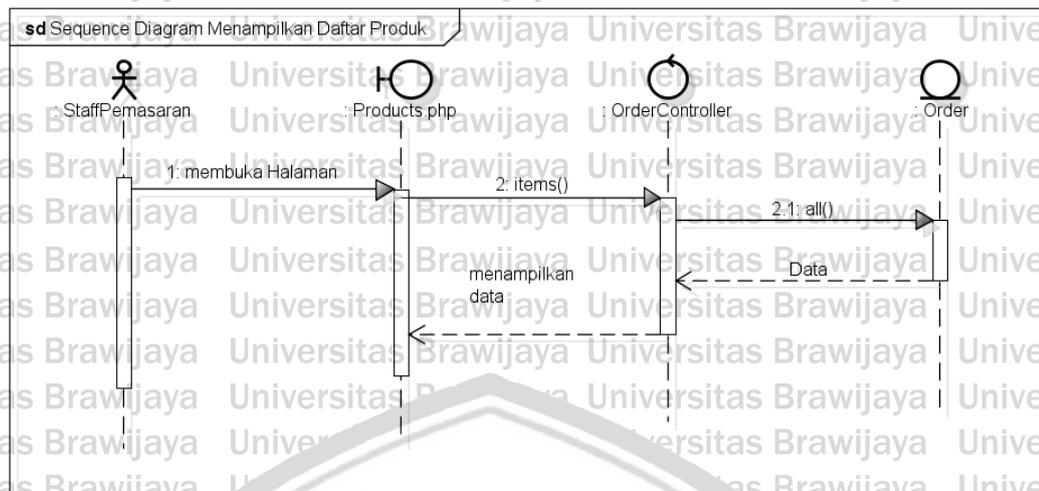
Diagram sekuen menambah pemesanan baru adalah event ketika staf pemasaran melakukan transaksi menambah pemesanan produk. Gambar 4.15 menjelaskan urutan interaksi objek sistem untuk membuat pemesanan baru oleh staf pemasaran.



Gambar 4.15 Diagram Sekuen Membuat Pemesanan Baru

i. Diagram Sekuen Menampilkan daftar produk

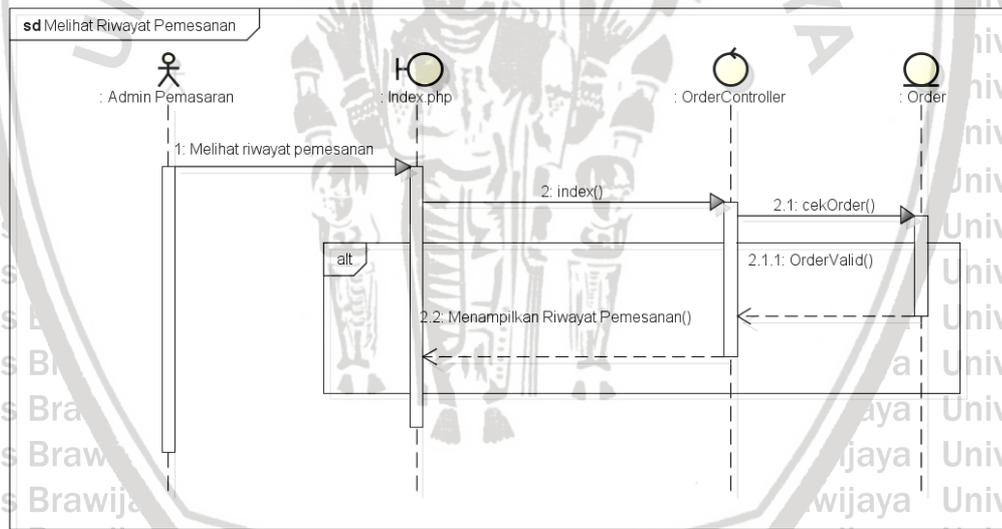
Diagram sekuen melihat informasi produk adalah event ketika staf gudang ingin melihat informasi data produk yang ada di PT. Pancamas Elite. Gambar 4.16 menjelaskan urutan interaksi objek sistem untuk melihat informasi produk oleh staf gudang.



Gambar 4.16 Diagram Sekuen Menampilkan Daftar Produk

j. Diagram Sekuen Melihat Riwayat Pemesanan

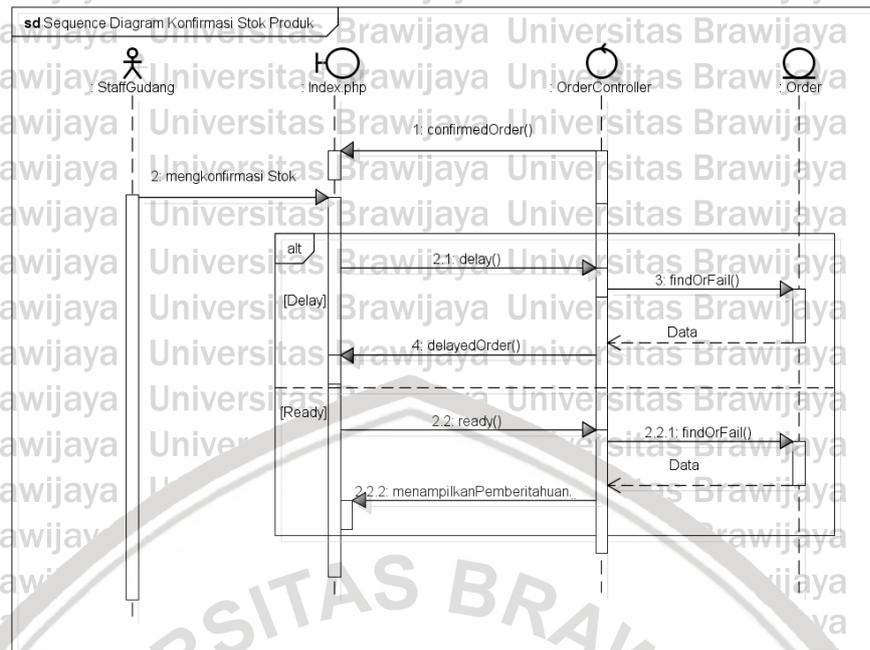
Diagram sekuen subsistem pemasaran ini menggambarkan urutan interaksi antar objek pada fitur melihat riwayat pemesanan. Aktor yang berinteraksi di sini adalah staf pemasaran dan admin pemasaran. Gambar 4.12 menjelaskan urutan interaksi objek sistem untuk melihat riwayat pemesanan.



Gambar 4.17 Diagram Sekuen Melihat Riwayat Pemesanan

k. Diagram Sekuen Konfirmasi Stok Produk

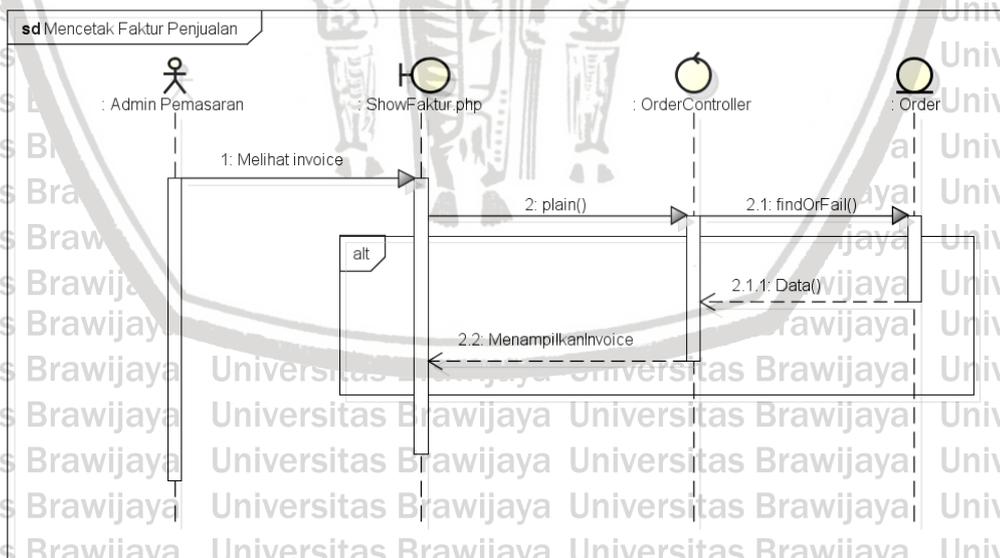
Diagram sekuen subsistem gudang ini menggambarkan urutan interaksi antar objek pada fitur konfirmasi stok produk. Aktor yang berinteraksi di sini adalah staf gudang. Gambar 4.18 menjelaskan urutan interaksi objek sistem untuk konfirmasi stok produk.



Gambar 4.18 Diagram Sekuen Konfirmasi Stok Produk

I. Diagram Sekuen Mencetak Faktur Penjualan

Diagram sekuen subsistem pemasaran ini menggambarkan urutan interaksi antar objek pada fitur mencetak faktur penjualan. Aktor yang berinteraksi di sini adalah staf pemasaran dan admin pemasaran. Gambar 4.19 menjelaskan urutan interaksi objek sistem untuk mencetak faktur penjualan.



Gambar 4.19 Diagram Sekuen Mencetak Faktur Penjualan

4.2.2 Diagram Aktivitas

Diagram aktivitas menggambarkan aktivitas yang dapat dilakukan oleh sistem. Diagram ini mengacu kepada skenario *use case* yang telah dibuat sebelumnya.

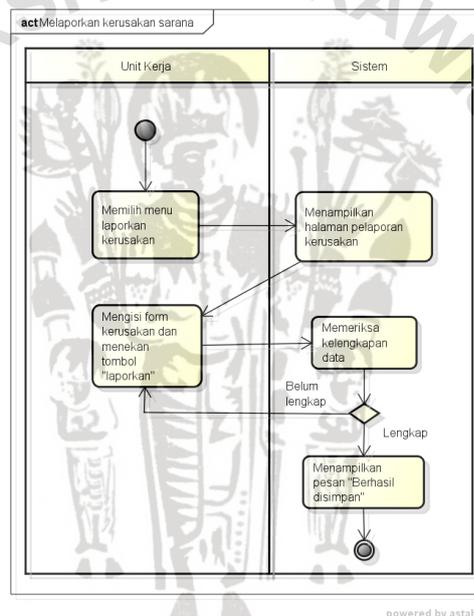
1. Diagram Aktivitas Sistem Pemeliharaan Sarana

Diagram aktivitas sistem pemeliharaan sarana menggambarkan proses bisnis yang dilakukan oleh sistem beserta interaksinya dengan aktor. Aktor yang mengoperasikan modul ini adalah unit kerja dan staf pemeliharaan.

a. Diagram Aktivitas Melaporkan kerusakan sarana

Diagram aktivitas melaporkan kerusakan merupakan aktivitas ketika unit kerja ingin mengajukan perbaikan dan perawatan terhadap sarana yang mengalami kerusakan untuk kemudian diserahkan kepada bagian staf pemeliharaan.

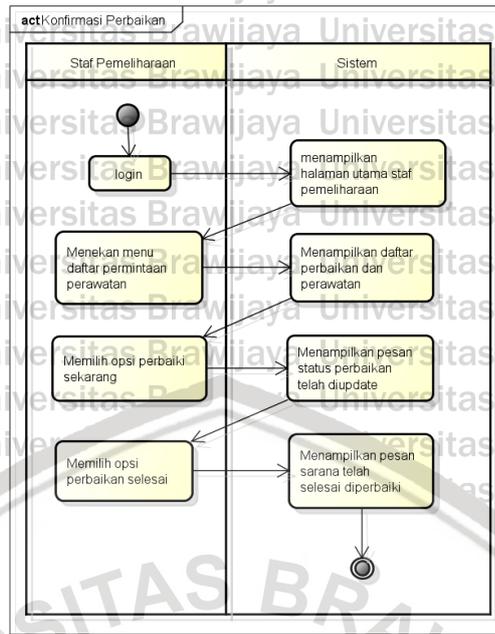
Gambar 4.20 menjelaskan aktivitas melaporkan kerusakan.



Gambar 4.20 Diagram Aktivitas Melaporkan kerusakan sarana

b. Diagram Aktivitas Konfirmasi perbaikan

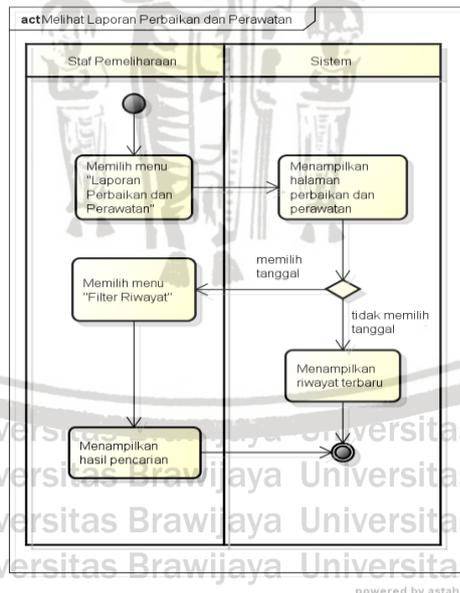
Diagram aktivitas konfirmasi perbaikan merupakan aktivitas ketika staf pemeliharaan melakukan konfirmasi perbaikan terhadap pengajuan kerusakan dari unit kerja maupun dari perawatan rutin Gambar 4.21 menjelaskan aktivitas konfirmasi perbaikan



Gambar 4.21 Diagram Aktivitas Konfirmasi perbaikan

c. Diagram Aktivitas Melihat Laporan Perbaikan dan Perawatan

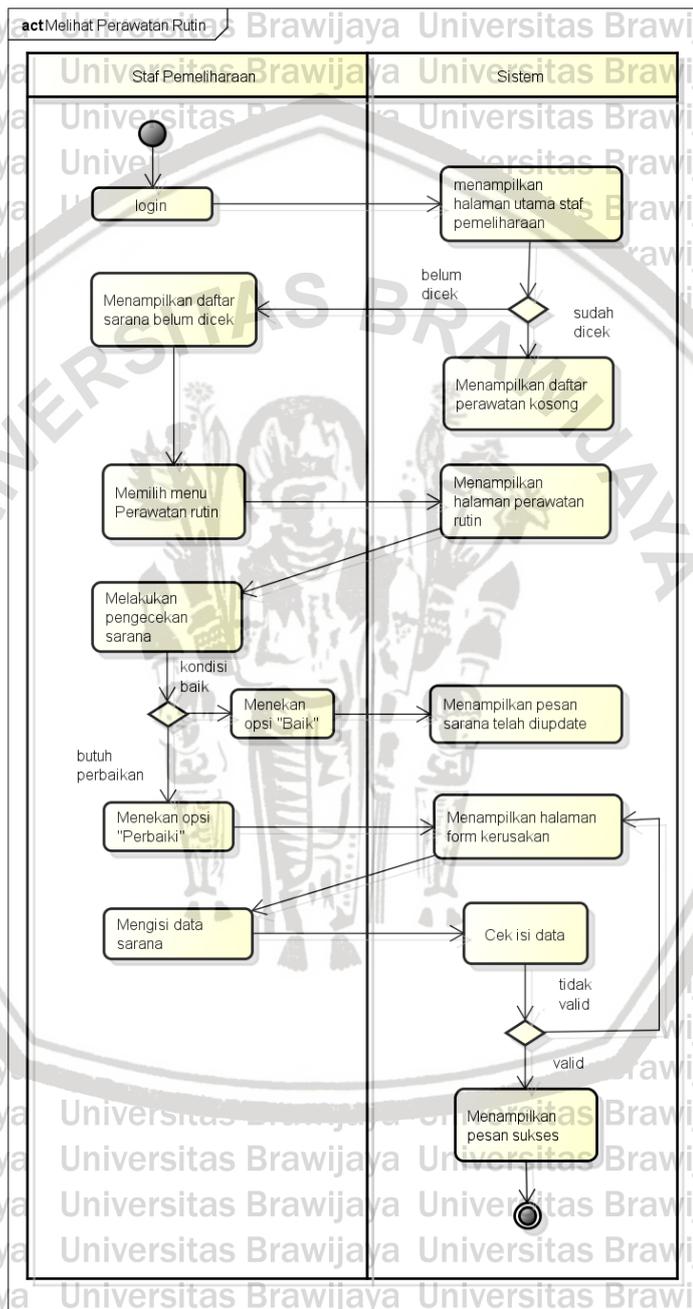
Diagram aktivitas melihat laporan perbaikan dan perawatan merupakan aktivitas ketika staf pemeliharaan ingin melihat riwayat terhadap perbaikan dan perawatan yang telah dilakukan selama periode waktu tertentu. Gambar 4.22 menjelaskan aktivitas melihat laporan perbaikan dan perawatan.



Gambar 4.22 Diagram Aktivitas Melihat Laporan Perbaikan dan Perawatan

d. Diagram Aktivitas Melihat Perawatan Rutin

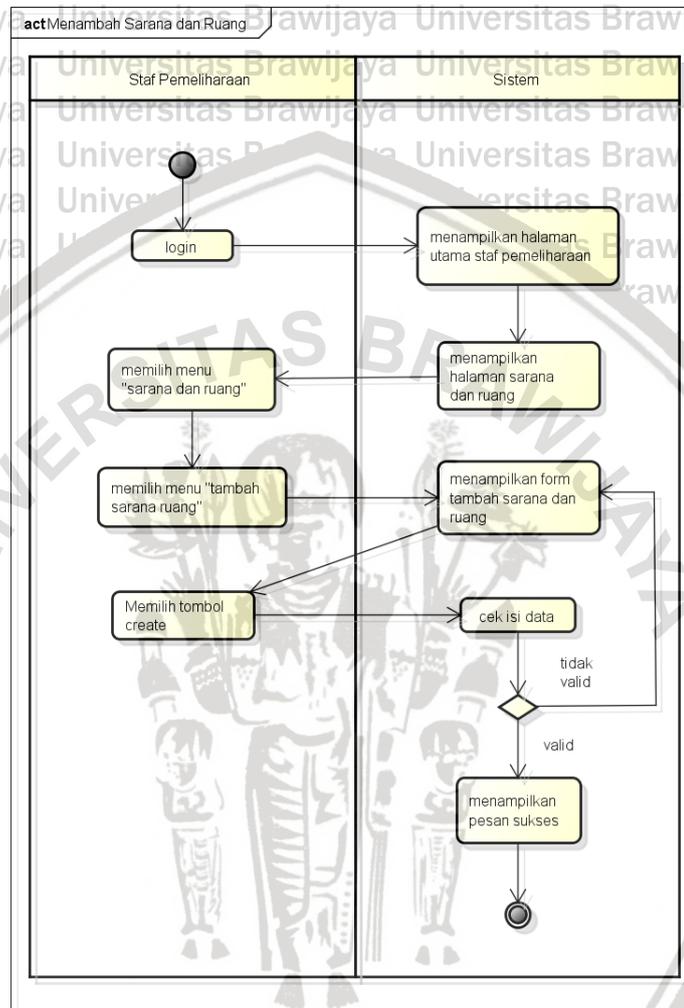
Diagram aktivitas melihat perawatan rutin merupakan aktivitas ketika staf pemeliharaan ingin melihat perawatan rutin yang telah dilakukan atau belum dilakukan selama jadwal yang ditetapkan. Gambar 4.23 menjelaskan aktivitas melihat perawatan rutin.



Gambar 4.23 Diagram Aktivitas Melihat Perawatan Rutin

e. Diagram Aktivitas Menambah Sarana dan Ruang

Diagram aktivitas menambah sarana dan ruang merupakan aktivitas ketika staf pemeliharaan ingin menambah sarana dan ruang kedalam sistem. Gambar 4.24 menjelaskan aktivitas menambah sarana dan ruang.



Gambar 4.24 Diagram Aktivitas Menambah Sarana dan Ruang

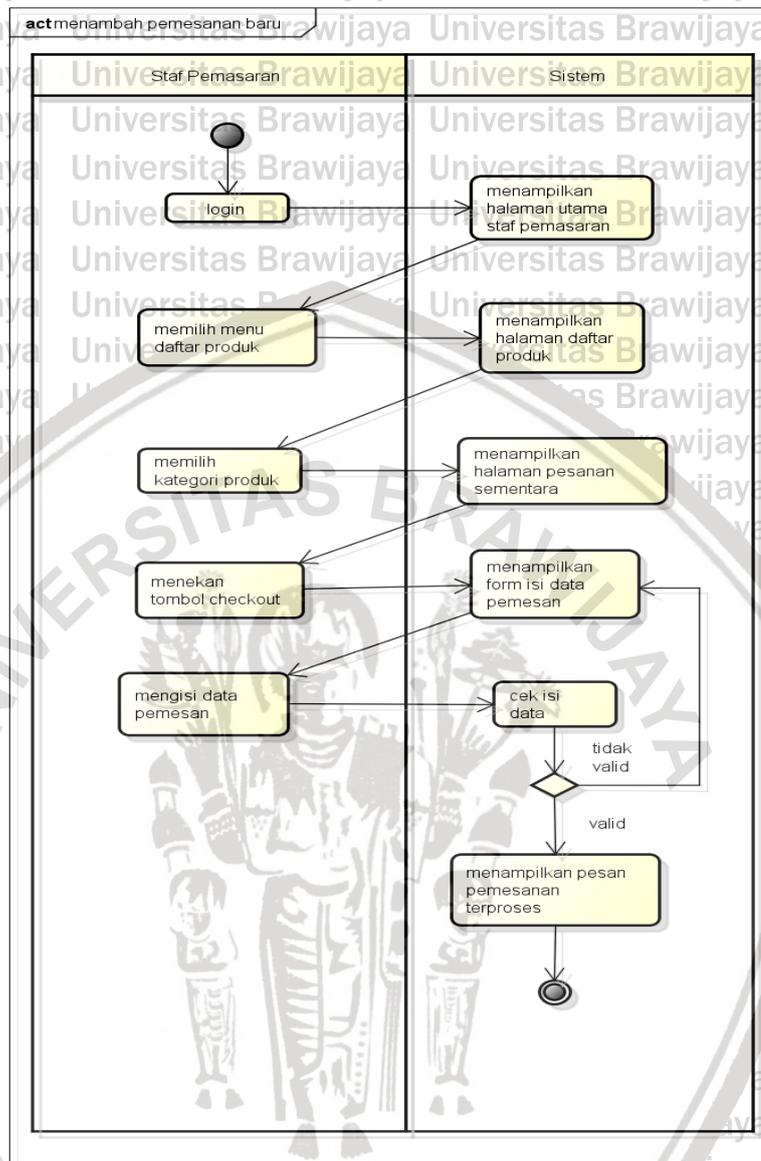
2. Diagram Aktivitas Sistem Pemasaran

Diagram aktivitas sistem pemasaran menggambarkan aktivitas yang dilakukan oleh sistem beserta interaksinya dengan aktor. Aktor yang mengoperasikan modul ini adalah staf pemasaran, admin pemasaran dan staf gudang.

a. Diagram Aktivitas Menambah pemesanan baru

Diagram aktivitas menambah pemesanan baru merupakan aktivitas ketika staf pemasaran ingin membuat pemesanan produk untuk kemudian dilakukan

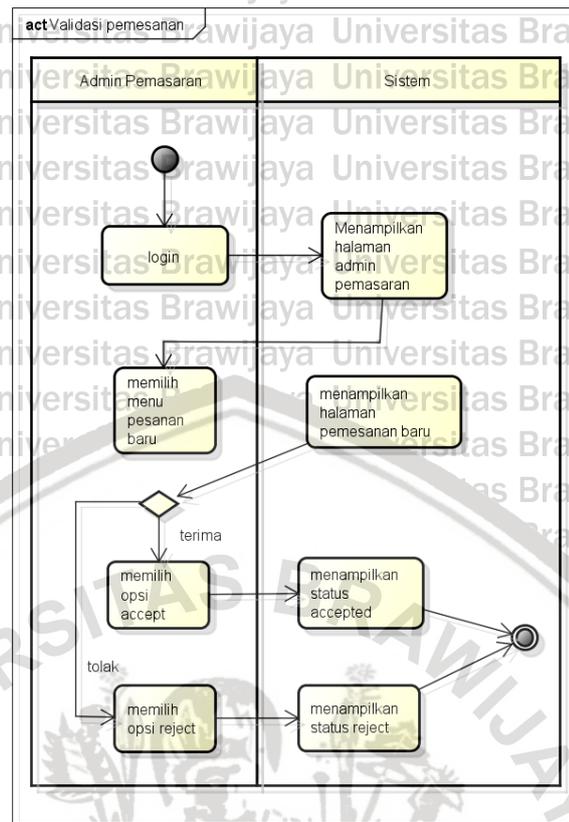
konfirmasi kepada admin pemasaran. Gambar 4.25 menjelaskan aktivitas sistem dalam menambah pemesanan baru.



Gambar 4.25 Diagram Aktivitas Menambah pemesanan baru

b. Diagram Aktivitas Validasi pemesanan

Diagram aktivitas validasi pemesanan merupakan aktivitas ketika admin pemasaran ingin memberikan persetujuan pemesanan produk untuk kemudian diteruskan kepada staf gudang. Gambar 4.26 menjelaskan aktivitas sistem dalam validasi pemesanan.



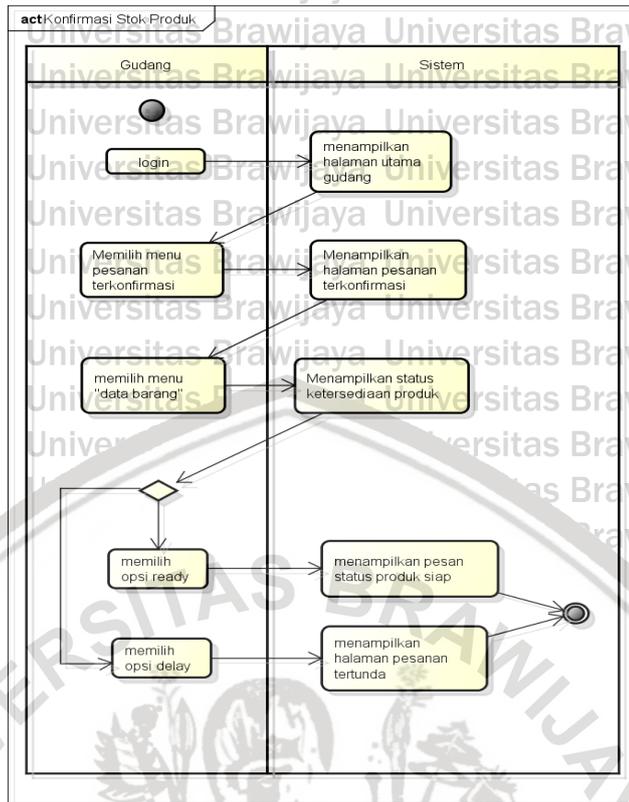
Gambar 4.26 Diagram Aktivitas Validasi pemesanan

c. Diagram Aktivitas Konfirmasi stok produk

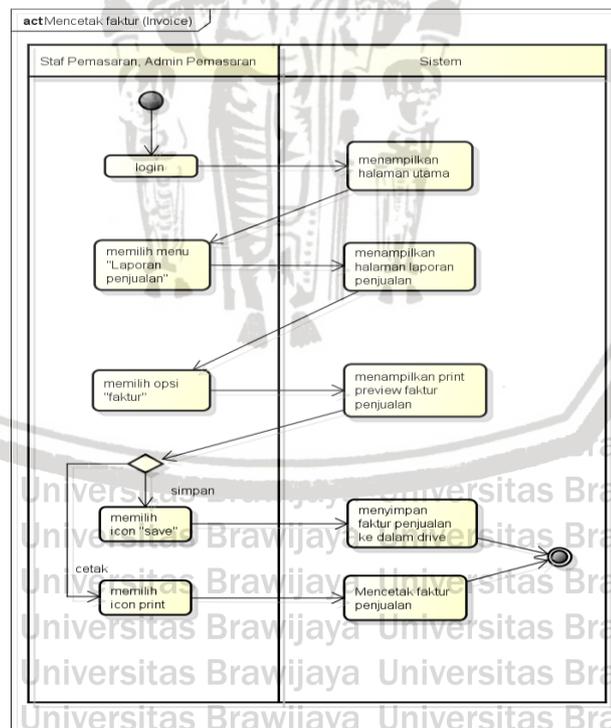
Diagram aktivitas Konfirmasi stok produk merupakan aktivitas ketika gudang menerima pesanan terkonfirmasi dari admin pemasaran untuk selanjutnya diperiksa ketersediaannya oleh staf gudang. Gambar 4.27 menjelaskan aktivitas sistem ketika staf gudang telah menerima notifikasi konfirmasi stok produk.

d. Diagram Aktivitas Mencetak Faktur (*Invoice*)

Diagram aktivitas mencetak *invoice* merupakan aktivitas ketika staf gudang telah memeriksa ketersediaan produk di gudang untuk selanjutnya dibuat *invoice* sebagai tanda bukti pemesanan yang akan diambil oleh staf pemasaran atau admin pemasaran. Gambar 4.28 menjelaskan aktivitas sistem ketika staf gudang membuat *invoice*.



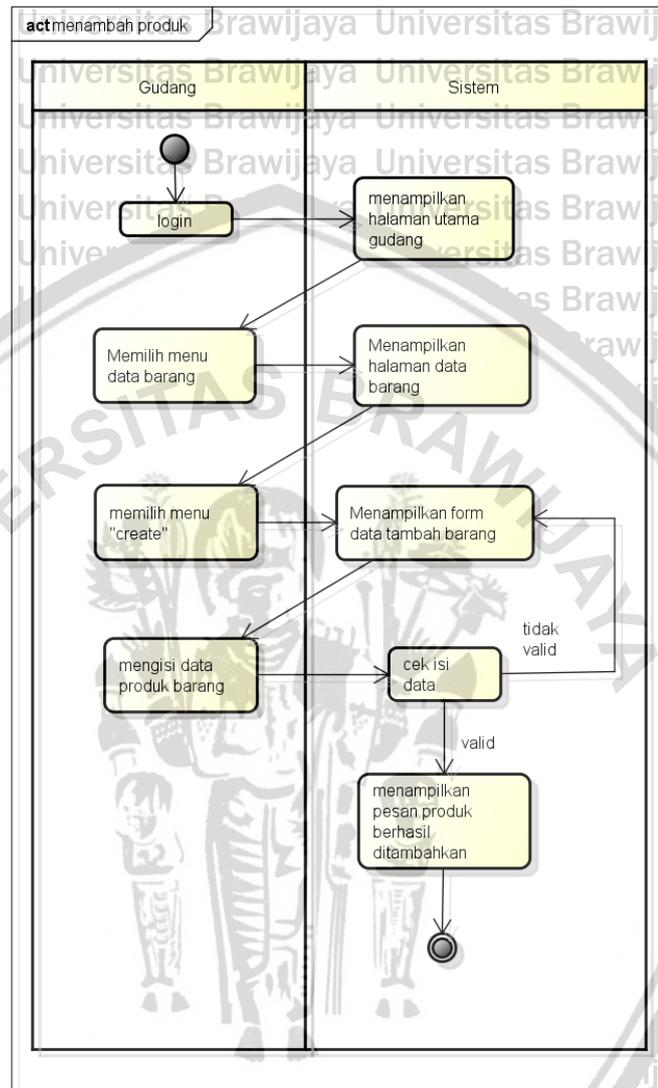
Gambar 4.27 Diagram Aktivitas Konfirmasi stok produk



Gambar 4.28 Diagram Aktivitas Mencetak Faktur (Invoice)

e. Diagram Aktivitas Menambah produk

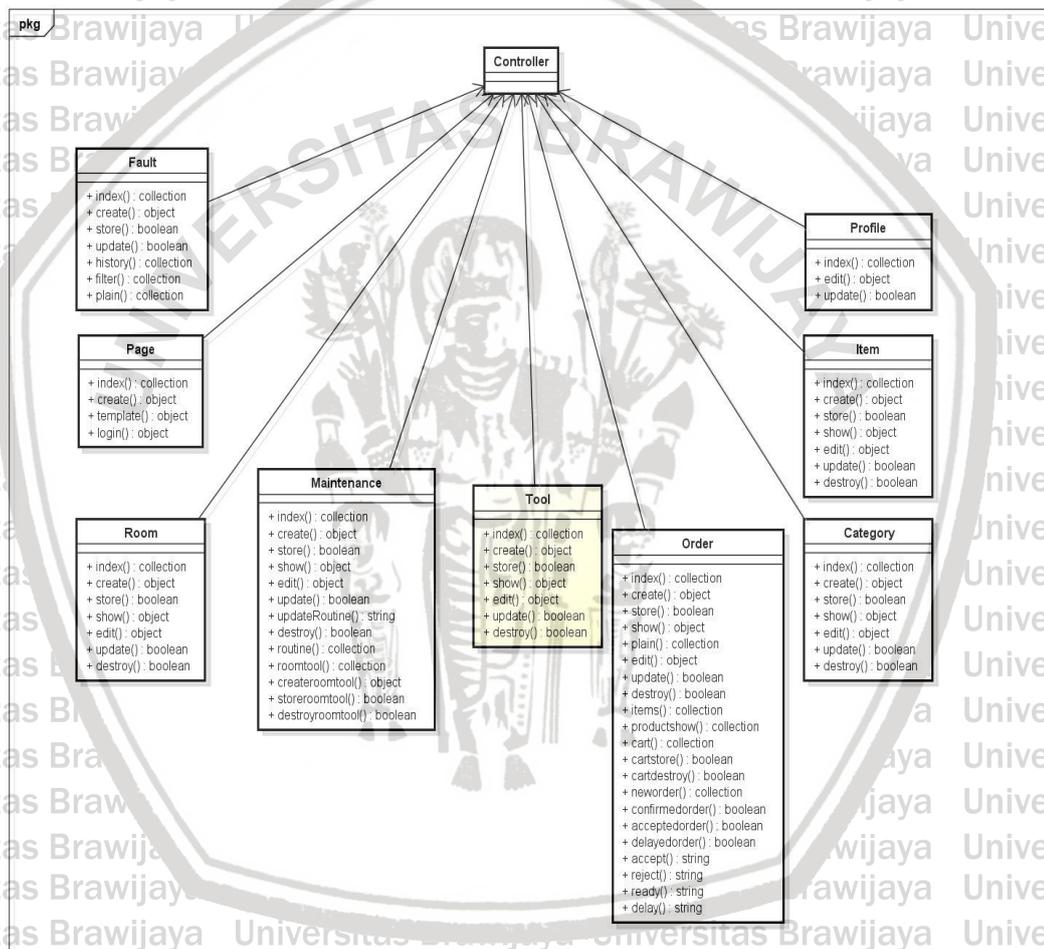
Diagram aktivitas menambah produk merupakan aktivitas ketika staf gudang ingin menambah daftar produk yang ada di dalam gudang. Gambar 4.29 menjelaskan aktivitas sistem menambah produk.



Gambar 4.29 Diagram Aktivitas Menambah produk

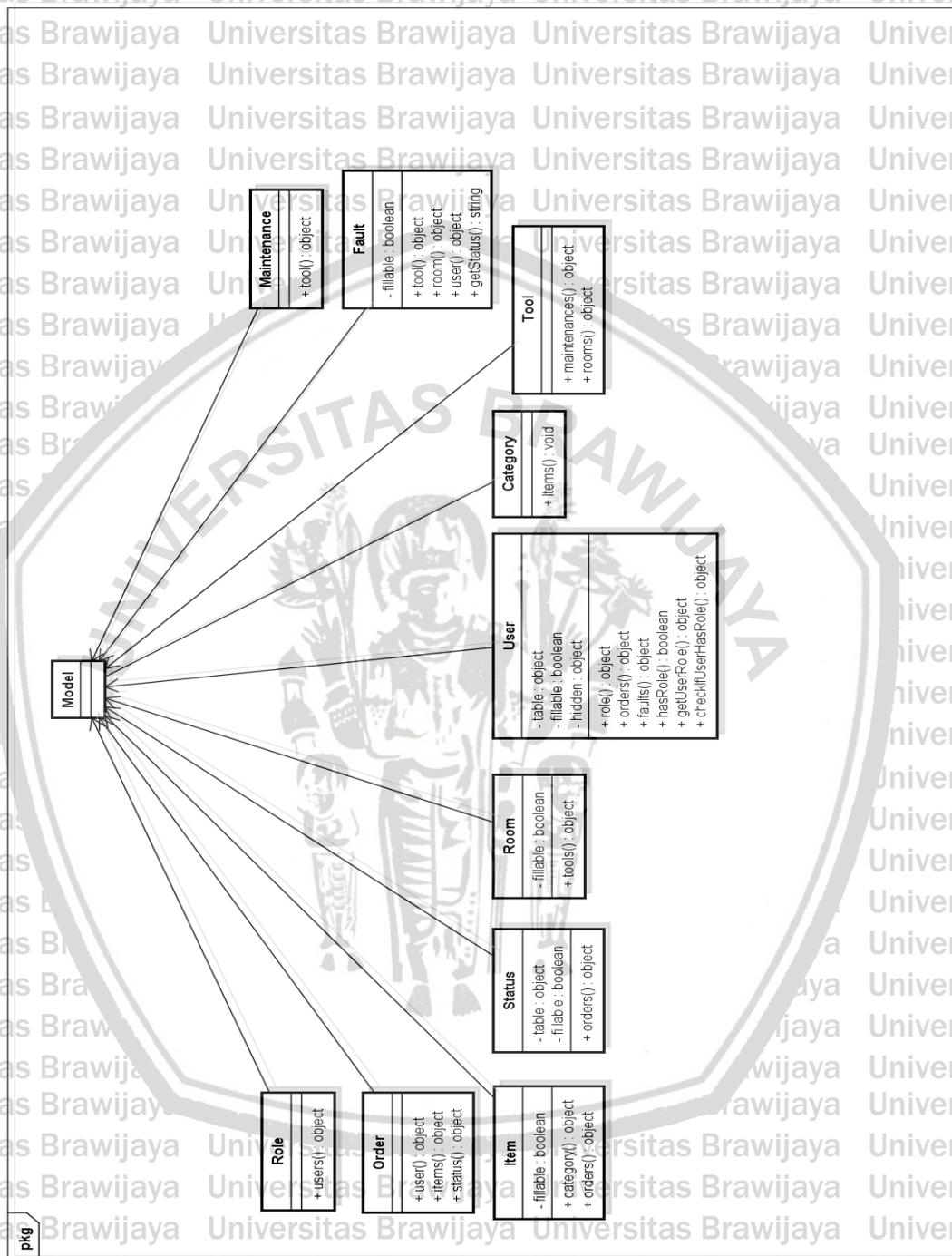
4.2.3 Class Diagram

Class diagram memberikan gambaran pemodelan elemen-elemen *class* yang membentuk sebuah perangkat lunak. Class diagram ini dibuat dari *method*-*method* serta *class* yang akan dibuat pada sistem sesuai dengan kebutuhan. Class Diagram ini berfungsi sebagai gambaran *object* yang ada dalam tahapan implementasi. Sistem dibuat dengan pengembangan berorientasi *object* maka class diagram direpresentasikan dengan *model*, *view* dan *controller*. Berikut merupakan diagram kelas dari sistem pemeliharaan sarana dan pemantauan pemasaran produk yang akan dijelaskan rancangan kelas-kelas *controller* aplikasi pada Gambar 4.30



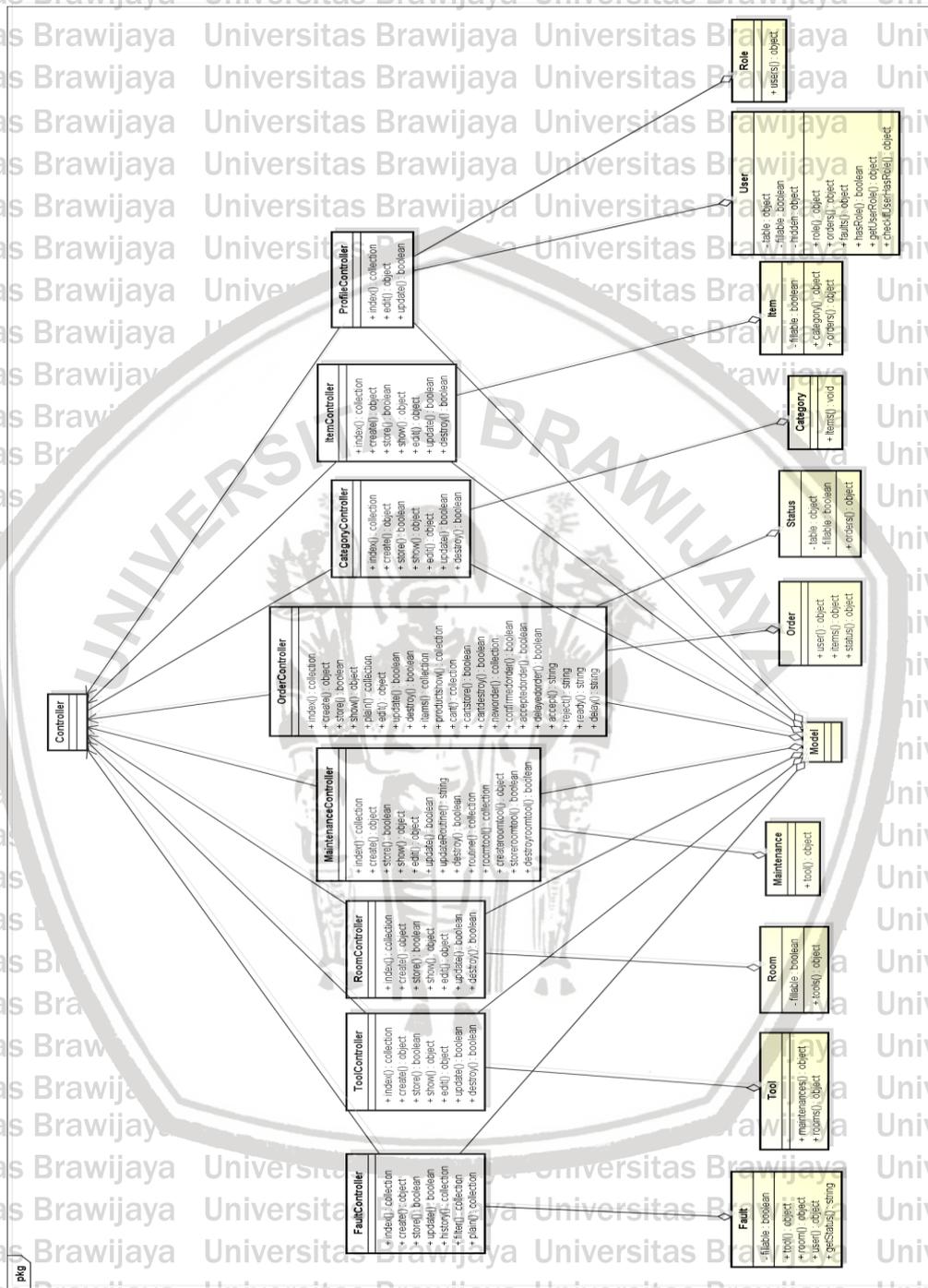
Gambar 4.30 Class Diagram pada *Controller*

Pada Gambar 4.31 berikut akan dijelaskan rancangan kelas-kelas model pada aplikasi yang nantinya akan diimplementasikan pada saat proses pembuatan aplikasi.



Gambar 4.31 Class Diagram pada *Model*

Pada Gambar 4.32 akan dijelaskan rancangan kelas-kelas keseluruhan hubungan dari controller dan model pada aplikasi yang nantinya akan diimplementasikan pada saat proses pembuatan aplikasi



Gambar 4.32 Perancangan Class Diagram

4.2.4 Perancangan Komponen

Perancangan komponen merupakan dekomposisi sub-sistem menjadi komponen detail. Perancangan komponen menjelaskan atribut dan algoritma method dalam sebuah class yang telah dimodelkan sebelumnya pada pemodelan class diagram. Dalam perancangan komponen pada sistem ini mengambil sample 3 *method* utama pada class controller ProfileController.

4.2.4.1 Algoritma Method index()

Nama *Class* : ProfileController

Nama *Method* : index()

Algoritma : (ALGO_CD_01)

```

1 Run library autentikasi
2   switch ($profile, $role, $name)
3     case 'staff'
4       Get $faults = Auth::user()->username)
5       foreach ($faults as $fault)
6         Tampilkan notifikasi
7         case 'maintenance'
8           Get data pemeliharaan
9           if (count ($count1) == 0
10            Semua sarana telah dilakukan pengecekan rutin
11          else
12            Terdapat sarana yang belum dilakukan pengecekan rutin
13            if ($count2 == 0)
14              Tidak ada sarana rusak
15          else
16            Terdapat permintaan perbaikan
17            case 'marketing'
18              getCountTotal, $countReady, $countAccepted, $countNotProcessed,
19              $notification
19            case 'manager'
20              getCountTotal, $countReady, $countAccepted, $countNotProcessed,
21              $notification
21            case 'storage'
22              getCountTotal, $countAccepted, $countCategory, $countNotProcessed,
23              $countItem, $notification
23            default:
24              Halaman Tidak Tersedia
25          return view ('profile.index', compact ('profile', 'notification'));
26 END

```

4.2.4.2 Algoritma Method store()

Nama Class : OrderController

Nama Method : store()

Algoritma : (ALGO_CD_02)

```

1  Input $produk, $jumlah
2  If ($order==1)
3      Isi identitas pembeli
4  Else if (($order==2)
5      Pesanan Sementara
6  If ($order=1, $identity=true)
7      Pesanan telah diinputkan
8  Else if ($order=1, $identity=false)
9      Isi kelengkapan data
10 ($pemesananbaru)
11 If ($order==1 && $acceptance==true)
12     pemesanan dikirim ke gudang
13 Else if ($order==1 && $acceptance==false)
14     pemesanan ditolak
15     ($gudang)
16 If ($order==1 && $stok>=1)
17     pesan stok tersedia dan mengirim faktur penjualan
18 Else if ($order==1 && $stok==0)
19     Pesanan Tertunda
20 END

```

4.2.4.3 Algoritma Method storeroomtool()

Nama Class : MaintenanceController

Nama Method : storeroomtool()

Algoritma : (ALGO_CD_03)

```

1  Run library validasi
2  'inventaris_code.required'
3  Kode Inventaris harus diisi,
4  'inventaris_code.unique'
5  Kode Inventaris harus unik (tidak boleh kembar)
6  ($request)
7  get ($request->input('tool_id'))
8  $room->tools attach($stool)
9  get $request input('inventaris_code');
10 return redirect ('pemeliharaan.roomtool');
11 menampilkan pesan sarana ruang telah ditambahkan
12 END

```

4.2.5 Perancangan Antarmuka

Pada bagian ini akan dijelaskan tentang perancangan antarmuka dari sistem pemeliharaan sarana dan pemantauan pemasaran produk berbasis web dengan kerangka kerja laravel. Antarmuka yang dimodelkan yaitu antarmuka untuk halaman *login*, halaman utama dari admin pemasaran, staf pemasaran, staf gudang, unit kerja dan staf pemeliharaan.

1. Perancangan Antarmuka *Login*

Antarmuka login merupakan halaman user sebelum masuk kedalam sistem Pemeliharaan Sarana dan Pemantauan Pemasaran Produk . Gambar 4.33 merepresentasikan rancangan antarmuka untuk *login* ke sistem.



The image shows a web browser window with the title "Please Sign In". The page features a large, faint watermark of a person's face in the background. Below the watermark, there is a login form with the following elements: a "Username" input field, a "Password" input field, a checked "Remember Me" checkbox, and a "Login" button.

Gambar 4.33 Perancangan Antarmuka *Login*

2. Perancangan Staf Pemasaran

Antarmuka staf pemasaran adalah halaman staf pemasaran yang bertujuan untuk menambah pemesanan baru . Gambar 4.34 merepresentasikan rancangan antarmuka untuk aktor staf pemasaran

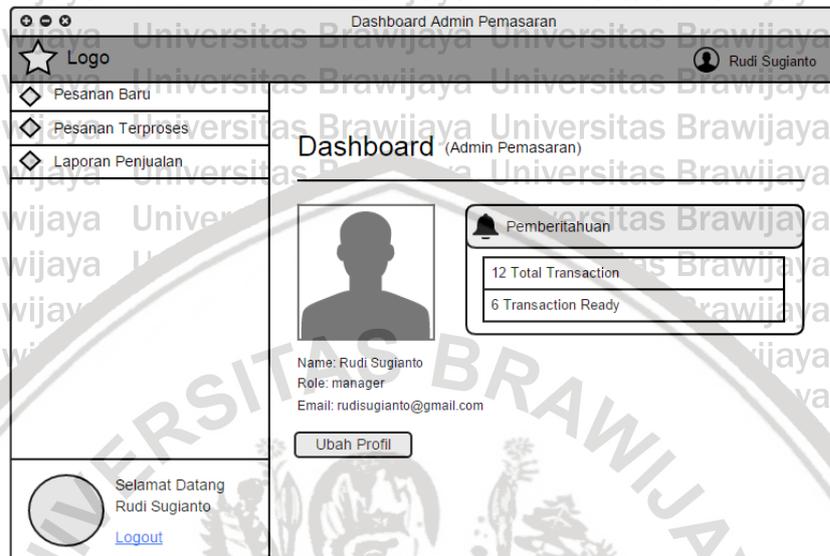


The image shows a web browser window titled "Dashboard Staf Marketing". The interface includes a top navigation bar with a "Logo" and a user profile icon labeled "Rani". On the left, there is a sidebar menu with items: "Daftar Produk", "Pesanan Sementara", and "Laporan Penjualan". The main content area is titled "Dashboard" and contains a user profile card for "Rani" with fields for "Name: Rani", "Role: Marketing", and "Email: rani09@gmail.com", along with a "Change Profile" button. To the right of the profile card is a "Notification" box with two placeholder lines of text.

Gambar 4.34 Perancangan Antarmuka Staf Pemasaran

3. Perancangan Admin Pemasaran

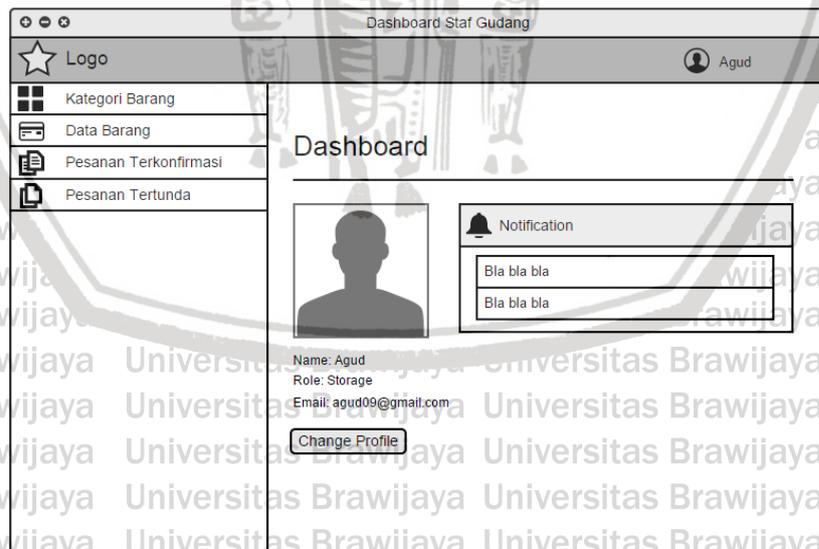
Antarmuka admin pemasaran adalah halaman admin pemasaran yang bertujuan melakukan konfirmasi terhadap pemesanan baru. Gambar 4.35 merepresentasikan rancangan antarmuka untuk aktor admin pemasaran



Gambar 4.35 Perancangan Antarmuka Admin Pemasaran

3. Perancangan Antarmuka Staf Gudang

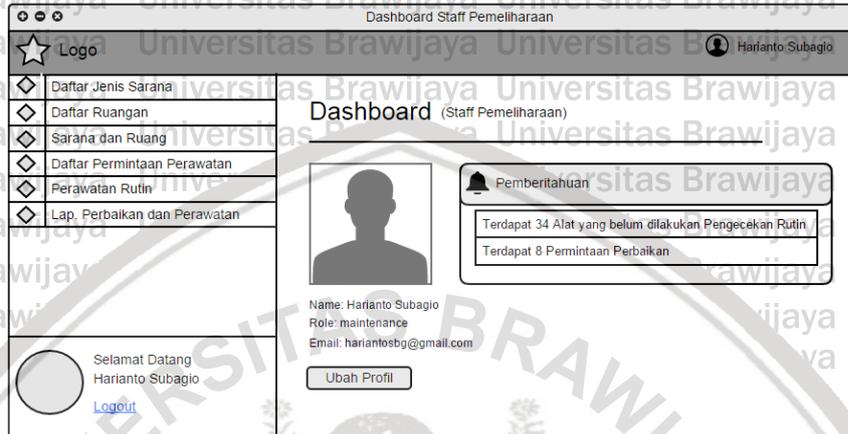
Antarmuka staf gudang adalah halaman staf gudang yang bertujuan untuk melihat ketersediaan produk yang ada di PT. Pancamas Elite Gambar 4.36 merepresentasikan rancangan antarmuka untuk aktor staf gudang



Gambar 4.36 Perancangan Antarmuka Staf Gudang

4. Perancangan Antarmuka Staf Pemeliharaan

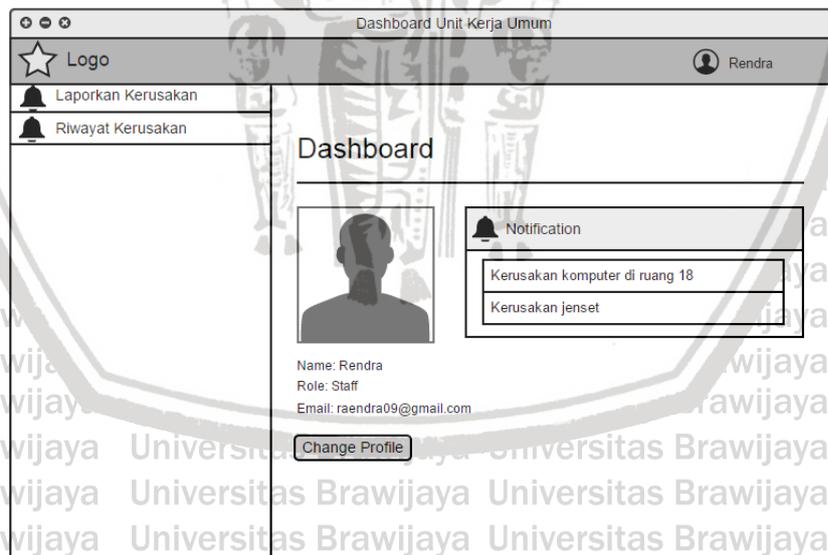
Antarmuka staf pemeliharaan adalah halaman staf pemeliharaan yang bertujuan untuk melakukan kegiatan peeliharaan sarana yang ada di PT.Pancamas Elite. Gambar 4.37 merepresentasikan rancangan antarmuka untuk aktor staf pemeliharaan



Gambar 4.37 Perancangan Antarmuka Staf Pemeliharaan

5. Perancangan Antarmuka Unit Kerja

Antarmuka unit kerja adalah halaman unit kerja yang bertujuan untuk melakukan pengajuan perbaikan kerusakan sarana yang ada di PT.Pancamas Elite. Gambar 4.38 merepresentasikan rancangan antarmuka untuk aktor unit kerja.



Gambar 4.38 Perancangan Antarmuka Unit Kerja

BAB 5 IMPLEMENTASI

Pada bab ini akan menjelaskan tentang implementasi sistem informasi sistem pemeliharaan sarana dan pemantauan pemasaran produk yang ada di PT.Pancama Elite. Pada bab ini implementasi yang akan dibahas meliputi lingkungan perangkat lunak, perangkat keras dan implementasi antarmuka.

5.1 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam analisa dan perancangan adalah :

1. Sistem operasi Windows 8 Pro 64bit
2. XAMPP versi 5.5.24

XAMPP adalah perangkat lunak yang berguna untuk pengembangan website berbasis PHP dan MySQL. Perangkat lunak komputer ini memiliki kelebihan untuk bisa berperan sebagai server web Apache untuk simulasi pengembangan website. *Tool* pengembangan web ini mendukung teknologi web populer seperti PHP, MySQL, dan Perl

3. Sublime Text 3

Sublime Text 3 adalah sebuah HTML editor professional untuk mendesain dan mengelola situs web maupun halaman web.

4. Google Chrome versi 46.0.2490.86

Google Chrome adalah sebuah *browser* atau secara sederhana didefinisikan sebagai sebuah perangkat lunak yang bertujuan untuk menampilkan berbagai dokumen yang tersedia pada *web server*, proses menampilkan berbagai dokumen ini melewati sebuah interaksi antara *browser* dengan *web server*. Fungsi dari *browser* antara lain adalah untuk melakukan atau menampilkan halaman web atau interaksi antara dokumen yang disediakan oleh *server*.

5.2 Lingkungan Perangkat Keras

Spesifikasi perangkat keras yang dipakai dalam proses pengembangan sistem informasi dijelaskan pada tabel 5.1

Tabel 5.1 Spesifikasi perangkat keras komputer

PC Toshiba L745	
Processor	Intel® Core™ i3-2310M CPU @2.10GHz 2.10GHz
Memory (RAM)	4096 RAM
Harddisk	500GB
Monitor	Generic PnP Monitor

5.3 Batasan Implementasi

Beberapa batasan dalam mengimplementasikan sistem informasi ini adalah sebagai berikut:

1. Sistem dibangun khususnya menangani di bagian sistem informasi pemeliharaan sarana, pelaporan kerusakan dan pemesanan produk.
2. Platform pembuatan sistem informasi yang digunakan adalah PHP dengan kerangka kerja *Laravel*.
3. *Database Management System* yang digunakan adalah MySQL yang terletak di localhost.

5.4 Implementasi Class Pada File Program

Implementasi Sistem Informasi Pemeliharaan Sarana dan Pemantauan Pemasaran Produk PT.Pancamas Elite ini menggunakan kerangka kerja *Laravel* yang menerapkan pola MVC. Untuk memudahkan pengembangan sistem ini akan dijelaskan masing-masing file class *controllers* dan *models* yang terdapat dalam folder "skripsi". Berikut ini merupakan hubungan antar class dengan file pada tabel 5.2

Tabel 5.2 Struktur Class dan lokasi file

No	Package	Nama Class	Lokasi File	Nama File Program
1	Controllers	CategoryController	/http/controllers	CategoryController.php
2	Controllers	FaultController	/http/controllers	FaultController.php
3	Controllers	ItemController	/http/controllers	ItemController.php
4	Controllers	MaintenanceController	/http/controllers	MaintenanceController.php
5	Controllers	OrderController	/http/controllers	OrderController.php
6	Controllers	PageController	/http/controllers	PageController.php
7	Controllers	ProfileController	/http/controllers	ProfileController.php
8	Controllers	RoomController	/http/controllers	RoomController.php
9	Controllers	ToolController	/http/controllers	ToolController.php
10	Models	Category	/app/providers	Category.php
11	Models	Fault	/app/providers	Fault.php
12	Models	Item	/app/providers	Item.php
13	Models	Maintenance	/app/providers	Maintenance.php
14	Models	Order	/app/providers	Order.php
15	Models	Room	/app/providers	Room.php
16	Models	Role	/app/providers	Role.php
17	Models	Status	/app/providers	User.php

18	Models	User	/app/providers	User.php
19	Models	Tool	/app/providers	Tool.php

5.5 Implementasi Kerangka Kerja Laravel pada Program

Pada bagian ini akan dijelaskan implementasi MVC (*Model, View, Controller*) pada kerangka kerja *Laravel* yang diterapkan dalam sistem pemeliharaan sarana dan pemantauan pemasaran produk berbasis web PT.Pancamas Elite.

5.5.1 Implementasi Eloquent ORM (*Object Relational Mapping*) pada Model

Implementasi Sistem Informasi Pemeliharaan Sarana dan Pemantauan Pemasaran Produk PT.Pancamas Elite ini menggunakan kerangka kerja *Laravel* yang menerapkan pola MVC. *Eloquent* adalah sebuah ORM (*Object Relational Mapping*) yang *powerful* dan *expressive*. Pada *Laravel* setiap model yang merupakan turunan dari *Eloquent* mewakili sebuah tabel pada database. Dengan menggunakan relationship kita bisa melakukan pengambilan data dari banyak tabel sekaligus. Dengan menggunakan *eloquent model* program yang dibuat bisa lebih rapi dan mudah dibaca serta cepat dan mudah digunakan kembali.

Tabel 5.3 Tabel Eloquent

No	Eloquent	Relasi
1	hasOne	One to One
2	hasMany	One to Many
3	belongsTo	Many to One
4	belongsToMany	Many to Many

Laravel menyediakan sebuah fitur ORM (*Object Relational Mapping*) yang telah terintegrasi didalamnya. Untuk membuat relasi antar tabel dituliskan dalam bentuk sintak *hasOne* untuk relasi *One to One*, *hasMany* untuk relasi *One to Many*, *belongsTo* untuk relasi *Many to One* dan *belongsToMany* untuk relasi *Many to Many*. Tabel 5.4 sampai dengan 5.11 merupakan penerapan relasi antar tabel di dalam Sistem Informasi Pemeliharaan Sarana dan Pemantauan Pemasaran Produk.

Tabel 5.4 Implementasi ORM pada Model

No	Class	Method	Eloquent ORM
1	Category	item	hasMany
2	Fault	tool	belongsTo

		room	belongsTo
		user	belongsTo
3	Item	category	belongsTo
		order	belongsToMany
4	Maintenance	tool	belongsTo
5	Order	user	belongsTo
		items	belongsToMany
		status	belongsToMany
6	Role	users	hasMany
7	Room	tools	belongsToMany
8	Status	order	belongsToMany
9	Tool	maintenance	hasMany
		rooms	belongsToMany
10	User	roles	belongsTo
		orders	hasMany
		faults	hasMany

5.5.2 Implementasi Routing dan Controller

Router merupakan bagian yang penting dari aplikasi terutama untuk aplikasi yang berbasis web. Router berfungsi menangani *request* aplikasi kemudian mengarahkan aplikasi untuk memanggil halaman atau *resource* tertentu. Pada *Laravel* semua routes diletakkan dalam satu file sehingga memudahkan modifikasi. Selain dengan membuka file tersebut, juga bisa melihat routes apa saja yang terdapat pada aplikasi anda dengan menggunakan perintah pada *command line*. Bentuk dasar routing pada *Laravel* adalah cukup mudah karena hanya dengan memanggil Controller saja. Tabel 5.5 dibawah adalah implementasi *Pattern Front Controller* pada *Laravel Route* Sistem Pemeliharaan Sarana dan Pemantauan Pemasaran Produk Berbasis Web dengan Kerangka Kerja *Laravel*.

Tabel 5.5 Pattern Front Controller

No	Fungsi	Pattern Front Controller pada <i>Laravel Route</i>
1	Profile	<pre>Route::get('profile', 'ProfileController@index'); Route::get('profile/edit', 'ProfileController@edit');</pre>

		<pre>Route::put('profile/update', 'ProfileController@update');</pre>
2	Login dan Logout	<pre>Route::get('login', ['as' => 'login', 'uses' => 'Auth\AuthController@getLogin']); Route::post('login', ['as' => 'post.login', 'uses' => 'Auth\AuthController@postLogin']); Route::get('logout', ['as' => 'logout', 'uses' => 'Auth\AuthController@getLogout']);</pre>
3	Pengolahan Master Data Produk	<pre>Route::group(['roles' => 'storage'], function() { Route::get('orders/accepted', ['as' => 'orders.accepted', 'uses' => 'OrderController@acceptedOrder']); Route::get('orders/delayed', ['as' => 'orders.delayed', 'uses' => 'OrderController@delayedOrder']); Route::put('orders/ready', ['as' => 'orders.ready', 'uses' => 'OrderController@ready']); Route::put('orders/delay', ['as' => 'orders.delay', 'uses' => 'OrderController@delay']); Route::get('orders/storage/{order}', ['as' => 'orders.showstorage', 'uses' => 'OrderController@show']); Route::resource('categories', 'CategoryController'); Route::resource('items', 'ItemController');});</pre>
4	Pengolahan Data Peralatan & Perawatan	<pre>Route::group(['roles' => ['maintenance']], function() { Route::resource('rooms', 'RoomController'); Route::resource('tools', 'ToolController'); Route::get('maintenances/roomtool', ['as' => 'maintenances.roomtool', 'uses' => 'MaintenanceController@roomtool']); Route::get('maintenances/roomtool/create', ['as' => 'maintenances.createroomtool', 'uses' => 'MaintenanceController@createroomtool']); Route::post('maintenances/roomtool/create', ['as' => 'maintenances.storerroomtool', 'uses' => 'MaintenanceController@storerroomtool']); Route::delete('maintenances/roomtool/destroy', ['as' => 'maintenances.destroyroomtool',</pre>

		<pre> 'uses' => 'MaintenanceController@destroyroomtool']); Route::get('maintenances/messages', ['as' => 'maintenances.messages', 'uses' => 'MaintenanceController@messages']); Route::get('maintenances/routines', ['as' => 'maintenances.routines', 'uses' => 'MaintenanceController@routines']); Route::put('maintenance/routines', ['as' => 'maintenance.updateroutines', 'uses' => 'MaintenanceController@updateRoutine']); Route::get('/maintenances/history', 'FaultController@h istory; Route::resource('maintenances', 'MaintenanceController'); Route::put('/faults/{id}', 'FaultController@update'); Route::get('/faults/index', 'FaultController@index'); Route::post('/faults/index', 'FaultController@filter'); Route::get('faults/plain', 'FaultController@plain');}); </pre>
5	<p>Pelaporan Sarana Rusak</p>	<pre> Route::group(['roles' => ['staff', 'maintenance']], function(){ Route::post('/faults', 'FaultController@store'); Route::get('/faults', 'FaultController@create'); Route::post('/faults/index', 'FaultController@filter'); Route::get('/faults/history', 'FaultController@index'); }); </pre>
6	<p>Pengolahan data pemesanan khusus Manager</p>	<pre> Route::group(['roles' => ['manager']], function(){ Route::get('orders/new', ['as' => 'orders.new', 'uses' => 'OrderController@newOrder']); Route::get('orders/confirm', ['as' => 'orders.confirm', 'uses' => 'OrderController@confirmedOrder']); Route::put('orders/accept', ['as' => 'orders.accept', 'uses' => 'OrderController@accept']); Route::put('orders/reject', ['as' => 'orders.reject', 'uses' => 'OrderController@reject']); }); </pre>

7.	Pemesanan Produk	<pre>Route::group(['roles' => ['marketing']], function() { Route::get('products', ['as' => 'products', 'uses' => 'OrderController@items']); Route::get('products/cat/{id}', ['as' => 'products.category', 'uses' => 'OrderController@itemsCategory']); Route::get('products/{id}', ['as' => 'products.show', 'uses' => 'OrderController@productShow']); Route::get('carts', ['as' => 'carts', 'uses' => 'OrderController@carts']); Route::post('carts', ['as' => 'carts.store', 'uses' => 'OrderController@cartsStore']); Route::get('orders/create', ['as' => 'orders.create', 'uses' => 'OrderController@create']); Route::delete('carts/{carts}', ['as' => 'carts.destroy', 'uses' => 'OrderController@cartsDestroy']);});</pre>
8	Pengolahan data order	<pre>Route::group(['roles'=>['marketing', 'manager']], function(){ Route::get('orders/plain/{id}', ['as' => 'orders.plain', 'uses' => 'OrderController@plain']); Route::get('orders/{order}', ['as' => 'orders.show', 'uses' => 'OrderController@show']); Route::resource('orders', 'OrderController'); });});</pre>

5.5.3 Implementasi Blade Templating pada View

Blade adalah template *engine* bawaan dari kerangka kerja *laravel*. Dengan fitur ini memungkinkan pengembang untuk membuat template yang lebih terstruktur dan dinamis. *Blade* diibaratkan sebagai sebuah pedang, yang akan memotong - motong struktur pada template sehingga lebih mudah dalam pengembangan. Dengan menggunakan fitur *blade* memungkinkan pengembang membuat satu *layout* yang dapat digunakan oleh beberapa view sekaligus, sehingga dapat dengan mudah mengatur *layout web* yang akan dibangun.

Tabel 5.6 dibawah menunjukkan *layout blade templating* yang diimplementasikan pada sistem pemeliharaan sarana dan pemantauan pemasaran produksi berbasis web PT.Pancamas Elite. Untuk mengimplementasikan view *layout blade templating* dimulai dengan membuat sebuah *master layout* yang berisi beberapa bagian atau potongan *layout* yang akan digunakan dalam implementasi sistem, *layout* dalam file ini dapat digunakan oleh view lain dengan cara memanggil command *extend*. Sedangkan command

section('content') digunakan untuk memanggil view yang diletakan ditempat dimana ada perintah yield('content') pada layout yang telah didefinisikan sebelumnya dalam master layout.

Tabel 5.6 Implementasi Blade Templating

No	Class	Nama File Blade View	Blade Template Command
1	categories	create.blade.php index.blade.php edit.blade.php show.blade.php	@extends('admin') @section('content')
2	faults	create.blade.php	@extends('admin') @section('additionalJs') @section('content')
		history.blade.php index.blade.php	@extends('admin') @section('content')
		report.blade.php	-
3	items	create.blade.php index.blade.php edit.blade.php show.blade.php	@extends('admin') @section('content')
4	maintenances	create.blade.php edit.blade.php show.blade.php createroomtool.blade.php index.blade.php messages.blade.php roomtool.blade.php	@extends('admin') @section('content')
5	orders	create.blade.php edit.blade.php show.blade.php carts.blade.php index.blade.php showStorage.blade.php	@extends('admin') @section('content')

		showPlain.blade.php invoice.blade.php	
		products.blade.php productshow.blade.php	@section('additionalJs') @extends('admin') @section('content')
6	profile	index.blade.php edit.blade.php	@extends('admin') @section('content')
7	rooms	create.blade.php index.blade.php edit.blade.php show.blade.php	@extends('admin') @section('content')
8	tools	create.blade.php index.blade.php edit.blade.php show.blade.php	@extends('admin') @section('content')

5.6 Implementasi Antarmuka

Implementasi antarmuka pada sistem informasi ini terdiri atas 5 bagian, yaitu Antarmuka Staf Pemeliharaan, Antarmuka Unit Kerja, Antarmuka Staff Pemasaran, Antarmuka Admin Pemasaran dan Antarmuka Gudang.

5.6.1 Implementasi Antarmuka Staff Pemeliharaan

a. Login

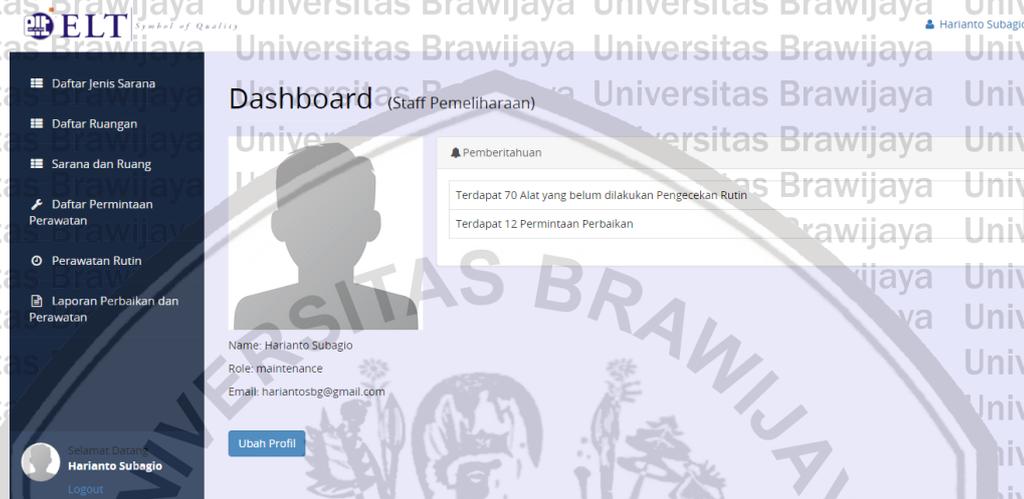
Sebelum aktor bisa masuk ke dalam sistem, aktor perlu memasukkan data *username* dan *password* yang dimiliki. Gambar 5.1 menunjukkan tampilan login sistem.



Gambar 5.1 Antarmuka Login

b. Dashboard

Di dalam antarmuka dashboard ini pengguna akan langsung masuk pada fitur dashboard user staff pemeliharaan sehingga dapat melihat informasi tentang profil user beserta fitur yang disediakan. Pada profil user pengguna bisa melakukan update terhadap informasi dari pengguna. Selain itu terdapat fitur pemberitahuan untuk melihat status sarana yang ada di PT.Pancamas Elite. Gambar 5.2 menunjukkan tampilan dashboard sistem.



Gambar 5.2 Antarmuka Dashboard Staf Pemeliharaan

c. Daftar Jenis Sarana

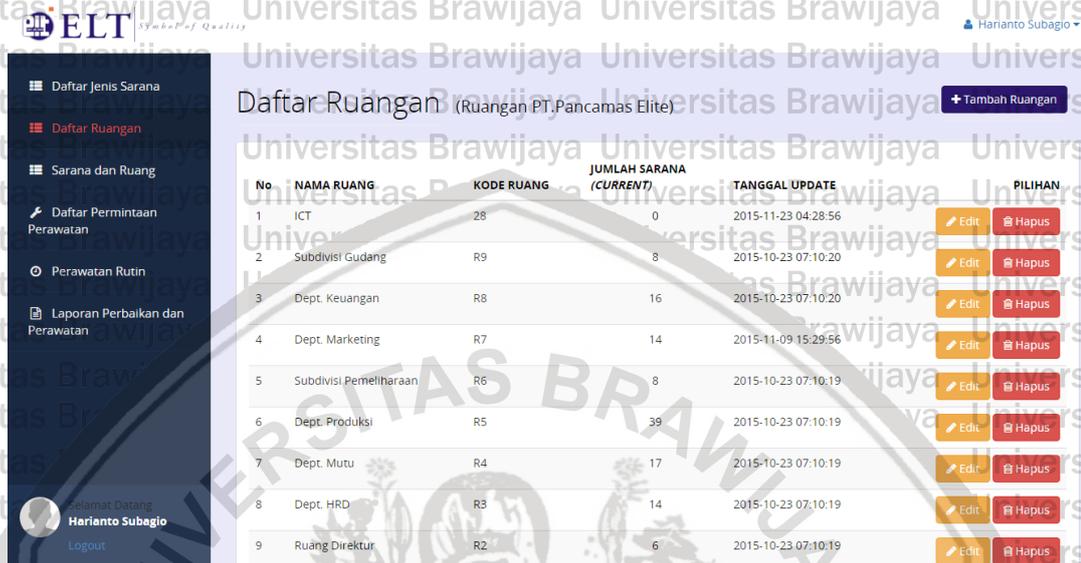
Di dalam antarmuka ini pengguna dapat melihat informasi tentang data jenis sarana yang ada di dalam perusahaan. Gambar 5.3 menunjukkan tampilan daftar jenis sarana pada sistem



Gambar 5.3 Antarmuka Daftar Jenis Sarana

d. Daftar Ruangan

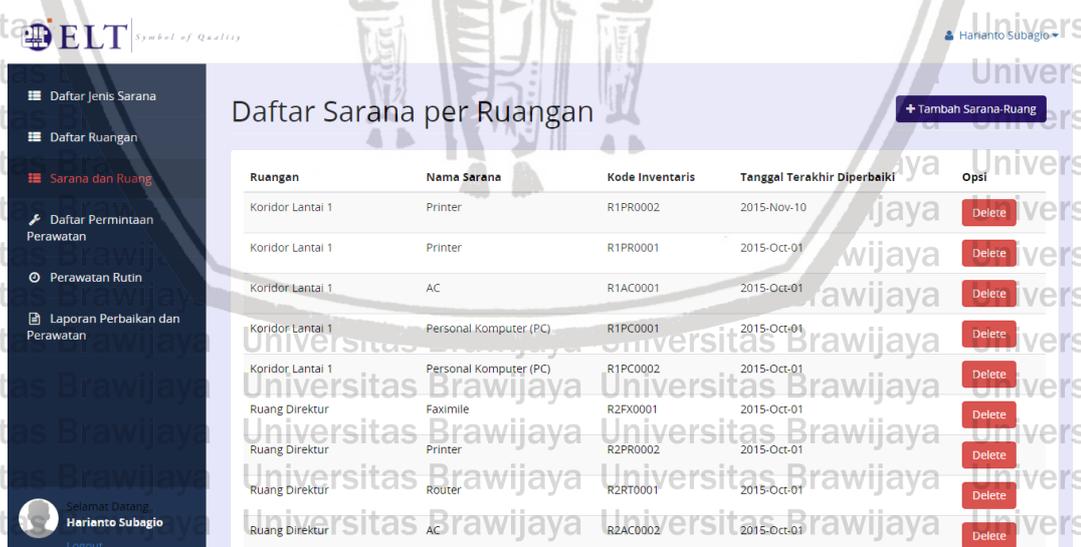
Di dalam antarmuka ini pengguna dapat melihat informasi tentang data daftar ruangan beserta jumlah sarana yang ada di dalam ruangan tersebut dan informasi lengkapnya. Gambar 5.4 menunjukkan tampilan daftar jenis sarana pada sistem



Gambar 5.4 Antarmuka Daftar Ruangan

e. Sarana dan Ruang

Di dalam antarmuka ini pengguna dapat melihat informasi tentang data jenis sarana tiap ruangan beserta tanggal terakhir diperbaiki. Gambar 5.5 menunjukkan tampilan daftar jenis sarana per ruangan pada sistem



Gambar 5.5 Antarmuka Sarana dan Ruang

f. Daftar Permintaan Perawatan

Di dalam antarmuka ini staff pemeliharaan dapat melihat informasi tentang daftar permintaan perbaikan yang diajukan oleh unit kerja. Gambar 5.6 menunjukkan tampilan daftar permintaan perawatan pada sistem.

Daftar Permintaan Perbaikan (Perawatan Sarana PT.Pancamas Elite)

Nama Sarana	Ruangan	Kode Inventaris	Tanggal Kerusakan	Keterangan	Status	Pegawai	Opsi
Mesin bubut	Dept. Produksi	R5B0003	2015-Dec-18	macet	Perbaikan Selesai	Tika Maharani	Tidak Ada Opsi Lagi
Personal Komputer (PC)	Dept. Marketing	R7PC0020	2015-Dec-18	kena virus	Perbaikan Selesai	Tika Maharani	Tidak Ada Opsi Lagi
Mesin las	Dept. Produksi	R5LS0003	2015-Dec-18	macet	Perbaikan Selesai	Tika Maharani	Tidak Ada Opsi Lagi
Printer	Dept. HRD	R3PR0004	2015-Dec-18	tinta habis	Perbaikan Selesai	Tika Maharani	Tidak Ada Opsi Lagi
Mesin milling	Dept. Mutu	R4ML0001	2015-Dec-17	macet	Sedang Diperbaiki	Tika Maharani	Perbaikan Selesai
Mesin bor	Dept. Produksi	R5BR0003	2015-Nov-25	berkarat	Belum Diproses	Tika Maharani	Perbaiki sekarang
Personal Komputer (PC)	Dept. Mutu	R4PC0009	2015-Nov-25	terkena virus	Belum Diproses	Tika Maharani	Perbaiki sekarang

Gambar 5.6 Antarmuka Daftar Permintaan Perawatan

g. Perawatan Rutin

Di dalam antarmuka ini staff pemeliharaan dapat melihat informasi tentang daftar sarana yang harus diperiksa untuk perawatan rutin beserta status perawatan sarana yang ada. Gambar 5.7 menunjukkan tampilan daftar perawatan rutin sarana yang ada di PT.Pancamas Elite.

Daftar Perawatan Rutin (Sarana PT. Pancamas Elite)

Terdapat 70 peralatan yang belum dilakukan perbaikan rutin.

Ruangan	Nama Sarana	Kode Inventaris	Lama Belum diperbaiki	Terakhir Diperbaiki	Opsi
Dept. Produksi	Mesin CNC	R5CN0003	49 dari 30 hari	2015-11-12	Baik Perbaiki
Dept. Produksi	Mesin CNC	R5CN0004	49 dari 30 hari	2015-11-12	Baik Perbaiki
Dept. Produksi	Mesin CNC	R5CN0005	49 dari 30 hari	2015-11-12	Baik Perbaiki
Dept. Produksi	Mesin Asah	R5AS0001	49 dari 30 hari	2015-11-12	Baik Perbaiki
Dept. Produksi	Mesin Asah	R5AS0002	49 dari 30 hari	2015-11-12	Baik Perbaiki
Dept. Produksi	Mesin Asah	R5AS0003	49 dari 30 hari	2015-11-12	Baik Perbaiki
Dept. Produksi	Mesin Asah	R5AS0004	49 dari 30 hari	2015-11-12	Baik Perbaiki
Dept. Mutu	Mesin Frais	R4FR0001	49 dari 30 hari	2015-11-12	Baik Perbaiki

Gambar 5.7 Antarmuka Daftar Perawatan Rutin

h. Laporan Perbaikan Perawatan

Di dalam antarmuka ini staff pemeliharaan dapat melihat informasi tentang daftar laporan sarana yang telah dilakukan perbaikan dan perawatan. Gambar 5.8 menunjukkan tampilan daftar laporan perbaikan dan perawatan sarana yang ada di PT.Pancamas Elite

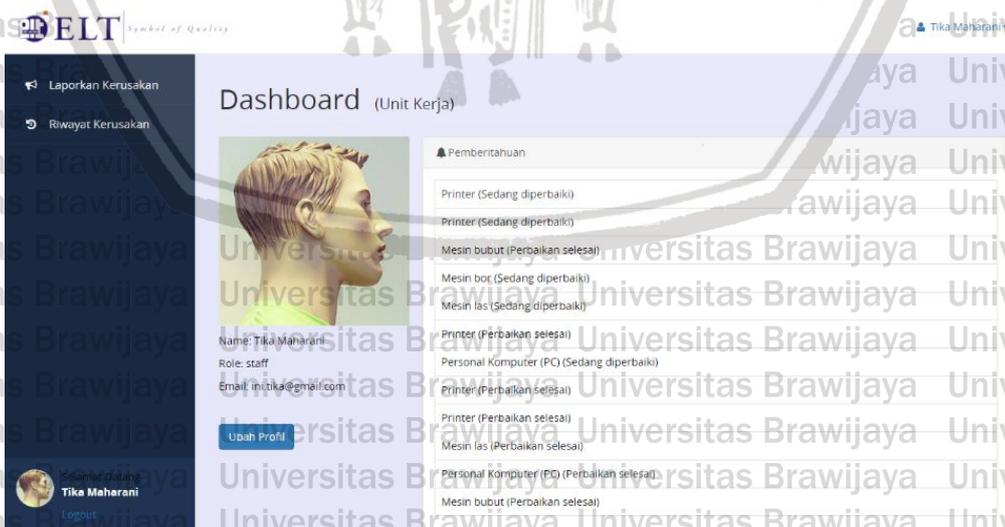


Gambar 5.8 Antarmuka Daftar Laporan Perbaikan dan Perawatan

5.6.2 Implementasi Antarmuka Unit Kerja

a. Dashboard

Di dalam antarmuka dashboard ini pengguna akan langsung masuk pada fitur dashboard user unit kerja sehingga dapat melihat informasi tentang profil user beserta fitur yang disediakan. Pada profil user pengguna bisa melakukan update terhadap informasi dari pengguna. Selain itu terdapat fitur pemberitahuan untuk melihat status sarana yang diajukan pengajuan perbaikan sarana. Gambar 5.9 menunjukkan tampilan dashboard sistem.



Gambar 5.9 Antarmuka Dashboard Unit Kerja

b. Laporkan Kerusakan

Di dalam antarmuka ini unit kerja dapat mengisi laporan permintaan perbaikan yang selanjutnya akan muncul pemberitahuan di staf pemeliharaan.

Gambar 5.10 menunjukkan tampilan form laporan kerusakan.

Formulir Permintaan Perbaikan (Perawatan Sarana PT.Pancamas Elite)

Nama Ruangant: --- Pilih Ruangant ---

Nama Sarana: --- Pilih Sarana ---

Kode Inventaris: --- Pilih Kode Inventaris ---

Tanggal Kerusakan: 2015-12-31 14:46:06

Keterangan: (Opsional)

Buttons: Back, Laporkan

Gambar 5.10 Antarmuka Form Permintaan Perbaikan

c. Daftar Permintaan Perbaikan

Di dalam antarmuka ini unit kerja dapat melihat daftar riwayat pengajuan perbaikan yang telah dilakukan oleh user tersebut. Gambar 5.11 menunjukkan tampilan riwayat pengajuan perbaikan.

Daftar Permintaan Perbaikan (Perawatan Sarana PT.Pancamas Elite)

Filter Riwayat: 2015 Jan Q Filter! Clear Filter

Nama Sarana	Ruangant	Kode Inventaris	Tanggal Kerusakan	Keterangan	Status
Mesin bubut	Dept. Produksi	R5BB0003	2015-Dec-18	macet	Perbaikan Selesai
Personal Komputer (PC)	Dept. Marketing	R7PC0020	2015-Dec-18	kena virus	Perbaikan Selesai
Mesin las	Dept. Produksi	R5LS0003	2015-Dec-18	macet	Perbaikan Selesai
Printer	Dept. HRD	R3PR0004	2015-Dec-18	tinta habis	Perbaikan Selesai
Mesin milling	Dept. Mutu	R4ML0001	2015-Dec-17	macet	Sedang Diperbaiki
Mesin bor	Dept. Produksi	R5BR0003	2015-Nov-25	berkarat	Belum Diproses
Personal Komputer (PC)	Dept. Mutu	R4PC0009	2015-Nov-25	terkena virus	Belum Diproses
Printer	Subdivisi Pemeliharaan	R6PR0008	2015-Nov-16	tinta habis	Perbaikan Selesai
Scanner	Dept. HRD	R3SC0001	2015-Nov-13	Macet	Belum Diproses
Mesin las	Dept. Produksi	R5LS0001	2015-Nov-12	rusak	Sedang Diperbaiki
Printer	Dept. Keuangan	R8PR0012	2015-Nov-06	Tinta Habis	Sedang Diperbaiki

Gambar 5.11 Antarmuka Daftar Permintaan Perbaikan

5.6.3 Implementasi Antarmuka Staff Pemasaran

a. Dashboard

Di dalam antarmuka dashboard ini pengguna akan langsung masuk pada fitur dashboard user staff pemasaran sehingga dapat melihat informasi tentang profil user beserta fitur yang disediakan. Pada profil user pengguna bisa melakukan update terhadap informasi dari pengguna. Selain itu terdapat fitur pemberitahuan untuk melihat status pemesanan produk. Gambar 5.12 menunjukkan tampilan dashboard sistem.



Gambar 5.12 Antarmuka Dashboard Staff Pemasaran

b. Kategori Produk

Di dalam antarmuka ini staf pemasaran dapat melihat kategori produk yang ada di PT.Pancamas Elite untuk selanjutnya dipilih untuk pemesanan. Gambar 5.13 dan 5.14 menunjukkan tampilan kategori produk yang ada di PT.Pancamas Elite.



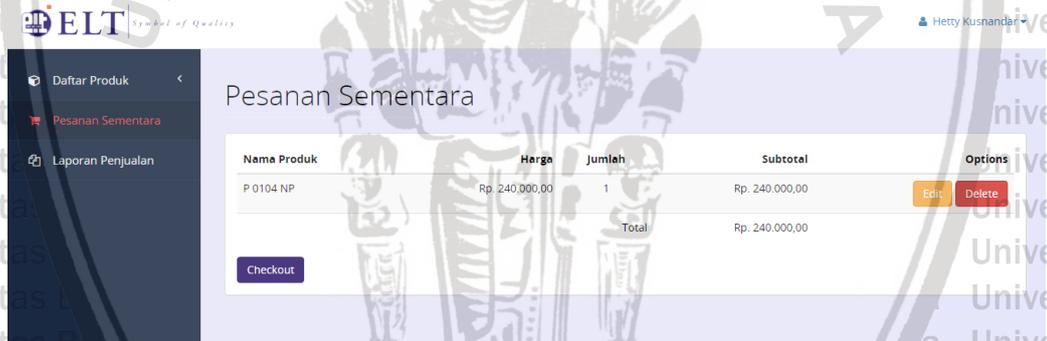
Gambar 5.13 Antarmuka Kategori Produk



Gambar 5.14 Antarmuka Kategori Produk

c. Pesanan Sementara

Di dalam antarmuka ini staf pemasaran dapat melihat daftar produk yang masuk dalam pesanan sementara sebelum dilakukan *checkout*. Gambar 5.15 menunjukkan tampilan daftar pesanan sementara.



Gambar 5.15 Antarmuka Pesanan Sementara

d. Tambah Identitas Pemesan

Di dalam antarmuka ini staf pemasaran dapat menambah pemesanan baru dengan mengisi identitas pemesan setelah dilakukan *checkout* pada halaman pesanan sementara. Gambar 5.16 menunjukkan tampilan halaman tambah identitas pemesan

Tambah Identitas Pemesan

NAMA PEMBELI

ALAMAT

KOTA

KODE POS

TELEPON

EMAIL

NAMA BANK

NAMA AKUN BANK

Back Create

Gambar 5.16 Antarmuka Tambah Identitas Pemesan

d. Laporan Penjualan

Di dalam antarmuka ini staf pemasaran dapat melihat daftar laporan penjualan produk dan melihat faktur penjualan atau *invoice*. Faktur penjualan akan tampil jika pemesanan produk sudah dikonfirmasi oleh admin pemasaran dan gudang. Gambar 5.17 menunjukkan tampilan daftar laporan penjualan.

Laporan Penjualan

Nama Petugas	Pembeli	Alamat Pembeli	Total Item	Telepon	Nama Bank	Total Harga	PILIHAN
hetty	Nabara Saraswati	Jl Puncak Borobudur 1, Mojolangu, Sukun, Malang, Jawa Timur 65142	1	081445127675	bri	Rp. 150.000,00	Hapus
hetty	Nita Daniarti	Jl Pahlawan Trip 2, Oro-oro Dowo, Klojen, Malang, Jawa Timur 65112	3	081445127675	bca	Rp. 480.000,00	Hapus
hetty	Birma Mentaya UD	Ds Tasikmadu, Tasikmadu, Sukun, Malang, Jawa Timur 65143	4	+60 341 486679	bni	Rp. 800.000,00	Hapus
hetty	Anita Junardi	Jl. Bendungan Sigura-gura 18 Lowokwaru, Malang	1	081445127675	bni	Rp. 150.000,00	Hapus
hetty	Ari Sudrajat	Jl. Panglima Sudirman No.9 RT/RW 13/03 Kemirahan, Malang	2	085643213442	bni	Rp. 300.000,00	Hapus
hetty	Siti Sumarti	Jl. Danau Maninjau No.23 Sawojajar, Malang	2	081334126543	bri	Rp. 510.000,00	Faktur Hapus
hetty	Lita Surdani	Jl. Kapten Untung 18 RT/RW 12/02 Lowokwaru, Malang	2	089567452123	mandiri	Rp. 205.000,00	Faktur Hapus

Gambar 5.17 Antarmuka Laporan Penjualan

5.6.4 Implementasi Antarmuka Admin Pemasaran

a. Dashboard

Di dalam antarmuka dashboard ini pengguna akan langsung masuk pada fitur dashboard user admin pemasaran sehingga dapat melihat informasi tentang profil user beserta fitur yang disediakan. Pada profil user pengguna bisa melakukan update terhadap informasi dari pengguna. Selain itu terdapat fitur pemberitahuan untuk melihat status ketersediaan produk yang dipesan dari bagian gudang. Gambar 5.18 menunjukkan tampilan dashboard sistem.

Dashboard (Admin Pemasaran)

▲ Pemberitahuan

- 18 Total Transaction
- 6 Transaction Ready
- 2 Transaction Accepted
- 7 Transaction Not Processed

Name: Rudi Sugianto
Role: manager
Email: rudisugianto@gmail.com

Ubah Profil

Selamat Datang
Rudi Sugianto
Logout

Gambar 5.18 Antarmuka Dashboard Admin Pemasaran

b. Pesanan Baru

Di dalam antarmuka ini admin pemasaran dapat melihat daftar laporan pesanan baru dari staf pemasaran untuk kemudian dilakukan konfirmasi. Gambar 5.19 menunjukkan tampilan daftar laporan pesanan baru

Pesanan Baru

Nama Petugas	Pembeli	Alamat Pembeli	Total Item	Telepon	Nama Bank	Total Harga	PILIHAN
hetty	CV. Pance Jaya	Jl. Selat,Sunda III/D4-2, Lesanpuro - Kedungkandang	1	081567123765	bni	Rp. 160.000,00	Accept Reject
hetty	Dina Sari	Jl. Meretojoyo BS Malang	1	+6287456345235	bni	Rp. 160.000,00	Accept Reject
hetty	Kanthy Raras	Jl. Koptu Djalis RT/RW 14/04 Malang	1	+6287456345235	bni	Rp. 150.000,00	Accept Reject
hetty	hani rinda	Jl.koptu jatemo, mergan	1	+6287456345235	bni	Rp. 150.000,00	Accept Reject
hetty	Kanthy raras	Jl. malabar	1	089234123	bni	Rp. 300.000,00	Accept Reject
hetty	Kanthy Raras	Jl. Iskandar Muda 18 Malang	1	+6287456345235	bni	Rp. 150.000,00	Accept Reject
hetty	Muhammad Azam	Jl. Koptu Djalis RT/RW 14/04 Sedayu, Turen	1	+6287456345235	bni	Rp. 150.000,00	Accept Reject
hetty	Yuli Rahmawati	Jl. Letjen Sungkono, Kedungkandang	1	+6282337453993	mandiri	Rp. 150.000,00	Accept Reject
hetty	Fani Bagus Ramadani	Jl. Pandanwangi, Pronojowo	1	+828556789815	boa	Rp. 150.000,00	Accept Reject

Selamat Datang
Rudi Sugianto
Logout

Gambar 5.19 Antarmuka Pesanan Baru

c. Pesanan Terproses

Di dalam antarmuka ini admin pemasaran dapat melihat daftar laporan status pesanan baru yang telah diproses untuk melihat ketersediaan produk tersebut yang dihubungkan dengan aktor gudang. Gambar 5.20 menunjukkan tampilan daftar laporan status pesanan terproses.

Pesanan Terproses

Nama Pecugas	Nama Pembeli	Alamat Pembeli	Kota	Total Item	Telepon	Nama Bank	Total Harga	Status	PILIHAN
hetty	Kantheni Raras	Jl. Mekarsari Gg. XI, Bangil	Pasuruan	1	+6287456345235	BNI	Rp. 160.000,00	accepted delayed ready	Faktur
hetty	Supriyono	Jl. Masjid Besar 12, Wonokromo	Surabaya	1	+6287456345235	BRI	Rp. 270.000,00	accepted ready	Faktur
hetty	Wawan Gunawan	Jl. Dirgantara 45, Sukun	Malang	1	+628970315183	BCA	Rp. 270.000,00	accepted ready	Faktur
hetty	Rizki Sepasthika Fatmapiaria	Jl. Mercubuana 32	Tulungagung	1	+6287031518328	BNI	Rp. 260.000,00	accepted ready	Faktur
hetty	Doni Chandra	Jl. Danau Maninjau 14, Sawojajar	Malang	1	+6287456345235	BNI	Rp. 170.000,00	accepted delayed ready	Faktur

Gambar 5.20 Antarmuka Pesanan Terproses

d. Faktur Penjualan

Di dalam antarmuka ini admin pemasaran dapat melihat *print preview* faktur penjualan sebelum dicetak Gambar 5.21 menunjukkan tampilan antarmuka faktur pembayaran.

Invoice #00031

Nama Pembeli: Kantheni Raras
Jl. Jendral Sudirman 32, Turen Malang

Product By: PT. Pancamas Elite
Alamat: Jl. Perusahaan 32, Tunjungtirto, Karanglo, Malang, 65153
Phone: (0341) 494187

Pembayaran:
Nama Bank: BCA
A/N: Raras

Nama Produk	Harga	Jumlah	Subtotal
R 1002 AB	Rp. 160.000,00	1	Rp. 160.000,00
HB 08	Rp. 120.000,00	2	Rp. 240.000,00
R 2002 AC	Rp. 190.000,00	2	Rp. 380.000,00
Total			Rp. 780.000,00

Informasi Bank
Bank: BRI
No Rekening: 3279-01-021391-53-1
a/n: PT. Pancamas Elite

Kontak
PT Pancamas Elite
Alamat: Jl. Perusahaan 32, Tunjungtirto, Karanglo, Malang, 65153
Phone: (0341) 494187

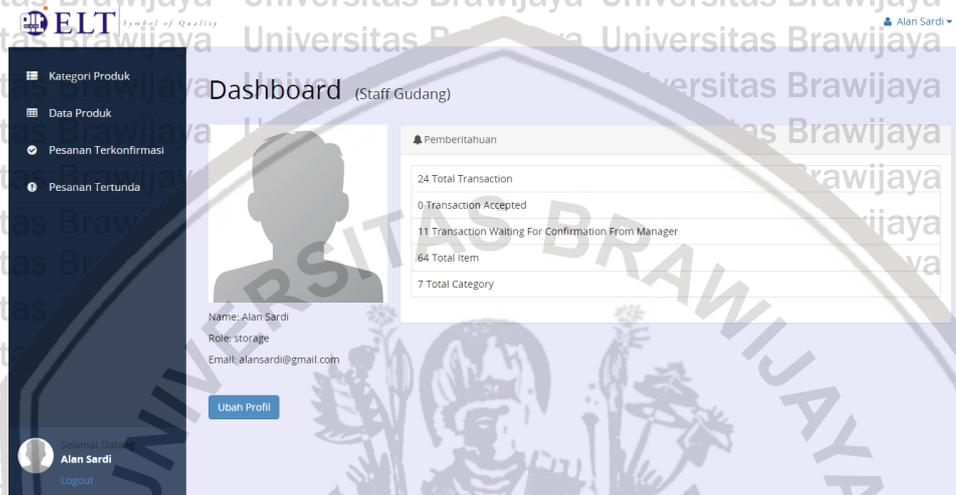
Malang,
hetty

Gambar 5.21 Antarmuka Faktur Penjualan

5.6.5 Implementasi Antarmuka Gudang

a. Dashboard

Di dalam antarmuka dashboard ini pengguna akan langsung masuk pada fitur dashboard user gudang sehingga dapat melihat informasi tentang profil user beserta fitur yang disediakan. Pada profil user pengguna bisa melakukan update terhadap informasi dari pengguna. Selain itu terdapat fitur pemberitahuan untuk melihat status ketersediaan produk yang ada di dalam gudang. Gambar 5.22 menunjukkan tampilan dashboard sistem.



Gambar 5.22 Antarmuka Dashboard Antarmuka Gudang

b. Kategori Produk

Di dalam antarmuka ini gudang dapat melihat daftar kategori produk produk yang dihasilkan di PT.Pancamas Elite . Gambar 5.23 menunjukkan tampilan daftar kategori produk.



Gambar 5.23 Antarmuka Kategori Produk

c. Daftar Produk

Di dalam antarmuka ini gudang dapat melihat daftar data produk beserta jumlah ketersediaannya dan status stok dari produk tersebut . Gambar 5.24 menunjukkan tampilan daftar data produk yang ada di PT.Pancamas Elite.



Gambar 5.24 Antarmuka Daftar Produk

d. Pesanan Terkonfirmasi

Di dalam antarmuka ini gudang dapat melihat daftar data produk yang telah disetujui atau dikonfirmasi oleh admin pemasaran untuk selanjutnya dilihat ketersediaan produknya di menu data produk. Gambar 5.25 menunjukkan tampilan daftar pesanan terkonfirmasi



Gambar 5.25 Antarmuka Pesanan Terkonfirmasi

e. Pesanan Tertunda

Di dalam antarmuka ini gudang dapat melihat daftar data produk yang masuk dalam daftar *delay* karena ketersediaan produk yang masih kosong atau masih tertunda ketersediaannya. Gambar 5.26 menunjukkan tampilan daftar pesanan tertunda.

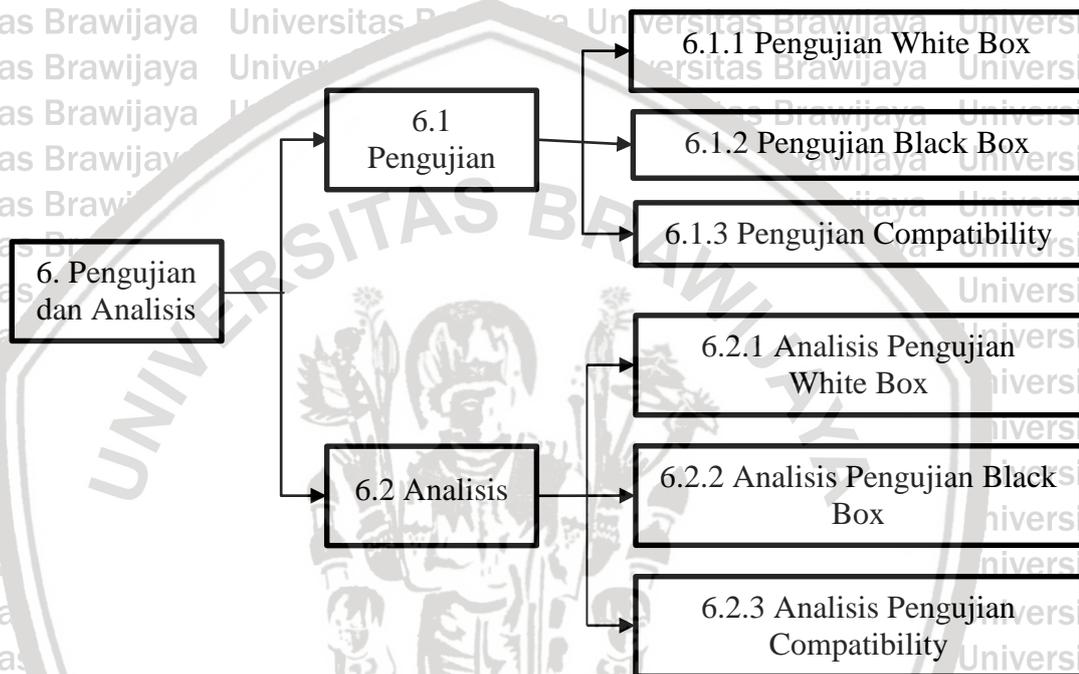
Pesanan Tertunda

Nama Petugas	Pembeli	Alamat Pembeli	Kota	Total Item	Telepon	Nama Bank	Total Harga	PILIHAN
hetty	Mayantara	Jl. LA. Sudipto 359 Pandanwangi Blimbing	Malang	1	+6287456345235	MANDIRI	Rp. 160.000,00	Ready Delete
hetty	Agung Sudrajat	Jl. Wahid Hasyim Gang 08 Belung-Poncokusumo-Kab.malang	Malang	1	+6287456345235	BCA	Rp. 160.000,00	Ready Delete
hetty	Anita Santi	Perum Asrikaton Indah Blok K1 No.10 Pakis Malang	Malang	2	+6287456333359	BNI	Rp. 310.000,00	Ready Delete

Gambar 5.26 Antarmuka Daftar Pesanan Tertunda

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas tahapan Pengujian dan Analisis Sistem Informasi Pemeliharaan Sarana dan Pemantauan Pemasaran Produk PT.Pancamas Elite. Pembahasan terdiri atas penjelasan tentang algoritma pengujian yang digunakan beserta hasil pengujiannya dan analisa dari hasil pengujian.



Gambar 6.1 Diagram Blok Bab Pengujian dan Analisis

6.1 Pengujian Sistem

Pengujian dilakukan dengan tiga tahapan yaitu pengujian *white box*, pengujian *black box* dan *compatibility*. Seluruh proses pengujian ini dilakukan dengan menggunakan spesifikasi perangkat keras dan perangkat lunak sesuai dengan sub bab 5.1 yang telah dijelaskan sebelumnya. Pengujian *Black Box* dilakukan dengan menguji satu persatu fitur yang ada sesuai dengan kebutuhan fungsional, pengujian *white box* dilakukan dengan melihat ke dalam modul untuk meneliti kode kode program dan menganalisis ada kesalahan atau tidak dan pengujian *compatibility* dilakukan dengan menguji sistem pada sistem operasi yang berbeda.

6.1.1 Pengujian *WhiteBox*

Pengujian *whitebox* dilakukan dengan pengujian *Basis Path*. Proses pengujian *basis path* merupakan proses pengujian unit yang dilakukan dengan menggunakan pseudocode yang menerapkan metode atau logika yang digunakan oleh sistem. Pengujian basis path ditunjukkan pada gambar 6.1 hingga gambar 6.4.

6.1.1.1 Tujuan

Tujuan dilakukannya pengujian white box adalah untuk menguji semua statement program pada beberapa algoritma yang memiliki prioritas *high* atau *medium* yang telah diimplementasikan sesuai dengan yang diharapkan.

6.1.1.2 Mekanisme

Pengujian *whitebox* dilakukan dengan membuat *flowgraph* dari pseudocode dilanjutkan dengan menghitung nilai *cyclomatic complexity* dan menentukan jalur independen *path* serta memaparkan kasus ujiannya. Pengujian *whitebox* ini dilakukan pada tiga fitur yaitu menampilkan pemberitahuan, pemesanan produk dan menambah sarana dan ruang.

6.1.1.3 Hasil Pengujian

a. Kasus uji Halaman Pemberitahuan Pengguna

Berikut adalah pseudo code menampilkan pemberitahuan yang nantinya digunakan untuk mencari *flowgraph* dan melanjutkan pada tahap berikutnya ditunjukkan pada tabel 6.1 berikut :

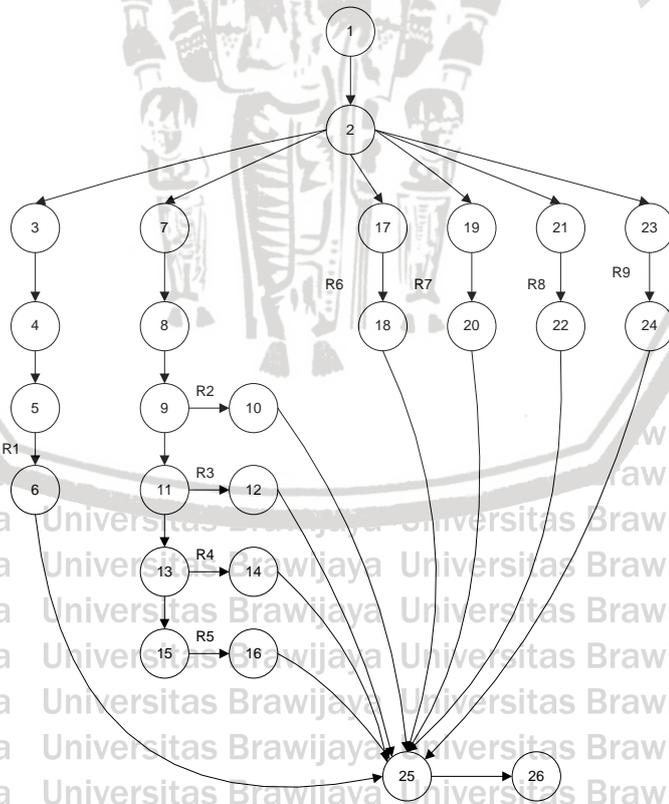
Tabel 6.1 Pseudocode Menampilkan Pemberitahuan

	Nama Algoritma: ALGO_CD_01
	Deskripsi: Algoritma untuk melakukan proses pemberitahuan
	Masukan: role, fault, count, countReady, countAccepted, countNotProcessed
	Proses:
1	Run library autentikasi
2	switch (\$profile, \$role, \$name)
3	case 'staff'
4	Get \$faults = Auth::user()->username)
5	foreach (\$faults as \$fault)
6	Tampilkan notifikasi
7	case 'maintenance'
8	Get data pemeliharaan
9	if (count(\$count1) == 0
10	Semua sarana telah dilakukan pengecekan rutin
11	else
12	Terdapat sarana yang belum dilakukan pengecekan rutin

```

13 if ($count2 == 0)
14     Tidak ada sarana rusak
15 else
16     Terdapat permintaan perbaikan
17     case 'marketing'
18     get$countTotal,$countReady,$countAccepted,$countNotProcessed,
19     $notification
20     case 'manager'
21     get$countTotal,$countReady,$countAccepted,$countNotProcessed,
22     $notification
23     case 'storage'
24     get$countTotal,$countAccepted,$countCategory,$countNotProcessed,
25     $countItem, $notification
26     default:
27     Halaman Tidak Tersedia
28 return view ('profile.index', compact ('profile', 'notification'));
29 END
    
```

Flowgraph dari pseudocode kasus uji kasus menampilkan pemberitahuan akan ditunjukkan pada gambar 6.2 dibawah ini



Gambar 6.2 Flowgraph Menampilkan Pemberitahuan

Berdasarkan *flowgraph* diatas dapat dihitung nilai *cyclomatic complexity* sebagai berikut :

$$V(G) = 9 \text{ region}$$

$$V(G) = \text{Jumlah region}$$

$$= 9$$

$$V(G) = E - N + 2$$

$$= 33 - 26 + 2$$

$$= 9$$

Sehingga berdasarkan hasil persamaan *cyclomatic complexity* maka didapatkan 9 jalur independen yaitu :

1. Jalur 1 : 1-2-3-4-5-6-25-26

2. Jalur 2 : 1-2-7-8-9-10-25-26

3. Jalur 3 : 1-2-7-8-9-11-12-25-26

4. Jalur 4 : 1-2-7-8-9-11-13-14-25-26

5. Jalur 5 : 1-2-7-8-9-11-13-15-16-25-26

6. Jalur 6 : 1-2-17-18-25-26

7. Jalur 7 : 1-2-19-20-25-26

8. Jalur 8 : 1-2-21-22-25-26

9. Jalur 9 : 1-2-23-24-25-26

Berdasarkan 9 jalur independen yang telah didefinisikan diatas maka dapat dibentuk kasus ujinya.

Tabel 6.2 Kasus Uji Halaman Pemberitahuan Pengguna

Jalur 1	<p>Nilai (role=staff, fault=status)</p> <p>Hasil yang diharapkan = Sistem menampilkan halaman dashboard unit kerja dan menampilkan notifikasi status pengajuan sarana</p> <p>Hasil Uji = valid</p>
Jalur 2	<p>Nilai (role=maintenance, count1=0) = perawatan rutin selesai</p> <p>Hasil yang diharapkan = Sistem menampilkan halaman dashboard staf pemeliharaan dan menampilkan notifikasi semua sarana telah dilakukan pengecekan rutin</p> <p>Hasil Uji = valid</p>
Jalur 3	<p>Nilai (role=maintenance, count1=ada) = perawatan rutin belum selesai</p> <p>Hasil yang diharapkan = Sistem akan menampilkan halaman dashboard staf pemeliharaan dan menampilkan notifikasi terdapat alat yang belum dilakukan pengecekan rutin</p>

Hasil Uji = valid

Jalur 4 Nilai (role=maintenance, count2=0) = tidak ada permintaan perbaikan

Hasil yang diharapkan = Sistem akan menampilkan halaman dashboard staf pemeliharaan dan menampilkan notifikasi tidak ada sarana yang rusak

Hasil Uji = valid

Jalur 5 Nilai (role=maintenance, count2=ada) = terdapat permintaan perbaikan

Hasil yang diharapkan = Sistem akan menampilkan halaman dashboard staf pemeliharaan dan menampilkan notifikasi terdapat permintaan perbaikan

Hasil Uji = valid

Jalur 6 Nilai (role=marketing, countTotal, countReady, countAccepted, countNotProcessed, notification) = true

Hasil yang diharapkan = Sistem akan menampilkan halaman dashboard staf pemasaran dan menampilkan notifikasi pemesanan yang diajukan

Hasil Uji = valid

Jalur 7 Nilai (role=manager, countTotal, countReady, countAccepted, countNotProcessed, notification) = true

Hasil yang diharapkan = Sistem akan menampilkan halaman dashboard admin pemasaran dan menampilkan notifikasi untuk konfirmasi pemesanan baru

Hasil Uji = valid

Jalur 8 Nilai (role= storage, countTotal, countAccepted, countNotAccepted, countItem, countCategory, notifications) = true

Hasil yang diharapkan = Sistem akan menampilkan halaman dashboard gudang dan menampilkan notifikasi pemesanan yang telah dikonfirmasi oleh admin pemasaran

Hasil Uji = valid

Jalur 9 Nilai (role) = "

Hasil yang diharapkan = Sistem akan menampilkan pesan halaman tidak tersedia

Hasil Uji = valid

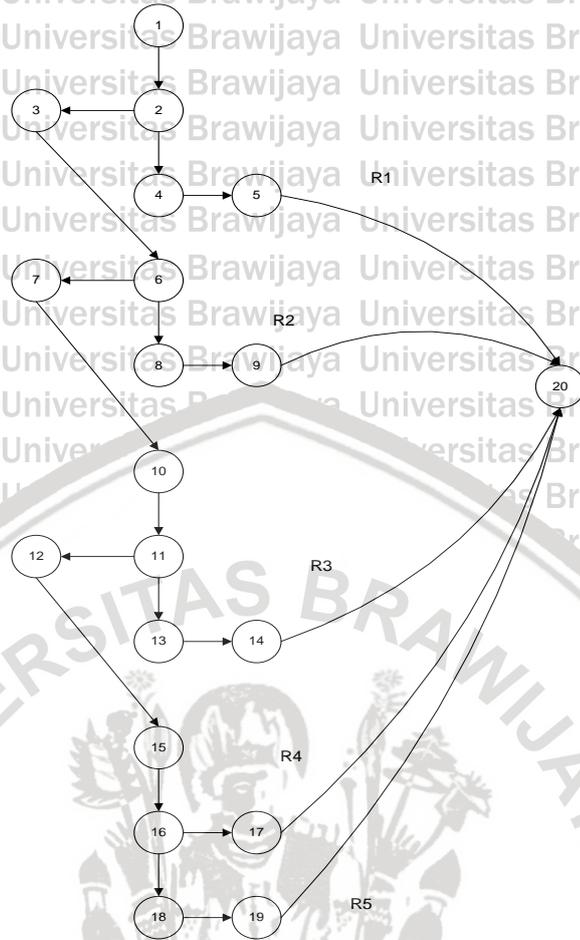
b. Kasus Uji Menambah Pemesanan Produk

Berikut ini adalah pseudo code pada kasus uji Menambah Pemesanan Produk ini yang nantinya digunakan untuk mencari flowgraph dan melanjutkan pada tahap berikutnya dijelaskan pada tabel 6.3

Tabel 6.3 Pseudo ode Menambah Pemesanan Produk

	Nama Algoritma: ALGO_CD_02
	Deskripsi: Algoritma untuk melakukan proses pemesanan produk
	Masukan: name, address, city, postalcode, telephone, email, bankname, bankaccountname, total
	Proses:
1	Input \$produk, \$jumlah
2	If (\$order==1)
3	Isi identitas pembeli
4	Else if ((\$order==2)
5	Pesanan Sementara
6	If (\$order=1, \$identity=true)
7	Pesanan telah diinputkan
8	Else if (\$order=1, \$identity=false)
9	Isi kelengkapan data
10	(\$pemesananbaru)
11	If (\$order==1 && \$acceptance==true)
12	pemesanan dikirim ke gudang
13	Else if (\$order==1 && \$acceptance==false)
14	pemesanan ditolak
15	(\$gudang)
16	If (\$order==1 && \$stok>=1)
17	pesan stok tersedia dan mengirim faktur penjualan
18	Else if (\$order==1 && \$stok==0)
19	Pesanan Tertunda
20	END

Flowgraph dari pseudo code kasus uji menambah pemesanan produk akan ditunjukkan pada gambar 6.3



Gambar 6.3 Flowgraph Menambah Pemesanan produk

Dari flowgraph pemesanan produk yang sudah dijelaskan pada gambar 6.3 dapat dihitung nilai *cyclomatic complexity* sebagai berikut :

$$V(G) = 5 \text{ regions}$$

$$V(G) = E - N + 2$$

$$= 23 - 20 + 2$$

$$= 3 + 2$$

$$= 5$$

$$V(G) = P + 1$$

$$= 4 + 1$$

$$= 5$$

Sehingga berdasarkan hasil persamaan *cyclomatic complexity* maka didapatkan

5 jalur independen yaitu :

1. Jalur 1 : 1-2-4-5-20

2. Jalur 2 : 1-2-3-6-8-9-20

3. Jalur 3 : 1-2-3-6-7-10-11-13-14-20

4. Jalur 4 : 1-2-3-6-7-10-11-12-15-16-17-20
 5. Jalur 5 : 1-2-3-6-7-10-11-12-15-16-18-19-20

Berdasarkan 5 jalur independen yang telah didefinisikan tersebut maka dapat dibentuk kasus ujinya. Tabel 6.4 memaparkan kasus uji dari menambah pemesanan produk.

Tabel 6.4 Kasus Uji menambah pemesanan produk

	<p>Nilai (order, item) = ada</p> <p>Hasil yang diharapkan = Sistem akan menampilkan halaman pesanan sementara</p> <p>Hasil Uji = valid</p>
Jalur 1	
	<p>Nilai (name, address, city, postalcode, telephone, email, bankname, bankaccountname, total) = tidak lengkap</p> <p>Hasil yang diharapkan = Sistem akan menampilkan pesan untuk mengisi kelengkapan data</p> <p>Hasil Uji = valid</p>
Jalur 2	
	<p>Nilai (name, address, city, postalcode, telephone, email, bankname, bankaccountname, total, request=rejected)</p> <p>Hasil yang diharapkan = Sistem akan menampilkan pesan pemesanan ditolak</p> <p>Hasil Uji = valid</p>
Jalur 3	
	<p>Nilai (name, address, city, postalcode, telephone, email, bankname, bankaccountname, total, request=accepted, status=ready)</p> <p>Hasil yang diharapkan = Sistem akan mengirimkan notifikasi status produk <i>ready</i> dan mengirimkan faktur penjualan</p> <p>Hasil Uji = valid</p>
Jalur 4	
	<p>Nilai (name, address, city, postalcode, telephone, email, bankname, bankaccountname, total, request=accepted, status=delayed)</p> <p>Hasil yang diharapkan = Sistem akan mengirimkan notifikasi status produk <i>delay</i></p> <p>Hasil Uji = valid</p>
Jalur 5	

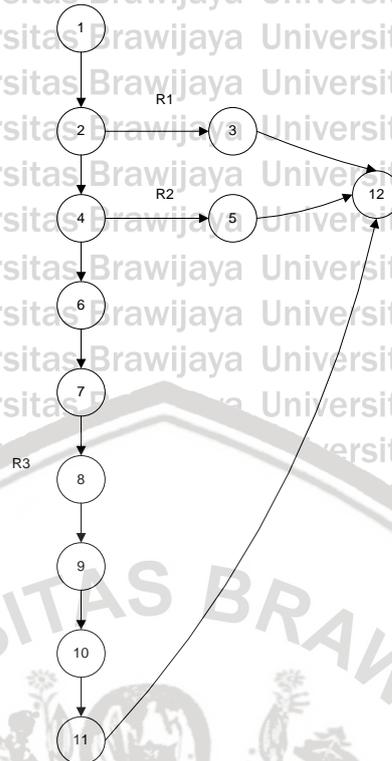
c. Kasus uji Menambah Sarana dan Ruang

Berikut adalah pseudocode menambah sarana dan ruang yang nantinya digunakan untuk mencari *flowgraph* dan melanjutkan pada tahap berikutnya ditunjukkan pada tabel 6.5 berikut :

Tabel 6.5 Pseudocode Menambah Sarana dan Ruang

	Nama Algoritma: ALGO_CD_03
	Deskripsi: Algoritma untuk menambah sarana dan ruang
	Masukan: <i>inventaris_code</i>
	Proses:
1	Run library validasi
2	'inventaris_code.required'
3	Kode Inventaris harus diisi,
4	'inventaris_code.unique'
5	Kode Inventaris harus unik (tidak boleh kembar)
6	(\$request)
7	get (\$request->input('tool_id')
8	\$room->tools attach(\$tool)
9	get \$request input('inventaris_code'),
10	return redirect ('pemeliharaan.roomtool')
11	menampilkan pesan sarana ruang telah ditambahkan
12	END

Flowgraph dari pseudocode kasus uji kasus menambah sarana dan ruang akan ditunjukkan pada gambar 6.4 dibawah ini



Gambar 6.4 Flowgraph Menambah Sarana dan Ruang

Berdasarkan flowgraph diatas dapat dihitung nilai *cyclomatic complexity* sebagai berikut :

$$V(G) = 3 \text{ Region}$$

$$V(G) = \text{Jumlah Region}$$

$$= 3$$

$$V(G) = E - N + 2$$

$$= 13 - 12 + 2$$

$$= 3$$

Sehingga berdasarkan hasil persamaan *cyclomatic complexity* maka didapatkan 3 jalur independen yaitu :

$$1. \text{ Jalur 1 : } 1 - 2 - 3 - 12$$

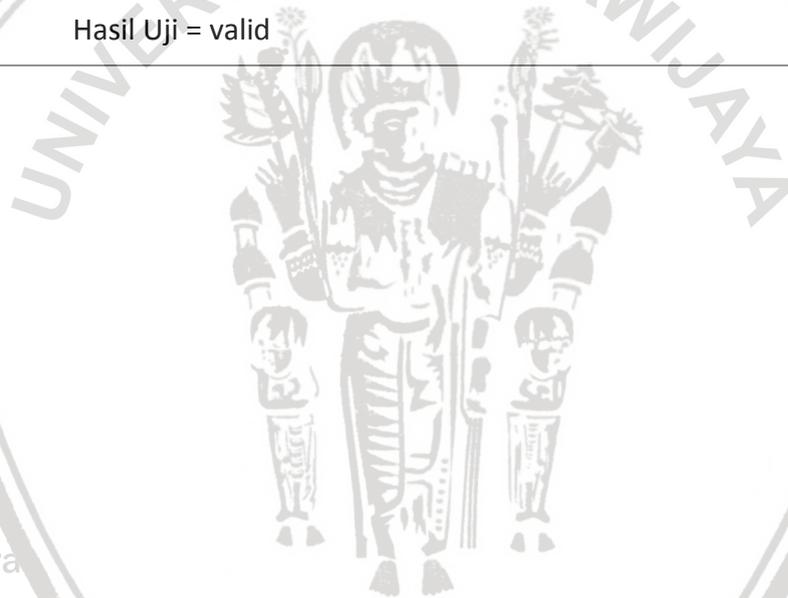
$$2. \text{ Jalur 2 : } 1 - 2 - 4 - 5 - 12$$

$$3. \text{ Jalur 3 : } 1 - 2 - 4 - 6 - 7 - 8 - 9 - 10 - 11 - 12$$

Berdasarkan 3 jalur independen yang telah didefinisikan tersebut maka dapat dibentuk kasus uji pada tabel 6.6

Tabel 6.6 Kasus Uji menambah sarana dan ruang

Jalur 1	<p>Nilai (inventaris_code) = ""</p> <p>Hasil yang diharapkan = Sistem menampilkan pesan 'Kode Inventaris harus diisi'</p> <p>Hasil Uji = valid</p>
Jalur 2	<p>Nilai (inventaris_code) = sudah ada</p> <p>Hasil yang diharapkan = Sistem menampilkan pesan 'Kode Inventaris harus unik (tidak boleh kembar)'</p> <p>Hasil Uji = valid</p>
Jalur 3	<p>Nilai (inventaris_code) = valid</p> <p>Hasil yang diharapkan = Sistem akan menyimpan data sarana-ruang baru dan menampilkan pesan 'Sarana-Ruang telah ditambahkan'</p> <p>Hasil Uji = valid</p>



6.1.2 Pengujian Validasi dengan Metode *BlackBox*

6.1.2.1 Tujuan

Tujuan dilakukannya pengujian validasi adalah untuk memastikan setiap spesifikasi kebutuhan yang didefinisikan telah sesuai dengan yang diharapkan dengan menguji seluruh skenario use case yang telah dibuat pada bab dari perancangan sebelumnya.

6.1.2.2 Mekanisme

Setiap skenario use case yang didefinisikan pada tahap analisis kebutuhan akan diujikan pada pengujian ini dengan cara mendefinisikan kasus uji dan hasil yang didapat terhadap setiap kebutuhan tersebut.

6.1.2.3 Hasil Pengujian

Pengujian black box secara validasi dijelaskan pada Tabel 6.7 berikut.

Tabel 6.7 Pengujian Black Box Sistem Pemeliharaan Sarana dan Pemantauan Pemasaran Produk PT.Pancamas Elite

KODE	FUNGSI	KASUS UJI	YANG DIHARAPKAN	HASIL YANG DIDAPAT	STATUS
KF_01	Login	-membuka sistem dan pilih menu 'login' -input <i>Username</i> dan <i>Password</i> -menekan tombol 'login'	-Sistem dapat memeriksa kebenaran input data user di database sehingga user dapat masuk ke dalam sistem.	-Sistem memeriksa kebenaran input data user di database sehingga user dapat masuk ke dalam sistem	Valid
		Alternative 1 -membuka sistem dan pilih menu 'login'	-sistem dapat menampilkan pesan 'kolom harus diisi'	-sistem menampilkan pesan 'kolom harus diisi'	

		<ul style="list-style-type: none"> -tidak mengisi salah satu kolom input -menekan tombol 'login' <p>Alternative 2</p> <ul style="list-style-type: none"> -membuka sistem dan pilih menu 'login' -input <i>username</i> dan <i>password</i> yang salah -menekan tombol 'login' 			
KF_02	Mengubah <i>profile</i> pengguna	<ul style="list-style-type: none"> -membuka sistem -memilih menu 'edit profile' -menekan tombol 'save' 	-Sistem dapat mengubah informasi 'profile' yang diinginkan	-sistem menampilkan pesan "These credentials do not match our record"	Valid
KF_03	Melaporkan Kerusakan Sarana	<ul style="list-style-type: none"> -login ke sistem sebagai unit kerja -memilih menu 'Laporkan Kerusakan' -menampilkan form 'Laporkan Kerusakan' -mengisi data form -memilih tombol 'Laporkan' 	-sistem dapat menampilkan pesan 'Kerusakan telah dilaporkan' dan dapat menambah data ke dalam database	-sistem menampilkan pesan 'Kerusakan telah dilaporkan' dan menambah data ke dalam database	Valid

		<p>Alternative 1</p> <ul style="list-style-type: none"> -login ke sistem sebagai unit kerja -memilih menu 'Laporkan Kerusakan' -menampilkan form 'Laporkan Kerusakan' -tidak mengisi salah satu data form -memilih tombol 'Laporkan' 	-sistem dapat menampilkan pesan 'kolom harus diisi'	-sistem menampilkan pesan 'kolom harus diisi'	
KF_04	Pemberitahuan Kerusakan	<ul style="list-style-type: none"> -login ke sistem sebagai staf pemeliharaan -menampilkan halaman dashboard -masuk ke menu "daftar request perawatan" -Klik opsi perbaikan 	-sistem dapat menampilkan pemberitahuan pengajuan kerusakan sarana dan dapat melakukan opsi perbaikan	-sistem menampilkan pemberitahuan pengajuan kerusakan dan dapat melakukan opsi perbaikan	Valid
KF_05	Melihat Riwayat	<ul style="list-style-type: none"> -login ke sistem sebagai unit kerja -memilih menu 'riwayat kerusakan' 	-sistem dapat menampilkan halaman riwayat pengajuan kerusakan	-sistem dapat menampilkan halaman riwayat pengajuan kerusakan	Valid

	Pengajuan Kerusakan	Alternative 1 -login ke sistem sebagai unit kerja -memilih menu 'riwayat kerusakan' -memilih submenu 'Filter Riwayat'	-sistem dapat menampilkan halaman riwayat pengajuan kerusakan sesuai periode waktu yang diinginkan	-sistem menampilkan halaman riwayat pengajuan kerusakan sesuai periode waktu yang diinginkan	
KF_06	Melihat Daftar Sarana	-login ke sistem sebagai staf pemeliharaan -memilih menu 'daftar Jenis sarana'	-sistem dapat menampilkan data sarana yang ada	-sistem menampilkan data sarana yang ada	Valid
KF_07	Menambah Sarana dan Ruang	-login ke sistem sebagai staf pemeliharaan -memilih menu 'daftar sarana ruang' -memilih submenu 'tambah jenis sarana ruang' -mengisi halaman form 'tambah sarana ruang' -menekan tombol 'Create'	-sistem dapat menampilkan pesan 'sarana ruang berhasil ditambahkan' dan dapat menambah data ke dalam database	-sistem menampilkan pesan 'sarana ruang berhasil ditambahkan' dan dapat menambah data ke dalam database	Valid

		<p>Alternative 1</p> <ul style="list-style-type: none"> -login ke sistem sebagai staf pemeliharaan -memilih menu 'daftar jenis sarana ruang' -memilih submenu 'tambah jenis sarana ruang' -mengisi halaman form 'tambah sarana ruang' lalu tidak mengisi salah satu field -menekan tombol 'Create' 	<ul style="list-style-type: none"> -sistem dapat menampilkan pesan 'kolom harus diisi' 	<ul style="list-style-type: none"> -sistem menampilkan pesan 'kolom harus diisi' 	
KF_08	<p>Mengubah Informasi Sarana</p>	<ul style="list-style-type: none"> -login ke sistem sebagai staf pemeliharaan -memilih menu 'daftar sarana' pada sistem -memilih submenu 'Edit' sarana yang ingin diubah lalu memilih icon 'edit' -mengisi perubahan informasi sarana lalu pilih 'save' 	<ul style="list-style-type: none"> -sistem dapat mengubah data sarana yang diinginkan 	<ul style="list-style-type: none"> -sistem mengubah data sarana yang diinginkan 	Valid

KF_09	Menghapus Sarana	<ul style="list-style-type: none"> -login ke sistem sebagai staf pemeliharaan -memilih menu 'daftar sarana' pada sistem -memilih icon 'delete' <p>Alternative 1</p> <ul style="list-style-type: none"> -memilih menu 'daftar sarana' pada sistem -memilih icon 'delete' -menekan tombol 'cancel' 	<ul style="list-style-type: none"> -sistem dapat menampilkan pesan 'sarana berhasil dihapus' dan berita dihapus dari database <p>-sistem dapat membatalkan aksi menghapus sarana</p>	<ul style="list-style-type: none"> -sistem menampilkan pesan 'sarana berhasil dihapus' dan berita dihapus dari database <p>-sistem membatalkan aksi menghapus sarana</p>	Valid
KF_10	Melihat Ruang	<ul style="list-style-type: none"> - login ke sistem sebagai staf pemeliharaan -memilih menu 'Daftar Ruang' pada sistem 	-sistem dapat menampilkan data ruangan yang ada	-sistem dapat menampilkan data ruangan yang ada	Valid
KF_11	Mengubah informasi ruangan	<ul style="list-style-type: none"> -login ke sistem sebagai staf pemeliharaan -memilih menu 'daftar ruangan' -memilih submenu 'edit ruangan' -mengisi halaman form 'edit ruangan' -menekan tombol 'Create' 	-sistem dapat menampilkan pesan 'ruangan berhasil diubah' dan dapat menambah data ke dalam database	-sistem menampilkan pesan 'ruangan berhasil diubah' dan dapat menambah data ke dalam database	

		<p>Alternative 1</p> <ul style="list-style-type: none"> -login ke sistem sebagai staf pemeliharaan -memilih menu 'edit ruangan ruangan' -memilih submenu 'edit ruangan' -mengisi halaman form 'edit ruangan' lalu tidak mengisi salah satu field -menekan tombol 'Create' 	-sistem dapat menampilkan pesan 'kolom harus diisi'	-sistem dapat menampilkan pesan 'kolom harus diisi'	
KF_12	Menghapus Ruang	<ul style="list-style-type: none"> -login ke sistem sebagai staf pemeliharaan -memilih menu 'daftar ruangan' pada sistem -memilih icon 'delete' 	-sistem dapat menampilkan pesan 'ruangan berhasil dihapus' dan ruangan dihapus dari database	-sistem menampilkan pesan 'ruangan berhasil dihapus' dan ruangan dihapus dari database	Valid
KF_13	Melihat Status Terakhir Sarana	<ul style="list-style-type: none"> -login ke sistem sebagai staf pemeliharaan -memilih menu 'Sarana Ruang' pada sistem 	-sistem dapat menampilkan status terakhir perawatan sarana	-sistem menampilkan status terakhir perawatan sarana	Valid

KF_14	Konfirmasi Perbaikan	<ul style="list-style-type: none"> -login ke sistem sebagai staf pemeliharaan -memilih menu 'Daftar Request Perawatan' -Klik opsi 'perbaiki sekarang' <p>Alternative 1</p> <ul style="list-style-type: none"> -login ke sistem sebagai staf pemeliharaan -memilih menu 'Daftar Request Perawatan' -Klik opsi 'perbaiki selesai' 	<ul style="list-style-type: none"> -sistem dapat menampilkan pesan bahwa 'status perbaikan telah diupdate' dan memberikan notifikasi kepada pelapor <p>Alternative 1</p> <ul style="list-style-type: none"> -sistem dapat menampilkan pesan bahwa 'status perbaikan telah selesai' dan memberikan notifikasi kepada pelapor 	<ul style="list-style-type: none"> -sistem menampilkan pesan bahwa 'status perbaikan telah diupdate' dan memberikan notifikasi kepada pelapor <p>Alternative 1</p> <ul style="list-style-type: none"> -sistem menampilkan pesan bahwa 'status perbaikan telah selesai' dan memberikan notifikasi kepada pelapor 	Valid
KF_15	Menerima Status Perbaikan	<ul style="list-style-type: none"> -login ke sistem sebagai unit kerja -menampilkan halaman dashboard pemberitahuan 	<ul style="list-style-type: none"> -Sistem dapat menampilkan pemberitahuan status pengajuan sarana yang dilakukan perbaikan 	<ul style="list-style-type: none"> -Sistem menampilkan pemberitahuan status pengajuan sarana yang dilakukan perbaikan 	Valid
KF_16	Melihat Laporan Perbaikan dan Perawatan	<ul style="list-style-type: none"> -login ke sistem sebagai staf pemeliharaan -memilih menu 'riwayat pengajuan perbaikan' <p>Alternative 1</p> <ul style="list-style-type: none"> -login ke sistem sebagai unit kerja 	<ul style="list-style-type: none"> -sistem dapat menampilkan halaman laporan perbaikan dan perawatan <p>Alternative 1</p> <ul style="list-style-type: none"> -sistem dapat menampilkan halaman laporan perbaikan 	<ul style="list-style-type: none"> -sistem menampilkan halaman laporan perbaikan dan perawatan <p>Alternative 1</p> <ul style="list-style-type: none"> -sistem menampilkan halaman laporan perbaikan 	Valid

		<ul style="list-style-type: none"> -memilih menu 'laporan perbaikan dan perawatan' -memilih submenu 'Filter Riwayat' 	dan perawatan sesuai periode waktu yang diinginkan	dan perawatan sesuai periode waktu yang diinginkan	
KF_17	Melihat Pemeliharaan Rutin	<ul style="list-style-type: none"> -login ke sistem sebagai staf pemeliharaan -memilih menu 'perawatan rutin' pada sistem 	-sistem dapat menampilkan halaman jadwal 'perawatan rutin'	-sistem menampilkan halaman jadwal 'perawatan rutin'	Valid
KF_18	Konfirmasi Perawatan Rutin	<ul style="list-style-type: none"> -login ke sistem sebagai staf pemeliharaan -memilih menu 'perawatan rutin' -memilih opsi 'Baik' <p>Alternative 1</p> <ul style="list-style-type: none"> -login ke sistem sebagai staf pemeliharaan -memilih menu 'perawatan rutin' -memilih opsi 'perbaiki' -menampilkan halaman form perbaikan -menekan tombol 'Laporkan' 	<ul style="list-style-type: none"> -sistem dapat menampilkan halaman jadwal 'perawatan rutin' dan menyimpan data perawatan kedalam database -sistem dapat menampilkan halaman 'perawatan rutin' dan menyimpan riwayat kerusakan ke dalam database 	<ul style="list-style-type: none"> -sistem menampilkan halaman jadwal 'perawatan rutin' dan menyimpan data perawatan kedalam database -sistem menampilkan halaman 'perawatan rutin' dan menyimpan riwayat kerusakan ke dalam database 	Valid

KF_19	Pemberitahuan jadwal pemeliharaan	-login ke sistem sebagai staf pemeliharaan -menampilkan halaman dashboard pemberitahuan jadwal perawatan	-sistem dapat menampilkan halaman pemberitahuan perawatan -sistem menampilkan dashboard jadwal perawatan	-sistem menampilkan halaman pemberitahuan jadwal perawatan	Valid
KF_20	Menampilkan daftar produk	-login ke sistem sebagai staf pemasaran -memilih menu 'Daftar Produk' -memilih submenu kategori produk pada sistem	-sistem dapat menampilkan halaman daftar produk	-sistem menampilkan halaman daftar produk	Valid
KF_21	Menambah produk	-login ke sistem sebagai gudang -memilih menu 'data produk' -memilih submenu 'tambah produk' -mengisi halaman form 'tambah produk' -menekan tombol 'Create' Alternative 1 -login ke sistem sebagai gudang -memilih menu 'data produk'	-sistem dapat menampilkan pesan 'produk berhasil ditambahkan' dan dapat menambah data ke dalam database -sistem dapat menampilkan pesan "kolom harus diisi"	-sistem menampilkan pesan 'produk berhasil ditambahkan' dan dapat menambah data ke dalam database -sistem menampilkan pesan "kolom harus diisi"	Valid

		<ul style="list-style-type: none"> -memilih submenu 'tambah produk' -mengisi halaman form 'tambah produk' lalu tidak mengisi salah satu field -menekan tombol 'Create' 			
KF_22	Mengubah Informasi Produk	<ul style="list-style-type: none"> -login ke sistem sebagai gudang -memilih menu 'data produk' pada sistem -memilih submenu 'Edit' produk yang ingin diubah lalu memilih icon 'edit' -mengisi perubahan informasi produk lalu pilih 'save' <p>Alternative 1</p> <ul style="list-style-type: none"> -memilih menu 'data produk' pada sistem -memilih submenu 'Edit' produk yang ingin diubah lalu memilih icon 'edit' -tidak mengisi semua kolom edit informasi produk lalu pilih 'save' 	<ul style="list-style-type: none"> -sistem dapat mengubah data produk yang diinginkan 	<ul style="list-style-type: none"> -sistem mengubah data produk yang diinginkan 	Valid

KF_23	Menghapus Produk	<ul style="list-style-type: none"> -login ke sistem sebagai gudang -memilih menu 'daftar produk' pada sistem -memilih submenu 'Delete' produk yang ingin dihapus lalu memilih icon 'delete' 	-sistem dapat menampilkan pesan 'produk berhasil dihapus' dan produk dihapus dari database	-sistem menampilkan pesan 'produk berhasil dihapus' dan produk dihapus dari database	Valid
KF_24	Menambah Pemesanan Produk	<ul style="list-style-type: none"> -login ke sistem sebagai staffpemasaran -memilih menu 'daftar produk' -memilih submenu 'daftar item' -mengisi halaman form 'tambah pemesanan' dan menekan tombol 'checkout' -mengisi halaman form data pemesanan -menekan tombol 'Create' <p>Alternative 1</p> <ul style="list-style-type: none"> -login ke sistem sebagai staffpemasaran -memilih menu 'daftar produk' -memilih submenu 'daftar item' 	-sistem dapat menampilkan pesan 'pemesanan berhasil ditambahkan' dan dapat menambah data ke dalam database	-sistem menampilkan pesan 'pemesanan berhasil ditambahkan' dan dapat menambah data ke dalam database	Valid

		<ul style="list-style-type: none"> -mengisi halaman form 'tambah pemesanan' dan menekan tombol 'checkout' -mengisi halaman form data pemesan dan tidak mengisi seluruh field data -menekan tombol 'Create' 			
KF_25	Validasi Pemesanan	<ul style="list-style-type: none"> -login ke sistem sebagai adminpemasaran -memilih menu 'pemesanan baru' -Klik opsi 'accept' <p>Alternative 1</p> <ul style="list-style-type: none"> -login ke sistem sebagai admin pemasaran -memilih menu 'pemesanan baru' -Klik opsi 'reject' 	<ul style="list-style-type: none"> -sistem dapat menampilkan pesan 'pemesanan diterima' dan status data berhasil dirubah di database -sistem dapat menampilkan pesan 'pemesanan ditolak' dan status data berhasil dirubah di database 	<ul style="list-style-type: none"> -sistem menampilkan pesan 'pemesanan diterima' dan status data berhasil dirubah di database -sistem menampilkan pesan 'pemesanan ditolak' dan status data berhasil dirubah di database 	Valid
KF_26	Melihat Pemesanan Sementara	<ul style="list-style-type: none"> -login ke sistem sebagai staf pemasaran -memilih menu 'pemesanan sementara' pada sistem 	<ul style="list-style-type: none"> -sistem dapat menampilkan halaman pemesanan sementara 	<ul style="list-style-type: none"> -sistem menampilkan halaman pemesanan sementara 	Valid

		<p>Alternative 1</p> <ul style="list-style-type: none"> -memilih menu 'pemesanan sementara' pada sistem -menekan tombol 'delete' 	<ul style="list-style-type: none"> -sistem dapat menghapus pesanan sementara dari daftar 	<ul style="list-style-type: none"> -sistem menghapus pesanan sementara dari daftar 	
KF_27	<p>Melihat riwayat Pemesanan</p>	<ul style="list-style-type: none"> -login ke sistem sebagai staf pemasaran -memilih menu 'laporan penjualan' pada sistem 	<ul style="list-style-type: none"> -sistem dapat menampilkan halaman laporan penjualan 	<ul style="list-style-type: none"> -sistem menampilkan halaman laporan penjualan 	Valid
KF_28	<p>Pemberitahuan status pengajuan pemesanan</p>	<ul style="list-style-type: none"> -login ke sistem sebagai staf pemasaran -menampilkan dashboard pemberitahuan status pengajuan pemesanan 	<ul style="list-style-type: none"> -sistem dapat menampilkan halaman pemberitahuan status pengajuan pemesanan 	<ul style="list-style-type: none"> -sistem menampilkan halaman pemberitahuan status pemesanan 	Valid
KF_29	<p>Pemberitahuan Pemesanan baru</p>	<ul style="list-style-type: none"> -login ke sistem sebagai admin pemasaran -menampilkan dashboard pemberitahuan pemesanan baru 	<ul style="list-style-type: none"> -sistem dapat menampilkan halaman pemberitahuan pemesanan baru 	<ul style="list-style-type: none"> -sistem menampilkan halaman pemberitahuan pemesanan baru 	Valid

KF_30	Melihat Status stok produk	-login ke sistem sebagai amin pemasaran -memilih menu 'pesanan terproses' pada sistem	-sistem dapat menampilkan halaman pesanan terproses yang berisi status ketersediaan produk yang ada dalam gudang	-sistem menampilkan halaman pesanan terproses yang berisi status ketersediaan produk yang ada dalam gudang	Valid
KF_31	Melihat informasi produk	-login ke sistem sebagai gudang -memilih menu 'data produk' pada sistem	-sistem dapat menampilkan informasi produk	-sistem menampilkan informasi produk	Valid
KF_32	Konfirmasi Stok produk	-login ke sistem sebagai gudang -memilih menu 'pesanan terkonfirmasi' pada sistem -menekan tombol 'Ready' pada sistem Alternative 1 -login ke sistem sebagai gudang -memilih menu 'pesanan terkonfirmasi' pada sistem -menekan tombol 'Delay' pada sistem	-sistem dapat menampilkan pesan 'produk sudah ditandai bahwa telah siap' dan mengirim status terbaru ke halaman amin pemasaran -sistem dapat menampilkan pesan 'produk sudah ditandai untuk delay' dan mengirim status terbaru ke halaman adminpemasaran serta menyimpan informasi ke halaman 'pesanan tertunda'	-sistem menampilkan pesan 'produk sudah ditandai bahwa telah siap' dan mengirim status terbaru ke halaman amin pemasaran -sistem menampilkan pesan 'produk sudah ditandai untuk delay' dan mengirim status terbaru ke halaman adminpemasaran	Valid

		<p>Alternative 2</p> <ul style="list-style-type: none"> -login ke sistem sebagai gudang -memilih menu 'pesanan tertunda' pada sistem -menampilkan data pemesanan produk yang tertunda -menekan tombol 'Ready' pada sistem 	<ul style="list-style-type: none"> -sistem dapat menampilkan pesan 'produk sudah ditandai bahwa telah siap' dan mengirim status terbaru ke halaman amin pemasaran 	<ul style="list-style-type: none"> -sistem menampilkan pesan 'produk sudah ditandai bahwa telah siap' dan mengirim status terbaru ke halaman amin pemasaran 	
KF_33	Mencetak Faktur (<i>Invoice</i>)	<ul style="list-style-type: none"> -login ke sistem sebagai staf pemasaran atau admin pemasaran -memilih menu 'laporan penjualan' -menekan tombol 'Faktur' pada sistem -menampilkan halaman informasi pemesanan -menekan tombol 'Print Preview' -menampilkan print preview faktur penjualan 	<ul style="list-style-type: none"> -sistem dapat menampilkan halaman <i>print preview</i> faktur penjualan yang siap untuk dicetak 	<ul style="list-style-type: none"> -sistem menampilkan halaman print preview faktur penjualan yang siap untuk dicetak 	Valid

	-menekan icon 'print' pada layar print preview		
	Alternative 1		
	-login ke sistem sebagai staf pemasaran atau admin pemasaran	-sistem dapat menampilkan halaman <i>print preview</i> faktur penjualan dan menyimpan faktur penjualan dalam bentuk file pdf	-sistem menampilkan halaman <i>print preview</i> faktur penjualan dan menyimpan faktur penjualan dalam bentuk file pdf
	-memilih menu 'laporan penjualan'		
	-menekan tombol 'Faktur' pada sistem		
	-menampilkan halaman informasi pemesanan		
	-menekan tombol 'Print Preview'		
	-menampilkan print preview faktur penjualan		
	-menekan icon 'save' pada layar print preview		

6.1.3 Pengujian *Compatibility*

6.1.3.1 Tujuan

Tujuan dilakukannya pengujian *compatibility* adalah untuk mengetahui apakah system bias berjalan normal pada sistem operasi, *browser* dan perangkat yang berbeda-beda karena banyaknya pengguna sistem.

6.1.3.2 Mekanisme

Mekanisme pengujian dengan melakukan pendefinisian daftar kebutuhan yang diperlukan untuk menguji sistem dan membuat skenario pengujian serta hasil yang diperoleh terhadap setiap pengujian tersebut. Pengujian dilakukan dengan 3 sistem operasi dan 3 *browser* yang berbeda.

6.1.3.3 Hasil Pengujian

1. Daftar kebutuhan pengujian pada sistem operasi Linux di Ubuntu dijelaskan pada table 6.8

Tabel 6.8 Daftar kebutuhan pengujian linux ubuntu

Daftar kebutuhan pengujian compatibility	
1	Linux Ubuntu
2	XAMPP version: 5.5
3	Mozilla firefork 15

Setelah didapatkan perangkat lunak yang diperlukan untuk menguji sistem, langkah selanjutnya adalah membuat skenario pengujian *compatibility* sistem dijelaskan pada tabel 6.9

Tabel 6.9 skenario pengujian linux ubuntu

Skenario pengujian di Ubuntu	Hasil yang diperoleh
<ol style="list-style-type: none"> 1. Instalasi Linux Ubuntu 12.04 2. Instalasi Xampp version 5.5 3. Instalasi Laravel 4. Salin file proyek ke dalam folder htdocs 5. Import basisdata ke phpmyadmin 6. Buka localhost/skripsi/public 	Berdasarkan pengujian ini maka diperoleh suatu hasil yakni sistem dapat berjalan dengan lancar pada sistem operasi Linux Ubuntu 12.04

Berdasarkan skenario pengujian maka hasil pengujian ditunjukkan pada Gambar 6.5 dibawah ini



Gambar 6.5 Halaman login di linux Ubuntu

2. Daftar kebutuhan pengujian pada sistem operasi Windows 8 Pro

Tabel 6.10 Daftar kebutuhan pengujian windows 8 pro

Daftar kebutuhan pengujian compatibility	
1	Windows 8 pro 64bit
2	XAMPP version: 5.5
3	Mozilla firefork 15
4	Google Chrome version 46.0.2490.86
5	Internet Explorer 10

Setelah didapatkan perangkat lunak yang diperlukan untuk menguji sistem, langkah selanjutnya adalah membuat skenario pengujian *compatibility* sistem dijelaskan pada table 6.11

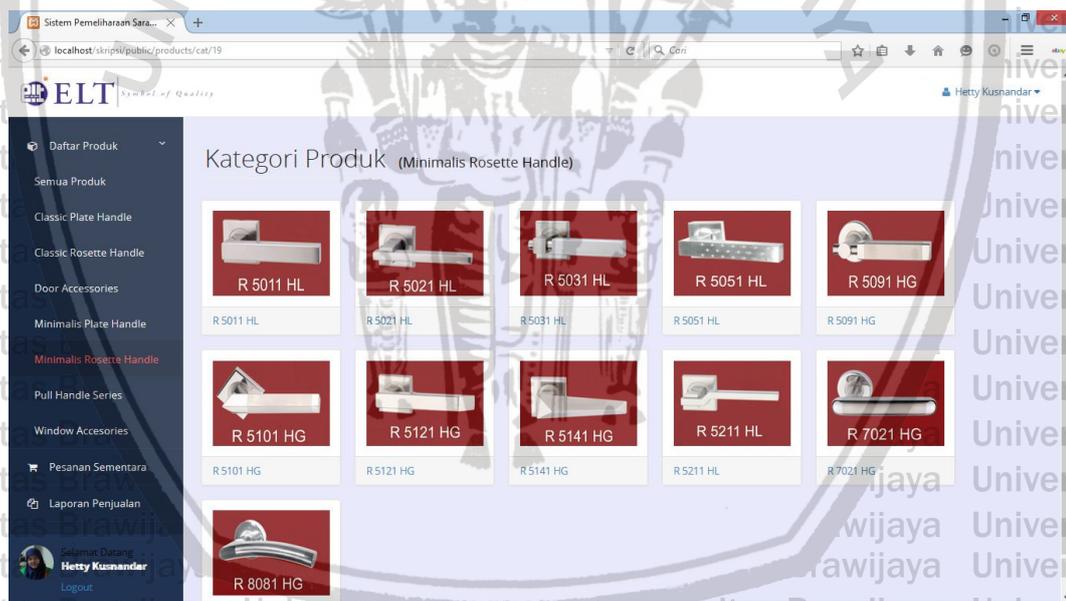
Tabel 6.11 skenario pengujian windows 8 pro

Skenario pengujian di windows	Hasil yang diperoleh
1. Installasi Windows 8 Pro 64bit	Berdasarkan pengujian ini maka diperoleh suatu hasil yakni sistem dapat berjalan dengan lancar pada sistem operasi Windows 8 Pro
2. Installasi Xampp version 5.5	
3. Instalasi Laravel	
4. Salin file projek ke dalam folder htdocs	
5. Import basisdata ke phpmyadmin	
6. Buka localhost/skripsi/public di browser google chrome	

Berdasarkan skenario pengujian maka hasil pengujian ditunjukkan pada Gambar 6.6-6.8 dibawah ini.



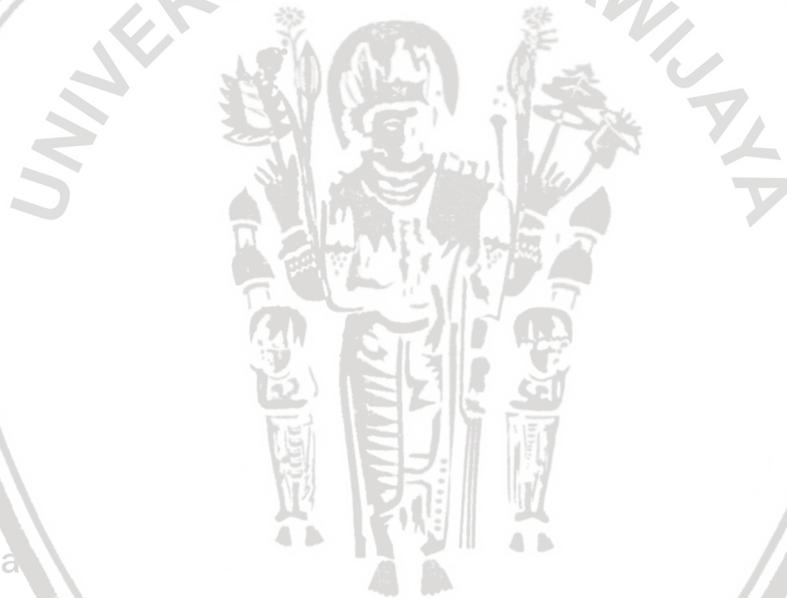
Gambar 6.6 Halaman daftar perawatan rutin di browser google chrome



Gambar 6.7 Halaman kategori produk di browser mozilla firefox

ID	Nama	Jumlah	Harga	Kategori	Status	OPTIONS
186	PHP 3019 SS	28	180000	Pull Handle Series	Ready	View Edit Delete
187	B 11	19	60000	Window Accesories	Ready	View Edit Delete
188	B 20	4	150000	Window Accesories	Ready	View Edit Delete
189	B 22	35	260000	Window Accesories	Ready	View Edit Delete
190	B 25	23	160000	Window Accesories	Ready	View Edit Delete
191	B 32	13	50000	Window Accesories	Ready	View Edit Delete
192	ES 09 AC	36	160000	Window Accesories	Ready	View Edit Delete
193	ES 900 SS	42	100000	Window Accesories	Ready	View Edit Delete
194	HB 08	3	120000	Window Accesories	Ready	View Edit Delete
195	HB 10	10	160000	Window Accesories	Ready	View Edit Delete

Gambar 6.8 Halaman daftar produk di browser internet explorer



6.2 Analisis Sistem

Pada bagian ini akan dilakukan seluruh analisa hasil dari semua pengujian yang telah dilakukan, analisis bertujuan untuk menganalisa data hasil pengujian hingga menghasilkan kesimpulan-kesimpulan. Analisis ini dilakukan berdasarkan hasil implementasi hingga pengujian yang telah dilakukan terhadap sistem dan mengacu pada dasar teori. Proses analisis yang dilakukan antara lain adalah analisa hasil pengujian *whitebox*, analisa hasil dari pengujian *blackbox* dan analisa hasil pengujian *compatibility*.

6.2.1 Analisis Pengujian *Whitebox*

Berdasarkan hasil pengujian *whitebox* yang dilakukan, menunjukkan bahwa semua jalur algoritma telah diuji, dengan didapatkan hasil berupa *cyclomatic complexity* yang berbeda dari setiap data uji. Dimana untuk fitur menerima pemberitahuan didapatkan *cyclomatic complexity* sebanyak 9, fitur menambah sarana dan ruang didapatkan 3 *cyclomatic complexity*, fitur menambah pemesanan produk didapatkan *cyclomatic complexity* sebanyak 5.

6.2.2 Analisis Pengujian *Black Box*

Berdasarkan hasil pengujian *blackbox* secara validasi, implementasi perancangan sistem telah sesuai dengan kebutuhan fungsional yang ada, hasil pengujian menunjukkan dari 33 kebutuhan fungsional yang diuji menunjukkan bahwa 100% kebutuhan fungsional dari sistem telah valid ketika diuji dengan kasus tertentu. Sehingga dapat disimpulkan bahwa Sistem Informasi Pemeliharaan Sarana dan Pemantauan Pemasaran Produk berbasis web dengan kerangka kerja Laravel dengan studi kasus di PT Pancamas Elite yang dibuat telah berjalan dengan baik sesuai dengan yang diharapkan.

6.2.3 Analisis Pengujian *Compatibility*

Berdasarkan hasil pengujian yang dilakukan dengan melihat hasil kinerja sistem yang dilakukan terhadap 2 sistem operasi dan 4 *browser*. Berdasarkan hasil pengujian *compatibility* dapat disimpulkan bahwa sistem informasi pemeliharaan sarana dan pemantauan pemasaran produk berbasis web dengan kerangka kerja Laravel dengan studi kasus di PT.Pancamas Elite dapat berjalan secara normal pada 2 sistem operasi dan 4 *browser* yang berbeda tersebut.

BAB 7 PENUTUP

7.1 Kesimpulan

Sebagaimana tujuan dari penelitian ini dimaksudkan untuk melakukan implementasi kerangka kerja *Laravel* dalam pembangunan sistem berbasis web maka berikut kesimpulan dari penelitian yang telah dilakukan:

1. Rancangan sistem dibuat berdasarkan analisis kebutuhan, menghasilkan 5 aktor yaitu Staff Pemasaran, Admin Pemasaran, Staff Gudang, Unit Kerja, dan Staff Pemeliharaan; 33 kebutuhan fungsional; 1 kebutuhan nonfungsional yaitu *compability*; 8 *controller*; 11 *model*.
2. Implementasi sistem menggunakan kerangka kerja *Laravel* terbukti sangat membantu, hal ini dapat ditunjukkan dengan adanya beberapa fungsi bawaan *Laravel* yang dapat mempersingkat kode program dan komunikasi antar objek serta memetakan tabel pada basisdata dengan singkat menggunakan ORM (*Object Relational Mapping*) *Laravel*. Implementasi sistem menggunakan *Laravel* dapat mempermudah pembangun sistem untuk mengatur hak akses pengguna (aktor) melalui fitur *Routes*.
3. Sistem yang dibuat dapat membantu meminimalisir kendala pada bagian pemeliharaan, hal ini dapat ditunjukkan dengan adanya fitur notifikasi perbaikan inventaris yang tampil pada dashboard masing-masing aktor pemeliharaan, serta adanya Laporan Riwayat Pengajuan Kerusakan sehingga dapat diketahui inventaris apa saja, di ruang apa yang sedang rusak, sedang diperbaiki, dan telah diperbaiki. Sistem juga dapat membantu memantau pemasaran produk pada bagian pemasaran, ditunjukkan dengan adanya fitur validasi yang diberikan untuk Admin Pemasaran dan fitur milik Staff Gudang untuk memberitahukan stok produk siap memenuhi permintaan atau tidak. Fitur Faktur juga memberikan kontribusi dalam transparansi total produk yang dikeluarkan dan jumlah pendapatan yang harus diterima perusahaan.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah:

1. Untuk penelitian lebih lanjut, diharapkan sistem ini dapat mencakup semua aspek manajemen di dalam Sistem Pemeliharaan Sarana dan Pemantauan Pemasaran Produk PT.Pancamas Elite seperti mengelola pemasukan dalam penjualan, mengelola kehadiran staf unit kerja di dalam setiap divisi, mengelola distributor sarana dan mengelola konsumen.
2. Untuk pengembangan lebih lanjut sebaiknya menggunakan *web service* agar keamanan data dan permintaan data mudah dilakukan.
3. Pengembangan sistem dapat dilakukan dengan menambahkan fitur *sms gateway* untuk penjadwalan rutin sarana.

DAFTAR PUSTAKA

- Aminuddin, Salim. 2010. *Perawatan Sistem Informasi Perawatan Mesin Produksi*, [e-journal]. Tersedia melalui: < eprints.undip.ac.id/6958/1/L2H_003_616.doc > [Diakses 1 Desember 2015]
- Barbara, H. and Tergeden, D. (2009). *System Analysis & Design UML Version 2.0: An Object-Oriented Approach*. Bonn : Jhon Wiley & Sons, Inc.
- Bean, Martin. 2014. *Laravel 5 Essentials*. New York : Stack Publishing
- Dharwiyanti, S. and Wahono, R. (2003). *Pengantar Unified Modeling Language*. Bandung : Modula
- Endah, Enita., 2014. *Sistem Informasi Pemeliharaan Sarana dan Prasarana Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya*. S1. Universitas Brawijaya
- Fadmara, Irfan. 2010. *Sistem Informasi Manajemen Pemasaran Berbasis Web*, [e-journal]. Tersedia melalui: < repository.uinjkt.ac.id > [Diakses 1 Desember 2015]
- Goyal, Arunes. 2009. *System Analysis and Design Eastern Economy Edition*. Portland : International Press
- Lindley R. Higgin & R. Keith Mobley. 2002. *Maintenance Engineering Handbook, Sixth Edition*. New York : McGraw-Hill
- Malatesta, Francesco. 2014. *PHP and Website Fundamental*. New York : Stack Publishing
- Matula, Terry. 2013. *Laravel Application Development Cookbook*. Anaheim: Beverly Hill
- Petersen, K., Wohlin, C., & Baca, D., 2008. *The Waterfall Model in Large-Scale Development*, [e-journal]. Tersedia melalui: [Blekinge Institute of Technology<www.divaportal.org/smash/get/diva2:835760/FULLTEXT02.pdf>](http://Blekinge%20Institute%20of%20Technology/www.divaportal.org/smash/get/diva2:835760/FULLTEXT02.pdf) [Diakses 25 Oktober 2015]
- Prayitno, Wendhie; S.Kom. 2005 . *Desain Model Sistem Perangkat Lunak dengan UML*. Yogyakarta : Informatika
- Pressman, Roger. 2010. *Software Engineering: A Practitioner's Approach*. New York : Mc Graw-Hill
- Rosa, A.S & Shalahuddin, M. 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung : Modula
- Rosa, A.S & Shalahuddin, M. 2013. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung : Informatika
- Roebuck, Kevin. 2012. *Systems Development Life Cycle (SDLC): High-impact Strategies - What You Need to Know (Definitions, Adoptions, Impact, Benefits, Maturity, Vendors)*. Boston : Emereo Pty Limited

Rohman, Abdul. *Mengenal Framework "Laravel" (Best PHP Frameworks For 2014)*, [e-journal]. Tersedia melalui: <<http://www.abdulrohman.web.id>> [Diakses 1 Juli 2015]

Sanjib, Sinha. 2014. *Laravel Unfolded (Design, Debug and Deploy Web Appilcation)*. Seattle : Apress

Sabharwai, Sangeeta. 2009. *Software Engineering*. London: New Age International Pvt Ltd Publisher

Simamarta, Janner. 2010. *Rekayasa Web*. Yogyakarta : CV Andi Offset

Sommerville, I. (2007). *Software engineering. 8th ed. Harlow*. England: Addison-Wesley, pp.117-125

Rubin, Frances. 1995. *Basic Guide to Evaluation for Development Workers*. Oxfam GB Practical Action Publishing

Utama, Agung. 2003. *Upaya Meningkatkan Keunggulan Kompetitif Perusahaan Dalam Era Persaingan Global Melalui Aliansi Strategis*, [e-journal]. Tersedia melalui:<<http://staff.uny.ac.id/UpayaMeningkatkanKeunggulanKompetitif.pdfid>> (Diakses 20 Januari 2016)

Whitten et. al. 2004. *System Analysis and Design Methods, 6 th edition*. New York: McGraw-Hill Companies, Inc

Wulandari, Novita. 2015. *Rancang Bangun Sistem Informasi Penjualan Produk Menggunakan Kerangka Kerja Laravel*, [e-journal]. Tersedia melalui: <eprints.mdp.ac.id/1461/> [Diakses 1 Juli 2015]

LAMPIRAN A : WAWANCARA

Lampiran ini berisi pertanyaan-pertanyaan yang diajukan ketika melakukan wawancara secara terstruktur. Wawancara dilakukan dengan perwakilan Staff Pemeliharaan dan Pemasaran PT.Pancamas Elite Malang. Wawancara dilakukan guna mengetahui informasi secara umum seputar tugas mereka dan permasalahan yang dihadapi terkait tugas-tugas yang dilakukan. Berikut adalah beberapa pertanyaan yang penulis utarakan kepada narasumber.

1. Bagaimana struktur organisasi di PT.Pancamas Elite-Malang ?
Bagian-bagian mana yang saling terkait satu sama lain ?
2. Bagaimana kendala saat ini di PT.Pancamas Elite – Malang ?
3. Apa penanganan yang dilakukan terhadap kendala tersebut ?
4. Bagaimana pemahaman pegawai tentang Teknologi Informasi?
5. Apa saja sarana yang membutuhkan perawatan di PT. Pancamas Elite?
6. Apa saja produk yang dihasilkan PT.Pancamas Elite ?
7. Bagaimana prosedur dalam menjalankan proses bisnis dan apakah menggunakan standar operasional prosedur ?
8. Bagaimana proses pemeliharaan sarana dan prasarana di PT. Pancamas Elite ?
9. Bagaimana proses pemasaran produk yang ada saat ini ?
10. Setelah dilakukan analisa, menurut anda proses mana yang penting atau sering digunakan dalam menjalankan proses bisnis ?

Jawaban dari pertanyaan-pertanyaan wawancara secara jelas dan singkat.

1. Struktur organisasi masih menggunakan yang lama (bisa dilihat pada bab 2 pada dasar teori) dan bagian-bagian terpenting yang saling terkait satu sama lain seperti bagian unit kerja dengan bagian divisi pemeliharaan, bagian staf pemasaran dengan bagian admin pemasaran dan gudang.
2. Beberapa kendala saat ini mengenai kebutuhan akan pemeliharaan sarana prasarana yang kurang memadai seperti dalam proses pengajuan perbaikan sarana unit kerja atau pegawai umum harus melapor langsung ke bagian divisi pemeliharaan. Selain itu, pada bagian pemasaran, admin Pemasaran kesulitan dalam merekap dan mengevaluasi pemasaran yang sudah dilakukan oleh Staff berdasarkan laporan tertulis yang telah dibuat oleh Staff tersebut. Kegiatan pemantauan menjadi terhambat ketika

Admin Pemasaran tidak memiliki kesempatan untuk bertatap muka dengan Staff Pemasaran yang ditugaskan.

3. Untuk divisi pemeliharaan penanganan saat ini adalah dengan memberikan form untuk pengajuan perbaikan sarana yang ditulis manual oleh unit kerja dan pada bagian pemasaran dengan melakukan rapat rutin
4. Tidak adanya divisi khusus pada bagian IT menyebabkan pegawai kurang adanya pemahaman mengenai penggunaan teknologi informasi
5. Sarana yang membutuhkan perawatan di PT.Pancamas Elite antara lain beberapa jenis mesin produksi seperti mesin CNC, mesin asah dan lain-lain serta sarana kantor seperti PC, Printer, Scanner, Faximile dan AC.
6. Produk yang dihasilkan PT.Pancamas Elite antara lain Door Accessories, Rosette Door Handle, Plate Door Handle, Round Pull Handle, Square Pull Handle, Windows Accessories.
7. Menggunakan prosedur lama yang telah disusun dalam SOP.
8. Proses pemeliharaan sarana dan prasarana saat ini adalah divisi pemeliharaan hanya menangani perbaikan jika ada laporan perbaikan atau kerusakan dari unit kerja atau pegawai umum.
9. Bagian pemasaran produk ditangani oleh staf pemasaran dengan cara pemesanan (*order*) atau melakukan distribusi langsung kepada toko yang bekerja sama dengan PT.Pancamas Elite. Admin Pemasaran akan merekap dan mengevaluasi pemasaran produk yang sudah dilakukan oleh Staff Pemasaran berdasarkan laporan penjualan yang dibuat manual.
10. Untuk divisi pemeliharaan meliputi proses perbaikan dan perawatan inventaris dan untuk pemasaran meliputi proses pemesanan dan pemantauan produk.