

# ANALISIS DAN PERANCANGAN SISTEM INFORMASI BANK DARAH (STUDI KASUS PMI KOTA MALANG)

SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Dwi Retno Palupi  
NIM: 125150400111064



PROGRAM STUDI SISTEM INFORMASI  
JURUSAN SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2016

## PENGESAHAN

ANALISIS DAN PERANCANGAN SISTEM INFORMASI BANK DARAH  
(STUDI KASUS PMI KOTA MALANG)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh:

Dwi Retno Palupi

NIM: 125150400111064

Skripsi ini telah diuji dan dinyatakan lulus pada  
08 Agustus 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Ismiarta Aknuranda, S.T, M.Sc, Ph.D

NIK: 201006 740719 1 001

Satrio Agung W., S.Kom, M.kom

NIP: 19860521 201212 1 001

Mengetahui

Ketua Jurusan Sistem Informasi

Herman Tolle, Dr. Eng., S.T, M.T

NIP: 19740823 200012 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, 08 Agustus 2016



Dwi Retno Palupi

NIM: 125150400111064

## KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul **“Analsis dan Perancangan Sistem Informasi Bank Darah (Studi Kasus PMI Kota Malang)”**.

Penyusunan skripsi ini juga tak lepas dari bantuan semua pihak yang telah memberikan dukungan, doa, bimbingan, kritik, saran maupun motivasi dari banyak pihak. Oleh karena itu, ucapan terima kasih penulis sampaikan kepada:

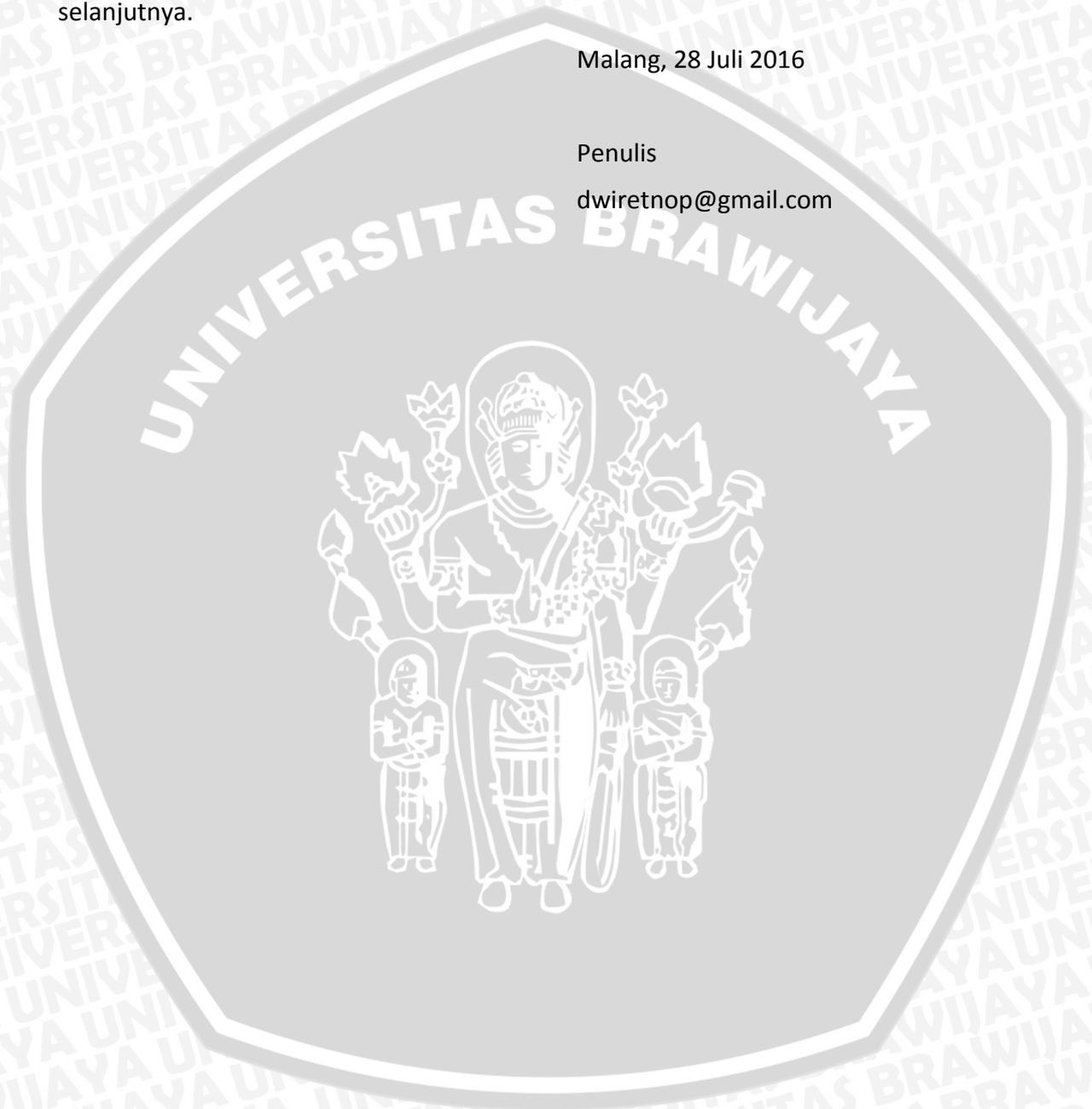
1. Bapak Ismiarta Aknuranda, S.T, M.Sc, Ph.D selaku dosen pembimbing I, atas segala bimbingan dan waktu yang telah diluangkan serta kritik dan saran yang telah diberikan sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak Satrio Agung W., S.Kom, M.kom selaku dosen pembimbing II, atas segala bimbingan dan waktu yang telah diluangkan serta kritik dan saran yang telah diberikan sehingga penulis dapat menyelesaikan skripsi ini.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer.
4. Bapak Herman Tolle, Dr. Eng., S.T,M.T dan Bapak Ismiarta Aknuranda, S.T, M.Sc, Ph.D selaku Ketua dan Sekretaris Jurusan Sistem Informasi Universitas Brawijaya.
5. Bapak Suprpto, S.T, M.T selaku Ketua Program Studi Sistem Informasi Universitas Brawijaya.
6. Seluruh Bapak dan Ibu dosen Fakultas Ilmu Komputer Universitas Brawijaya atas segala bimbingan serta ilmu yang telah diajarkan kepada penulis.
7. Kedua orang tua penulis, Bapak Sudarso dan Ibu Mujiartik yang telah memberikan motivasi, kasih sayang serta dukungan moril dan meteril. Seluruh keluarga besar yang telah memberikan semangat, doa dan dukungan penuh dalam penyusunan skripsi ini.
8. Riska Puji Sudarwati, Liana Rahmadhani, Alzalea Belvanny Permana, Fakhry Ikhsan Firdaus yang senantiasa memberikan dukungan dan banyak bantuan.
9. Seluruh pegawai dan staf Fakultas Ilmu Komputer Universitas Brawijaya.
10. Seluruh teman-teman mahasiswa Program Studi Sistem Informasi khususnya angkatan 2012.
11. Teman seperjuangan; Mini, Voni, Nely, Icha, Putri, Rara, Isti, Bela, Lia, Anita, Mila, Dio, Eta.
12. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Dengan segala kerendahan hati, penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari sempurna serta banyak kekurangan disebabkan oleh keterbatasan kemampuan dan pengalaman, dengan kerendahan hati penulis mengharapkan kritik dan saran yang bersifat membangun dari pembaca. Semoga penulisan laporan skripsi ini bermanfaat bagi pembaca dan untuk pengembangan selanjutnya.

Malang, 28 Juli 2016

Penulis

[dwiretnop@gmail.com](mailto:dwiretnop@gmail.com)



## ABSTRAK

Palang Merah Indonesia (PMI) merupakan lembaga sosial kemanusiaan yang netral dan mandiri didirikan dengan tujuan membantu meringankan penderitaan sesama manusia. Salah satu tugas yang dilakukan PMI adalah pelayanan transfusi darah, pengolahan dan penyediaan darah, yang terkadang dalam beberapa kesempatan kehabisan stok darah. Terdapat beberapa pasien kesulitan mendapatkan informasi ketersediaan darah, sehingga harus pergi mencari ke PMI atau ke Rumah Sakit lain. Hal ini membutuhkan waktu yang lama, mengingat rentannya pasien yang tidak bisa diselamatkan karena terlambat mendapatkan pertolongan. Informasi mengenai jumlah stok darah di PMI dan Rumah Sakit yang dapat diakses oleh masyarakat merupakan hal yang utama, informasi tersebut juga berguna bagi PMI untuk mengontrol ketersediaan darah Rumah Sakit guna menghindari keterlambatan penyediaan darah, meminimalkan ketidaksesuaian antara darah yang dibutuhkan dan yang diberikan, jika Rumah Sakit menghubungi mendadak.

Analisis dan perancangan merupakan salah satu tahapan umum dalam rekayasa perangkat lunak. Perangkat lunak yang dibangun tanpa melalui tahapan dalam rekayasa perangkat lunak yang baik, hampir dapat dipastikan tidak akan mempunyai kualitas yang baik. Oleh karena itu, dilakukan penelitian analisis dan perancangan sistem informasi bank darah berbasis web menggunakan pendekatan berorientasi objek (*Object Oriented Analysis and Design*) menggunakan *Rational Unified Process (RUP) style* untuk memodelkan persyaratan sistem. Hasil penelitian berupa spesifikasi persyaratan sistem, rancangan sistem dan prototipe yang dievaluasi untuk mendapatkan masukan dari pengguna dan menggunakan matriks kerunutan untuk menelusuri hasil persyaratan ke perancangan sistem. Hasil evaluasi antarmuka pengguna menunjukkan bahwa terdapat masukan dari pengguna terkait perancangan antarmuka selanjutnya, untuk matriks kerunutan menunjukkan bahwa seluruh persyaratan dapat ditelusuri ke perancangan sistem.

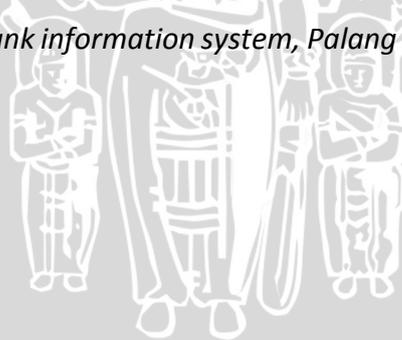
**Kata Kunci:** OOAD, sistem informasi bank darah, Palang Merah Indonesia (PMI).

## ABSTRACT

*Palang Merah Indonesia (PMI) is a social institution humanitarian that is neutral and independent established with the aim to help alleviate the suffering of our fellow human beings as a result of the disaster and for whatever reason. One of the tasks performed by the PMI is the blood transfusion service, processing and the supply of blood. The blood supply is also owned hospital that has a blood bank. There are some patients have difficulty in obtaining information on the availability of blood, so it had to go looking to the PMI or to other hospitals. It takes a long time, considering the patients can not be saved because it too late to get a blood transfusion. Information on blood stock available at the PMI and Hospital, which are accessible by the public is the main thing.*

*Analysis and design is one of the common steps in software engineering. The software is built without going through the stages in software engineering that is good, almost certainly will not have a good quality. Therefore, research analysis and design of information system blood bank web based approach to object Oriented Analysis and Design or OOAD using the Rational Unified Process (RUP) style for modeling system requirements. The results of this study are the system requirements specification, system design and a prototype is evaluated to get feedback from users on the system design to specification and design requirements. The results of the evaluation made by the user indicates that the system design has been as expected. Additionally in this study is used matrices keruntan to discover the result of design to system requirements.*

*Key Words: OOAD, blood bank information system, Palang Merah Indonesia (PMI).*



## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN .....	xix
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan .....	3
1.4 Manfaat.....	3
1.5 Batasan masalah .....	3
1.6 Sistematika pembahasan.....	3
<b>BAB 2 LANDASAN KEPUSTAKAAN .....</b>	<b>5</b>
2.1 Kajian pustaka .....	5
2.2 Profil Palang Merah Indonesia (PMI).....	5
2.3 Definisi pelayanan transfusi darah .....	5
2.4 Proses bisnis.....	6
2.4.1 BPMN (Business Process Model and Notation) .....	6
2.5 Konsep Dasar Sistem Informasi .....	8
2.5.1 Komponen Sistem Informasi.....	8
2.6 Analisis dan Perancangan Sistem Informasi .....	8
2.6.1 Pengertian Analisis Sistem .....	8
2.6.2 Definisi Desain Sistem .....	9
2.7 Analisis dan Desain Berorientasi Objek .....	9
2.7.1 Analisis Berorientasi Objek .....	9
2.7.2 Desain Berorientasi Objek.....	9

2.8 Rational Unified Process RUP .....	9
2.9 Langkah-langkah dalam menganalisis sistem .....	11
2.9.1 Pemangku kepentingan (Stakeholder).....	11
2.9.2 Mengidentifikasi tipe pemangku kepentingan (stakeholder).....	11
2.9.3 Mengidentifikasi perwakilan dan peran pemangku kepentingan .....	11
2.9.4 Analisis permasalahan.....	12
2.9.5 Kebutuhan pemangku kepentingan dan pengguna.....	12
2.9.6 Fitur .....	12
2.9.7 Pernyataan posisi produk.....	13
2.10 Langkah-langkah perancangan sistem.....	13
2.10.1 Spesifikasi persyaratan dan pemodelan <i>use case</i> .....	14
2.10.2 Analisis arsitektur.....	14
2.10.3 Analisis use case.....	14
2.10.4 Identifikasi elemen desain .....	15
2.10.5 Perancangan database.....	15
2.11 Unified Modeling Language (UML).....	15
2.12 Evaluasi .....	21
2.12.1 Tinjauan (Review).....	21
2.12.2 Evaluasi antarmuka pengguna .....	22
2.12.3 Verifikasi dan Validasi .....	25
<b>BAB 3 METODOLOGI .....</b>	<b>26</b>
3.1 Studi literatur .....	26
3.2 Pengumpulan data.....	26
3.3 Analisis proses bisnis.....	27
3.4 Analisis persyaratan sistem .....	27
3.5 Perancangan sistem .....	27
3.6 Evaluasi .....	28
3.7 Kesimpulan dan saran.....	28
<b>BAB 4 PEMODELAN PROSES BISNIS DAN ANALISIS PERSYARATAN SISTEM .....</b>	<b>29</b>
4.1 Analisis proses bisnis.....	29
4.1.1 Proses bisnis mengelola stok darah kondisi saat ini (as is).....	29



4.1.2	Proses bisnis mengelola stok darah kondisi usulan (to be) .....	29
4.1.3	Proses bisnis mengelola data rumah sakit kondisi saat ini (as is)30	
4.1.4	Proses bisnis mengelola data rumah sakit kondisi usulan (to be) .....	30
4.1.5	Proses bisnis mengelola data pendonor kondisi saat ini (as is)..	31
4.1.6	Proses bisnis mengelola data pendonor kondisi usulan (to be) .	31
4.1.7	Mengelola darah keluar dan mengecek ketersediaan darah rumah sakit kondisi saat ini (as is) .....	32
4.1.8	Mengelola darah keluar dan mengecek ketersediaan darah rumah sakit kondisi usulan (to be) .....	32
4.1.9	Proses bisnis pencarian darah kondisi saat ini (as is) .....	33
4.1.10	Proses bisnis pencarian darah kondisi usulan (to be).....	33
4.2	Pemetaan proses bisnis dengan persyaratan fungsional .....	34
4.3	Gambaran umum sistem .....	34
4.4	Identifikasi aktor .....	35
4.5	Analisis masalah.....	36
4.6	Analisis pemangku kepentingan .....	36
4.6.1	Tipe pemangku kepentingan.....	36
4.7	Kebutuhan pengguna dan pemangku kepentingan .....	37
4.8	Pernyataan posisi produk .....	38
4.9	Fitur .....	39
4.10	Persyaratan fungsional .....	40
4.11	Use case diagram .....	41
4.12	Tabel pemetaan diagram use case .....	43
4.13	Spesifikasi use case .....	43
4.14	Activity Diagram.....	56
4.15	Perancangan kelas analisis.....	64
4.15.1	Mekanisme analisis .....	67
4.15.2	Pengintegrasian kelas analisis.....	67
4.15.3	Package diagram .....	68
<b>BAB 5 PERANCANGAN SISTEM</b> .....		<b>70</b>
5.1	Perancangan interaksi dan kelas .....	70
5.1.1	Diagram interaksi (sequence diagram) .....	70



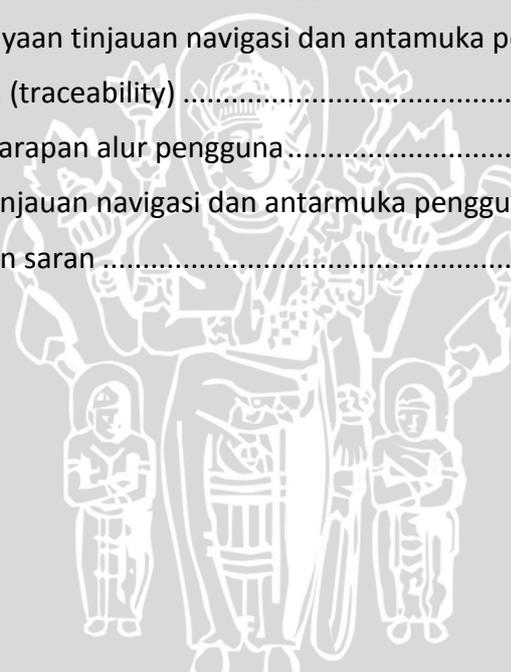
5.1.2 Diagram Kelas.....	82
5.2 Perancangan basis data .....	83
5.3 Perancangan antarmuka sistem .....	85
<b>BAB 6 EVALUASI DAN ANALISIS.....</b>	<b>90</b>
6.1 Tinjauan (review) .....	90
6.1.1 Traceability .....	90
6.1.2 Walkthrough .....	90
6.2 Evaluasi prototipe .....	93
6.3 Hasil evaluasi.....	94
6.3.1 Hasil tinjauan pengecekan traceability .....	94
6.3.2 Hasil tinjauan walkthrough .....	97
6.3.3 Hasil evaluasi prototipe.....	97
6.4 Analisis hasil evaluasi.....	98
6.4.1 Analisis hasil tinjauan.....	98
6.4.2 Analisis hasil prototipe .....	98
6.5 Analisis hasil evaluasi keseluruhan.....	98
<b>BAB 7 PENUTUP .....</b>	<b>102</b>
7.1 Kesimpulan.....	102
7.2 Saran .....	103
<b>BAB 8 DAFTAR PUSTAKA .....</b>	<b>104</b>
<b>LAMPIRAN A HALAMAN ANTARMUKA SISTEM .....</b>	<b>106</b>
A.1 Mengecek ketersediaan darah Rumah Sakit.....	106
A.2 Mengecek ketersediaan darah PMI.....	106
A.3 Autentikasi.....	107
A.4 Menambah darah masuk.....	107
A.5 Melihat data darah .....	108
A.6 Menambah darah keluar atau transaksi ke Rumah Sakit.....	108
A.7 Menambah darah keluar atau transaksi ke pasien .....	109
A.8 Menambah darah keluar atau transaksi untuk darah rusak .....	109
A.9 Melihat data darah keluar atau transaksi .....	109
A.10 Menambah Rumah Sakit .....	110
A.11 Memperbarui data Rumah Sakit .....	111

A.12 Melihat data Rumah Sakit .....	111
A.13 Menambah pendonor .....	112
A.14 Melihat data pendonor .....	112
A.15 RS Mengecek ketersediaan darah Rumah Sakit .....	113
A.16 RS menambah darah keluar atau transaksi ke pasien .....	113
A.17 RS menambah darah keluar atau transaksi darah rusak.....	113
A.18 Menambah user atau pengguna .....	114
A.19 Melihat data user atau pengguna .....	115
<b>LAMPIRAN B HASIL EVALUASI PENGGUNA .....</b>	<b>116</b>
B.1 Hasil tinjauan prototipe oleh responden 1.....	116
B.2 Hasil tinjauan prototipe oleh responden 1.....	117
B.3 Hasil tinjauan prototipe oleh responden 2.....	118
B.4 Hasil tinjauan prototipe oleh responden 2.....	119
B.5 Hasil tinjauan prototipe oleh responden 3.....	120
B.6 Hasil tinjauan prototipe oleh responden 3.....	121
B.7 Hasil tinjauan prototipe oleh responden 4.....	122
B.8 Hasil tinjauan prototipe oleh responden 4.....	123
B.9 Hasil tinjauan prototipe oleh responden 5.....	124
B.10 Hasil tinjauan prototipe oleh responden 5.....	125

## DAFTAR TABEL

Tabel 2.1 Template problem statement .....	12
Tabel 2.2 Template fitur.....	13
Tabel 2.3 Template pernyataan posisi produk .....	13
Tabel 2.4 Notasi kelas diagram .....	16
Tabel 2.5 Notasi use case diagram.....	17
Tabel 2.6 Notasi sequence diagram.....	18
Tabel 2.7 Notasi activity diagram.....	20
Tabel 2.8 Kuisisioner WEBUSE per kategori .....	23
Tabel 2.9 Kuisisioner WEBUSE kategori navigasi dan antarmuka pengguna .....	24
Tabel 4.1 Pemetaan proses bisnis dengan persyaratan fungsional.....	34
Tabel 4.2 Identifikasi aktor.....	35
Tabel 4.3 Analisis masalah .....	36
Tabel 4.4 Analisis masalah .....	36
Tabel 4.5 Tipe pemangku kepentingan.....	37
Tabel 4.6 Kebutuhan pemangku kepentingan.....	37
Tabel 4.7 Pernyataan posisi produk.....	38
Tabel 4.8 Fitur .....	39
Tabel 4.9 Persyaratan fungsional sistem .....	40
Tabel 4.10 Pemetaan use case dengan persyaratan fungsional.....	43
Tabel 4.11 Spesifikasi use case mengelola data stok darah .....	43
Tabel 4.12 Spesifikasi use case mengelola data pendonor.....	46
Tabel 4.13 Spesifikasi use case mengelola data rumah sakit .....	48
Tabel 4.14 Spesifikasi use case mengelola user.....	50
Tabel 4.15 Spesifikasi use case mengelola darah keluar .....	52
Tabel 4.16 Skenario use case mengecek data darah rumah sakit .....	54
Tabel 4.17 Spesifikasi use case mengecek ketersediaan darah pmi dan rumah sakit .....	54
Tabel 4.18 Spesifikasi use case autentikasi.....	55
Tabel 4.19 Kelas analisis.....	64
Tabel 4.20 Kelas analisis.....	67

Tabel 4.21 Kelas analisis.....	69
Tabel 5.1 Detail atribut perancangan basis data .....	84
Tabel 5.2 Keterangan antarmuka autentikasi.....	85
Tabel 5.3 Keterangan antarmuka mengelola stok darah.....	86
Tabel 5.4 Keterangan antarmuka menampilkan data darah masuk.....	87
Tabel 5.5 Keterangan antarmuka memperbaiki data Rumah Sakit .....	88
Tabel 5.6 Keterangan antarmuka mengecek ketersediaan darah PMI .....	89
Tabel 6.1 Daftar pertanyaan traceability .....	90
Tabel 6.2 Daftar tugas kesesuaian harapan pengguna .....	90
Tabel 6.3 Detail alur sistem dan harapan pengguna .....	91
Tabel 6.4 Daftar pertanyaan harapan alur pengguna.....	93
Tabel 6.5 Daftar pertanyaan tinjauan navigasi dan antarmuka pengguna.....	94
Tabel 6.6 Hasil tinjauan (traceability) .....	94
Tabel 6.8 Daftar hasil harapan alur pengguna.....	99
Tabel 6.9 Daftar hasil tinjauan navigasi dan antarmuka pengguna.....	100
Tabel 6.10 Masukan dan saran .....	101



## DAFTAR GAMBAR

Gambar 2.1 Notasi BPMN flow object .....	6
Gambar 2.2 Notasi BPMN data .....	7
Gambar 2.3 Notasi BPMN connecting objects.....	7
Gambar 2.4 Notasi BPMN swimlanes .....	7
Gambar 2.5 Notasi BPMN artifacts .....	7
Gambar 2.6 RUP .....	10
Gambar 2.7 Contoh kelas diagram.....	16
Gambar 2.8 Contoh use case diagram .....	18
Gambar 2.9 Contoh sequence diagram .....	19
Gambar 2.10 Contoh activity diagram .....	21
Gambar 3.1 Metodologi Penelitian.....	26
Gambar 4.1 Proses bisnis mengelola stok darah kondisi saat ini (as is).....	29
Gambar 4.2 Proses bisnis mengelola stok darah kondisi usulan (to be) .....	30
Gambar 4.3 Proses bisnis mengelola data rumah sakit kondisi saat ini (as is).....	30
Gambar 4.4 Proses bisnis mengelola data rumah sakit kondisi usulan (to be) ....	31
Gambar 4.5 Proses bisnis mengelola data pendonor kondisi saat ini (as is).....	31
Gambar 4.6 Proses bisnis mengelola data pendonor kondisi usulan (to be) .....	32
Gambar 4.7 Proses bisnis mengelola darah keluar dan mengecek ketersediaan darah rumah sakit kondisi usulan kondisi saat ini (as is).....	32
Gambar 4.8 Proses bisnis mengelola darah keluar dan mengecek ketersediaan darah rumah sakit kondisi usulan kondisi usulan (to be) .....	33
Gambar 4.9 Proses bisnis pencarian darah kondisi saat ini (as is).....	33
Gambar 4.10 Proses bisnis pencarian darah kondisi usulan (to be).....	34
Gambar 4.11 Diagram use case .....	42
Gambar 4.12 Activity diagram mengelola stok darah (darah masuk) .....	57
Gambar 4.13 Activity diagram mengelola stok darah (darah keluar).....	58
Gambar 4.14 Activity diagram mengelola data pendonor .....	59
Gambar 4.15 Activity diagram mengelola data pendonor .....	60
Gambar 4.16 Activity diagram mengelola user.....	61
Gambar 4.17 Activity darah keluar .....	62



Gambar 4.18 Activity mengecek data darah Rumah Sakit .....	63
Gambar 4.19 Activity mengecek ketersediaan darah PMI dan Rumah Sakit .....	64
Gambar 4.20 Activity autentikasi .....	64
Gambar 4.21 Pengintegrasian kelas analisis .....	68
Gambar 4.22 Package diagram .....	69
Gambar 5.1 Sequence diagram autentikasi .....	70
Gambar 5.2 Sequence diagram menambah darah masuk .....	71
Gambar 5.3 Sequence diagram memperbarui data darah masuk .....	71
Gambar 5.4 Sequence melihat data darah masuk .....	72
Gambar 5.5 Sequence diagram menambah transaksi .....	72
Gambar 5.6 Sequence diagram memperbarui data transaksi .....	73
Gambar 5.7 Sequence diagram melihat data transaksi .....	73
Gambar 5.8 Sequence diagram menambah data pendonor .....	74
Gambar 5.9 Sequence diagram memperbarui data pendonor .....	75
Gambar 5.10 Sequence diagram melihat data pendonor .....	75
Gambar 5.11 Sequence diagram menambah data Rumah Sakit .....	76
Gambar 5.12 Sequence diagram memperbarui Rumah Sakit .....	76
Gambar 5.13 Sequence diagram melihat Rumah Sakit .....	77
Gambar 5.14 Sequence diagram menambah user .....	77
Gambar 5.15 Sequence diagram memperbarui user .....	78
Gambar 5.16 Sequence diagram melihat user .....	78
Gambar 5.17 Sequence diagram menambah darah keluar .....	79
Gambar 5.18 Sequence diagram memperbarui darah keluar .....	80
Gambar 5.19 Sequence diagram melihat darah keluar .....	80
Gambar 5.20 Sequence diagram mengecek data darah Rumah Sakit .....	81
Gambar 5.21 Sequence diagram mengecek ketersediaan darah PMI dan Rumah Sakit .....	81
Gambar 5.22 Kelas diagram .....	82
Gambar 5.23 Physycal Data Model .....	83
Gambar 5.24 Antarmuka autentikasi .....	85
Gambar 5.25 Antarmuka mengelola stok darah .....	86
Gambar 5.26 Antarmuka menampilkan data darah masuk .....	87



Gambar 5.27 Antarmuka memperbarui data Rumah Sakit .....	88
Gambar 5.28 Antarmuka mengecek ketersediaan darah PMI .....	89
Gambar 5.1 Traceability .....	96
Gambar 7.1 Antarmuka mengecek ketersediaan darah Rumah Sakit.....	106
Gambar 7.2 Antarmuka mengecek ketersediaan darah PMI .....	106
Gambar 7.3 Antarmuka autentikasi .....	107
Gambar 7.4 Antarmuka menambah darah masuk .....	107
Gambar 7.5 Antarmuka melihat data darah .....	108
Gambar 7.6 Antarmuka transaksi ke Rumah Sakit .....	108
Gambar 7.7 Antarmuka transaksi ke pasien .....	109
Gambar 7.8 Antarmuka darah rusak.....	109
Gambar 7.9 Antarmuka melihat transaksi .....	110
Gambar 7.10 Antarmuka menambah Rumah Sakit .....	110
Gambar 7.11 Antarmuka memperbarui data Rumah Sakit .....	111
Gambar 7.12 Antarmuka melihat data Rumah Sakit .....	111
Gambar 7.13 Antarmuka menambah pendonor .....	112
Gambar 7.14 Antarmuka melihat data pendonor .....	112
Gambar 7.15 Antarmuka Rumah Sakit mengecek ketersediaan darah Rumah Sakit .....	113
Gambar 7.16 Antarmuka Rumah Sakit transaksi ke pasien.....	113
Gambar 7.17 Antarmuka Rumah Sakit darah rusak.....	114
Gambar 7.18 Antarmuka menambah user atau pengguna .....	114
Gambar 7.19 Antarmuka melihat data user atau pengguna .....	115
Gambar 7.20 Evaluasi pengguna responden 1 .....	116
Gambar 7.21 Evaluasi pengguna responden 1 .....	117
Gambar 7.22 Evaluasi pengguna responden 2 .....	118
Gambar 7.23 Evaluasi pengguna responden 2 .....	119
Gambar 7.24 Evaluasi pengguna responden 3 .....	120
Gambar 7.25 Evaluasi pengguna responden 3 .....	121
Gambar 7.26 Evaluasi pengguna responden 4 .....	122
Gambar 7.27 Evaluasi pengguna responden 4 .....	123
Gambar 7.28 Evaluasi pengguna responden 5 .....	124





## DAFTAR LAMPIRAN

LAMPIRAN A HALAMAN ANTARMUKA SISTEM .....	106
A.1 Mengecek ketersediaan darah Rumah Sakit.....	106
A.2 Mengecek ketersediaan darah PMI.....	106
A.3 Autentikasi.....	107
A.4 Menambah darah masuk.....	107
A.5 Melihat data darah .....	108
A.6 Menambah darah keluar atau transaksi ke Rumah Sakit.....	108
A.7 Menambah darah keluar atau transaksi ke pasien .....	109
A.8 Menambah darah keluar atau transaksi untuk darah rusak .....	109
A.9 Melihat data darah keluar atau transaksi .....	109
A.10 Menambah Rumah Sakit .....	110
A.11 Memperbarui data Rumah Sakit .....	111
A.12 Melihat data Rumah Sakit .....	111
A.13 Menambah pendonor .....	112
A.14 Melihat data pendonor .....	112
A.15 RS Mengecek ketersersediaan darah Rumah Sakit .....	113
A.16 RS menambah darah keluar atau transaksi ke pasien .....	113
A.17 RS menambah darah keluar atau transaksi darah rusak.....	113
A.18 Menambah user atau pengguna .....	114
A.19 Melihat data user atau pengguna .....	115
LAMPIRAN B HASIL EVALUASI PENGGUNA.....	116
B.1 Hasil tinjauan prototipe oleh responden 1.....	116
B.2 Hasil tinjauan prototipe oleh responden 1.....	117
B.3 Hasil tinjauan prototipe oleh responden 2.....	118
B.4 Hasil tinjauan prototipe oleh responden 2.....	119
B.5 Hasil tinjauan prototipe oleh responden 3.....	120
B.6 Hasil tinjauan prototipe oleh responden 3.....	121
B.7 Hasil tinjauan prototipe oleh responden 4.....	122
B.8 Hasil tinjauan prototipe oleh responden 4.....	123
B.9 Hasil tinjauan prototipe oleh responden 5.....	124

B.10 Hasil tinjauan prototipe oleh responden 5..... 125



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Palang Merah Indonesia (PMI) merupakan lembaga sosial kemanusiaan yang netral dan mandiri. PMI didirikan dengan tujuan membantu meringankan penderitaan sesama manusia akibat bencana dan apapun sebabnya, tanpa membedakan latar belakang, politik, suku, ras maupun agama dari korban yang ditolong (Sapta, 2009). Salah satu tugas yang dilakukan oleh PMI adalah pelayanan transfusi darah, pengolahan dan penyediaan darah yang tepat bagi masyarakat yang membutuhkan. Palang Merah Indonesia (PMI) sebagai pusat penyimpanan stok darah, dalam beberapa kesempatan tidak jarang kehabisan stok darah, sehingga hal ini dapat menyulitkan bagi para pasien yang sedang membutuhkan transfusi darah (Utomo, 2010).

Penyediaan darah juga dimiliki Rumah Sakit yang memiliki Bank Darah. Diketahui terdapat beberapa pasien mengalami kesulitan dalam mencari darah yang sesuai dengan yang dibutuhkan, pada saat pasien berada di Rumah Sakit dan sedang membutuhkan darah, namun Bank Darah yang terdapat di Rumah Sakit telah kehabisan stok darah, sehingga pasien harus mencari ke PMI atau ke Rumah Sakit lain yang memiliki bank darah. Hal ini membutuhkan waktu yang lama, mengingat rentannya pasien yang tidak bisa diselamatkan karena terlambat mendapatkan transfusi darah. Kebutuhan akan pelayanan darah yang berkualitas, mudah di akses serta dalam jumlah yang cukup semakin meningkat khususnya dalam menurunkan angka Kematian Ibu (AKI) yang terbilang masih cukup tinggi di Indonesia (Depkes RI, 2008).

Informasi mengenai jumlah stok darah yang tersedia di PMI dan Rumah Sakit yang dapat diakses oleh masyarakat merupakan hal yang utama yang dibutuhkan oleh masyarakat. Di PMI kota Malang terdapat sistem informasi mengenai jumlah stok darah, namun hanya terbatas jumlah darah yang ada di PMI. Di Rumah Sakit yang memiliki Bank Darah belum terdapat sistem informasi mengenai jumlah ketersediaan darah yang dapat di akses oleh masyarakat. Kegunaan sistem informasi tersebut adalah untuk memudahkan masyarakat mendapatkan informasi mengenai tempat ketersediaan darah yang dibutuhkan, sehingga masyarakat dapat langsung menuju ke tempat ketersediaan darah (PMI atau Rumah Sakit). Selain berguna untuk masyarakat, sistem informasi tersebut diharapkan dapat membantu pihak PMI untuk mengontrol ketersediaan darah yang ada di Rumah Sakit untuk menghindari keterlambatan dalam penyediaan darah, meminimalkan ketidaksesuaian antara darah yang dibutuhkan dengan darah yang diberikan. Selama ini Rumah Sakit menghubungi PMI untuk mendapatkan darah melalui via telepon, hal ini memungkinkan ketidaksesuaian permintaan dengan ketersediaan jika Rumah Sakit menghubungi dalam keadaan mendadak.

Dari masalah yang telah dijelaskan terkait kesulitan masyarakat dalam mendapatkan informasi mengenai tempat ketersediaan darah dan

ketidaksihayaan antara permintaan darah dan ketersediaan darah, maka dalam penelitian ini dilakukan sebuah analisis dan perancangan sebelum membangun suatu sistem informasi yang di dalamnya terdapat perangkat lunak. Analisis dan perancangan merupakan salah satu tahapan umum dalam rekayasa perangkat lunak. Perangkat lunak yang dibangun tanpa melalui tahapan-tahapan dalam rekayasa perangkat lunak yang baik, hampir dapat dipastikan tidak akan mempunyai kualitas yang baik (Rosa & Shalahuddin, 2014).

Penelitian pernah dilakukan oleh Rian Fitriah dengan judul analisis dan perancangan perpustakaan digital dan sistem otomatisasi perpustakaan SMAN 3 Palembang. Penelitian ini membahas tentang analisis dan perancangan sebuah sistem informasi untuk perpustakaan SMAN 3 Palembang dengan menggunakan metode *Object Oriented* (Fitriah, 2012). Keterkaitan dengan penelitian yang dilakukan oleh yaitu berupa metode *Object Oriented* yang digunakan. Selain itu penelitian juga dilakukan oleh Nanda Fitriano dengan judul analisis dan perancangan wiki budaya dalam rangka melestarikan budaya bangsa dan kearifan lokal nusantara (Nugraha, et al., 2014). Keterkaitan dengan penelitian yang dilakukan penulis yaitu berupa bahasa pemodelan sistem yang menggunakan UML dan evaluasi dengan menggunakan matriks keruntutan.

Dengan adanya masalah yang terjadi berdasarkan pengalaman penulis, dan mengingat perkembangan teknologi informasi saat ini telah merambah ke seluruh sektor kehidupan manusia. Maka dilakukan penelitian analisis dan perancangan sistem informasi bank darah berbasis web dengan judul **“Analisis dan Perancangan Sistem Informasi Bank Darah (Studi Kasus PMI Kota Malang)”** menggunakan pendekatan analisis dan perancangan berorientasi objek (*Object Oriented Analysis and Design* atau OOAD) dan menggunakan *Rational Unified Process* (RUP) style untuk memodelkan persyaratan sistem. Hasil penelitian ini berupa proses bisnis usulan, spesifikasi persyaratan sistem, rancangan sistem dan sebuah prototipe yang dievaluasi untuk mendapatkan saran dan masukan dari pengguna. Selain itu evaluasi juga menggunakan matriks keruntutan atau *traceability* untuk menelusuri hasil perancangan ke persyaratan sistem. Hasil evaluasi antarmuka pengguna menunjukkan bahwa terdapat beberapa saran dan masukan dari pengguna guna memperbaiki perancangan antarmuka selanjutnya, hasil untuk evaluasi *traceability* menunjukkan bahwa seluruh rancangan dapat ditelusuri ke persyaratan sistem.

## 1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan, dapat dirumuskan permasalahan dalam penelitian ini adalah sebagai berikut:

1. Bagaimanakah proses bisnis yang terjadi saat ini dan proses bisnis yang diusulkan?
2. Bagaimanakah spesifikasi persyaratan dari sistem yang akan dibuat?
3. Bagaimanakah rancangan dari sistem yang akan dibuat?
4. Bagaimana hasil evaluasi dari perancangan yang telah dibuat?

### 1.3 Tujuan

Tujuan dari analisis dan perancangan terkait dengan penelitian yang dilakukan adalah:

1. Untuk menganalisis proses bisnis yang terjadi saat ini dan membuat model proses bisnis usulan.
2. Untuk membuat persyaratan sistem dalam membuat perancangan sistem bank darah.
3. Untuk membuat rancangan antarmuka yang sesuai dengan persyaratan sistem.
4. Untuk menghasilkan evaluasi dari persyaratan sistem yang telah dirancang.

### 1.4 Manfaat

Hasil dari analisis dan perancangan sistem informasi ini diharapkan dapat memberikan manfaat umum, antara lain:

1. Memudahkan untuk tahap pengembangan lebih lanjut sebagai acuan untuk memulai tahapan implementasi.
2. Memberikan wawasan terkait pemodelan proses bisnis, analisis dan perancangan dan evaluasi hasil analisis dan perancangan sistem.

### 1.5 Batasan masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sistem yang dirancang berbentuk prototipe berbasis *web*.
2. Sistem yang dirancang untuk memudahkan pasien mendapatkan informasi stok darah sesuai dengan yang dibutuhkan tanpa perlu pergi ke rumah sakit atau PMI terlebih dahulu, dengan studi kasus PMI Kota Malang.
3. Perancangan di fokuskan pada fungsional sistem.
4. Perancangan yang dilakukan merupakan perancangan awal yang sebaiknya dilanjutkan dengan perancangan detail di penelitian berikutnya sebelum memulai tahap implementasi.

### 1.6 Sistematika pembahasan

Adapaun sistematika laporan skripsi yang ditujukan untuk memberikan gambaran dan uraian secara garis besar adalah sebagai berikut:

BAB I : PENDAHULUAN

Bab ini diuraikan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan dalam penulisan laporan.

BAB II : LANDASAN KEPUSTAKAAN

Bab ini menjelaskan secara singkat mengenai profil instansi dan dasar teori mengenai perancangan sistem yang dibuat.

**BAB III : METODOLOGI**

Bab ini menguraikan tentang perancangan dan aplikasi sistem informasi pencari darah PMI Kota Malang.

**BAB IV : PEMODELAN PROSES BISNIS DAN ANALISIS PERSYARATAN SISTEM**

Bab ini berisi tentang analisis proses bisnis yang ada saat ini dan usulan, dan analisis persyaratan sistem.

**BAB V : PERANCANGAN SISTEM**

Bab ini dijelaskan mengenai perancangan sistem berdasarkan analisis yang telah dijabarkan.

**BAB VI : EVALUASI**

Bab ini dijelaskan mengenai evaluasi terkait analisis dan perancangan yang dilakukan.

**BAB VII : PENUTUP**

Bab ini dijelaskan kesimpulan dan saran dari proses yang didapat pada pelaksanaan penelitian dan laporan.



## BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan mengenai tinjauan pustaka berupa penelitian serupa yang pernah dilakukan dan dasar teori yang mendukung dan menjadi dasar penelitian.

### 2.1 Kajian pustaka

Berdasarkan penelitian sebelumnya yang pernah dilakukan oleh Rian Fitriah dengan judul analisis dan perancangan perpustakaan digital dan sistem otomasi perpustakaan SMAN 3 Palembang. Penelitian membahas tentang analisis dan perancangan yang dapat digunakan untuk memberikan informasi mengenai judul buku maupun referensi buku yang ada di perpustakaan yang dapat bisa diakses langsung oleh siswa tanpa harus datang langsung ke perpustakaan. Penelitian ini menggunakan metode *Object Oriented*. Hasil dari penelitian ini berupa suatu perancangan sistem informasi berbasis *web* dengan menggunakan bahasa pemodelan UML (Fitriah, 2012).

Penelitian serupa dilakukan oleh Nanda Fitriano dengan judul analisis dan perancangan wiki budaya dalam rangka melestarikan budaya bangsa dan kearifan lokal nusantara. Penelitian membahas tentang hilangnya kebudayaan indonesia. Penelitian ini dilakukan dengan menggunakan metode spiral, menggunakan UML sebagai bahasa pemodelan sistem yang berparadigma berorientasi objek dan menggunakan evaluasi dengan menggunakan matriks kerunutan (Nugraha, et al., 2014). Keterkaitan kedua penelitian yang telah disebutkan dengan penelitian yang dilakukan penulis yaitu berupa bahasa pemodelan sistem yang menggunakan UML dan evaluasi dengan menggunakan matriks kerunutan.

### 2.2 Profil Palang Merah Indonesia (PMI)

Palang Merah Indonesia berdiri pada tahun 1960 yang dipimpin oleh seorang komisar dr. Angka Nitisastro. Program-program yang dilaksanakan oleh PMI Provinsi Jawa Timur dalam rentang waktu lima tahun terakhir meliputi:

1. Penanggulangan bencana (PB).
2. Program pelayanan kesehatan.
3. Program pelayanan sosial.
4. Program peningkatan fungsi atau peran komunikasi dan informasi.

### 2.3 Definisi pelayanan transfusi darah

Pelayanan transfusi darah merupakan upaya pelayanan kesehatan yang terdiri dari serangkaian kegiatan mulai dari pengerahan dan pelestarian donor, pengambilan, pengamanan, pengolahan, penyimpanan darah dan tindakan medis pemberian darah kepada resipien untuk tujuan penyembuhan penyakit dan pemulihan kesehatan.

Setiap kegiatan pelayanan transfusi darah harus dikerjakan sesuai dengan Standar Prosedur Operasional (SOP) karena jika terjadi kesalahan yang terjadi pada setiap langkah kegiatan tersebut akan berakibat fatal bagi resipien, dan hal

ini juga dapat membahayakan pendonor maupun petugas kesehatan yang melaksanakan. Rangkaian kegiatan distribusi darah sampai ke pasien atau resipien harus dilakukan hanya oleh petugas dengan menggunakan peralatan khusus dan sesuai Standar Prosedur Operasional (SOP) (Depkes RI, 2008).

## 2.4 Proses bisnis

Proses bisnis (*business process*) merupakan kumpulan dari proses yang mendukung proses-proses operasional dalam perusahaan. Proses bisnis berisi kumpulan aktivitas (*task*) yang saling berhubungan satu sama lain untuk menghasilkan suatu hasil yang mendukung pada tujuan dan sasaran strategis dari organisasi. Suatu proses bisnis dibangun untuk mendefinisikan secara spesifik aktivitas-aktivitas yang ada didalamnya. Proses bisnis memiliki tiga komponen utama yaitu:

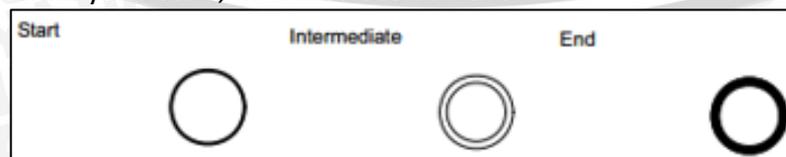
1. *Input*, merupakan material atau *user requirement*.
2. Proses, merupakan material sejumlah aktivitas atau pekerjaan untuk melakukan transformasi input. Dapat berupa serangkaian proses atau tahapan tertentu.
3. *Output*, merupakan hasil akhir yang diinginkan.

Proses bisnis dapat dipahami sebagai kumpulan aktivitas yang dapat mendefinisikan *business event* dan pekerjaan yang dilakukan oleh sebuah sistem untuk merubah *input* menjadi *output* yang bernilai tambah bagi pengguna. Model proses bisnis bisa digunakan sebagai dasar dalam mengidentifikasi kebutuhan pengguna yang merupakan bagian utama dalam pengembangan sistem *enterprise* (Marnasusanto, 2014).

### 2.4.1 BPMN (Business Process Model and Notation)

Menurut OMG (2013) Tujuan dari BPMN adalah untuk memberikan notasi yang mudah dipahami oleh semua pengguna bisnis, mulai dari bisnis analis yang membuat konsep awal dari sebuah proses, ke para pengembang teknis yang bertanggung jawab untuk mengimplementasikan suatu teknologi dari proses-proses tersebut. Dengan demikian, BPMN menciptakan suatu jembatan standar antara desain bisnis proses dan implementasi proses. Terdapat lima dasar kategori elemen yang digunakan, diantaranya:

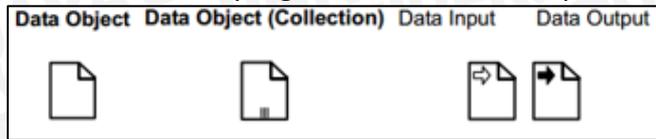
1. *Flow Object* terdiri dari *event*, *activities* dan *gateways*.  
*Event* menggambarkan sesuatu yang terjadi selama berlangsungnya proses bisnis. *Event* ini mempengaruhi aliran proses dan biasanya memiliki penyebab (*trigger*) atau hasil (*result*). Terdapat 3 tipe *event* yang ditunjukkan pada gambar 2.1 yaitu *start*, *intermediate* dan *end*.



Gambar 2.1 Notasi BPMN flow object

Sumber: OMG (2013)

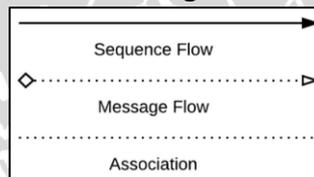
2. Data yang terdiri dari data *objects*, data *inputs*, data *outputs* dan data *stores* ditunjukkan pada gambar 2.2. Data *object* memberikan informasi tentang kegiatan apa yang perlu dilakukan atau apa yang perlu dihasilkan. Data *object* dapat mewakili objek tunggal atau koleksi dari objek. Data *input* dan data *output* memberikan informasi yang sama untuk sebuah proses.



**Gambar 2.2 Notasi BPMN data**

Sumber: OMG (2013)

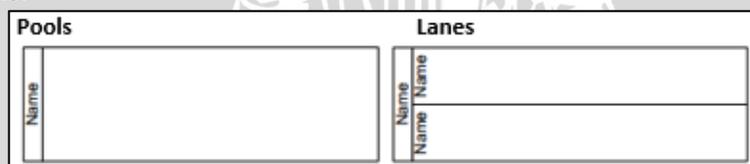
3. *Connecting objects* terdiri dari *sequence flows*, *messages flows* data *associations* ditunjukkan pada gambar 2.3. *Connecting object* merupakan elemen yang menghubungkan *flow object*. *Sequence flow* digunakan untuk menunjukkan urutan yang kegiatan akan dilakukan dalam sebuah proses. *Message flow* digunakan untuk menunjukkan aliran pesan antara dua entitas yang siap untuk mengirim dan menerima. *Association* digunakan untuk asosiasi data, informasi dan artefak dengan aliran beda.



**Gambar 2.3 Notasi BPMN connecting objects**

Sumber: OMG (2013)

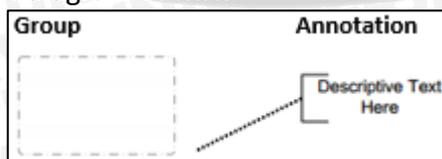
4. *Swimlanes* terdiri dari *pools* dan *lanes* ditunjukkan pada gambar 2.4. *Pool* yang mewakili partisipan dalam sebuah proses dan *lane* merupakan sub-bagian dalam sebuah *pool* dan akan menambah panjang dari *pool* baik vertikal maupun horizontal. *Lanes* digunakan untuk mengatur dan mengkategorikan aktivitas.



**Gambar 2.4 Notasi BPMN swimlanes**

Sumber: OMG (2013)

5. *Artifacts* terdiri dari *group* dan *annotation* ditunjukkan pada gambar 2.5. *Group* digunakan untuk tujuan dokumentasi atau analisis. *Annotation* merupakan mekanisme untuk pemodel memberikan informasi teks tambahan untuk pembaca dari diagram BPMN.



**Gambar 2.5 Notasi BPMN artifacts**

Sumber: OMG (2013)

## 2.5 Konsep Dasar Sistem Informasi

Menurut O'Brien dan Marakas (2014) sistem merupakan seperangkat komponen yang saling berkaitan, dengan sebuah batasan yang relatif jelas, bekerja bersama-sama untuk mencapai sebuah tujuan yang sama.

Sebuah sistem informasi (SI) bisa terdiri atas kombinasi terorganisasi berupa manusia, perangkat keras atau hardware, perangkat lunak, jaringan komunikasi, sumber data dan kebijakan serta prosedur yang terorganisasi yang menyimpan, mengambil, mengubah dan memisahkan informasi dalam sebuah organisasi.

### 2.5.1 Komponen Sistem Informasi

Komponen sistem informasi terdiri dari:

1. Sumber daya manusia, dibutuhkan untuk pengoperasian semua sistem informasi. Sumber daya manusia meliputi pemakai akhir dan pakar SI.
  - a. Pemakai akhir adalah orang yang menggunakan sistem informasi atau informasi yang dihasilkan sistem tersebut.
  - b. Pakar SI adalah orang yang mengembangkan dan mengoperasikan sistem informasi. Meliputi analisis sistem dan staf administrasi SI.
2. Sumber daya *hardware*, meliputi semua peralatan dan bahan fisik yang digunakan dalam pemrosesan informasi.
3. Sumber daya *software*, meliputi semua rangkaian perintah pemrosesan informasi.
4. Sumber daya data. Data membentuk suatu sumber daya organisasi yang berharga. Sehingga data harus dilihat sebagai sumber daya data yang harus dikelola secara efektif agar dapat memberi manfaat para pemakai akhir dalam sebuah organisasi.
5. Sumber daya jaringan. Sumber daya jaringan menekankan bahwa teknologi komunikasi dan jaringan adalah komponen sumber daya dasar dari semua sistem informasi.

## 2.6 Analisis dan Perancangan Sistem Informasi

### 2.6.1 Pengertian Analisis Sistem

Menurut Rosa & Shalahudin (2014) analisis sistem merupakan kegiatan untuk melihat sistem yang sudah berjalan, melihat bagian mana yang bagus dan bagian mana yang tidak bagus, dan kemudian mendokumentasikan kebutuhan yang akan dipenuhi dalam sistem yang baru. Analisis sistem dilakukan untuk mengetahui kebutuhan apa saja yang diperlukan oleh sistem untuk mengembangkan sebuah aplikasi, sehingga hasil akhir dari tahap analisis sistem yaitu berupa dokumen yang menjelaskan mengenai spesifikasi kebutuhan sistem informasi atau SRS (*Software Requirement Specification*). Analisis sistem merupakan salah satu tahapan dalam mengembangkan sebuah sistem. Tujuan dari analisis sistem yaitu untuk mencari solusi dari masalah atau kelemahan yang terdapat di sistem lama.

## 2.6.2 Definisi Desain Sistem

Desain atau perancangan adalah tahapan selanjutnya yang harus dilakukan setelah analisis desain, desain atau perancangan dalam pembangunan perangkat lunak merupakan upaya untuk mengkonstruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu dan perangkat. Kualitas perangkat lunak biasanya dinilai dari segi kepuasan pengguna perangkat lunak terhadap perangkat lunak yang digunakan.

## 2.7 Analisis dan Desain Berorientasi Objek

### 2.7.1 Analisis Berorientasi Objek

Analisis berorientasi objek atau *Object Oriented Analysis* (OOA) merupakan tahapan untuk menganalisis spesifikasi atau kebutuhan akan sistem yang akan dibangun dengan konsep berorientasi objek, apakah benar kebutuhan yang ada dapat diimplementasikan menjadi sebuah sistem berorientasi objek.

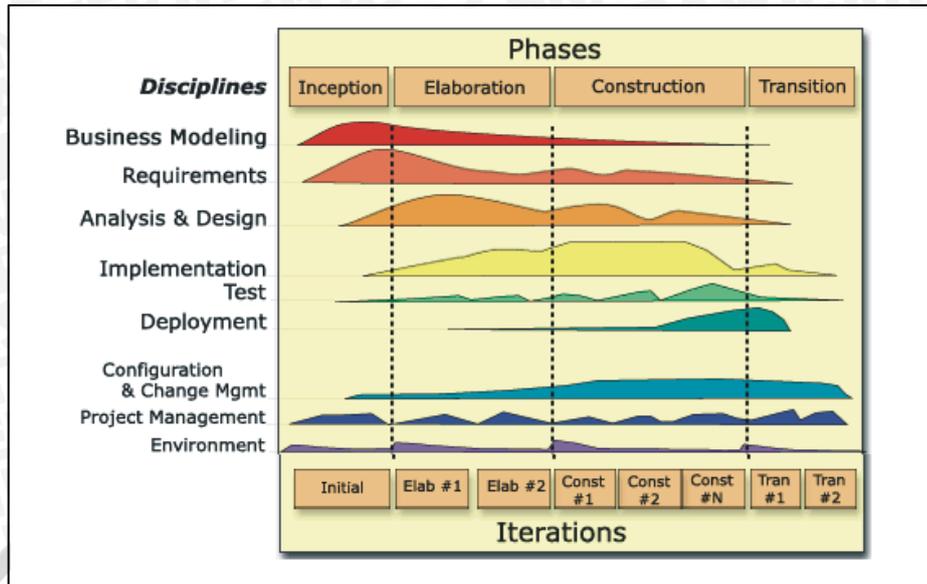
### 2.7.2 Desain Berorientasi Objek

Desain berorientasi objek atau *Object Oriented Design* (OOD) merupakan tahapan perantara untuk memetakan spesifikasi atau kebutuhan sistem yang akan dibangun dengan konsep berorientasi objek ke desain pemodelan agar lebih mudah diimplementasikan dengan pemrograman berorientasi objek. Pemodelan berorientasi objek biasanya dituangkan dalam dokumentasi perangkat lunak dengan menggunakan perangkat pemodelan berorientasi objek, di antaranya adalah UML (Unified Modeling Language).

OOA dan OOD dalam proses yang berulang-ulang seringkali memiliki batasan yang samar, sehingga kedua tahapan tersebut sering disebut dengan OOAD (*Object Oriented Analysis and Design*) atau Analisis dan Desain Berorientasi Objek.

## 2.8 Rational Unified Process RUP

*Rational Unified Process* (RUP) menurut Siahaan (2012) merupakan salah satu kerangka kerja untuk melakukan proses rekayasa kebutuhan. Standar ini menyediakan aturan pendekatan untuk membagi tugas dalam pembangunan perangkat lunak. Tujuan utama dari standar RUP adalah untuk memastikan bahwa perangkat lunak yang sampai kepada pengguna adalah perangkat lunak yang berkualitas baik. Termasuk yang ada di dalamnya adalah untuk memastikan jadwal dan biaya yang tepat. Template RUP memberikan arahan serta gambaran untuk membangun aktivitas yang penting. Pembangunan RUP mengacu pada bahasa Unified Modeling Language (UML). Gambar 2.6 merupakan gambar arsitektur RUP. Menurut IBM (2003) RUP terorganisir menjadi dua struktur yaitu waktu dan isi atau konten. Dimensi waktu merujuk pada aspek siklus hidup proses, sedangkan dimensi konten merujuk pada disiplin berupa aktivitas kelompok yang logis.



**Gambar 2.6 RUP**

Sumber: IBM (2003)

Menurut IBM (2003) konten RUP diatur dalam disiplin ilmu (*disciplines*) yang merupakan semua kegiatan semua yang terkait dengan memperhatikan area utama. Disiplin ilmu (*disciplines*) RUP adalah sebagai berikut:

1. *Business modeling*
2. *Requirement*
3. *Analysis & design*
4. *Implementation*
5. *Test*
6. *Deployment*
7. *Configuration & change management*
8. *Project management*
9. *Environment*

RUP memiliki beberapa fase diantaranya:

1. *Inception* tahap dalam menjelaskan batasan proyek, menjelaskan mengenai kebutuhan primer, sekunder dan vital.
2. *Elaboration* tahap dalam menjelaskan lebih detail mengenai kebutuhan pada fase *inception*.
3. *Construction* tahap membangun sistem secara utuh, dengan menjalankan iterasi, program dan kode.
4. *Transition* tahap dilakukan pengujian sistem. Pengguna melakukan uji coba sistem dan mengadakan pelatihan.

Pada penelitian yang dilakukan, penulis memfokuskan untuk menggunakan standar atau template RUP untuk membuat diagram use case dan spesifikasi use case. Menurut IBM (2007) Template RUP memberikan cara yang baik dalam menuliskan sebuah use case, memberikan kemudahan untuk memahami audience.

## 2.9 Langkah-langkah dalam menganalisis sistem

Sub bab ini menjelaskan mengenai tahap-tahap yang dilakukan dalam menganalisis sistem. Dalam melakukan analisis sistem pada penelitian ini mengacu pada Bitner & Spence (2002).

### 2.9.1 Pemangku kepentingan (Stakeholder)

Pemangku kepentingan merupakan seorang individu yang secara material terpengaruh oleh hasil dari sistem atau proyek. Masyarakat pemangku kepentingan terdiri dari kedua individu yang secara langsung dipengaruhi oleh sistem dengan keterlibatan mereka dalam proyek tersebut dan orang-orang yang secara tidak langsung dipengaruhi oleh sistem.

### 2.9.2 Mengidentifikasi tipe pemangku kepentingan (stakeholder)

Langkah pertama untuk memahami pemangku kepentingan adalah dengan mengidentifikasi tipe pemangku kepentingan yang mempengaruhi sistem. Tipe pemangku kepentingan merupakan klasifikasi pemangku kepentingan dari berbagai karakteristik yang berhubungan dengan sistem.

Berikut merupakan kategori pemangku kepentingan:

1. *User*: jenis yang paling jelas dari pemangku kepentingan adalah pengguna sebenarnya dari sistem yang nantinya akan mengambil peran yang didefinisikan sebagai aktor dalam model use case.
2. *Sponsor*: manajer bisnis, keuangan, pemegang saham dan orang lain yang berinvestasi dalam produksi sistem. Pemangku kepentingan ini merupakan pengguna tidak langsung dari sistem.
3. *Developers*: manajer proyek, pengelola sistem, pengujian, staf pendukung dan jenis yang lainnya dari pengembang yang terlibat langsung dalam produksi sistem.
4. *Authorities*: ahli dalam aspek tertentu dari masalah atau solusi domain. Termasuk departemen tata kelola organisasi, badan pengawas eksternal dan internal, ahli domain dan ahli teknologi.
5. *Customers*: orang atau organisasi yang akan membeli sistem, dapat berupa pembeli, evaluator, akuntan dan organisasi.

Pada dasarnya daftar tipe pemangku kepentingan proyek akan lebih konkrit dari ini. unciannya adalah untuk memastikan semua yang terlibat dalam proyek. Dalam mengidentifikasi tipe stakeholder fokus pada pemahaman bagaimana mereka dipengaruhi oleh proyek dan sistem yang akan dihasilkan.

### 2.9.3 Mengidentifikasi perwakilan dan peran pemangku kepentingan

Tujuan dari mengidentifikasi perwakilan dan peran pemangku kepentingan adalah untuk merekrut pemangku kepentingan untuk terlibat langsung dalam proyek. Pemangku kepentingan merupakan sumber utama dari persyaratan.

#### 2.9.4 Analisis permasalahan

Masalah dapat didefinisikan sebagai perbedaan antara hal-hal seperti yang dirasakan dan hal-hal yang diinginkan. Cara terbaik untuk menangkap masalah adalah untuk membangun sebuah pernyataan terkait masalah. Terdapat template sederhana yang dijelaskan pada tabel 2.1.

**Tabel 2.1 Template problem statement**

The problem of	[describe the problem]
Affects	[the stakeholders affected by the problem]
The impact of which is	[what is the impact of the problem?]
A successful solution would	[list some key benefits of a successful solution]

Sumber: Bitner & Spence (2002)

#### 2.9.5 Kebutuhan pemangku kepentingan dan pengguna

Biasanya, para pemangku kepentingan memiliki perspektif yang berbeda pada masalah dan kebutuhannya. Kebutuhan pemangku kepentingan merupakan sebuah refleksi dari bisnis, masalah pribadi atau operasional yang harus diatasi untuk mempertimbangkan sistem baru. Deskripsi kebutuhan pemangku kepentingan harus mencakup alasan dibalik kebutuhan dan jelas menunjukkan mengapa hal itu penting. Terdapat aturan MoSCoW untuk memprioritaskan kebutuhan pemangku kepentingan. Diantaranya ([www.dsdm.org](http://www.dsdm.org)):

1. **Must Have**  
*Must have* digunakan ketika suatu persyaratan yang harus ada atau harus diimplementasikan, jika persyaratan ini tidak ada maka tidak akan memberikan solusi.
2. **Should Have**  
*Should have* digunakan ketika suatu persyaratan sistem yang penting namun tidak vital. Jika suatu persyaratan tidak diimplementasikan, maka suatu proyek masih bisa dikatakan layak atau masih dapat menyelesaikan permasalahan yang ada.
3. **Could Have**  
*Could have* digunakan ketika suatu persyaratan sistem diinginkan untuk diimplementasikan tetapi persyaratan tersebut kurang penting dan tidak begitu memiliki dampak yang besar jika tidak diimplementasikan.
4. **Won't Have this time**  
*Won't have this time* digunakan ketika suatu persyaratan disepakati untuk tidak dikerjakan pada jangka waktu saat ini.

#### 2.9.6 Fitur

Fitur merupakan kemampuan tingkat tinggi dari sistem untuk memberikan manfaat kepada pengguna dan yang membantu untuk memenuhi pemangku kepentingan dan kebutuhan pengguna. Terdapat cara dalam mendokumentasikan fitur, yaitu:

1. Sertakan deskripsi fungsi dan masalah yang relevan yang harus diatasi.
2. Hindari desian dengan mendeskripsikan fitur pada tingkat umum. Fokus pada kemampuan yang diperlukan dan mengapa (bukan bagaimana) mereka harus dilaksanakan.
3. Memberikan identifier yang unik untuk memudahkan pelacakan.

Dalam memprioritaskan fitur digunakan aturan MoSCoW. Tabel 2.2 merupakan template fitur yang akan digunakan.

**Tabel 2.2 Template fitur**

Identifier	Description	Priority
FEAT1	The system shall allow the caller to place local calls.	Must
FEAT2	The system shall allow the caller to place long-distance calls.	Must
FEAT3	The system shall select the cheapest routing for all long-distance calls.	Should
FEAT4	The system shall provide a continuously up-to-date call history for all accounts.	Must
FEAT5	The system shall be continuously available 24 hours a day, seven days a week.	Should

Sumber: Bitner & Spence (2002)

### 2.9.7 Pernyataan posisi produk

Setiap sistem dibangun dengan tujuan setidaknya untuk satu alasan yang baik. Kita dapat menyatakan dalam hal yang jelas apa yang sistem lakukan dan mengapa melakukannya. Tabel 2.3 merupakan template yang dapat digunakan untuk menyatakan posisi produk.

**Tabel 2.3 Template pernyataan posisi produk**

For	(target customer)
Who	(statement of the need or opportunity)
For	(target customer)
Who	(statement of the need or opportunity)
The	(product name) is a (product category)
That	(statement of key benefit, that is, compelling reason to buy)
Unlike	(primary competitive alternative)
Our product	(statement of primary differentiation)

Sumber: Bitner & Spence (2002)

### 2.10 Langkah-langkah perancangan sistem

Sub bab ini menjelaskan mengenai tahap-tahap yang dilakukan dalam melakukan perancangan sistem. Perancangan sistem yang dilakukan pada penelitian ini mengacu pada IBM (2004).

### 2.10.1 Spesifikasi persyaratan dan pemodelan *use case*

Pemodelan *use case* dan spesifikasi persyaratan yang dilakukan yaitu:

1. Mendefinisikan persyaratan yang dibutuhkan oleh sistem.
2. Memodelkan *use case* dari sistem yang dirancang, yang merupakan dasar untuk proses perancangan selanjutnya.

### 2.10.2 Analisis arsitektur

Analisis arsitektur merupakan aktivitas untuk mendefinisikan pola arsitektur yang akan digunakan dalam pengembangan. Beberapa contoh dari pola arsitektur adalah *layers*, *Model View Controller (MVC)*, *pipes and filters* dan *blackboard*. Di dalam analisis arsitektur terdapat mekanisme arsitektur yang terdiri dari tiga tahapan yaitu:

1. Mekanisme analisis: dilakukan untuk menyediakan perilaku khusus untuk domain yang berkaitan atau sesuai dengan implementasi dari hubungan dengan kelas atau komponen. Beberapa contoh dari mekanisme analisis adalah *persistency*, *distribution*, *communication (IPC and RPC)*, *message routing*, *transaction management*, *security*, *legacy interface* dan lain sebagainya.
2. Mekanisme perancangan: dilakukan dengan mengasumsikan detail dari lingkungan implementasi namun tidak spesifik pada implementasi tertentu. Terdapat beberapa teknologi yang digunakan pada mekanisme perancangan diantaranya *RDBMS*, *OODBMS*, *flash card*, *in-memory storage*, *shared memory* dan lain sebagainya.
3. Mekanisme implementasi: digunakan selama proses implementasi. Mekanisme implementasi digunakan untuk memperbaiki mekanisme perancangan, dan menspesifikasikan implementasi yang tepat. Beberapa contoh pada mekanisme implementasi diantaranya *COTS products*, *database (Oracle, sybase)* dan lain sebagainya.

### 2.10.3 Analisis *use case*

Analisis *use case* merupakan sebuah cara untuk mengidentifikasi kelas awal dari sistem yang akan dibangun. Tujuan dari analisis *use case* adalah sebagai berikut:

1. Mengidentifikasi kelas yang melakukan aliran *use case*.
2. Mendistribusikan perilaku *use case* untuk kelas-kelas dengan menggunakan realisasi *use case*.
3. Mengidentifikasi tanggung jawab, atribut dan asosiasi kelas.

Langkah-langkah dalam menganalisis *use case* adalah sebagai berikut:

1. Melengkapi deskripsi *use case*.
2. Menemukan kelas dari perilaku *use case* yang digunakan untuk mengidentifikasi serangkaian kelas analisis. Kelas analisis merupakan suatu model konseptual awal untuk "sesuatu yang ada dalam sistem yang

memiliki tanggung jawab dan perilaku”. Kelas analisis digunakan untuk menangkap “draft-pertama” dari model object sistem. Kelas analisis menggambarkan persyaratan fungsional dari sistem. Terdapat teknik dalam menemukan kelas analisis dengan menggunakan 3 perspektif. Yaitu:

- a. *Boundary* atau batas antara sistem dengan aktor.
- b. Informasi yang digunakan oleh sistem.
- c. *Control* logik dari sistem.

Jenis kelas analisis diidentifikasi dengan *stereotype* sebagai berikut:

- a. *Boundary*
- b. *Entity*
- c. *Control*

Selain itu juga dapat direpresentasikan dengan menggunakan nama *stereotype* di dalam *Guillements* (<<>>): <<boundary>>, <<control>>, <<entity>>.

3. Untuk setiap kelas analisis yang dihasilkan, jelaskan responsibilities, atribut dan asosiasi.
4. Mengintegrasikan kelas analisis yang digunakan untuk memastikan seluruh kelas analisis telah diintegrasikan.
5. Checkpoints

#### 2.10.4 Identifikasi elemen desain

Elemen desain dapat berupa identifikasi kelas dan *package*. Identifikasi kelas ditunjukkan berupa pemodelan diagram kelas. *Package* merupakan mekanisme untuk mengorganisir unsur ke dalam kelompok.

#### 2.10.5 Perancangan database

Perancangan database merupakan tahapan yang digunakan untuk memastikan bahwa data yang persisten disimpan secara konsisten, efisien dan untuk menentukan perilaku yang harus diimplementasikan pada sistem.

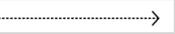
### 2.11 Unified Modeling Language (UML)

Menurut Rumbaugh, Jacobson & Booch (2005) Bahasa pemodelan UML merupakan bahasa yang digunakan untuk memvisualisasi, menentukan, membangun, mendokumentasikan artefak dari sistem perangkat lunak. UML mendefinisikan sejumlah diagram untuk memvisualisasikan sistem yang akan dibangun. Diagram yang baik membuat sistem yang akan dikembangkan lebih mudah dimengerti. UML mendefinisikan 9 macam diagram.

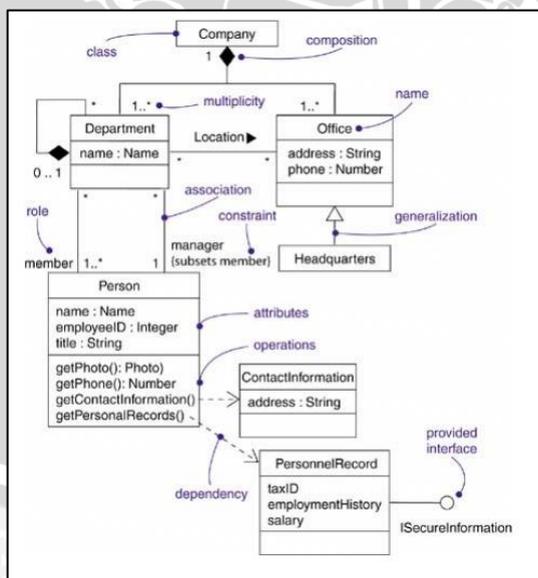
1. *Structure* Diagram digunakan untuk memvisualisasi, menentukan, membangun dan mendokumentasikan aspek statis sistem. Seperti class, interfaces, collaborations, components dan nodes. *Structure* diagram terdiri dari *class* diagram, *object* diagram, *component* diagram dan *deployment* diagram.
  - a. *Class* diagram digunakan untuk menggambarkan tampilan desain statis dari suatu sistem atau mendefinisikan kelas-kelas yang akan dibuat dalam

pembangunan sistem. Tabel 2.4 merupakan penjelasan notasi *class diagram* dan gambar 2.7 merupakan contoh kelas diagram.

**Tabel 2.4 Notasi kelas diagram**

No	Simbol	Deskripsi
1		Kelas pada struktur sistem
2		Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3		Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5		Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6		Relasi antar kelas dengan makna kebergantungan antar kelas.
7		Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> ).

Sumber: Rumbaugh, Jacobson & Booch (2005)



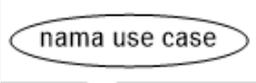
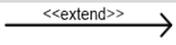
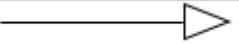
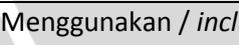
**Gambar 2.7 Contoh kelas diagram**

Sumber: Rumbaugh, Jacobson & Booch (2005)

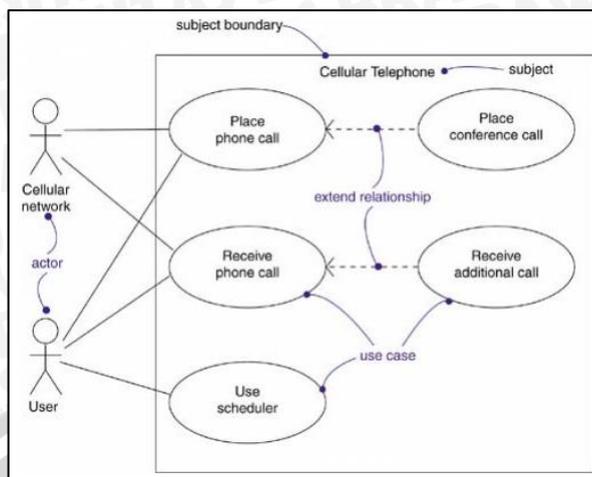
- b. *Object diagram* digunakan untuk memodelkan tampilan desain dan melihat proses dari suatu sistem.

- c. *Component* diagram menunjukkan satu set komponen dan hubungannya. Diagram komponen digunakan untuk menggambarkan implementasi sistem.
  - d. *Deployment* diagram menunjukkan satu set node dan hubungannya. Digunakan untuk menggambarkan arsitektur.
2. *Behavioral* Diagram digunakan untuk memvisualisasikan, menentukan, membangun dan mendokumentasikan aspek dinamis sistem. Seperti aliran sebuah pesan dari waktu ke waktu. *Behavioral* diagram terdiri dari *use case* diagram, *sequence* diagram, *activity* diagram, *collaboration* diagram dan *statechart* diagram.
- a. *Use case* diagram menunjukkan suatu *use case* dan aktor dan hubungannya. Digunakan untuk menggambarkan penggunaan dari suatu sistem. Tabel 2.5 merupakan penjelasan notasi *use case* diagram dan gambar 2.8 merupakan contoh *use case* diagram.

**Tabel 2.5 Notasi use case diagram**

No	Simbol	Deskripsi
1		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case.
2		Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
3		Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5		Hubungan generalisasi dan spesialisasi (umum khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

Sumber: Rumbaugh, Jacobson & Booch (2005)

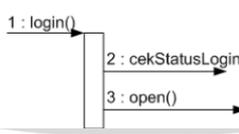
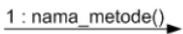


**Gambar 2.8** Contoh use case diagram

Sumber: Rumbaugh, Jacobson & Booch (2005)

- b. *Sequence* diagram atau diagram interaksi menunjukkan suatu sekumpulan objek dan pesan yang dikirim dan diterima oleh objek tersebut. digunakan untuk mengilustrasikan suatu sistem. Tabel 2.6 merupakan penjelasan notasi *sequence* diagram dan gambar 2.9 merupakan contoh *sequence* diagram.

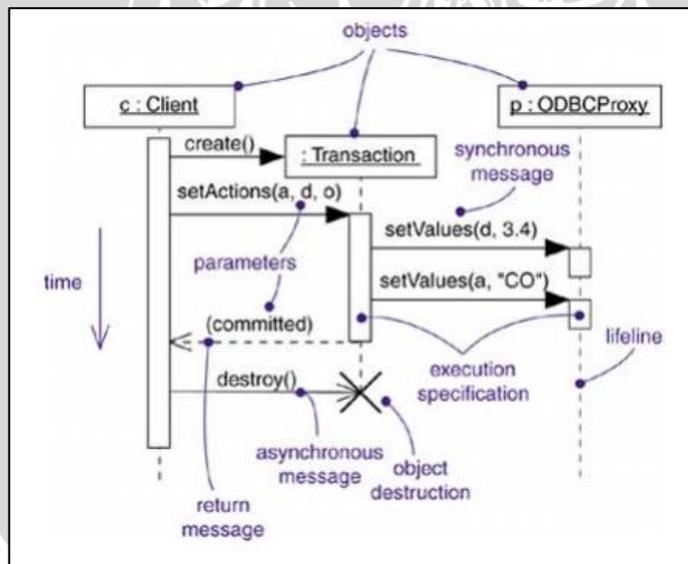
**Tabel 2.6** Notasi *sequence* diagram

No	Simbol	Deskripsi
1	Aktor  Atau 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
2	<i>Lifeline</i> 	Menyatakan kehidupan dalam suatu objek.
3	Objek 	Menyatakan objek yang berinteraksi pesan.
4	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya  Maka cekStatusLogin() dan open() dilakukan di dalam metode login(). Aktor tidak memiliki waktu aktif.
5	Pesan tipe create 	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.
6	Pesan tipe call 	Menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri

Tabel 2.6 Notasi sequence diagram (lanjutan)

No	Simbol	Deskripsi
		Arah panah mengarah pada objek yang memiliki operasi atau metode, karena ini memanggil operasi atau metode maka operasi atau metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7	Pesan tipe send 	Menyatakan bahwa suatu objek mengirimkan data atau masukan atau informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8	Pesan tipe return 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9	Pesan tipe destroy 	Menyatakan suatu objek megakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

Sumber: Rumbaugh, Jacobson & Booch (2005)



Gambar 2.9 Contoh sequence diagram

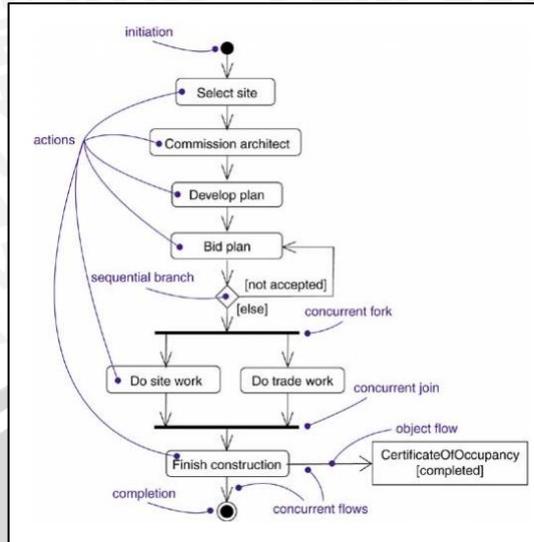
Sumber: Rumbaugh, Jacobson & Booch (2005)

- c. Activity diagram menunjukkan aliran atau aktivitas dari suatu sistem. Tabel 2.7 merupakan penjelasan notasi activity diagram dan gambar 2.10 merupakan contoh activity diagram.

**Tabel 2.7 Notasi activity diagram**

No	Simbol	Deskripsi
1	Status awal 	Start atau <i>initial state</i> adalah keadaan awal pada saat sistem mulai hidup.
2	Aktivitas 	Aktivitas yang dilakukan oleh sistem.
3	Percabangan atau <i>decision</i> 	Sebuah cabang mungkin memiliki satu arus masuk dan dua atau lebih arus keluar dan digambarkan dalam simbol <i>diamond</i> . Pada setiap aliran keluar, ditempatkan ekspresi <i>Boolean</i> , yang dievaluasi saat memasuki cabang. Pada kondisi tersebut cabang berperan sebagai <i>decision</i> . Ketika dua aliran kontrol kembali bergabung bersama-sama, simbol <i>diamond</i> dapat digunakan dengan dua arus masuk dan satu arus keluar. Pada kondisi ini cabang berperan sebagai <i>merge</i> .
4	Pengabungan atau <i>join</i> 	<i>Fork</i> merupakan pemisahan alur kontrol tunggal menjadi dua atau lebih alur kontrol bersamaan. Pada <i>fork</i> , mungkin memiliki satu transisi masuk dan dua atau lebih transisi keluar, masing-masing merepresentasikan aliran kontrol yang independen. <i>Join</i> merupakan sinkronisasi dua atau lebih alur kontrol yang bersamaan. Pada <i>join</i> mungkin memiliki dua atau lebih transisi masuk dan satu transisi keluar.
5	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Rumbaugh, Jacobson & Booch (2005)



**Gambar 2.10** Contoh activity diagram

Sumber: Rumbaugh, Jacobson & Booch (2005)

- d. *Collaboration* diagram menunjukkan ubungan antara objek-objek, pesan yang dikirim dan diterima oleh objek tersebut.
- e. *Statechart* diagram menggambarkan tampilan dinamis dari sebuah sistem.

## 2.12 Evaluasi

Menurut Dix, et al. (2004), Evaluasi tidak dianggap sebagai fase tunggal dalam proses desain. Pada dasarnya evaluasi terjadi sepanjang siklus hidup desain, sehingga desain dapat dimodifikasi sesuai dengan hasil evaluasi. Terdapat hubungan erat antara evaluasi dan prinsip-prinsip *prototyping* yang digunakan untuk memastikan bahwa desain dinilai secara terus menerus. Tujuan dari evaluasi yaitu untuk menilai sejauh mana aksesibilitas fungsi sistem, untuk menilai pengalaman pengguna dalam berinteraksi dengan sistem dan untuk mengidentifikasi masalah sistem. Fungsionalitas sistem merupakan hal yang terpenting sehingga hal tersebut harus sesuai dengan persyaratan pengguna. Dengan kata lain, desain sistem harus memungkinkan pengguna untuk melakukan tugas-tugas yang mereka harapkan dengan mudah. Hal ini tidak hanya membuat fungsi yang tepat namun juga harus membuatnya jelas bisa dicapai dengan pengguna. Hal ini melibatkan pencocokan antara alur yang disediakan sistem dengan harapan pengguna dari tugas yang dilakukan.

### 2.12.1 Tinjauan (Review)

Menurut Agarwal, Tayal dan Gupta (2010), tinjauan dapat didefinisikan sebagai sebuah pertemuan yang mempresentasikan kepada anggota tim proyek, manajer, pengguna, pelanggan. Sebuah tinjauan didefinisikan sebagai proses software engineering. Tujuan dari tinjauan adalah untuk menemukan kesalahan dalam analisis, desain, pengkodean, pengujian dan implementasi dari siklus pengembangan perangkat lunak. Tujuan dari tinjauan adalah untuk melihat apakah prosedur yang diterapkan telah sesuai dengan persyaratan yang telah di

definisikan. Tinjauan dibedakan menjadi *informal review* dan *formal review*. *Formal review* digunakan untuk mengecek *Verifiability*, *comprehensibility*, *traceability* dan *adaptability*. Tujuan review adalah:

1. untuk memastikan bahwa unsur-unsur perangkat lunak telah sesuai dengan spesifikasi.
2. Untuk memastikan bahwa pengembang perangkat lunak dilakukan sesuai dengan rencana, standar dan pedoman yang berlaku untuk pengembangan perangkat lunak tersebut.
3. Untuk memastikan bahwa perubahan pada elemen perangkat lunak diimplementasikan dengan benar.

Review memiliki 2 tipe yaitu *informal technical reviews* yaitu pertemuan dan pemeriksaan secara informal dan *formal technical review* yaitu kegiatan yang dilakukan secara formal mengenai kualitas perangkat lunak melalui berbagai pendekatan, seperti *structured*, *walkthrough* dan *inspections*. Penelitian ini melakukan tinjauan untuk melakukan pengecekan *traceability* dan tinjauan dengan menggunakan *walkthrough*.

#### **2.12.1.1 Traceability**

Menurut Agarwal, Tayal dan Gupta (2010) *Traceability* merupakan sifat untuk mendapatkan verifikasi dari sebuah desain. *Traceability* digunakan untuk memastikan bahwa seluruh elemen desain dapat ditelusuri ke persyaratan. Pengecekan *traceability* salah satunya menggunakan peta *traceability* dan *checklist* yang mengacu pada *checklist for requirement specification reviews* untuk kategori *traceability*.

#### **2.12.1.2 Walkthrough**

Menurut Naik dan Tripathy (2008) *Walkthrough* merupakan salah satu *formal technical review* dimana penulis memimpin tim dengan cara melakukan simulasi dari produk menggunakan skenario yang telah ditetapkan. Menurut Dix, et al. (2004), *walkthrough* merupakan urutan suatu tindakan yang mengacu pada langkah-langkah antarmuka yang dilakukan oleh pengguna untuk mencapai suatu tugas. Untuk dapat melakukan *walkthrough* diperlukan 4 hal, diantaranya:

1. Sebuah prototipe dari sistem.
2. Sebuah deskripsi tugas pengguna untuk melakukan suatu tugas pada sistem.
3. Lengkap, daftar tindakan yang diperlukan untuk menyelesaikan suatu tugas dengan alur sistem yang diusulkan.
4. Sebuah indikasi dari siapa pengguna dan jenis pengalaman dan pengetahuan.

#### **2.12.2 Evaluasi antarmuka pengguna**

Perancangan antarmuka pengguna dibuat untuk menciptakan komunikasi antara manusia dan komputer. Terdapat 3 aturan dalam prinsip perancangan antarmuka, diantaranya: pengguna mengetahui dimana tempat ia berada, mengurangi memori beban pengguna, membuat antarmuka secara konsisten.

Untuk membuat antarmuka yang efektif semua perancangan antarmuka dimulai dengan memahami maksud pengguna. Fokus pada evaluasi antarmuka adalah untuk mengetahui bagaimana pengguna berinteraksi dengan sistem. Dalam membuat antarmuka untuk pengguna, antarmuka tersebut harus di evaluasi untuk menentukan apakah telah memenuhi harapan pengguna. Terdapat beberapa cara dalam melakukan evaluasi antarmuka yaitu dengan menggunakan kuisisioner (Pressman, 2010).

### 2.12.2.1 WEBUSE (Website Usability Evaluation Tool)

Menurut Chiew & Salim (2003) WEBUSE cocok digunakan untuk mengevaluasi semua jenis situs web. Dalam melakukan evaluasi digunakan suatu alat atau *tool* yaitu berupa kuisisioner. Kuisisioner WEBUSE merupakan alat yang digunakan untuk melakukan evaluasi. Kuisisioner WEBUSE memiliki 24 pertanyaan yang terdiri dari 4 kategori, diantaranya *content, organisation and readability, navigation and links, user interface design dan performance and effectiveness*. Pada tabel 2.8 dijelaskan mengenai kuisisioner WEBUSE untuk setiap kategori. Pada penelitian yang dilakukan, penulis melakukan tinjauan antarmuka prototipe dengan menggunakan kuisisioner WEBUSE. Pemilihan kategori difokuskan pada kategori navigasi dan antarmuka, hal ini di sesuaikan dengan penelitian yang dilakukan yaitu analisis dan perancangan.

**Tabel 2.8 Kuisisioner WEBUSE per kategori**

Kategori	No.	Atribut
<i>Content, Organisation and Readability</i>	1.	<i>This website contains most of my interest material and topics and they are up-to-date.</i>
	2.	<i>I can easily find what I want at this website</i>
	3.	<i>The content of this website is well organised.</i>
	4.	<i>Reading content at this website is easy.</i>
	5.	<i>I am comfortable and familiar with the language used.</i>
	6.	<i>I need not scroll left and right when reading at this website.</i>
<i>Navigations and Links</i>	7.	<i>I can easily know where I am at this website.</i>
	8.	<i>This website provide useful cues and links for me to get the desired information.</i>
	9.	<i>It is easy to move around at this website by using the links or back button of the browser.</i>
	10.	<i>The links at this website are well maintained and updated.</i>
	11.	<i>The website does not open too many new browser windows when I am moving around.</i>
	12.	<i>Placement of links or menu is standard throughout the website and I can easily recognise them.</i>
	13.	<i>This website's interface design is attractive.</i>

User Interface Design	14.	<i>I am comfortable with the colours used at this website.</i>
	15.	<i>This website contains no feature that irritates me such as scrolling or blinking text and looping animations.</i>
	16.	<i>This website has a consistent feel and look.</i>
	17.	<i>This website does not contain too many Web advertisements.</i>
Performance and Effectiveness	18.	<i>The design of the website makes sense and it is easy to learn how to use it.</i>
	19.	<i>I need not wait too long to download a file or open a page.</i>
	20.	<i>I can easily distinguish between visited and not-visited links.</i>
	21.	<i>I can access this website most of the time.</i>
	22.	<i>This website responds to my actions as expected.</i>
	23.	<i>It is efficient to use this website.</i>
	24.	<i>This website always provides clear and useful messages when I don't know how to proceed.</i>

Sumber: Chiew dan Salim (2003)

Tabel 2.9 merupakan beberapa daftar pertanyaan yang telah diterjemahkan yang digunakan dalam penelitian ini untuk kategori *navigation and links* dan *user interface design*.

**Tabel 2.9** Kuisioner WEBUSE kategori navigasi dan antarmuka pengguna

Pertanyaan	S	N	TS	Keterangan
<b>Navigasi</b>				
Saya dapat dengan mudah mengetahui posisi saya pada sistem				
Sistem menyediakan navigasi menu dan <i>link</i> yang berguna untuk mendapatkan informasi				
Sistem mudah digunakan dengan menggunakan tombol kembali atau link				
Sistem tidak membuka banyak jendela browser ketika digunakan				
Peletakan menu atau link dapat dengan mudah dikenali				
Navigasi menu pada sistem sudah menunjukkan fungsi dengan jelas				
<b>Antarmuka pengguna</b>				
Perancangan antar muka sistem mudah untuk dipelajari				
Sistem tidak mengandung fitur yang membutuhkan scroll atau animasi yang berulang				
Sistem memiliki antar muka yang konsisten				



### 2.12.3 Verifikasi dan Validasi

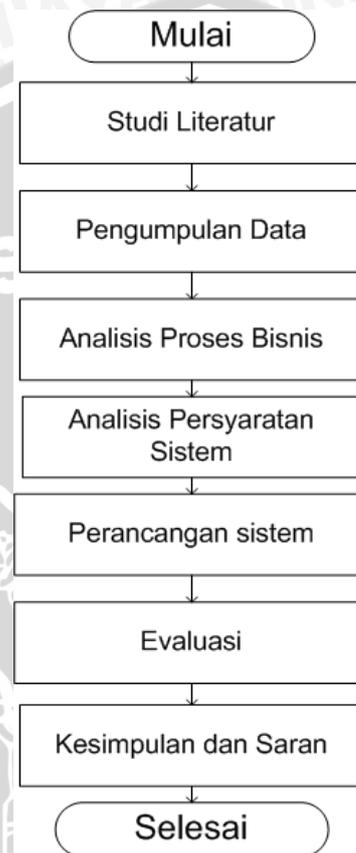
Menurut Sommerville (2001) verifikasi dan validasi adalah nama yang diberikan pada proses pemeriksaan dan analisis yang menjamin bahwa perangkat lunak sesuai dengan spesifikasinya dan memenuhi kebutuhan pelanggan. Verifikasi dan validasi merupakan proses siklus hidup penuh. Proses ini mulai dengan tinjauan persyaratan dan berlanjut melalui tinjauan rancangan dan pemeriksaan kode sampai pengujian produk. Harus ada kegiatan verifikasi dan validasi pada setiap tahap proses perangkat lunak.

Verifikasi melibatkan pemeriksaan yaitu bahwa perangkat lunak sesuai dengan spesifikasinya, apakah sistem memenuhi persyaratan fungsional dan non-fungsional yang ada dispesifikasi sistem. Ini dibedakan dengan validasi yang merupakan proses yang lebih umum, Validasi menjamin bahwa perangkat lunak telah memenuhi harapan klien.



## BAB 3 METODOLOGI

Pada bab ini akan dijelaskan mengenai tahapan atau langkah-langkah yang dilakukan oleh peneliti mulai dari proses analisis sampai hasil akhir dalam melakukan penelitian tugas akhir. Metodologi penelitian yang digunakan dapat dilihat pada gambar 3.1:



**Gambar 3.1 Metodologi Penelitian**

### 3.1 Studi literatur

Studi literatur dilakukan dengan mencari referensi untuk mendapatkan landasan kerangka berfikir yang berkaitan dengan penelitian yang dilakukan oleh penulis, baik dari segi konsep, dasar teori, metode yang akan digunakan yang dapat membantu dalam proses analisis, perancangan dan pengujian. Sumber pembelajaran yang digunakan pada penelitian ini didapatkan melalui buku tulis maupun media internet dalam bentuk jurnal ilmiah, maupun tugas akhir.

### 3.2 Pengumpulan data

Pengumpulan data dilakukan untuk mendapatkan data terkait proses bisnis yang terjadi saat ini mengenai proses pencarian darah yang dilakukan masyarakat saat membutuhkan darah, hal tersebut juga berkaitan dengan proses bisnis yang terjadi di perusahaan terkait proses pengolahan data darah beserta pengolahan

stok darah hingga menjadi sebuah informasi ketersediaan stok darah untuk masyarakat. Pengumpulan data pada penelitian ini dilakukan dengan teknik penelitian lapangan atau observasi. Penelitian lapangan atau observasi bertujuan untuk mengetahui secara langsung lingkungan penelitian dan objek yang diteliti. Pengumpulan data yang dilakukan oleh penulis mengenai bagaimana proses bisnis pencarian darah yang dilakukan masyarakat, beserta proses bisnis yang terjadi di PMI mengenai pengolahan data stok darah. Gambaran mengenai proses bisnis perusahaan akan digambarkan menggunakan notasi BPMN (*Business Process Model and Notation*).

### 3.3 Analisis proses bisnis

Pada tahap ini dilakukan analisis terhadap proses bisnis yang terjadi diperusahaan, setelah dilakukan pengumpulan data dengan menggunakan teknik wawancara, selanjutnya dilakukan analisis terhadap data tersebut. Dalam melakukan analisis proses bisnis penulis menggunakan notasi BPMN (*Business Process Model and Notation*). Dimana di dalam BPMN terdapat notasi yang menjelaskan bagaimana aktivitas bisnis berjalan sehingga memudahkan pembaca dalam mengenali aktivitas yang berjalan di PMI saat ini dan bagaimana aktivitas dari proses bisnis yang di usulkan. Proses bisnis yang diusulkan dibuat sesuai dengan masalah dan solusi yang didapatkan.

### 3.4 Analisis persyaratan sistem

Pada tahap ini dilakukan analisis persyaratan sistem yang dibutuhkan oleh pengguna berdasarkan analisis proses bisnis yang telah dilakukan sebelumnya. Analisis persyaratan sistem dilakukan dengan mengacu pada langkah-langkah yang terdapat pada buku Kurt Bittner dan Ian Spence 2002, yaitu:

1. Mengidentifikasi tipe pemangku kepentingan.
2. Melakukan analisis permasalahan.
3. Melakukan analisis kebutuhan pemangku kepentingan dan pengguna.
4. Melakukan analisis fitur-fitur yang harus disediakan sistem sesuai dengan kebutuhan pemangku kepentingan dan pengguna.
5. Menjelaskan posisi produk dari sistem yang dirancang.
6. Melakukan analisis *use case*.

Dalam melakukan analisis *use case*, mengacu pada coak atau model RUP *style*.

### 3.5 Perancangan sistem

Setelah menentukan jenis persyaratan yang diperlukan oleh sistem, tahap selanjutnya adalah melakukan perancangan sistem. Perancangan sistem yang dilakukan pada penelitian ini antara lain akan digambarkan menggunakan diagram UML. Pada tahap ini seluruh persyaratan yang telah didefinisikan akan dikonversi dalam bentuk diagram yang nantinya akan digunakan oleh programmer untuk

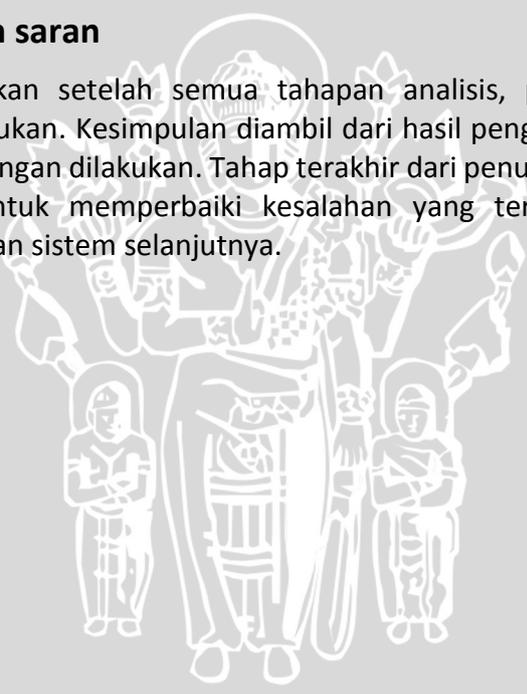
melakukan implementasi. Perancangan sistem yang dilakukan diantaranya berupa perancangan kelas, perancangan interaksi dan perancangan antarmuka pengguna.

### 3.6 Evaluasi

Evaluasi yang dilakukan menggunakan tinjauan (*review*) yang digunakan untuk mengecek *traceability* dan menggunakan *walkthrough* untuk kesesuaian antara alur yang disediakan sistem dengan alur yang diharapkan pengguna. Evaluasi yang dilakukan selanjutnya berupa evaluasi prototipe yang dilakukan dengan mengerjakan beberapa tugas atau mencoba sistem. Dalam penelitian ini evaluasi antarmuka dilakukan dengan menggunakan suatu alat atau *tools* yaitu berupa kuisisioner yang mengacu pada kuisisioner WEBUSE untuk mendapatkan saran dan masukan dari pengguna sistem. Jumlah responden yang digunakan untuk melakukan pengecekan *traceability* sejumlah 1 responden yaitu pengembang perangkat lunak, untuk *walkthrough* dan WEBUSE sebanyak 5 responden dengan 2 pegawai PMI dan 3 masyarakat umum.

### 3.7 Kesimpulan dan saran

Kesimpulan dilakukan setelah semua tahapan analisis, perancangan dan pengujian selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis setelah proses perancangan dilakukan. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan yang terjadi serta untuk memberikan perancangan sistem selanjutnya.



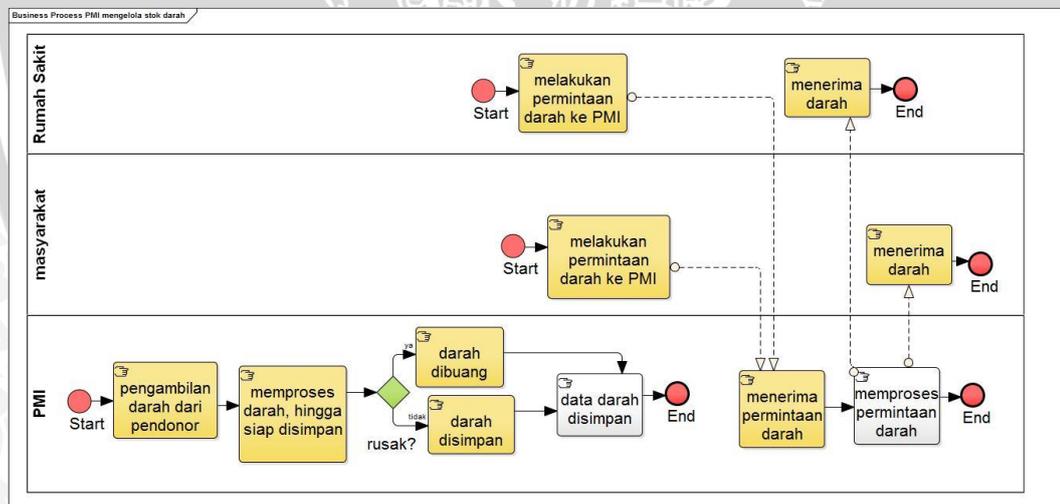
## BAB 4 PEMODELAN PROSES BISNIS DAN ANALISIS PERSYARATAN SISTEM

Bab ini menjelaskan mengenai pemodelan proses bisnis dan analisis persyaratan sistem. Pemodelan proses bisnis merupakan gambaran mengenai proses bisnis yang terjadi saat ini, analisis persyaratan sistem meliputi analisis permasalahan yang ada, gambaran umum sistem, identifikasi aktor, analisis persyaratan fungsional dan pemodelan use case.

### 4.1 Analisis proses bisnis

#### 4.1.1 Proses bisnis mengelola stok darah kondisi saat ini (as is)

Mengelola stok darah yaitu proses penyimpanan data darah mulai dari darah masuk hingga darah keluar. Proses mengelola stok darah yang terjadi di PMI saat ini dapat dilihat pada gambar 4.1. Setelah PMI mendapatkan darah dari pendonor, maka darah tersebut akan diproses hingga darah tersebut siap untuk disimpan. Setelah proses pengolahan darah selesai, baik darah tersebut dalam kondisi baik maupun rusak data darah tersebut akan disimpan, untuk darah yang rusak darah tersebut akan dibuang, sedangkan untuk darah yang baik maka darah tersebut akan disimpan ke ruang penyimpanan darah. Untuk pencatatan darah keluar terjadi saat adanya permintaan dari Rumah Sakit, masyarakat dan darah rusak, dengan demikian PMI akan melakukan pencatatan darah keluar.



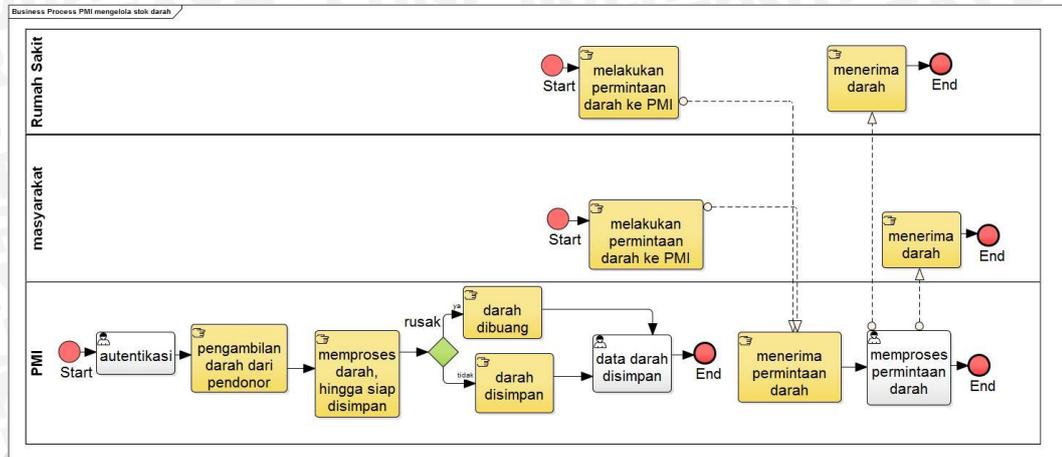
**Gambar 4.1** Proses bisnis mengelola stok darah kondisi saat ini (as is)

#### 4.1.2 Proses bisnis mengelola stok darah kondisi usulan (to be)

Pada gambar 4.2 dijelaskan bagaimana proses bisnis saat mengelola stok darah kondisi usulan. Mengelola stok darah yaitu proses penyimpanan data darah mulai dari darah masuk hingga darah keluar. Proses mengelola stok darah setelah adanya perancangan sistem hampir sama pada proses sebelum adanya perancangan. Perbedaannya terletak pada penyimpanan data darah. Setelah adanya sistem, penyimpanan data darah akan disimpan menggunakan sistem



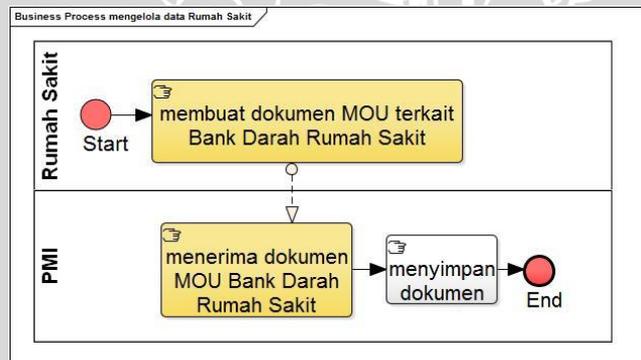
informasi Bank Darah menggunakan database, untuk pencatatan darah keluar secara otomatis akan mengurangi jumlah stok darah masuk. Setelah itu data tersebut akan ditampilkan ke sistem informasi Bank Darah untuk diakses oleh masyarakat.



**Gambar 4.2** Proses bisnis mengelola stok darah kondisi usulan (to be)

#### 4.1.3 Proses bisnis mengelola data rumah sakit kondisi saat ini (as is)

Pada gambar 4.3 dijelaskan bagaimana proses PMI dalam mengelola data Rumah Sakit. Pengolahan data Rumah Sakit dilakukan dengan mencatat daftar Rumah Sakit mana saja yang memiliki Bank Darah, hubungan Bank Darah Rumah Sakit dengan PMI harus dengan menggunakan dokumen MOU di antara kedua pihak. Setelah membuat dokumen MOU pihak PMI dan Rumah Sakit menyimpan dokumen tersebut.

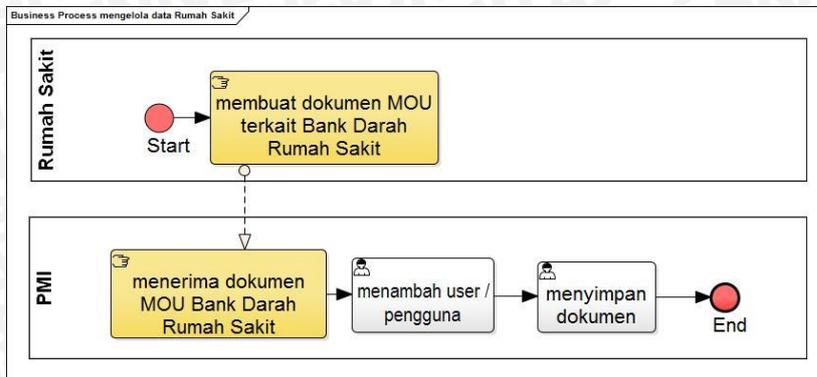


**Gambar 4.3** Proses bisnis mengelola data rumah sakit kondisi saat ini (as is)

#### 4.1.4 Proses bisnis mengelola data rumah sakit kondisi usulan (to be)

Pada gambar 4.4 dijelaskan bagaimana Proses Bisnis PMI dalam mengelola data Rumah Sakit setelah adanya perancangan sistem. Proses bisnis setelah adanya prancangan sistem hampir sama sebelum adanya perancangan. Perbedaannya terletak pada penyimpanan data Rumah Sakit. Data Rumah Sakit setelah adanya perancangan sistem akan disimpan kedalam sistem informasi Bank Darah, tujuannya adalah untuk mendapatkan informasi mengenai jumlah

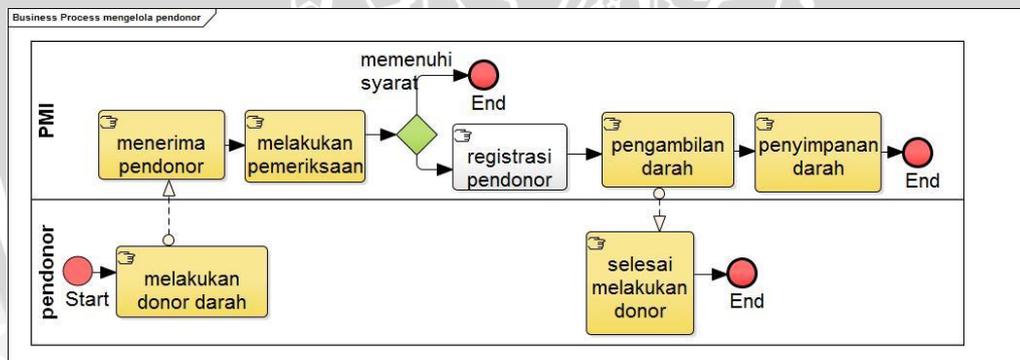
darah yang ada di Rumah Sakit tersebut, agar PMI dapat mengontrol ketersediaan darah yang ada.



Gambar 4.4 Proses bisnis mengelola data rumah sakit kondisi usulan (to be)

#### 4.1.5 Proses bisnis mengelola data pendonor kondisi saat ini (as is)

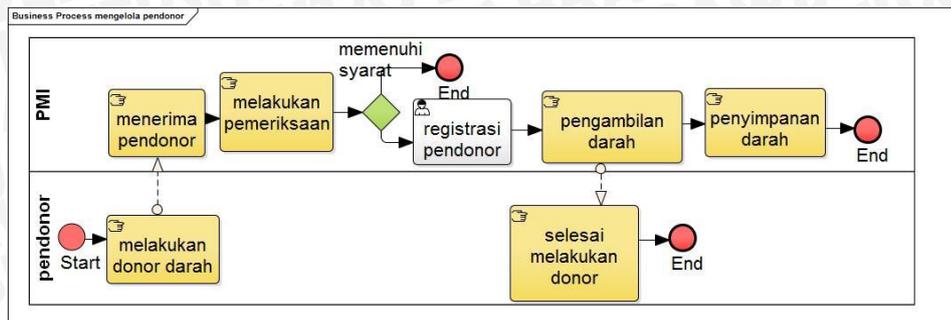
Pada gambar 4.5 dijelaskan bagaimana proses mengelola data pendonor saat ini. Seluruh data pendonor baik yang melakukan donor melalui mobil unit maupun PMI akan disimpan ke dalam database. Penyimpanan data pendonor dilakukan saat pendonor datang untuk melakukan donor, baik untuk pendonor yang baru maupun pendonor yang sudah pernah melakukan donor. Pencatatan pendonor digunakan untuk memudahkan PMI mendapatkan darah dari pendonor sesuai dengan waktu donor yang telah ditentukan.



Gambar 4.5 Proses bisnis mengelola data pendonor kondisi saat ini (as is)

#### 4.1.6 Proses bisnis mengelola data pendonor kondisi usulan (to be)

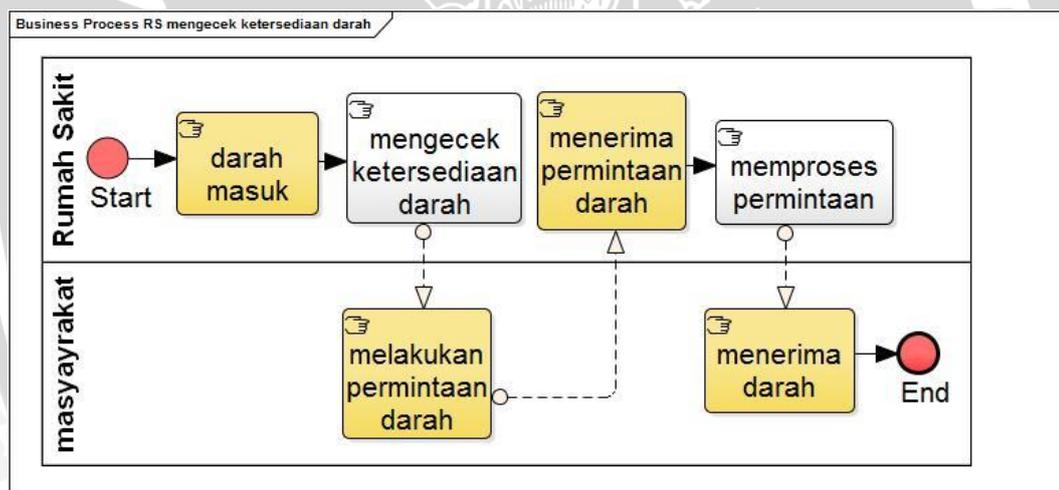
Pada gambar 4.6 dijelaskan bagaimana proses mengelola data pendonor setelah adanya perancangan sistem. Proses yang terjadi setelah adanya perancangan sistem sama dengan kondisi yang terjadi pada pencatatan pendonor saat ini.



Gambar 4.6 Proses bisnis mengelola data pendonor kondisi usulan (to be)

#### 4.1.7 Mengelola darah keluar dan mengecek ketersediaan darah rumah sakit kondisi saat ini (as is)

Pada gambar 4.7 dijelaskan bagaimana proses mengelola darah keluar yang terjadi di Rumah Sakit saat ini. Mengelola darah keluar yaitu berupa pencatatan darah keluar. Pada saat terjadi permintaan darah oleh pasien, pihak Rumah Sakit akan mengecek darah yang tersedia di tempat penyimpanan darah, hal ini dilakukan secara manual. Kemudian setelah darah tersebut tersedia, maka Rumah Sakit akan memproses permintaan darah.

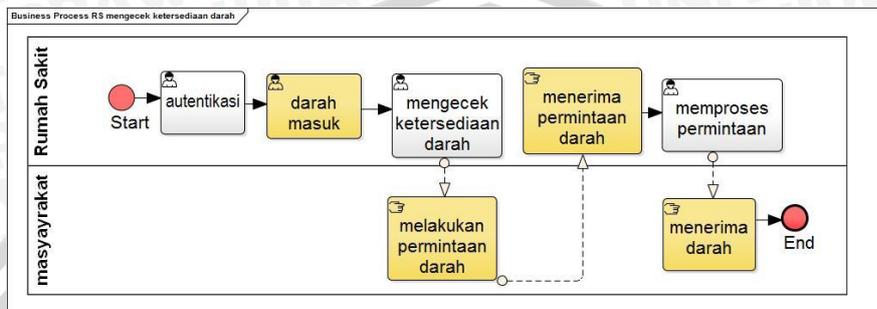


Gambar 4.7 Proses bisnis mengelola darah keluar dan mengecek ketersediaan darah rumah sakit kondisi usulan kondisi saat ini (as is)

#### 4.1.8 Mengelola darah keluar dan mengecek ketersediaan darah rumah sakit kondisi usulan (to be)

Pada gambar 4.8 dijelaskan bagaimana proses bisnis mengelola darah keluar yang dilakukan oleh Rumah Sakit. Darah keluar yang terjadi di Rumah Sakit terjadi karena adanya permintaan dari masyarakat yang membutuhkan atau pasien dan darah keluar yang terjadi karena rusak. Darah masuk yang terjadi di Rumah Sakit terjadi saat pengiriman darah yang dilakukan oleh pihak PMI, saat PMI melakukan pencatatan pengiriman darah yang dikirim ke Rumah Sakit, seara otomatis data tersebut masuk ke data stok darah milik Rumah Sakit, pihak Rumah Sakit dapat mengecek ketersediaan darah yang ada di Rumah Sakit tersebut tanpa harus mencari ke tempat penyimpanan. Tujuannya yaitu untuk memberikan informasi

yang dapat di akses oleh masyarakat dan dapat digunakan oleh PMI untuk mengontrol ketersediaan darah yang ada di setiap Rumah Sakit yang memiliki Bank Darah. Untuk pencatatan darah keluar terjadi jika ada permintaan pasien atau darah rusak, maka pihak Rumah Sakit akan mencatat data darah keluar. saat terjadi permintaan darah, maka Rumah Sakit akan memproses permintaan tersebut dan mencatatnya kembali kedalam pencatatan darah keluar, dengan demikian data stok darah yang ada di Rumah Sakit akan berubah.



**Gambar 4.8** Proses bisnis mengelola darah keluar dan mengecek ketersediaan darah rumah sakit kondisi usulan kondisi usulan (to be)

#### 4.1.9 Proses bisnis pencarian darah kondisi saat ini (as is)

Pada gambar 4.9 dijelaskan bagaimana proses pencarian darah yang terjadi saat ini. Saat masyarakat membutuhkan darah ketika berada di Rumah Sakit dan ketersediaan Rumah Sakit tidak ada, maka masyarakat harus pergi mencari darah yang dibutuhkan baik ke PMI maupun Rumah Sakit, setelah mendapatkan maka proses permintaan darah akan dilakukan sesuai dengan prosedur yang telah ditentukan.

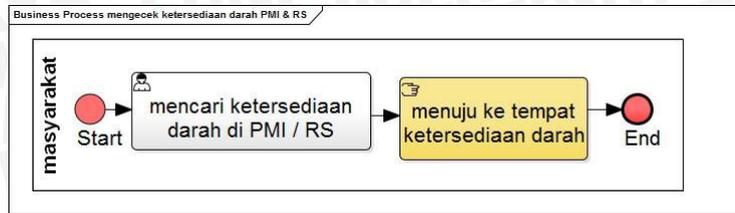


**Gambar 4.9** Proses bisnis pencarian darah kondisi saat ini (as is)

#### 4.1.10 Proses bisnis pencarian darah kondisi usulan (to be)

Pada gambar 4.10 dijelaskan bagaimana proses pencarian setelah ada perancangan saat pasien membutuhkan darah. Saat pasien berada di Rumah Sakit dan sedang membutuhkan darah, namun di Rumah Sakit tersebut tidak memiliki darah yang sesuai maka pasien harus mencari darah ke PMI atau Rumah Sakit lain, pasien dapat membuka sistem informasi Bank Darah untuk mengetahui stok darah yang ada di PMI maupun Rumah Sakit tanpa pergi terlebih dahulu ke tempat tersebut. Setelah pasien mengetahui tempat mana yang memiliki stok darah yang dibutuhkan beserta jumlah ketersediaan, pasien dapat langsung menuju ke tempat tersebut dengan surat keterangan dari Rumah Sakit yang terkait. Dengan

melalui sistem informasi tersebut pasien dapat mengetahui ketersediaan darah yang ada, pasien juga dapat mengetahui jika tidak ada darah yang dibutuhkan sehingga dapat mempersiapkan pendonor atau keluarga yang memiliki golongan darah yang sesuai.



Gambar 4.10 Proses bisnis pencarian darah kondisi usulan (to be)

## 4.2 Pemetaan proses bisnis dengan persyaratan fungsional

Pemetaan antara proses bisnis dengan persyaratan fungsional digambarkan pada tabel 4.1.

Tabel 4.1 Pemetaan proses bisnis dengan persyaratan fungsional

No	Proses Bisnis	Persyaratan Fungsional	Use Case
1	Data darah disimpan	SRS-F-01	Mengelola data stok darah
2	Registrasi pendonor	SRS-F-02	Mengelola data pendonor
3	Menyimpan dokumen	SRS-F-03	Mengelola data Rumah Sakit
4	Menambah user / pengguna	SRS-F-04	Mengelola user
5	Memproses permintaan	SRS-F-05	Mengelola darah keluar
6	Mengecek ketersediaan darah	SRS-F-06	Mengecek data darah Rumah Sakit
7	Mencari ketersediaan darah di PMI / Rumah Sakit	SRS-F-07	Mengecek ketersediaan darah PMI dan Rumah Sakit
8	Autentikasi	SRS-F-08	Autentikasi

## 4.3 Gambaran umum sistem

Sistem yang akan dirancang merupakan sistem informasi mengenai informasi persediaan darah di PMI dan Rumah Sakit yang dapat diakses oleh pasien atau masyarakat umum. Sistem yang dirancang juga membantu untuk mengintegrasikan antara pihak PMI dengan Rumah Sakit agar mudah dalam melakukan persediaan darah. Analisis dan perancangan sistem informasi Bank Darah memiliki beberapa fitur diantaranya:

1. Mengelola stok darah, digunakan oleh Admin PMI untuk mengelola stok darah yang ada saat ini. Mengelola stok darah nantinya dapat melakukan menambah data darah, memperbarui data darah, menambah darah keluar dan memperbarui data darah keluar.

2. Mengelola data pendonor, digunakan oleh Admin PMI untuk mengelola data pendonor. Mengelola data pendonor dapat berupa menambah data pendonor, memperbarui data pendonor dan menghapus data pendonor.
3. Mengelola data Rumah Sakit, digunakan oleh Admin PMI untuk mengelola data Rumah Sakit yang memiliki Bank Darah. Nantinya agar pihak PMI dapat memantau jumlah stok darah yang ada di Rumah Sakit agar tidak terjadi keterlambatan pengiriman darah. Mengelola data rumah sakit nantinya dapat berupa menambah data Rumah Sakit, memperbarui data Rumah Sakit.
4. Mengelola user, digunakan oleh Admin PMI untuk mengelola data user (Admin PMI dan Admin Rumah Sakit). Mengelola user nantinya dapat berupa menambah user dan memperbarui data user.
5. Mengelola darah keluar, digunakan oleh Admin Rumah Sakit untuk melakukan pencatatan data darah keluar yang ada di Rumah Sakit yang nantinya akan ditampilkan ke Sistem Informasi Bank Darah.
6. Mengecek data darah Rumah Sakit, digunakan oleh Admin Rumah Sakit untuk mengecek ketersediaan darah di Rumah Sakit tersebut.
7. Mengecek ketersediaan darah di PMI dan Rumah Sakit, digunakan oleh masyarakat untuk melakukan pencarian darah sesuai dengan yang dibutuhkan. Pasien dapat melihat stok darah di PMI maupun Rumah Sakit.

#### 4.4 Identifikasi aktor

Pada tabel 4.2 dijelaskan mengenai aktor-aktor yang berhubungan dengan sistem yang dirancang.

**Tabel 4.2 Identifikasi aktor**

No	Aktor	Deskripsi
1	Masyarakat	Masyarakat umum yang dapat menggunakan aplikasi tanpa harus melakukan login ke sistem terlebih dahulu. Pengguna hanya dapat melihat stok darah yang ada di Rumah Sakit dan PMI.
2	Admin PMI	Admin PMI merupakan orang yang dapat menjalankan sistem secara keseluruhan, Admin PMI perlu melakukan login atau autentikasi terlebih dahulu untuk dapat mengoperasikan sistem. Admin PMI dapat melakukan mengelola data stok darah, data pendonor dan Rumah Sakit yang memiliki Bank Darah berupa menambah dan memperbarui.
3	Admin Rumah Sakit	Admin Rumah Sakit merupakan orang yang dapat menggunakan sistem dengan wewenang untuk mengelola data stok darah yang ada di Rumah Sakit tersebut. Admin Rumah Sakit perlu melakukan login atau autentikasi terlebih dahulu untuk dapat mengoperasikan sistem. Admin Rumah Sakit dapat mengelola data stok darah yang berupa melakukan pencatatan darah keluar.

#### 4.5 Analisis masalah

Analisis masalah merupakan penjelasan dari identifikasi masalah yang terjadi saat ini sehingga dibutuhkan sebuah usulan sistem yang akan digunakan untuk menyelesaikan masalah yang ada. Tabel berikut menjelaskan mengenai masalah yang terjadi di PMI kota Malang saat ini.

1. Pada tabel 4.3 dijelaskan mengenai masalah yang terjadi saat ini yaitu tidak adanya informasi mengenai stok darah yang ada di Rumah Sakit.

**Tabel 4.3 Analisis masalah**

Masalah	Tidak adanya informasi mengenai informasi stok arah yang ada di Rumah Sakit sehingga masyarakat harus datang langsung untuk mengetahui informasi ketersediaan darah.
Efek	Pasien / orang yang mencari darah.
Dampak	Sulitnya pasien untuk mencari informasi mengenai ketersediaan darah.
Solusi	Menyediakan informasi stok darah yang ada di Rumah Sakit maupun PMI

2. Pada tabel 4.4 dijelaskan mengenai masalah yang terjadi saat ini yaitu terkait tidak adanya sistem yang dapat digunakan PMI untuk mengontrol ketersediaan darah di Rumah Sakit yang memiliki Bank Darah.

**Tabel 4.4 Analisis masalah**

Masalah	Pengiriman darah dari PMI ke Rumah Sakit masih menggunakan via telepon dalam menginformasikan kebutuhan darah sehingga ketersediaan kecukupan darah kurang bisa terkontrol.
Efek	PMI, Rumah Sakit, masyarakat.
Dampak	Sulitnya mengontrol ketersediaan stok darah.
Solusi	Menyediakan informasi stok darah Rumah Sakit, agar PMI dapat mengontrol stok darah yang ada di Rumah Sakit, sehingga PMI dapat melakukan persediaan darah sebelum terlambat.

#### 4.6 Analisis pemangku kepentingan

Pada sub bab ini menjelaskan mengenai pemangku kepentingan dan pengguna yang terlibat dalam pengerjaan proyek.

##### 4.6.1 Tipe pemangku kepentingan

Pada tabel 4.5 dijelaskan mengenai jenis pemangku kepentingan yang terlibat dalam pengerjaan proyek.

**Tabel 4.5 Tipe pemangku kepentingan**

No	Nama	Deskripsi	Contoh
1	Pengguna	Orang yang secara langsung menggunakan sistem informasi Bank Darah.	Masyarakat, Admin PMI dan Admin Rumah Sakit.
2	Pengembang	Orang yang membangun proyek atau sistem informasi Bank Darah.	Analisis dan programmer
3	Authorities / Pihak Berwenang	Pihak yang ahli dalam aspek tertentu terkait sistem informasi Bank Darah terkait mengelola data darah, pendonor, user atau pengguna dan Rumah Sakit.	PMI
4	Authorities / Pihak Berwenang	Pihak yang ahli dalam aspek tertentu terkait sistem informasi Bank Darah terkait ketersediaan darah yang ada di Rumah Sakit.	Rumah Sakit

#### 4.7 Kebutuhan pengguna dan pemangku kepentingan

Pada tabel 4.6 menggambarkan hubungan antara kebutuhan pengguna dan kebutuhan pemangku kepentingan dengan masalah yang ada. Sehingga dengan menangkap kebutuhan stakeholder dapat memungkinkan kita untuk memahami bagaimana dan sejauh mana masalah memengaruhi berbagai jenis pemangku kepentingan. Terdapat aturan penggunaan Moskow dalam pengukuran saat memprioritaskan kebutuhan pemangku kepentingan. Pada penelitian yang dilakukan penulis, prioritas untuk mendapatkan nilai Mo, S, Co dan W dilakukan dengan menyimpulkan hasil wawancara dengan pihak PMI dan masyarakat.

**Tabel 4.6 Kebutuhan pemangku kepentingan**

Need	Stakeholder	Prioritas	Situasi saat ini	Solusi yang ditawarkan
Kemudahan untuk mengelola data darah masuk, data darah keluar beserta jumlah stok darah yang tersedia dan menampilkan jumlah ketersediaan darah yang ada agar dapat diakses oleh masyarakat.	Admin PMI dan Admin Rumah Sakit	M	Data darah masuk dan darah keluar disimpan dengan menggunakan excel, sehingga masih belum bisa menampilkan informasi ketersediaan jumlah stok darah secara otomatis.	Sistem mampu menyediakan dan menampilkan informasi ketersediaan jumlah stok darah terbaru dan menampilkan jumlah ketersediaan dan tempat tersedianya darah.
Kemudahan untuk mengelola data Rumah Sakit yang	Admin PMI	M	Mencari data Rumah Sakit yang memiliki Bank Darah	Sistem mampu menyediakan dan menampilkan informasi mengenai

**Tabel 4.6 Kebutuhan pemangku kepentingan (lanjutan)**

Need	Stakeholder	Prioritas	Situasi saat ini	Solusi yang ditawarkan
memiliki bank darah			secara manual dalam buku pencatatan bak darah.	data bank darah Rumah Sakit, sehingga dapat mengetahui jumlah stok darah di setiap Rumah Sakit yang memiliki bank darah.
Kemudahan untuk mendapatkan informasi mengenai jumlah stok darah yang tersedia di PMI maupun Rumah Sakit	Admin PMI, Admin Rumah Sakit dan masyarakat	M	Informasi mengenai jumlah ketersediaan stok darah masih terbatas pada jumlah ketersediaan stok darah yang ada di PMI saja.	Menyediakan informasi mengenai jumlah stok darah yang ada di PMI maupun Rumah Sakit yang memiliki Bank Darah.

#### 4.8 Pernyataan posisi produk

Pernyataan posisi produk (*product positioning statement*) merupakan deskripsi singkat dan sederhana atas apa yang dilakukan oleh sistem. Pada tabel 4.7 dijelaskan mengenai *product overview* dari sistem informasi bank darah.

**Tabel 4.7 Pernyataan posisi produk**

Untuk	Masyarakat, pegawai PMI dan pegawai Rumah Sakit yang memiliki Bank Darah.
Yang	Membutuhkan informasi mengenai ketersediaan stok darah yang ada di PMI dan Rumah Sakit.
Sebuah	Sistem informasi bank darah adalah sistem informasi berbasis web yang bertujuan untuk memudahkan masyarakat mendapatkan informasi mengenai stok darah yang ada di PMI dan Rumah Sakit.
Yang	Menyediakan fitur yang mampu digunakan untuk mengelola data darah beserta stok darah (seperti menambah, memperbarui, melakukan pencatatan darah keluar), mengelola data Rumah Sakit yang memiliki Bank Darah (seperti menambah dan memperbarui data Rumah Sakit), mengelola data pendonor (seperti menambah dan memperbarui data pendonor) dan mampu memberikan informasi mengenai stok darah yang ada di PMI maupun Rumah Sakit.
Tidak seperti	Yang sedang berjalan saat ini, yaitu masih belum tersedianya informasi yang dapat di akses masyarakat untuk mengetahui jumlah stok darah yang ada di PMI dan Rumah Sakit.

**Tabel 4.7 Pernyataan posisi produk (lanjutan)**

Produk	Sistem informasi bank darah tersedia selama 24 jam selama terkoneksi dengan internet tanpa perlu datang ke PMI atau Rumah Sakit terlebih dahulu untuk mencari darah yang dibutuhkan.
--------	--

#### 4.9 Fitur

Fitur yang akan dibuat dalam perancangan sistem dijelaskan pada tabel 4.8. Dalam memberikan nilai prioritas, penulis melakukan wawancara terhadap pihak yang bersangkutan yaitu oleh PMI dan masyarakat, selain itu juga disesuaikan dengan yang dibutuhkan oleh pemangku kepentingan.

**Tabel 4.8 Fitur**

No	Fitur	Nama Fitur	Deskripsi	Prioritas
1	FT-01	Mengelola data stok darah	Digunakan oleh Admin PMI untuk mengelola stok darah yang ada saat ini. Mengelola stok darah nantinya dapat melakukan menambah data darah, memperbarui data darah, menambah darah keluar dan memperbarui data darah keluar	M
2	FT-02	Mengelola data pendonor	Digunakan oleh Admin PMI untuk mengelola data pendonor. Mengelola data pendonor dapat berupa menambah data pendonor, memperbarui data pendonor dan menghapus data pendonor	M
3	FT-03	Mengelola data Rumah Sakit	Digunakan oleh Admin PMI untuk mengelola data Rumah Sakit yang memiliki Bank Darah. Nantinya agar pihak PMI dapat memantau jumlah stok darah yang ada di Rumah Sakit agar tidak terjadi keterlambatan pengiriman darah. Mengelola data rumah sakit nantinya dapat berupa menambah data Rumah Sakit, memperbarui data Rumah Sakit	M
4	FT-04	Mengelola user	Digunakan oleh Admin PMI untuk mengelola data user, terkait tambah user dan memperbarui data user.	M
5	FT-05	Mengelola darah keluar	Digunakan oleh Admin Rumah Sakit untuk melakukan pencatatan data darah keluar yang ada di Rumah Sakit yang nantinya akan ditampilkan ke Sistem Informasi Bank Darah	M
6	FT-06	Mengecek data darah Rumah Sakit	Digunakan oleh Admin Rumah Sakit untuk mengecek ketersediaan darah di Rumah Sakit tersebut	M
7	FT-07	Mengecek ketersediaan darah PMI dan Rumah Sakit	Digunakan oleh masyarakat untuk melakukan pencarian darah sesuai dengan yang dibutuhkan. Pasien dapat melihat stok darah di PMI maupun Rumah Sakit	M

**Tabel 4.8 Fitur (lanjutan)**

No	Fitur	Nama Fitur	Deskripsi	Prioritas
8	FT-08	Autentikasi	Digunakan oleh Admin PMI dan Admin Rumah Sakit untuk masuk kedalam sistem dan menjalankan sistem	M

#### 4.10 Persyaratan fungsional

Tabel 4.9 merupakan persyaratan fungsional dari sistem yang dirancang. Persyaratan fungsional di dapatkan dari hasil menganalisis masalah yang ada berdasarkan dengan yang diharapkan pengguna.

**Tabel 4.9 Persyaratan fungsional sistem**

No	Kode fitur	Kode dasar persyaratan fungsional	Kode lengkap persyaratan fungsional	Deskripsi
1	FT-01	SRS-F-01	SRS-F-01-01	Sistem mampu menambah data darah masuk.
			SRS-F-01-02	Sistem mampu memperbarui data darah masuk.
			SRS-F-01-03	Sistem mampu menambah data darah keluar.
			SRS-F-01-04	Sistem mampu memperbarui data darah keluar.
			SRS-F-01-05	Sistem mampu menampilkan data darah.
2	FT-02	SRS-F-02	SRS-F-02-01	Sistem mampu menambah data pendonor.
			SRS-F-02-02	Sistem mampu memperbarui data pendonor.
			SRS-F-02-03	Sistem mampu menampilkan data pendonor.
3	FT-03	SRS-F-03	SRS-F-03-01	Sistem mampu menambah data Rumah Sakit yang Memiliki Bank Darah.
			SRS-F-03-02	Sistem mampu memperbarui data Rumah Sakit yang memiliki Bank Darah.
			SRS-F-03-03	Sistem mampu menampilkan data Rumah Sakit yang memiliki Bank Darah.
4	FT-04	SRS-F-04	SRS-F-04-01	Sistem mampu menambah data user PMI dan Rumah Sakit.
			SRS-F-04-02	Sistem mampu memperbarui data user.
			SRS-F-04-03	Sistem mampu menampilkan data user.
5	FT-05	SRS-F-05	SRS-F-05-01	Sistem mampu mencatat data darah keluar yang terjadi di Rumah Sakit.
6	FT-06	SRS-F-06	SRS-F-06-01	Sistem mampu menampilkan data stok darah yang ada di Rumah Sakit.
7	FT-07	SRS-F-07	SRS-F-07-01	Sistem mampu menampilkan data beserta jumlah stok darah yang tersedia

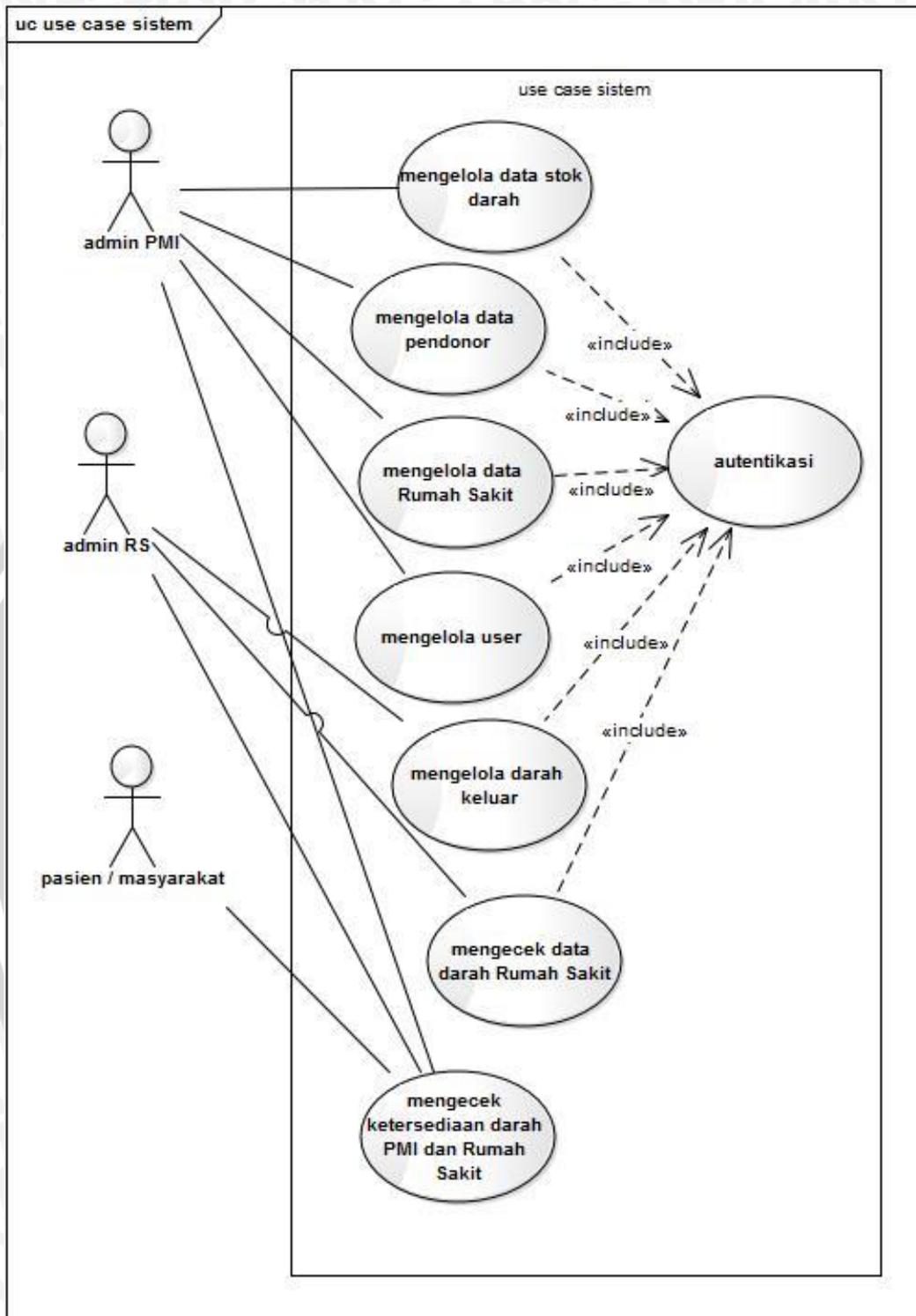
**Tabel 4.9 Persyaratan fungsional sistem (lanjutan)**

No	Kode fitur	Kode dasar persyaratan fungsional	Kode lengkap persyaratan fungsional	Deskripsi
				PMI dan Rumah Sakit yang memiliki Bank Darah.
8	FT-08	SRS-F-08	SRS-F-08-01	Sistem mampu melakukan autentikasi yang digunakan Admin PMI dan Admin Rumah Sakit untuk masuk ke dalam sistem dan dapat menjalankan fungsi yang ada di dalam sistem.

#### 4.11 Use case diagram

Diagram use case sistem digunakan untuk menggambarkan interaksi yang terjadi antara aktor dengan sistem. Beberapa aktor yang terlibat dalam sistem yaitu aktor Admin PMI merupakan Admin pengguna yang berada pada PMI untuk melakukan pengelolaan darah mulai dari darah masuk hingga darah keluar, mengelola informasi mengenai stok darah yang ada di PMI, dan mengelola data Rumah Sakit. Aktor Admin Rumah Sakit merupakan admin pengguna yang berada di Rumah Sakit yang terintegrasi dengan PMI. Aktor pengguna merupakan orang umum yang menggunakan sistem atau yang berkebutuhan untuk melakukan pencarian darah. Diagram use case perancangan sistem untuk perancangan sistem informasi bank darah ditunjukkan pada Gambar 4.11.





Gambar 4.11 Diagram use case

#### 4.12 Tabel pemetaan diagram use case

Tabel 4.10 merupakan tabel untuk pemetaan antara use case dengan persyaratan fungsional sistem.

**Tabel 4.10 Pemetaan use case dengan persyaratan fungsional**

No	Kode Fungsi	Kode Use Case	Use Case	Aktor
1	SRS-F-01	UC-01	Mengelola data stok darah	Admin PMI
2	SRS-F-02	UC-02	Mengelola data pendonor	Admin PMI
3	SRS-F-03	UC-03	Mengelola data Rumah Sakit	Admin PMI
4	SRS-F-04	UC-04	Mengelola user	Admin PMI
4	SRS-F-05	UC-05	Mengelola darah keluar	Admin Rumah Sakit
5	SRS-F-06	UC-06	Mengecek data darah Rumah Sakit	Admin Rumah Sakit
6	SRS-F-07	UC-07	Sistem mampu menampilkan data beserta jumlah stok darah yang tersedia PMI dan Rumah Sakit yang memiliki Bank Darah	Masyarakat, Admin PMI, Admin Rumah Sakit
7	SRS-F-08	UC-08	Autentikasi	Admin PMI, Admin Rumah Sakit

#### 4.13 Spesifikasi use case

Pada sub bab ini akan dijelaskan mengenai spesifikasi *use case* yang mengacu pada corak atau model RUP.

##### 1. Mengelola data stok darah

Tabel 4.11 merupakan spesifikasi *use case* untuk mengelola data stok darah yang di dalamnya terdapat aktivitas untuk mengelola data darah masuk dan aktivitas untuk mengelola data transaksi atau darah keluar.

**Tabel 4.11 Spesifikasi use case mengelola data stok darah**

Mengelola data stok darah	
<b>Deskripsi</b>	Use case ini mendeskripsikan tentang bagaimana mengelola data stok darah yang menyediakan kemampuan menambah stok darah, menampilkan data darah, memperbarui data darah, menambah darah keluar, menampilkan data darah keluar dan memperbarui data darah keluar.
<b>Preconditions</b>	1. Untuk mengakses sistem informasi bank darah harus tersambung dengan jaringan internet.
<b>Basic flow of event</b>	{Membuka sistem}

	<ol style="list-style-type: none"> <li>1. Use case dimulai ketika aktor Admin PMI membuka sistem dan memilih untuk melakukan autentikasi.</li> <li>2. Sistem meminta Admin untuk mengisi data yang digunakan untuk autentikasi. <b>{Autentikasi}</b></li> <li>3. Include ke use case autentikasi untuk masuk kedalam sistem. <b>{Memilih mengelola stok darah}</b></li> <li>4. Admin PMI masuk kedalam sistem dan mengakses untuk mengelola stok darah.</li> <li>5. Sistem menampilkan pilihan untuk melakukan mengelola stok darah. <b>{Memilih pilihan mengelola data stok darah}</b></li> <li>6. Sistem menampilkan pilihan untuk menambah data darah masuk dan menampilkan data darah yang memiliki fungsi untuk memperbarui data darah masuk, menambah data darah keluar dan menampilkan data darah keluar yang memiliki fungsi untuk memperbarui data darah keluar.             <ol style="list-style-type: none"> <li>a. Jika Admin PMI memilih untuk menambah data stok darah, maka <i>subflow</i> untuk menambah data stok darah akan dijalankan.</li> <li>b. Jika Admin PMI memilih untuk memperbarui data darah, maka <i>subflow</i> untuk memperbarui data darah akan dijalankan.</li> <li>c. Jika Admin PMI memilih untuk menambah data darah keluar, maka <i>subflow</i> untuk menambah darah keluar akan dijalankan.</li> <li>d. Jika Admin PMI memilih untuk memperbarui data darah keluar, maka <i>subflow</i> untuk memperbarui data darah keluar akan dijalankan.</li> </ol> </li> <li><b>{Use case selesai}</b></li> <li>7. Use Case mengelola data stok darah selesai.</li> </ol>
<p><b>Alternative flows</b></p>	<ol style="list-style-type: none"> <li>1. Data yang dimasukkan salah. Pada <b>{Autentikasi}</b>, jika data yang dimasukkan salah maka sistem akan menampilkan pesan kesalahan bahwa data yang dimasukkan salah, kemudian sistem meminta Admin untuk memasukkan data yang diperlukan untuk autentikasi.</li> <li>2. Data yang diisi belum lengkap saat menambah data darah masuk. Pada <i>subflow</i> menambah data stok darah saat kondisi Admin PMI memilih simpan data darah, sistem mendeteksi apakah data yang dimasukkan sudah lengkap. Jika data belum lengkap maka sistem akan menampilkan pesan kesalahan bahwa data yang diisikan belum lengkap. Kemudian kembali ke <b>{Mengisi data stok darah}</b>.</li> <li>3. Data yang diisi belum lengkap saat memperbarui data darah masuk. Pada <i>subflow</i> memperbarui data darah saat kondisi Admin PMI memilih simpan data darah, sistem mendeteksi apakah data yang dimasukkan sudah lengkap. Jika data belum</li> </ol>



	<p>lengkap maka sistem akan menampilkan pesan kesalahan bahwa data yang diisikan belum lengkap. Kemudian kembali ke <b>{Mengisi data darah terbaru}</b>.</p> <p>4. Data yang diisi belum lengkap saat menambah data darah keluar.                  Pada <i>subflow</i> menambah darah keluar saat kondisi Admin PMI memilih simpan data darah, sistem mendeteksi apakah data yang dimasukkan sudah lengkap. Jika data belum lengkap maka sistem akan menampilkan pesan kesalahan bahwa data yang diisikan belum lengkap. Kemudian kembali ke <b>{Mengisi data darah keluar}</b>.</p> <p>5. Data yang diisi belum lengkap saat memperbarui data darah keluar.                  Pada <i>subflow</i> memperbarui data darah keluar saat kondisi Admin PMI memilih simpan data darah keluar, sistem mendeteksi apakah data yang dimasukkan sudah lengkap. Jika data belum lengkap maka sistem akan menampilkan pesan kesalahan bahwa data yang diisikan belum lengkap. Kemudian kembali ke <b>{Mengisi data darah keluar terbaru}</b>.</p>
<p><b>Subflows</b></p>	<p>1. Menambah Data Stok Darah  <b>{Memilih Menambah Data Stok Darah}</b></p> <ol style="list-style-type: none"> <li>1. Admin memilih menambahkan data stok darah.</li> <li>2. Sistem meminta Admin untuk mengisi data darah.</li> </ol> <p><b>{Mengisi Data Stok Darah}</b></p> <ol style="list-style-type: none"> <li>3. Admin PMI mengisi data yang diminta oleh sistem terkait data darah.</li> </ol> <p><b>{Menyimpan Data Stok Darah}</b></p> <ol style="list-style-type: none"> <li>4. Admin PMI memilih simpan data darah.</li> <li>5. Sistem menyimpan data darah.</li> <li>6. Sistem menampilkan data darah masuk.</li> </ol> <p>2. Memperbarui Data Darah  <b>{Memilih Memperbarui Data Darah}</b></p> <ol style="list-style-type: none"> <li>1. Admin PMI memilih untuk menampilkan data darah masuk.</li> <li>2. Admin PMI memilih fungsi untuk memperbarui data darah.</li> <li>3. Admin PMI memilih data darah yang akan diperbarui.</li> </ol> <p><b>{Mengisi Data Darah Terbaru}</b></p> <ol style="list-style-type: none"> <li>4. Sistem menampilkan detail data darah yang akan diperbarui.</li> <li>5. Admin PMI mengisi data darah terbaru sesuai yang dibutuhkan oleh sistem.</li> </ol> <p><b>{Menyimpan Data Darah Terbaru}</b></p> <ol style="list-style-type: none"> <li>6. Admin PMI memilih simpan data darah terbaru.</li> <li>7. Sistem memperbarui data darah masuk.</li> <li>8. Sistem menampilkan data darah.</li> </ol> <p>3. Menambah Darah Keluar  <b>{Memilih Menambah Darah Keluar}</b></p> <ol style="list-style-type: none"> <li>1. Admin PMI memilih menambahkan darah keluar.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Sistem meminta Admin untuk mengisi data darah keluar.</li> </ol> <p><b>{Mengisi Data Darah Keluar}</b></p> <ol style="list-style-type: none"> <li>3. Admin PMI mengisi data yang diminta oleh sistem terkait data darah keluar.</li> </ol> <p><b>{Menyimpan Data Darah Keluar}</b></p> <ol style="list-style-type: none"> <li>4. Admin PMI memilih simpan data darah.</li> <li>5. Sistem menyimpan data darah keluar kedalam sistem.</li> <li>6. Sistem menampilkan data darah.</li> </ol> <ol style="list-style-type: none"> <li>4. Memperbarui Data Darah Keluar</li> </ol> <p><b>{Memilih Memperbarui Data Darah Keluar}</b></p> <ol style="list-style-type: none"> <li>1. Admin PMI memilih untuk menampilkan data darah keluar.</li> <li>2. Admin PMI memilih fungsi untuk memperbarui data darah keluar.</li> <li>3. Admin PMI memilih data darah keluar yang akan diperbarui.</li> </ol> <p><b>{Mengisi Data Darah Keluar Terbaru}</b></p> <ol style="list-style-type: none"> <li>4. Sistem menampilkan detail data darah keluar yang akan diperbarui Admin PMI.</li> <li>5. Admin PMI mengisi data darah keluar terbaru sesuai yang dibutuhkan oleh sistem.</li> </ol> <p><b>{Menyimpan Data Darah Terbaru}</b></p> <ol style="list-style-type: none"> <li>6. Admin PMI memilih simpan data darah keluar.</li> <li>7. Sistem memperbarui data darah keluar.</li> <li>8. Sistem menampilkan data darah.</li> </ol>
<b>Postcondition</b>	1. Admin PMI Sakit berhasil mengelola data stok darah.

<b>Glossary</b>
Data darah yang dimaksud terkait: id darah, golongan darah, nama pendonor, id transaksi, status transaksi, nama rumah sakit, tanggal masuk, tanggal transaksi

2. Mengelola data pendonor  
 Tabel 4.12 merupakan spesifikasi *use case* untuk mengelola data pendonor berupa menambah dan memperbarui data pendonor.

**Tabel 4.12 Spesifikasi use case mengelola data pendonor**

<b>Mengelola Data Pendonor</b>	
<b>Deskripsi</b>	Use case ini mendeskripsikan tentang bagaimana mengelola data pendonor dan menyediakan kemampuan untuk menambah data pendonor, menampilkan data pendonor dan memperbarui data pendonor.
<b>Preconditions</b>	1. Untuk mengakses Sistem Informasi Bank Darah harus tersambung dengan jaringan internet.
<b>Basic flow of event</b>	<p><b>{Membuka sistem}</b></p> <ol style="list-style-type: none"> <li>1. Use case dimulai ketika aktor Admin PMI membuka sistem dan memilih untuk melakukan autentikasi.</li> <li>2. Sistem meminta Admin untuk mengisi data yang digunakan untuk proses autentikasi.</li> </ol>

	<p><b>{Autentikasi}</b></p> <p>3. Include ke use case autentikasi untuk masuk kedalam sistem.</p> <p><b>{Memilih mengelola data pendonor}</b></p> <p>4. Admin PMI masuk kedalam sistem dan mengakses untuk mengelola data pendonor.</p> <p>5. Sistem menampilkan pilihan untuk melakukan mengelola data pendonor.</p> <p><b>{Memilih pilihan mengelola data pendonor}</b></p> <p>6. Sistem menampilkan pilihan untuk menambah data pendonor dan memperbarui data pendonor.</p> <p>a. Jika Admin PMI memilih untuk menambah data pendonor, maka <i>subflow</i> untuk menambah data pendonor akan dijalankan.</p> <p>b. Jika Admin PMI memilih untuk memperbarui data pendonor, maka <i>subflow</i> untuk memperbarui data pendonor akan dijalankan.</p> <p><b>{Use Case Selesai}</b></p> <p>7. Use Case mengelola data pendonor selesai.</p>
<p><b>Alternative flows</b></p>	<p>1. Data yang dimasukkan salah. Pada <b>{Autentikasi}</b>, jika data yang dimasukkan salah maka sistem akan menampilkan pesan kesalahan bahwa data yang dimasukkan salah, kemudian sistem menampilkan halaman untuk melakukan autentikasi.</p> <p>2. Data yang diisi belum lengkap saat menambah data pendonor. Pada <i>subflow</i> menambah data pendonor saat kondisi Admin memilih simpan data pendonor, sistem mendeteksi apakah data yang dimasukkan sudah lengkap. Jika data belum lengkap maka sistem akan menampilkan pesan kesalahan bahwa data yang diisikan belum lengkap. Kemudian kembali ke <b>{Mengisi Data Pendonor}</b>.</p> <p>3. Data yang diisi belum lengkap saat memperbarui data pendonor. Pada <i>subflow</i> memperbarui data pendonor saat kondisi Admin PMI memilih simpan data pendonor, sistem mendeteksi apakah data yang dimasukkan sudah lengkap. Jika data belum lengkap maka sistem akan menampilkan pesan kesalahan bahwa data yang diisikan belum lengkap. Kemudian kembali ke <b>{Mengisi Data Pendonor Terbaru}</b>.</p>
<p><b>Subflows</b></p>	<p>1. Menambah Data pendonor <b>{Memilih menambah data pendonor}</b></p> <p>1. Admin PMI memilih menambahkan data Pendonor. <b>{Mengisi data pendonor}</b></p> <p>2. Sistem meminta Admin PMI untuk mengisi data pendonor.</p> <p>3. Admin PMI mengisi data yang diminta oleh oleh sistem terkait data pendonor. <b>{Menyimpan data pendonor}</b></p> <p>4. Admin PMI memilih simpan data pendonor.</p>

	<ol style="list-style-type: none"> <li>5. Sistem menyimpan data pendonor.</li> <li>6. Sistem menampilkan data pendonor.</li> </ol> <ol style="list-style-type: none"> <li>2. Memperbarui Data pendonor  <b>{Memilih memperbarui data pendonor}</b> <ol style="list-style-type: none"> <li>1. Admin PMI memilih untuk menampilkan data pendonor.</li> <li>2. Admin PMI memilih fungsi untuk memperbarui data pendonor.</li> <li>3. Admin PMI memilih data pendonor yang akan diperbarui.  <b>{Mengisi data pendonor terbaru}</b> <ol style="list-style-type: none"> <li>4. Sistem menampilkan detail data pendonor yang akan diperbarui Admin PMI.</li> <li>5. Admin PMI mengisi data pendonor terbaru sesuai yang dibutuhkan oleh sistem.  <b>{Menyimpan data pendonor terbaru}</b> <ol style="list-style-type: none"> <li>6. Admin PMI memilih simpan data pendonor.</li> <li>7. Sistem memperbarui data pendonor.</li> <li>8. Sistem menampilkan data pendonor.</li> </ol> </li> </ol> </li> </ol> </li> </ol>
<b>Postcondition</b>	<ol style="list-style-type: none"> <li>1. Admin PMI berhasil mengelola data pendonor.</li> </ol>

<b>Glossary</b>
Data pendonor yang dimaksud terkait: id pendonor, nama pendonor, nomor telepon pendonor dan alamat pendonor.

3. Mengelola data Rumah Sakit  
Tabel 4.13 merupakan spesifikasi *use case* untuk mengelola data Rumah Sakit berupa menambah dan memperbarui data Rumah Sakit.

**Tabel 4.13 Spesifikasi use case mengelola data rumah sakit**

<b>Mengelola Data Rumah Sakit</b>	
<b>Deskripsi</b>	Use case ini mendeskripsikan tentang bagaimana mengelola data Rumah Sakit dan menyediakan kemampuan untuk menambah data Rumah Sakit, menampilkan data Rumah Sakit dan memperbarui data Rumah Sakit.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Untuk mengakses Sistem Informasi Bank Darah harus tersambung dengan jaringan internet.</li> </ol>
<b>Basic flow of event</b>	<b>{Membuka sistem}</b> <ol style="list-style-type: none"> <li>1. Use case dimulai ketika aktor Admin PMI membuka sistem dan memilih untuk melakukan autentikasi.</li> <li>2. Sistem meminta Admin untuk mengisi data yang digunakan untuk proses autentikasi.  <b>{Autentikasi}</b> <ol style="list-style-type: none"> <li>3. Include ke use case autentikasi untuk masuk kedalam sistem.  <b>{Memilih Mengelola Rumah Sakit}</b> <ol style="list-style-type: none"> <li>4. Admin PMI memilih untuk mengelola data Rumah Sakit.</li> <li>5. Sistem menampilkan pilihan untuk mengelola Rumah Sakit.  <b>{Memilih pilihan mengelola data pendonor}</b> <ol style="list-style-type: none"> <li>6. Sistem menampilkan pilihan untuk menambah data Rumah Sakit dan memperbarui data Rumah Sakit.</li> </ol> </li> </ol> </li> </ol> </li> </ol>



	<p>a. Jika Admin PMI memilih untuk menambah Rumah Sakit, maka subflow untuk menambah Rumah Sakit akan dijalankan.</p> <p>b. Jika Admin PMI memilih untuk memperbaiki data Rumah Sakit, maka <i>subflow</i> untuk memperbaiki data Rumah Sakit akan dijalankan.</p> <p><b>{Use Case Selesai}</b></p> <p>7. Use Case mengelola data Rumah Sakit selesai.</p>
<p><b>Alternative flows</b></p>	<p>1. Data yang dimasukkan salah. Pada <b>{otentikasi}</b>, jika data yang dimasukkan salah maka sistem akan menampilkan pesan kesalahan bahwa data yang dimasukkan salah, kemudian sistem menampilkan halaman untuk melakukan autentikasi.</p> <p>2. Data yang diisi belum lengkap saat menambah data Rumah Sakit. Pada <i>subflow</i> menambah data Rumah Sakit saat kondisi Admin PMI memilih simpan data Rumah Sakit, sistem mendeteksi apakah data yang dimasukkan sudah lengkap. Jika data belum lengkap maka sistem akan menampilkan pesan kesalahan bahwa data yang diisikan belum lengkap. Kemudian kembali ke <b>{Mengisi Data Rumah Sakit}</b>.</p> <p>3. Data yang diisi belum lengkap saat memperbaiki data Rumah Sakit. Pada <i>subflow</i> memperbaiki data Rumah Sakit saat kondisi Admin memilih simpan data Rumah Sakit, sistem mendeteksi apakah data yang dimasukkan sudah lengkap. Jika data belum lengkap maka sistem akan menampilkan pesan kesalahan bahwa data yang diisikan belum lengkap. Kemudian kembali ke <b>{Mengisi Data Rumah Sakit Terbaru}</b>.</p>
<p><b>Subflows</b></p>	<p>1. Menambah Data Rumah Sakit <b>{Memilih Menambah Data Rumah Sakit}</b></p> <ol style="list-style-type: none"> <li>1. Admin PMI memilih menambahkan data Rumah Sakit.</li> <li>2. Sistem meminta Admin PMI untuk mengisi data Rumah Sakit.</li> </ol> <p><b>{Mengisi Data Rumah Sakit}</b></p> <ol style="list-style-type: none"> <li>3. Admin PMI mengisi data yang diminta oleh oleh sistem terkait data Rumah Sakit.</li> </ol> <p><b>{Menyimpan Data Rumah Sakit}</b></p> <ol style="list-style-type: none"> <li>4. Admin PMI memilih simpan data Rumah Sakit.</li> <li>5. Sistem menyimpan data Rumah Sakit.</li> <li>6. Sistem menampilkan data Rumah Sakit.</li> </ol> <p>2. Memperbarui Data Rumah Sakit <b>{Memilih Memperbarui Data Rumah Sakit}</b></p> <ol style="list-style-type: none"> <li>1. Admin PMI memilih untuk menampilkan data Rumah Sakit.</li> <li>2. Admin PMI memilih fungsi untuk memperbaiki data Rumah Sakit.</li> <li>3. Admin PMI memilih data Rumah Sakit yang akan diperbarui.</li> </ol>

	<p><b>{Mengisi Data Rumah Sakit Terbaru}</b></p> <ol style="list-style-type: none"> <li>Sistem menampilkan detail data Rumah Sakit yang akan diperbarui Admin PMI.</li> <li>Admin PMI mengisi data Rumah Sakit terbaru sesuai yang dibutuhkan oleh sistem.</li> </ol> <p><b>{Menyimpan Data Rumah Sakit Terbaru}</b></p> <ol style="list-style-type: none"> <li>Admin PMI memilih simpan data Rumah Sakit.</li> <li>Sistem memperbarui data Rumah Sakit.</li> <li>Sistem menampilkan data Rumah Sakit.</li> </ol>
<b>Postcondition</b>	1. Admin PMI berhasil mengelola data Rumah Sakit.

<b>Glossary</b>
Data Rumah Sakit yang dimaksud terkait: id Rumah Sakit, nama Rumah Sakit, nomor telepon Rumah Sakit dan alamat Rumah Sakit.

4. Mengelola user

Tabel 4.14 merupakan spesifikasi *use case* untuk mengelola data user berupa menambah dan memperbarui data user.

**Tabel 4.14 Spesifikasi use case mengelola user**

<b>Mengelola user atau pengguna</b>	
<b>Deskripsi</b>	Use case ini mendeskripsikan tentang bagaimana mengelola data user dan menyediakan kemampuan untuk menambah data user, menampilkan data user dan memperbarui data user.
<b>Preconditions</b>	1. Untuk mengakses Sistem Informasi Bank Darah harus tersambung dengan jaringan internet.
<b>Basic flow of event</b>	<p><b>{Membuka sistem}</b></p> <ol style="list-style-type: none"> <li>Use case dimulai ketika aktor Admin PMI membuka sistem dan memilih untuk melakukan autentikasi.</li> <li>Sistem meminta Admin PMI untuk mengisi data yang digunakan untuk proses autentikasi.</li> </ol> <p><b>{Autentikasi}</b></p> <ol style="list-style-type: none"> <li>Include ke use case autentikasi untuk masuk kedalam sistem.</li> </ol> <p><b>{Memilih mengelola user}</b></p> <ol style="list-style-type: none"> <li>Admin PMI memilih untuk mengelola user.</li> <li>Sistem menampilkan pilihan untuk mengelola user.</li> </ol> <p><b>{Memilih pilihan mengelola data user}</b></p> <ol style="list-style-type: none"> <li>Sistem menampilkan pilihan untuk menambah data user dan memperbarui data user.                     <ol style="list-style-type: none"> <li>Jika Admin PMI memilih untuk menambah user, maka <i>subflow</i> untuk menambah user akan dijalankan.</li> <li>Jika Admin PMI memilih untuk memperbarui data user, maka <i>subflow</i> untuk memperbarui data user akan dijalankan.</li> </ol> </li> </ol> <p><b>{Use Case Selesai}</b></p> <ol style="list-style-type: none"> <li>Use Case mengelola data user selesai.</li> </ol>
<b>Alternative flows</b>	1. Data yang dimasukkan salah. Pada <b>{autentikasi}</b> , jika data yang dimasukkan salah maka sistem akan menampilkan pesan kesalahan bahwa data yang

	<p>dimasukkan salah, kemudian sistem menampilkan halaman untuk melakukan autentikasi.</p> <p>2. Data yang diisi belum lengkap saat menambah data user. Pada <i>subflow</i> menambah data user saat kondisi Admin PMI memilih simpan data user, sistem mendeteksi apakah data yang dimasukkan sudah lengkap. Jika data belum lengkap maka sistem akan menampilkan pesan kesalahan bahwa data yang diisikan belum lengkap. Kemudian kembali ke <b>{Mengisi Data user}</b>.</p> <p>3. Data yang diisi belum lengkap saat memperbarui data user. Pada <i>subflow</i> memperbarui data user saat kondisi Admin memilih simpan data user, sistem mendeteksi apakah data yang dimasukkan sudah lengkap. Jika data belum lengkap maka sistem akan menampilkan pesan kesalahan bahwa data yang diisikan belum lengkap. Kemudian kembali ke <b>{Mengisi Data user Terbaru}</b>.</p>
<b>Subflows</b>	<p>1. Menambah data user <b>{Memilih menambah data user}</b></p> <ol style="list-style-type: none"> <li>Admin PMI memilih menambahkan data user.</li> <li>Sistem meminta Admin PMI untuk mengisi data user. <b>{Mengisi data user}</b></li> <li>Admin PMI mengisi data yang diminta oleh sistem terkait data user. <b>{Menyimpan data user}</b></li> <li>Admin PMI memilih simpan data user.</li> <li>Sistem menyimpan data user.</li> <li>Sistem menampilkan data user.</li> </ol> <p>2. Memperbarui data user <b>{Memilih memperbarui data user}</b></p> <ol style="list-style-type: none"> <li>Admin PMI memilih untuk menampilkan data user.</li> <li>Admin PMI memilih fungsi untuk memperbarui data user.</li> <li>Admin PMI memilih data user yang akan diperbarui. <b>{Mengisi data user terbaru}</b></li> <li>Sistem menampilkan detail data user yang akan diperbarui Admin PMI.</li> <li>Admin PMI mengisi data user terbaru sesuai yang dibutuhkan oleh sistem. <b>{Menyimpan data user terbaru}</b></li> <li>Admin PMI memilih simpan data user.</li> <li>Sistem memperbarui data user.</li> <li>Sistem menampilkan data user.</li> </ol>
<b>Postcondition</b>	<p>1. Admin PMI berhasil mengelola data Rumah Sakit.</p>

**Glossary**

Data user yang dimaksud terkait: id admin, username, password dan level.

5. Mengelola darah keluar



Tabel 4.15 merupakan spesifikasi *use case* untuk mengelola darah keluar yang terjadi di Rumah Sakit berupa menambah dan memperbaiki data darah keluar.

**Tabel 4.15 Spesifikasi use case mengelola darah keluar**

<b>Mengelola Darah Keluar</b>	
<b>Deskripsi</b>	Use case ini mendeskripsikan tentang bagaimana mengelola darah keluar yang terjadi di Rumah Sakit dan menyediakan kemampuan untuk melakukan pencatatan data darah keluar, menampilkan data darah keluar dan memperbaiki data darah keluar.
<b>Preconditions</b>	1. Untuk mengakses Sistem Informasi Bank Darah harus tersambung dengan jaringan internet.
<b>Basic flow of event</b>	<p><b>{Membuka sistem}</b></p> <ol style="list-style-type: none"> <li>1. Use case dimulai ketika aktor Admin Rumah Sakit membuka sistem dan memilih untuk melakukan autentikasi.</li> <li>2. Sistem meminta Admin Rumah Sakit untuk mengisi data yang digunakan untuk proses autentikasi.</li> </ol> <p><b>{Autentikasi}</b></p> <ol style="list-style-type: none"> <li>3. Include ke use case autentikasi untuk masuk kedalam sistem.</li> </ol> <p><b>{Memilih mengelola darah keluar}</b></p> <ol style="list-style-type: none"> <li>4. Admin Rumah Sakit memilih untuk mengelola data darah keluar.</li> <li>5. Sistem menampilkan pilihan untuk melakukan mengelola darah keluar.</li> </ol> <p><b>{Memilih pilihan mengelola darah keluar}</b></p> <ol style="list-style-type: none"> <li>6. Sistem menampilkan pilihan untuk menambah darah keluar dan memperbaiki data darah keluar. <ol style="list-style-type: none"> <li>a. Jika Admin Rumah Sakit memilih menambah data darah keluar, maka subflow menambah data darah keluar akan dijalankan.</li> <li>b. Jika Admin Rumah Sakit memilih untuk memperbaiki data darah keluar, maka <i>subflow</i> untuk memperbaiki data darah keluar akan dijalankan.</li> </ol> </li> </ol> <p><b>{Use Case Selesai}</b></p> <ol style="list-style-type: none"> <li>7. Use Case mengelola darah keluar selesai.</li> </ol>
<b>Alternative flows</b>	<ol style="list-style-type: none"> <li>1. Data yang dimasukkan salah. Pada <b>{Autentikasi}</b>, jika ada data yang dimasukkan salah maka sistem akan menampilkan pesan kesalahan bahwa data yang dimasukkan salah, kemudian sistem menampilkan halaman untuk melakukan autentikasi.</li> <li>2. Data yang diisi belum lengkap saat menambah data darah keluar. Pada <i>subflow</i> menambah data darah keluar saat kondisi Admin Rumah Sakit memilih simpan data darah, sistem mendeteksi apakah data yang diisikan sudah lengkap. Jika data belum lengkap maka sistem akan menampilkan pesan kesalahan bahwa data yang diisikan belum lengkap. Kemudian kembali ke <b>{Mengisi Data Darah Keluar}</b>.</li> <li>3. Data yang diisi belum lengkap saat memperbaiki data darah keluar.</li> </ol>

	<p>Pada <i>subflow</i> memperbarui data darah keluar saat kondisi Admin Rumah Sakit memilih simpan data darah, sistem mendeteksi apakah data yang diisikan sudah lengkap. Jika data belum lengkap maka sistem akan menampilkan pesan kesalahan bahwa data yang diisikan belum lengkap. Kemudian kembali ke <b>{Mengisi Data Darah Keluar Terbaru}</b>.</p>
<b>Subflows</b>	<ol style="list-style-type: none"> <li>Menambah data darah keluar  <b>{Memilih Menambah Data Darah Keluar}</b> <ol style="list-style-type: none"> <li>Admin memilih menambahkan data darah keluar.</li> <li>Sistem meminta Admin Rumah Sakit untuk mengisi data darah keluar.</li> </ol> <b>{Mengisi Data Darah Keluar}</b> <ol style="list-style-type: none"> <li>Admin Rumah Sakit mengisi data yang diminta oleh sistem terkait data darah keluar yang terjadi di Rumah Sakit.</li> </ol> <b>{Menyimpan Data Darah Keluar}</b> <ol style="list-style-type: none"> <li>Admin Rumah Sakit memilih simpan data darah.</li> <li>Sistem menyimpan data darah keluar.</li> <li>Sistem menampilkan data darah keluar.</li> </ol> </li> <li>Memperbarui Data Darah Keluar  <b>{Memilih Memperbarui Data Darah Keluar}</b> <ol style="list-style-type: none"> <li>Admin Rumah Sakit memilih untuk menampilkan data darah keluar.</li> <li>Admin Rumah Sakit memilih fungsi untuk memperbarui data darah keluar.</li> <li>Admin Rumah Sakit memilih data darah yang akan diperbarui.</li> </ol> <b>{Mengisi Data Darah Keluar Terbaru}</b> <ol style="list-style-type: none"> <li>Sistem menampilkan detail data darah yang akan diperbarui Admin Rumah Sakit.</li> <li>Admin Rumah Sakit mengisi data darah keluar terbaru sesuai yang dibutuhkan oleh sistem.</li> </ol> <b>{Menyimpan Data Darah Keluar Terbaru}</b> <ol style="list-style-type: none"> <li>Admin Rumah Sakit memilih simpan data darah.</li> <li>Sistem memperbarui data darah.</li> <li>Sistem menampilkan data darah keluar.</li> </ol> </li> </ol>
<b>Postcondition</b>	<ol style="list-style-type: none"> <li>Admin Rumah Sakit berhasil Mengelola darah keluar yang meliputi menambah data darah keluar dan memperbarui data darah keluar.</li> </ol>

**Glossary**

Data darah keluar yang dimaksud terkait: id transaksi Rumah Sakit, id darah, tanggal keluar, status dan nama penerima.

6. Mengecek data darah Rumah Sakit

Tabel 4.16 merupakan spesifikasi *use case* untuk mengecek data stok darah yang terdapat di Rumah Sakit tersebut.

**Tabel 4.16 Skenario use case mengecek data darah rumah sakit**

Mengecek data darah Rumah Sakit	
<b>Deskripsi</b>	Use case ini mendeskripsikan tentang bagaimana mengecek data darah Rumah Sakit dan menyediakan kemampuan untuk memberikan informasi mengenai data stok darah yang ada di Rumah Sakit tersebut.
<b>Preconditions</b>	1. Untuk mengakses Sistem Informasi Bank Darah harus tersambung dengan jaringan internet.
<b>Basic flow of event</b>	<p><b>{Membuka sistem}</b></p> <ol style="list-style-type: none"> <li>1. Use case dimulai ketika aktor Admin Rumah Sakit membuka sistem dan memilih untuk melakukan autentikasi.</li> <li>2. Sistem meminta Admin Rumah Sakit untuk mengisi data yang digunakan untuk proses autentikasi.</li> </ol> <p><b>{Autentikasi}</b></p> <ol style="list-style-type: none"> <li>3. Include ke use case autentikasi untuk masuk kedalam sistem.</li> </ol> <p><b>{Memilih Mengecek Data Darah Rumah Sakit}</b></p> <ol style="list-style-type: none"> <li>4. Admin Rumah Sakit memilih untuk mengecek ketersediaan darah.</li> <li>5. Sistem menampilkan jumlah ketersediaan darah berupa ketersediaan darah yang ada di Rumah Sakit tersebut.</li> </ol> <p><b>{Use Case Selesai}</b></p> <ol style="list-style-type: none"> <li>6. Use Case mengecek ketersediaan darah Rumah Sakit selesai.</li> </ol>
<b>Alternative flows</b>	
<b>Subflows</b>	
<b>Postcondition</b>	1. Sistem menampilkan data darah yang ada di Rumah Sakit.

**Glossary**

Data ketersediaan darah yang dimaksud terkait: data darah yang ada di Rumah Sakit berupa golongan darah dan jumlah darah.

7. Mengecek ketersediaan darah PMI dan Rumah Sakit

Tabel 4.17 merupakan spesifikasi *use case* untuk mengecek ketersediaan darah yang ada di PMI dan Rumah Sakit.

**Tabel 4.17 Spesifikasi use case mengecek ketersediaan darah pmi dan rumah sakit**

Mengecek Ketersediaan Darah PMI dan Rumah Sakit	
<b>Deskripsi</b>	Use case ini mendeskripsikan tentang bagaimana mengecek ketersediaan darah yang ada di PMI dan Rumah Sakit dan menyediakan kemampuan memberikan informasi mengenai data stok darah yang ada di PMI maupun di Rumah Sakit.
<b>Preconditions</b>	1. Untuk mengakses Sistem Informasi Bank Darah harus tersambung dengan jaringan internet.
<b>Basic flow of event</b>	<p><b>{Membuka sistem}</b></p> <ol style="list-style-type: none"> <li>1. Use case dimulai ketika aktor masyarakat membuka sistem.</li> </ol>



	<p><b>{Memilih mengecek data stok darah}</b></p> <ol style="list-style-type: none"> <li>masyarakat memilih untuk mengecek ketersediaan darah.</li> <li>sistem menampilkan data ketersediaan darah.</li> </ol> <p><b>{Memilih pilihan tempat ketersediaan darah}</b></p> <ol style="list-style-type: none"> <li>Sistem menampilkan pilihan tempat ketersediaan darah.             <ol style="list-style-type: none"> <li>Jika masyarakat memilih untuk mengecek ketersediaan darah di PMI, maka subflow menampilkan ketersediaan darah di PMI akan dijalankan.</li> <li>Jika masyarakat memilih untuk mengecek ketersediaan darah di Rumah Sakit, maka <i>subflow</i> menampilkan ketersediaan darah di Rumah Sakit akan dijalankan.</li> </ol> </li> </ol> <p><b>{Use Case Selesai}</b></p> <ol style="list-style-type: none"> <li>Use Case mengecek ketersediaan darah selesai.</li> </ol>
<b>Alternative flows</b>	
<b>Subflows</b>	<ol style="list-style-type: none"> <li>Menampilkan ketersediaan darah PMI             <p><b>{Memilih Ketersediaan PMI}</b></p> <ol style="list-style-type: none"> <li>Masyarakat memilih ketersediaan darah yang ada di PMI.</li> <li>Sistem menampilkan ketersediaan darah yang ada di PMI.</li> </ol> </li> <li>Menampilkan ketersediaan Rumah Sakit             <p><b>{Memilih Ketersediaan Darah Rumah Sakit}</b></p> <ol style="list-style-type: none"> <li>Masyarakat memilih ketersediaan darah yang ada di Rumah Sakit.</li> <li>Sistem menampilkan data stok darah di Rumah Sakit.</li> </ol> </li> </ol>
<b>Postcondition</b>	<ol style="list-style-type: none"> <li>Sistem menampilkan data stok darah yang ada di PMI dan Rumah Sakit.</li> </ol>

<b>Glossary</b>
Data ketersediaan darah yang dimaksud terkait: data darah yang ada di PMI dan ketersediaan darah di Rumah Sakit berupa golongan darah dan jumlah.

## 8. Autentikasi

Tabel 4.18 merupakan spesifikasi *use case* untuk autentikasi, yang dilakukan oleh aktor admin PMI dan admin Rumah Sakit.

**Tabel 4.18 Spesifikasi use case autentikasi**

<b>Autentikasi</b>	
<b>Deskripsi</b>	Use case ini include dari use case lain. Mendeskripsikan bagaimana pengguna sistem masuk ke dalam jenis dengan pengguna yang berbeda-beda sehingga dapat menggunakan fungsi atau fitur yang disediakan sistem.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>Untuk mengakses Sistem Informasi Bank Darah harus tersambung dengan jaringan internet.</li> <li>Admin PMI atau Admin Rumah Sakit membuka sistem.</li> </ol>
<b>Basic flow of event</b>	<p><b>{Memilih autentikasi}</b></p> <ol style="list-style-type: none"> <li>Use case dimulai ketika aktor Admin PMI atau aktor Admin Rumah Sakit membuka sistem dan memilih untuk melakukan autentikasi untuk dapat menjalankan fungsi dalam sistem.</li> <li>Sistem meminta Admin PMI dan Admin Rumah Sakit untuk mengisi data yang digunakan untuk proses autentikasi.</li> </ol>

	<p><b>{Mengisi data yang diperlukan untuk autentikasi}</b></p> <p>3. Admin PMI atau Admin Rumah Sakit memasukkan data yang dibutuhkan sistem untuk melakukan autentikasi.</p> <p><b>{Memilih validasi}</b></p> <p>4. Admin PMI dan Admin Rumah Sakit memilih untuk validasi data autentikasi.</p> <p><b>{Melakukan validasi akun}</b></p> <p>5. Sistem mengecek id dan password valid.</p> <p><b>{Use Case Selesai}</b></p> <p>6. Use Case autentikasi selesai.</p>
<b>Alternative flows</b>	<p>1. Data yang dimasukkan salah Pada saat <b>{Melakukan validasi akun}</b>, jika data yang dimasukkan untuk melakukan autentikasi tidak sesuai dengan yang ada di sistem. Maka sistem akan menampilkan pesan kesalahan bahwa data yang dimasukkan salah.</p> <p>2. Form autentikasi tidak lengkap Pada saat <b>{Memilih validasi}</b>, jika data yang diisikan pada form tidak lengkap. Maka sistem akan menampilkan pesan kesalahan bahwa data yang diisikan tidak lengkap.</p> <p><b>{Use Case Selesai}.</b></p>
<b>Subflows</b>	
<b>Postcondition</b>	<p>1. Admin PMI atau Admin Rumah Sakit berhasil melakukan autentikasi, dapat masuk ke dalam sistem dan menjalankan sistem.</p>

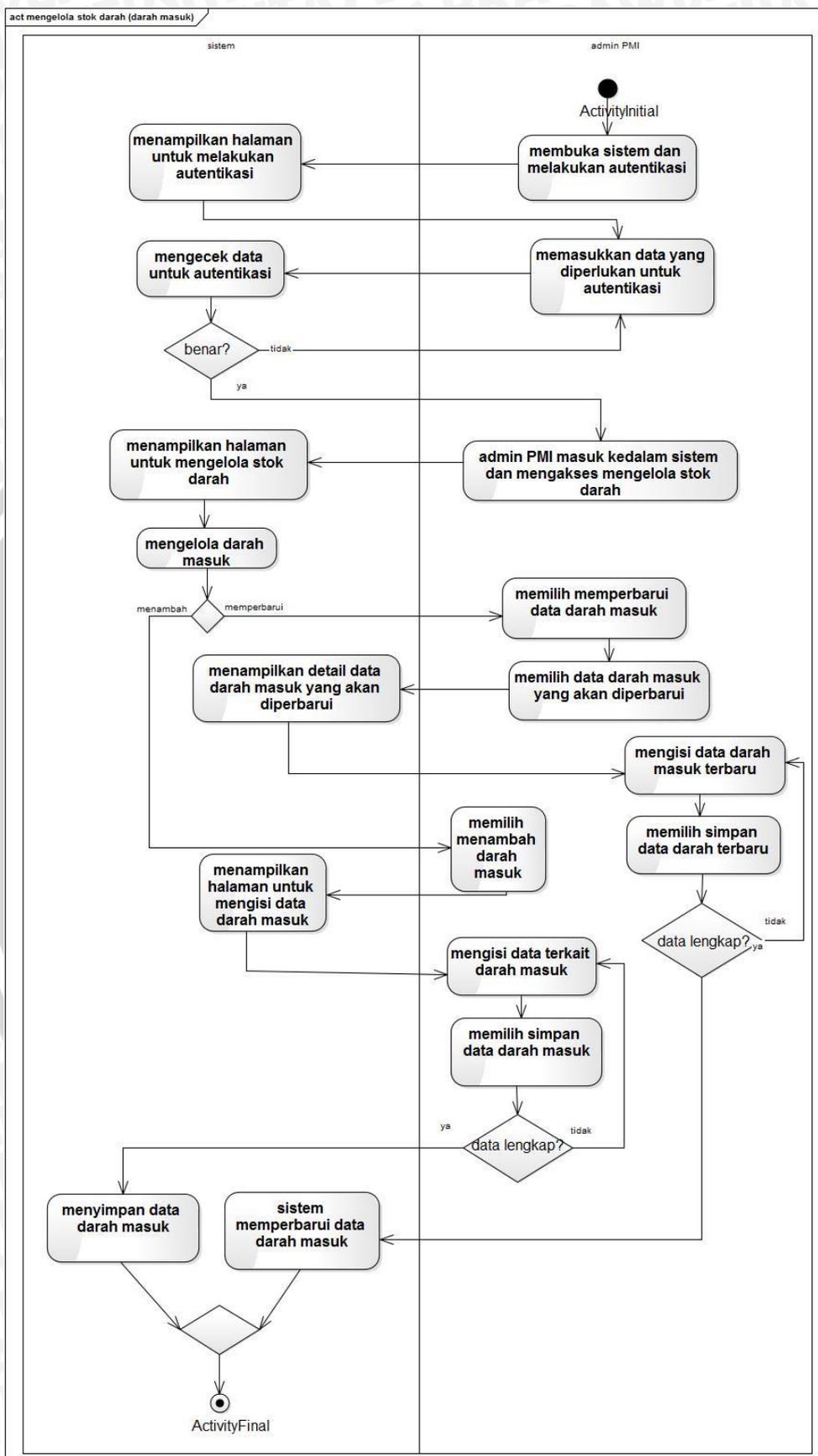
<b>Glossary</b>
Data autentikasi yang dimaksud terkait: id atau username dan password.

#### 4.14 Activity Diagram

*Activity Diagram* digunakan untuk menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau menu yang ada dalam perangkat lunak. Berikut adalah *activity diagram* dari sistem yang dirancang berdasarkan skenario use case yang telah dijelaskan.

##### 1. Activity diagram mengelola stok darah

Pada aktivitas mengelola stok darah, terdapat dua aktivitas yaitu untuk menambah data darah masuk dan menambah data darah keluar. Aktivitas tersebut dijalankan Admin PMI dalam keadaan telah melakukan autentikasi. Gambar 4.12 merupakan aktivitas proses mengelola stok darah untuk aktivitas menambah data darah masuk. Aktivitas dimulai ketika pengguna memilih mengelola stok darah dan melakukan untuk menambah darah masuk atau memperbarui data. Gambar 4.13 merupakan aktivitas saat melakukan transaksi, dapat menambah transaksi maupun memperbarui data transaksi.



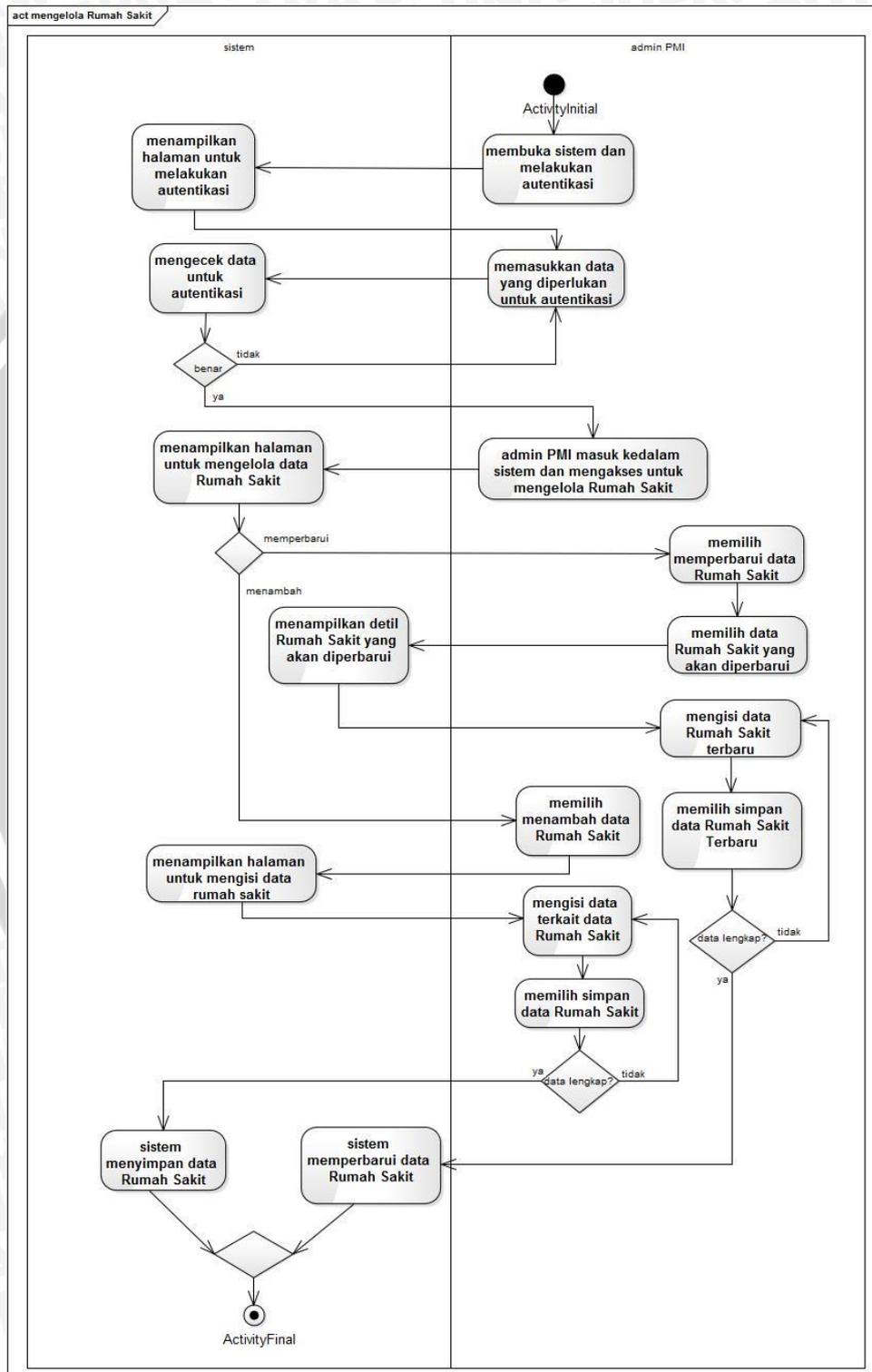
Gambar 4.12 Activity diagram mengelola stok darah (darah masuk)





### 3. Activity diagram mengelola data Rumah Sakit

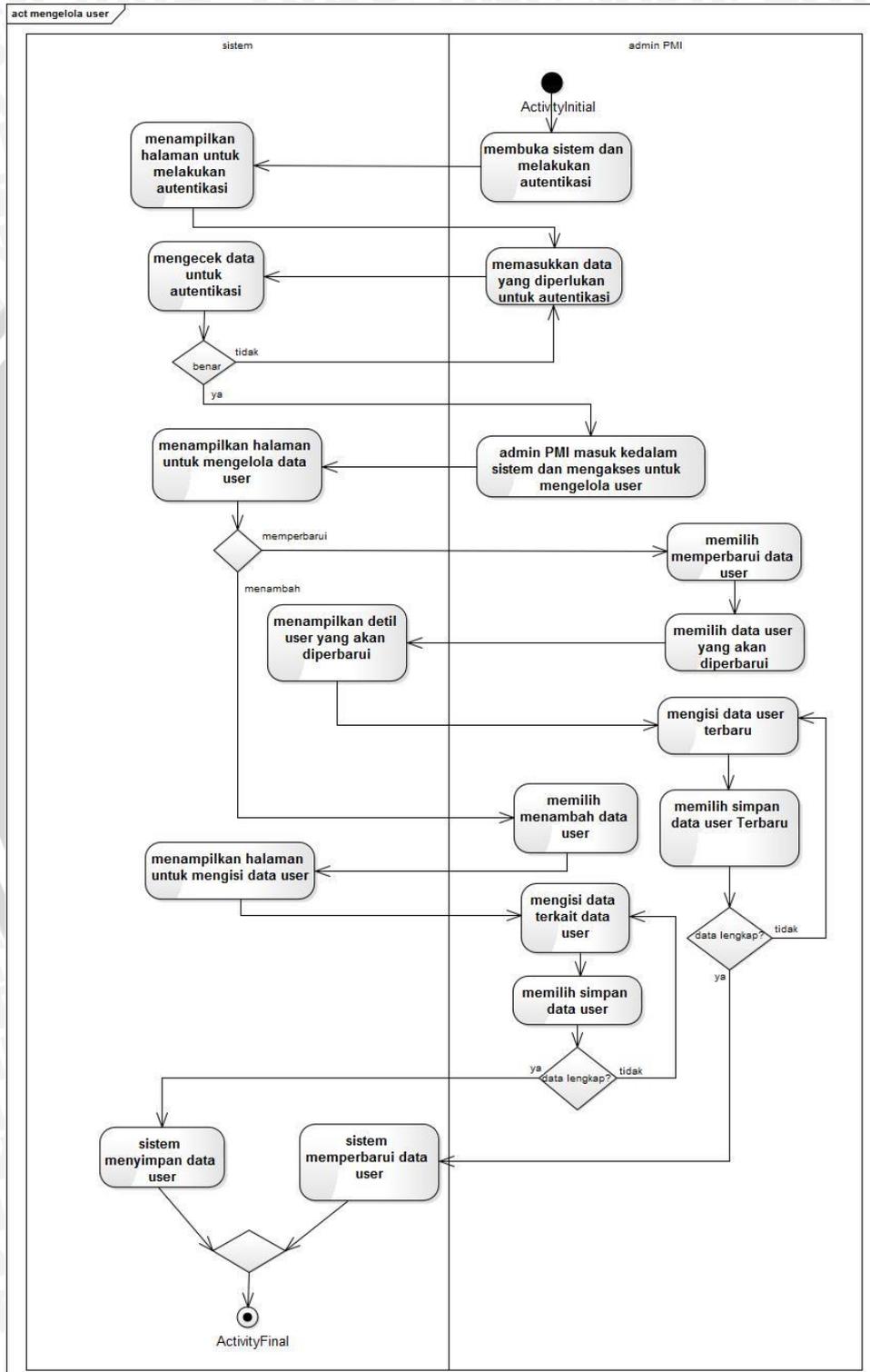
Gambar 4.15 merupakan aktivitas yang terjadi dalam proses mengelola data Rumah Sakit. Aktivitas ini dijalankan Admin PMI setelah autentikasi. Aktivitas dimulai ketika pengguna memilih untuk mengelola data Rumah Sakit yang nantinya dapat berupa menambah dan memperbaiki data Rumah Sakit.



Gambar 4.15 Activity diagram mengelola data pendonor

#### 4. Mengelola user

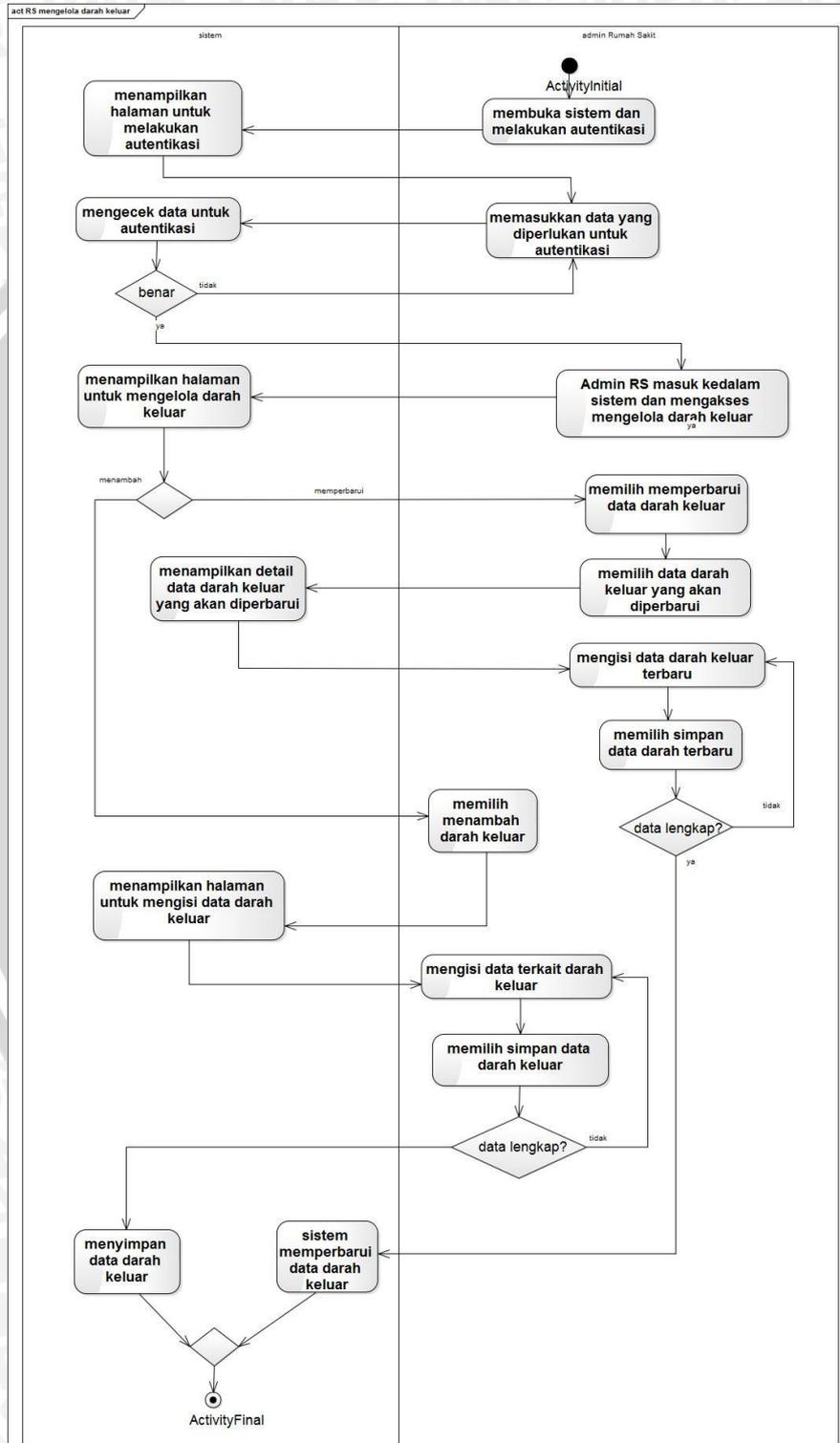
Gambar 4.16 merupakan aktivitas yang terjadi dalam proses mengelola data user. Aktivitas ini dijalankan Admin PMI setelah autentikasi. Aktivitas dimulai ketika pengguna memilih untuk mengelola data user yang nantinya dapat berupa menambah data user dan memperbarui data user.



Gambar 4.16 Activity diagram mengelola user

### 5. Mengelola darah keluar

Gambar 4.17 merupakan aktivitas yang terjadi dalam proses mengelola darah keluar. Aktivitas ini dijalankan Admin Rumah Sakit setelah autentikasi. Aktivitas dimulai ketika pengguna memilih untuk mengelola darah keluar yang nantinya dapat berupa menambah dan memperbarui data darah keluar.

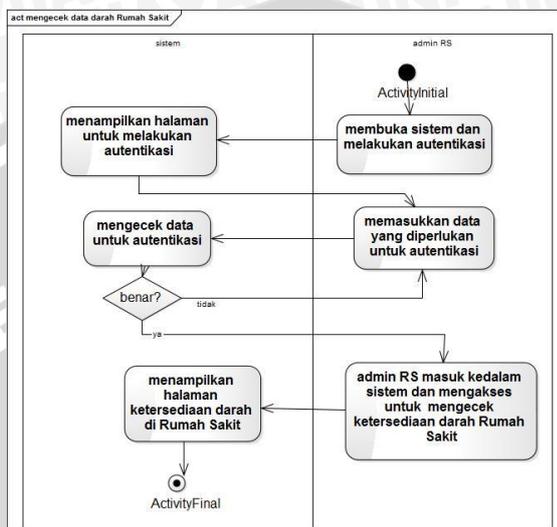


Gambar 4.17 Activity darah keluar



## 6. Mengecek data darah rumah sakit

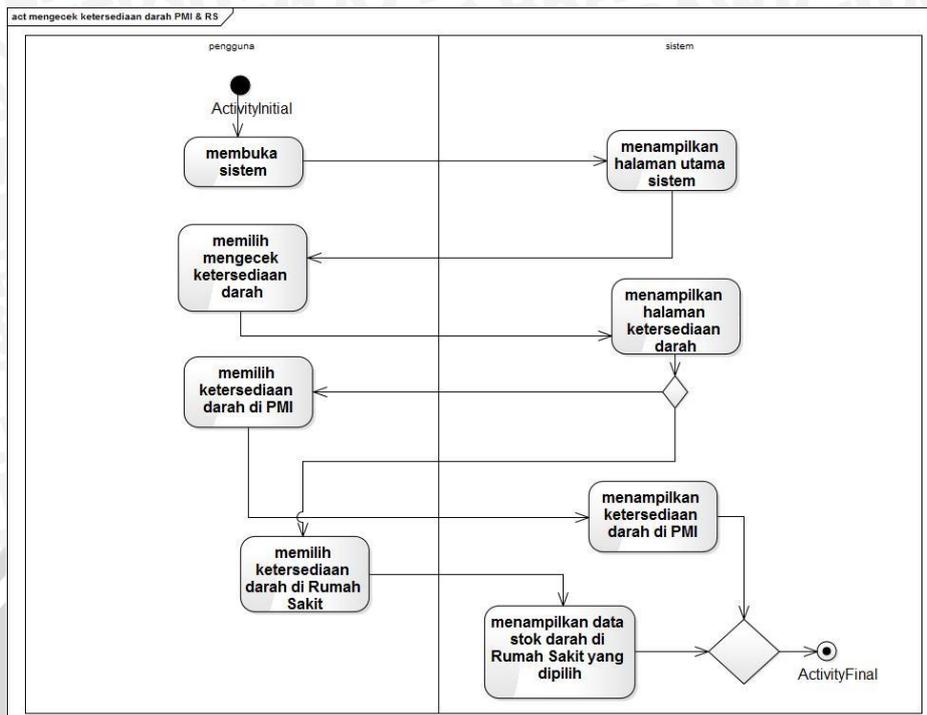
Gambar 4.18 merupakan aktivitas yang terjadi saat Admin Rumah Sakit mengecek ketersediaan darah yang ada di Rumah Sakit tersebut. Aktivitas ini dapat dijalankan jika Admin Rumah Sakit sudah melakukan autentikasi. Aktivitas dimulai ketika Admin Rumah Sakit memilih untuk mengecek ketersediaan darah yang ada di Rumah Sakit.



**Gambar 4.18 Activity mengecek data darah Rumah Sakit**

## 7. Mengecek ketersediaan darah PMI dan Rumah Sakit

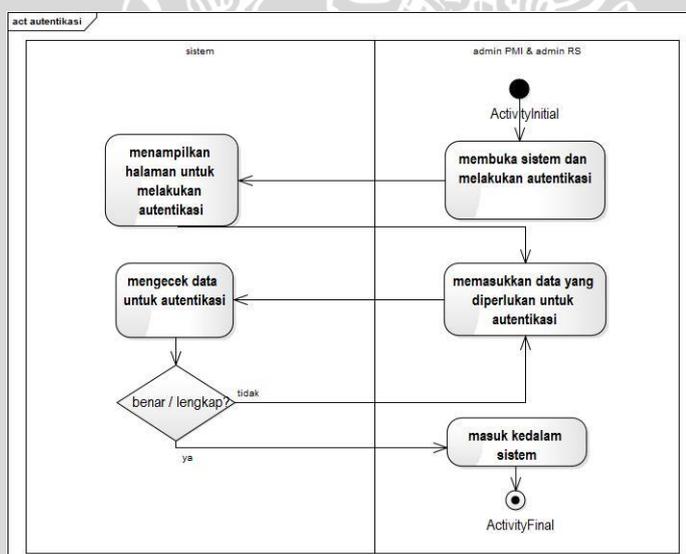
Gambar 4.19 merupakan aktivitas yang terjadi dalam proses mengecek ketersediaan darah PMI dan Rumah Sakit. Aktivitas ini dijalankan saat masyarakat membuka sistem dan mengecek ketersediaan darah yang ada di PMI maupun Rumah Sakit tanpa melakukan autentikasi.



Gambar 4.19 Activity mengecek ketersediaan darah PMI dan Rumah Sakit

### 8. Autentikasi

Gambar 4.20 merupakan aktivitas yang terjadi saat melakukan autentikasi.

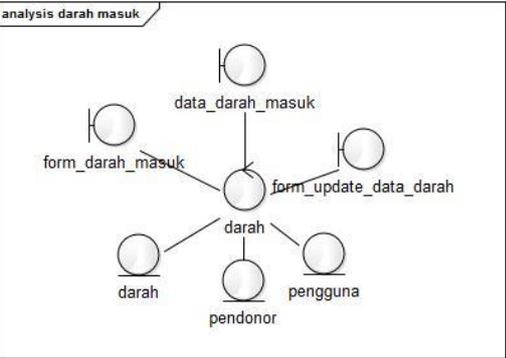
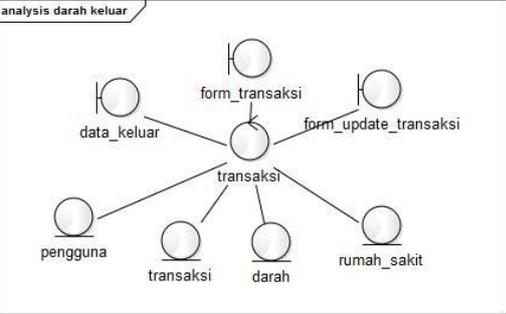
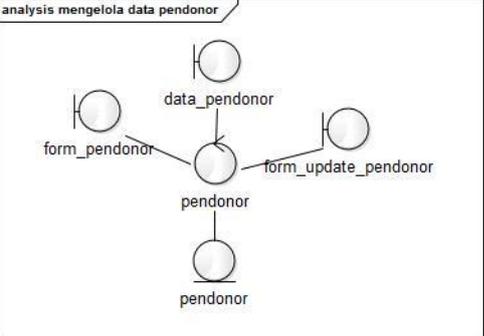
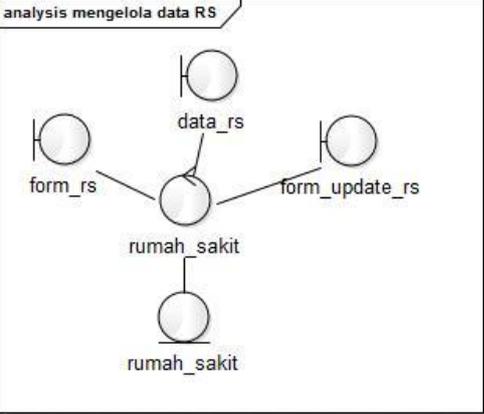


Gambar 4.20 Activity autentikasi

### 4.15 Perancangan kelas analisis

Pada tahapan ini dilakukan analisis kelas berdasarkan *use case* dan deskripsinya yang telah dibuat. Tabel 4.19 merupakan analisis kelas dari sistem yang akan dirancang.

Tabel 4.19 Kelas analisis

No	Use Case	Kelas Analisis
1	Mengelola data stok darah	<p><b>Darah masuk</b></p>  <p><b>Darah keluar</b></p> 
2	Mengelola data pendonor	<p><b>analysis mengelola data pendonor</b></p> 
3	Mengelola data Rumah Sakit	<p><b>analysis mengelola data RS</b></p> 

Tabel 4.19 Kelas analisis (lanjutan)

No	Use Case	Kelas Analisis
4	Mengelola user	<pre> classDiagram     class form_user     class data_user     class form_update_user     class pengguna     class pengguna      form_user -- data_user     data_user -- form_update_user     pengguna -- pengguna     </pre>
5	Mengelola darah keluar	<pre> classDiagram     class rs_form_transaksi     class rs_data_transaksi     class rs_update_transaksi     class transaksi_rs     class darah     class pengguna      rs_form_transaksi -- rs_data_transaksi     rs_data_transaksi -- rs_update_transaksi     transaksi_rs -- darah     transaksi_rs -- pengguna     </pre>
6	Mengecek data darah Rumah Sakit	<pre> classDiagram     class data_darah_rs     class darah_rs     class darah      data_darah_rs -- darah_rs     darah_rs -- darah     </pre>
7	Mengecek ketersediaan darah	<pre> classDiagram     class data_stok     class stok_darah     class darah      data_stok -- stok_darah     stok_darah -- darah     </pre>

**Tabel 4.19 Kelas analisis (lanjutan)**

No	Use Case	Kelas Analisis
8	Autentikasi	

**4.15.1 Mekanisme analisis**

Mekanisme analisis merupakan serangkaian layanan konseptual yang digunakan oleh objek analisis. Tujuan dari mekanisme analisis adalah untuk menunjukkan pola struktur, pola perilaku atau keduanya pada setiap kelas yang telah di analisis sebelumnya. Dalam penelitian ini mekanisme analisis yang digunakan adalah untuk kategori persistency yang digunakan untuk menjaga setiap elemen persisten, sehingga apabila terdapat setiap perubahan data baru, data akan tersimpan kedalam database dan security berfungsi untuk mengontrol setiap akses ke elemen. Tabel 4.20 merupakan mekanisme analisis yang digunakan dari penelitian yang dilakukan.

**Tabel 4.20 Kelas analisis**

Kelas Analisis	Mekanisme Analisis
pengguna	<i>Persistency, Security</i>
transaksi	<i>Security</i>
darah	<i>Security</i>
rumah_sakit	<i>Security</i>
pendonor	<i>Security</i>
transaksi_rs	<i>Security</i>

**4.15.2 Pengintegrasian kelas analisis**

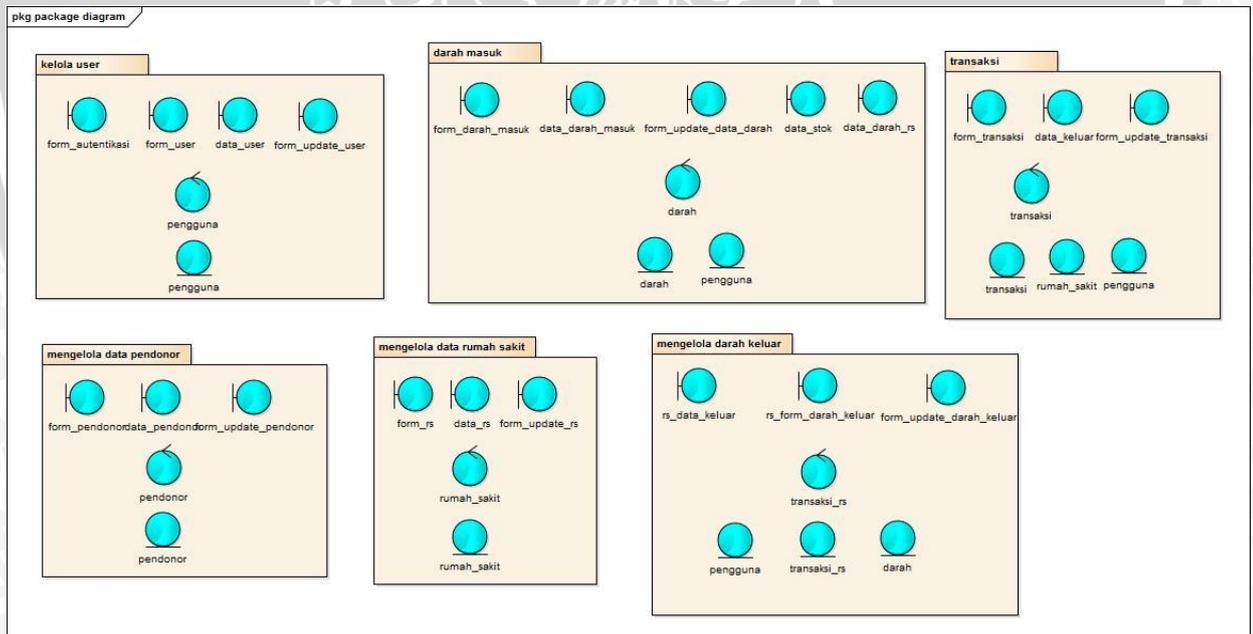
Pada gambar 4.21 merupakan pengintegrasian dari kelas analisis. Tujuan pengintegrasian kelas analisis adalah untuk memastikan bahwa setiap kelas analisis telah didefinisikan dengan baik.



Tabel 4.21 Kelas analisis

No	Kelas Analisis	Package
1	pengguna	::entity
2	transaksi_rs	::entity
3	rumah_sakit	::entity
4	transaksi	::entity
5	darah	::entity
6	pendonor	::entity
7	c_pendonor	::controller
8	c_darah	::controller
9	c_transaksi_rs	::controller
10	c_transaksi	::controller
11	c_rumah_sakit	::controller
12	c_pengguna	::controller

Pada gambar 4.22 merupakan gambar package diagram berdasarkan kelas analisis.



Gambar 4.22 Package diagram



## BAB 5 PERANCANGAN SISTEM

### 5.1 Perancangan interaksi dan kelas

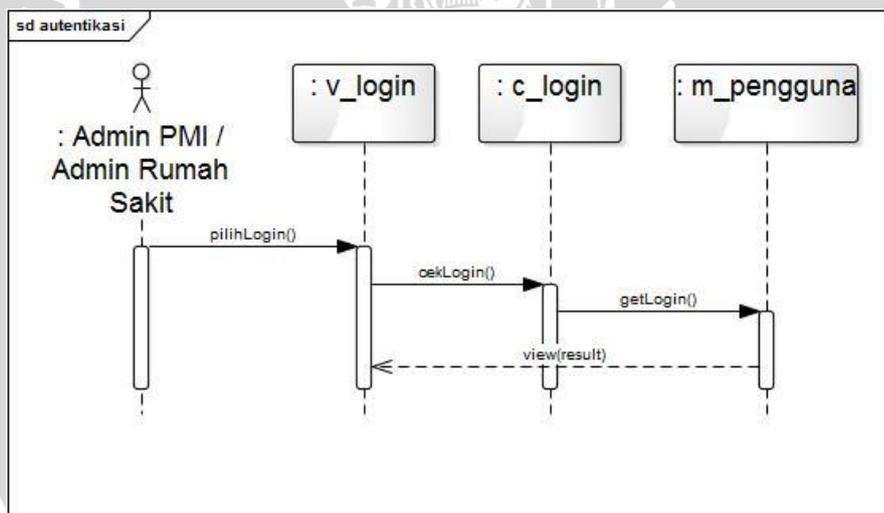
Bab ini menjelaskan mengenai perancangan interaksi dan kelas. Perancangan sistem dibuat berdasarkan analisis yang telah dilakukan sebelumnya dengan menggunakan *Unified Modelling Language* (UML).

#### 5.1.1 Diagram interaksi (sequence diagram)

Diagram interaksi menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Berikut adalah diagram interaksi dari sistem yang dirancang.

##### 1. Autentikasi

Gambar 5.1 merupakan diagram interaksi untuk melakukan autentikasi. Saat aktor memilih untuk melakukan autentikasi, halaman akan ditampilkan, kemudian setelah aktor memasukkan data yang diperlukan untuk autentikasi, objek *c\_login* akan melakukan fungsi *cekLogin* untuk memvalidasi apakah data yang dimasukkan sesuai atau tidak.

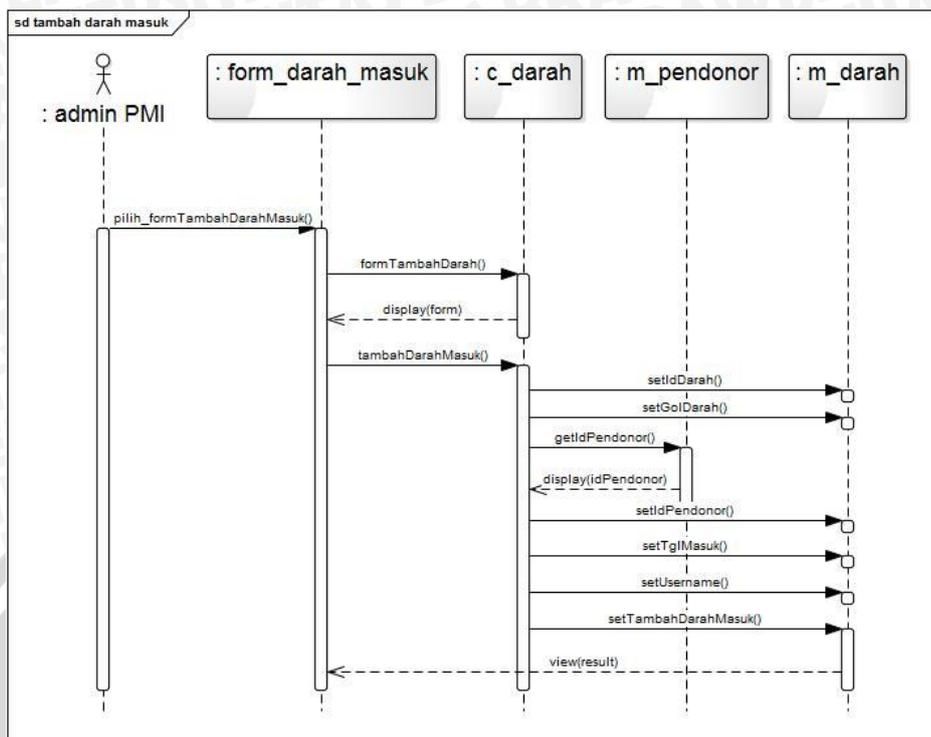


Gambar 5.1 Sequence diagram autentikasi

##### 2. Mengelola data stok darah

###### a. Menambah data darah masuk

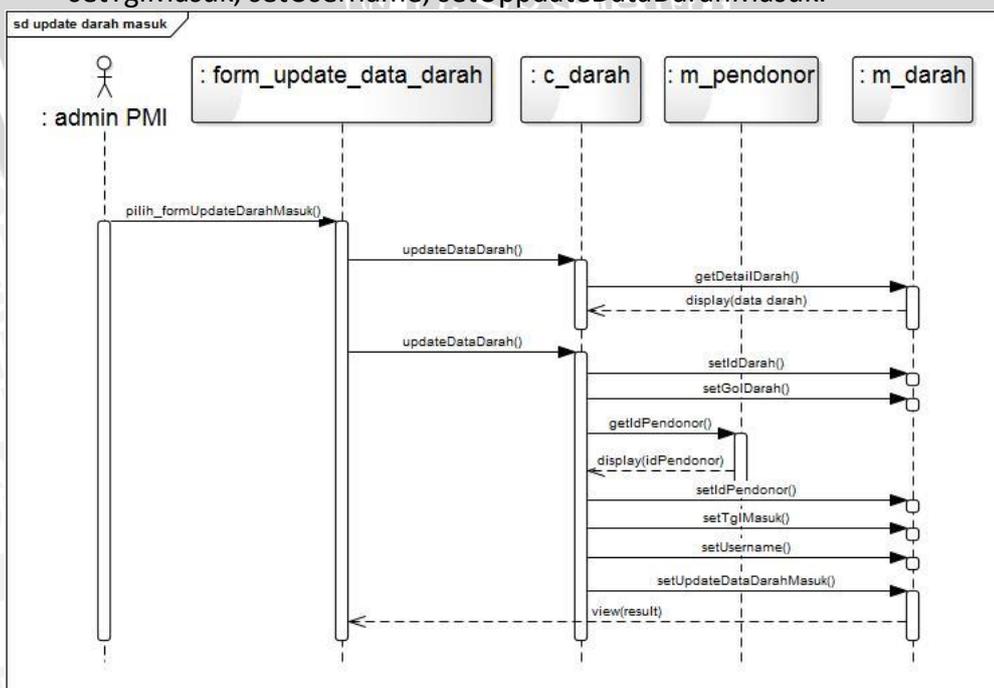
Gambar 5.2 merupakan diagram interaksi untuk proses menambah data darah masuk. Aktor memilih untuk menampilkan halaman untuk mengisi data yang dibutuhkan, kemudian setelah data diisi objek *c\_darah* akan menerima fungsi *tambahDarahMasuk* yang digunakan untuk mengirimkan data tersebut ke objek *m\_darah* dengan fungsi *setIdDarah*, *setIdPendoror*, *setTglMasuk*, *setTambahDarahMasuk*.



Gambar 5.2 Sequence diagram menambah darah masuk

b. Memperbarui data darah masuk

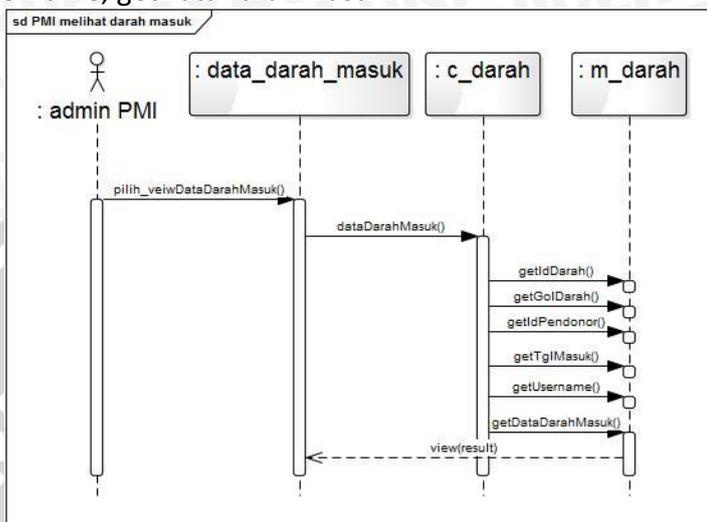
Gambar 5.3 merupakan diagram interaksi untuk proses memperbarui data darah masuk. Aktor memilih untuk menampilkan halaman untuk mengisi data yang dibutuhkan, kemudian setelah data diisi objek `c_darah` akan menerima fungsi `updateDataDarah` yang digunakan untuk mengirimkan data tersebut ke objek `m_darah` dengan fungsi `setIdDarah`, `setIdPendonor`, `setTglMasuk`, `setUpdateDataDarahMasuk`.



Gambar 5.3 Sequence diagram memperbarui data darah masuk

c. Melihat data darah masuk

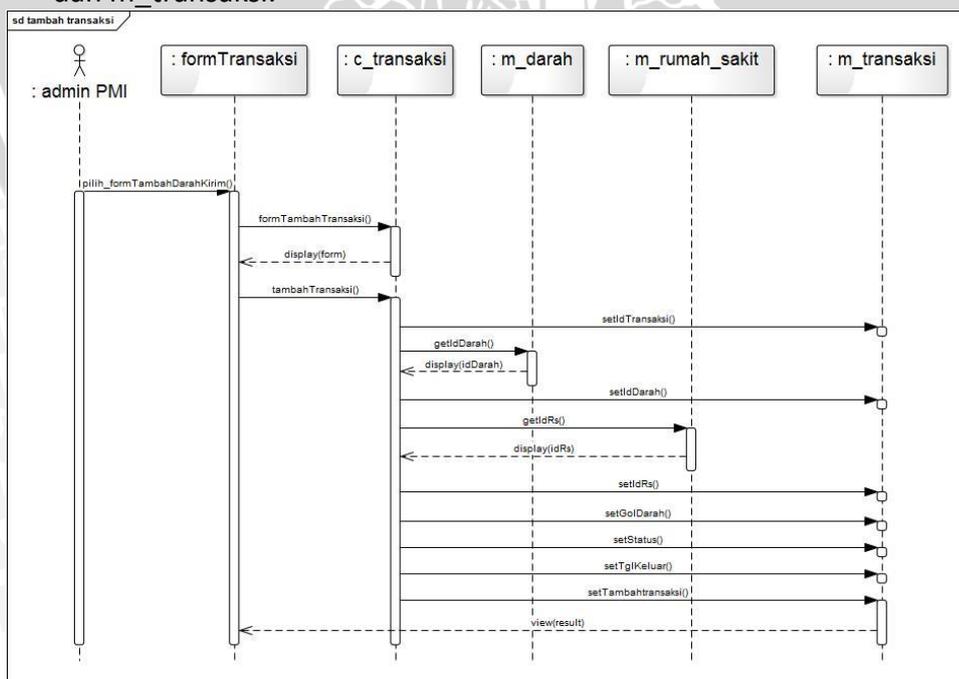
Gambar 5.4 merupakan diagram interaksi untuk proses melihat data darah masuk. Aktor memilih untuk menampilkan halaman data darah masuk, kemudian objek `c_darah` akan mendapatkan data dari objek `m_darah` berupa fungsi `getIdDarah`, `getGolDarah`, `getIdPendoror`, `getTglMasuk`, `getUsername`, `getDataDarahMasuk`.



Gambar 5.4 Sequence melihat data darah masuk

d. Menambah transaksi

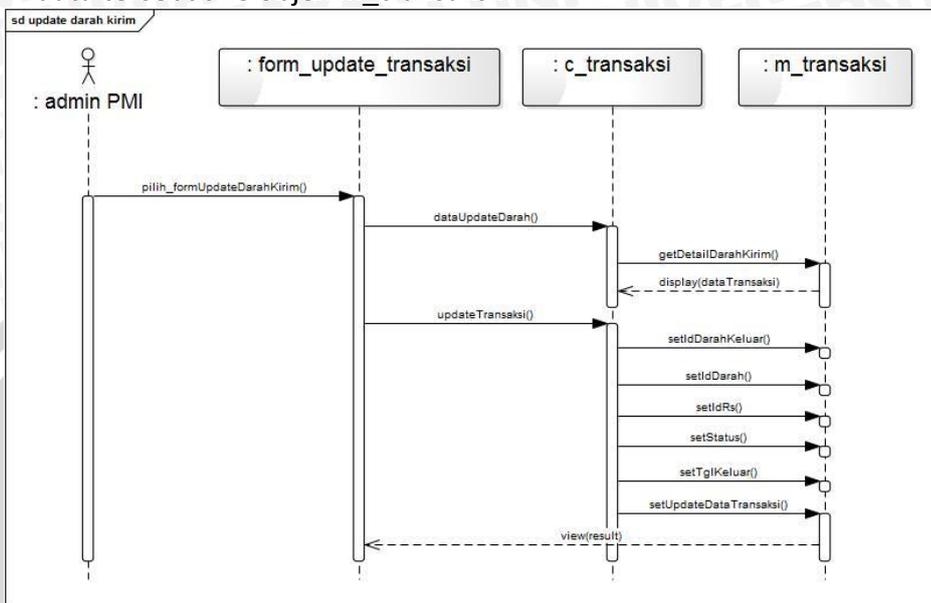
Gambar 5.5 merupakan diagram interaksi untuk proses menambah transaksi yang dilakukan oleh PMI. Aktor memilih untuk menampilkan halaman untuk mengisi data yang dibutuhkan terkait transaksi, kemudian setelah data diisi objek `c_transaksi` akan menerima fungsi `tambahTransaksi` yang digunakan untuk mengirimkan data tersebut ke objek `m_rumah_sakit` dan `m_transaksi`.



Gambar 5.5 Sequence diagram menambah transaksi

- e. Memperbarui data transaksi

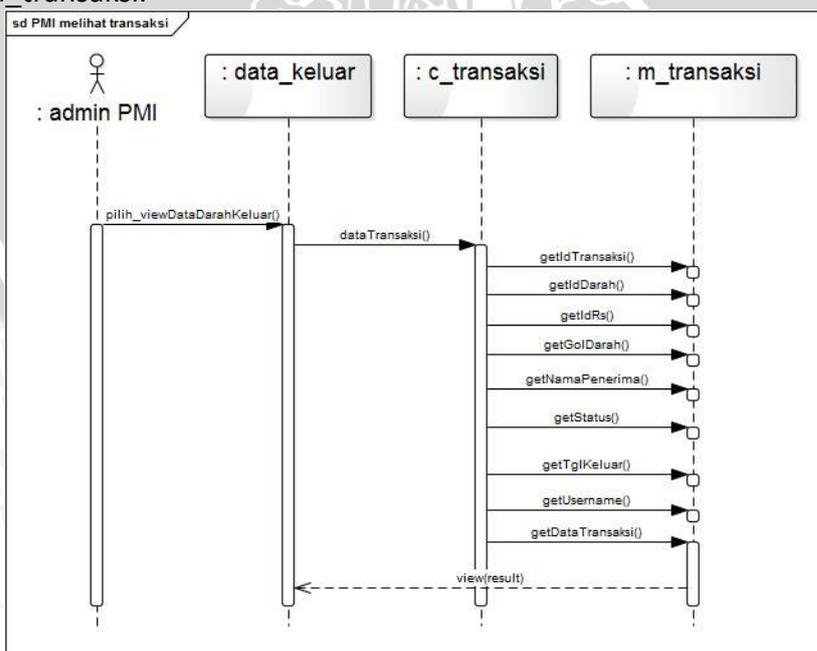
Gambar 5.6 merupakan diagram interaksi untuk proses memperbarui data transaksi. Aktor memilih untuk menampilkan halaman untuk mengisi data yang dibutuhkan, kemudian setelah data diisi objek c\_transaksi akan menerima fungsi updateTransaksi yang digunakan untuk mengirimkan data tersebut ke objek m\_transaksi.



Gambar 5.6 Sequence diagram memperbarui data transaksi

- f. Melihat data transaksi

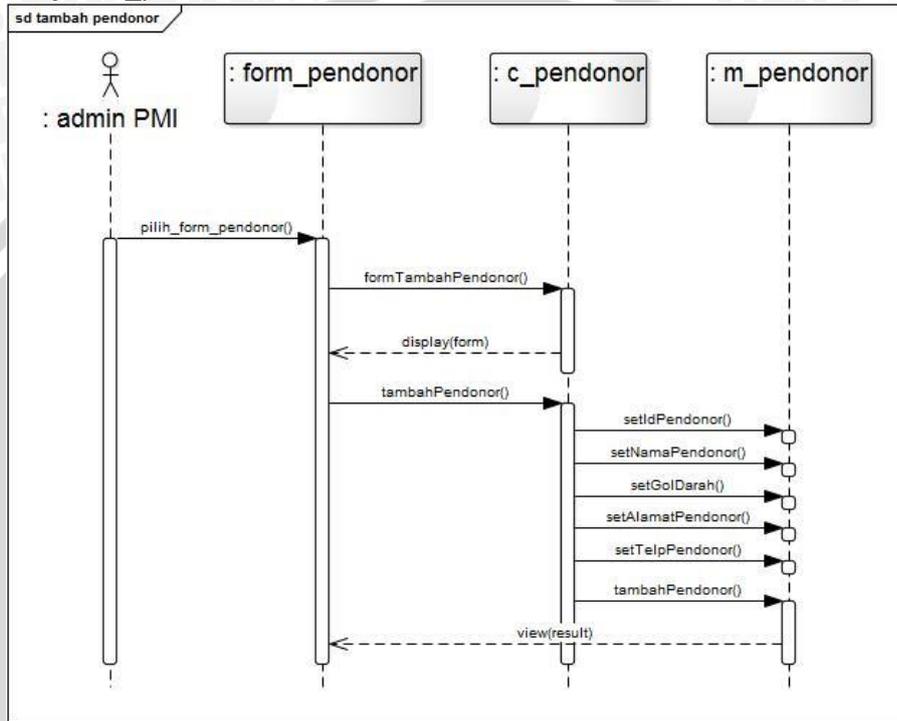
Gambar 5.7 merupakan diagram interaksi untuk proses melihat data transaksi. Aktor memilih untuk menampilkan halaman data darah keluar, kemudian objek c\_transaksi akan mendapatkan data dari objek m\_transaksi.



Gambar 5.7 Sequence diagram melihat data transaksi

- 3. Mengelola data pendonor
  - a. Menambah data pendonor

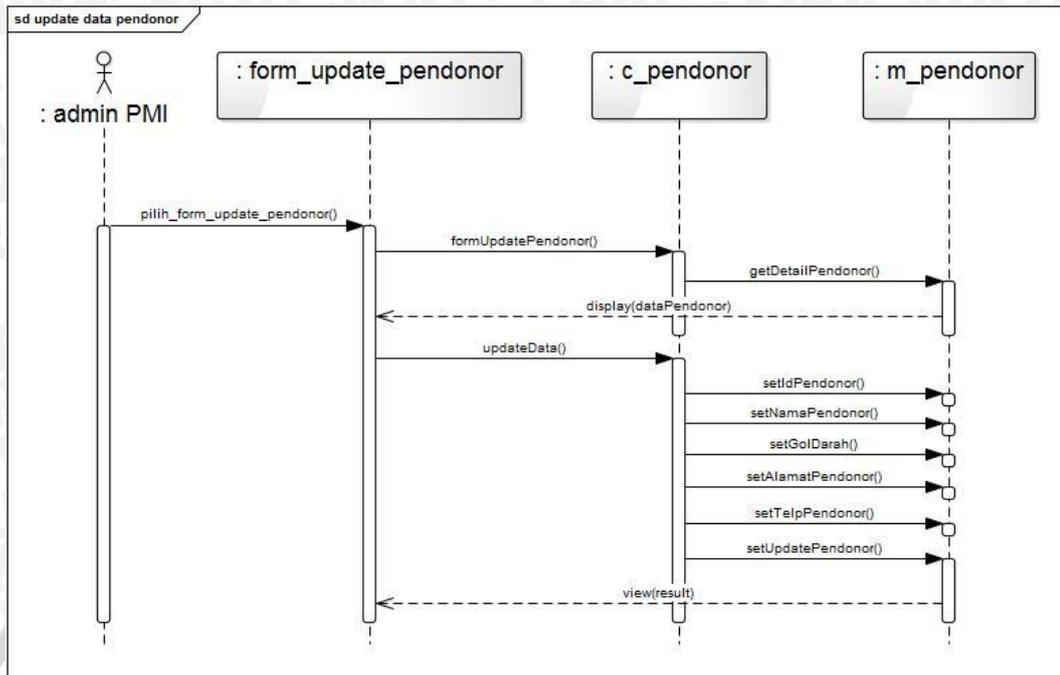
Gambar 5.8 merupakan diagram interaksi untuk proses menambah data pendonor yang dilakukan oleh PMI. Aktor memilih untuk menampilkan halaman untuk mengisi data yang dibutuhkan terkait data pendonor, kemudian setelah data diisi objek c\_pendonor akan menerima fungsi tambahPendonor yang digunakan untuk mengirimkan data tersebut ke objek m\_pendonor.



**Gambar 5.8 Sequence diagram menambah data pendonor**

- b. Memperbarui data pendonor

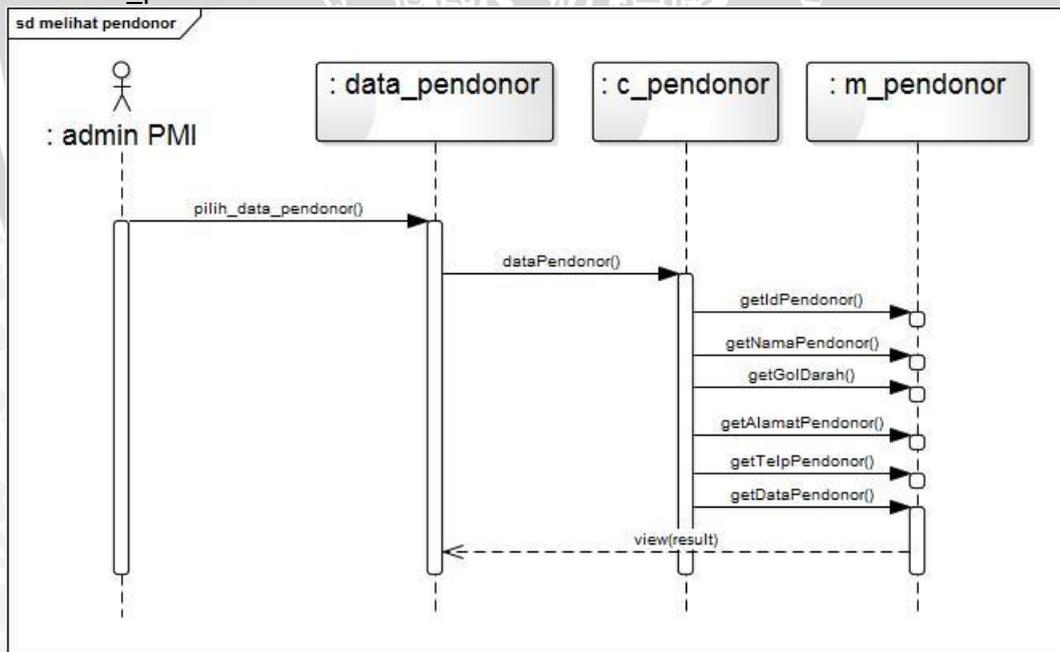
Gambar 5.9 merupakan diagram interaksi untuk proses memperbarui data pendonor. Aktor memilih untuk menampilkan halaman untuk mengisi data yang dibutuhkan, kemudian setelah data diisi objek c\_pendonor akan menerima fungsi updateData yang digunakan untuk mengirimkan data tersebut ke objek m\_pendonor



**Gambar 5.9** Sequence diagram memperbarui data pendonor

c. Melihat data pendonor

Gambar 5.10 merupakan diagram interaksi untuk proses melihat data pendonor. Aktor memilih untuk menampilkan halaman data pendonor, kemudian objek c\_pendonor akan mendapatkan data dari objek m\_pendonor.



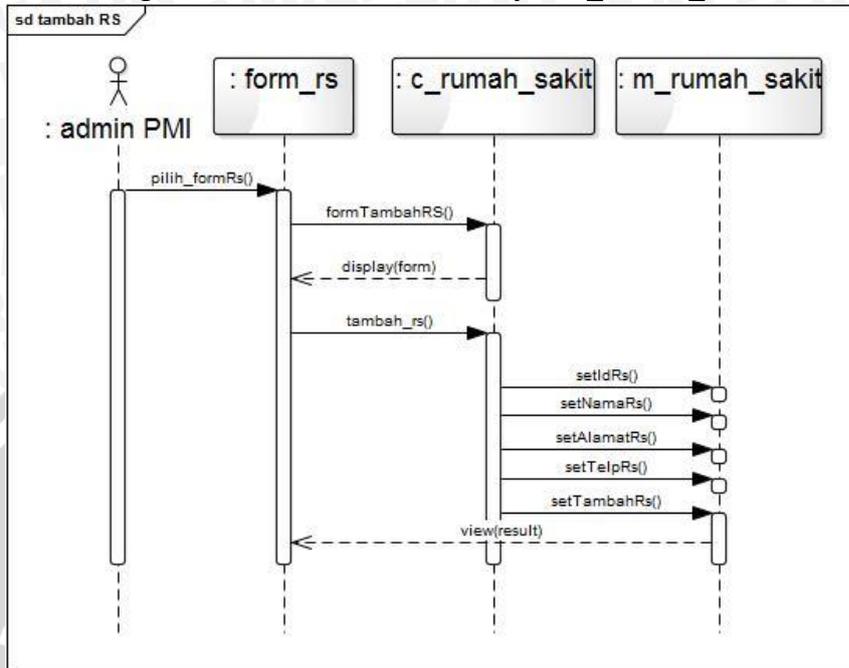
**Gambar 5.10** Sequence diagram melihat data pendonor

4. Mengelola data Rumah Sakit

a. Menambah Rumah Sakit

Gambar 5.11 merupakan diagram interaksi untuk proses menambah Rumah Sakit. Aktor memilih untuk menampilkan halaman untuk mengisi

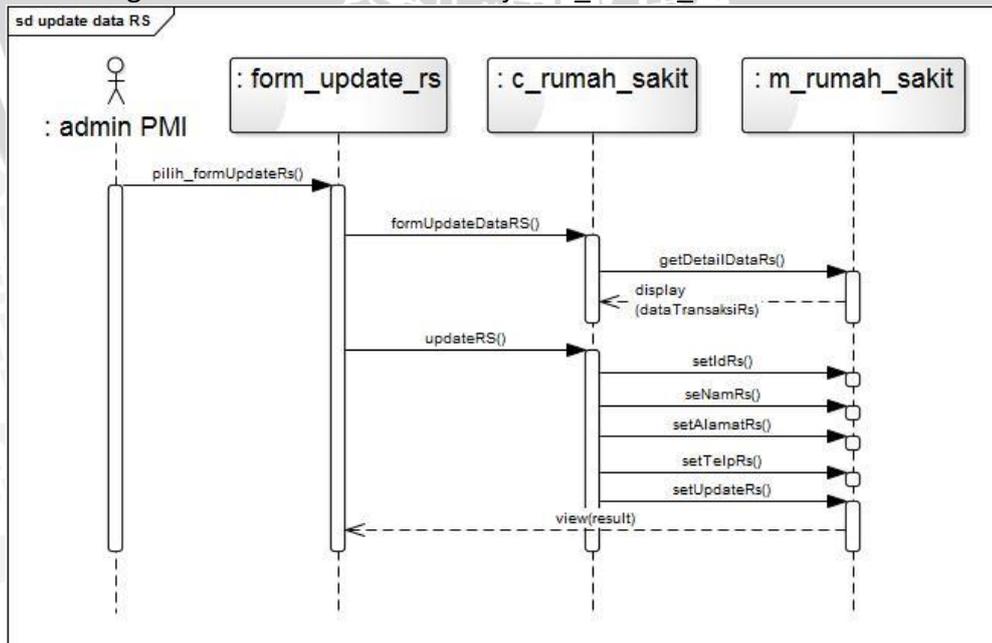
data yang dibutuhkan terkait data Rumah Sakit, kemudian setelah data diisi objek `c_rumah_sakit` akan menerima fungsi `tambah_rs` yang digunakan untuk mengirimkan data tersebut ke objek `m_rumah_sakit`.



**Gambar 5.11 Sequence diagram menambah data Rumah Sakit**

b. Memperbarui Rumah Sakit

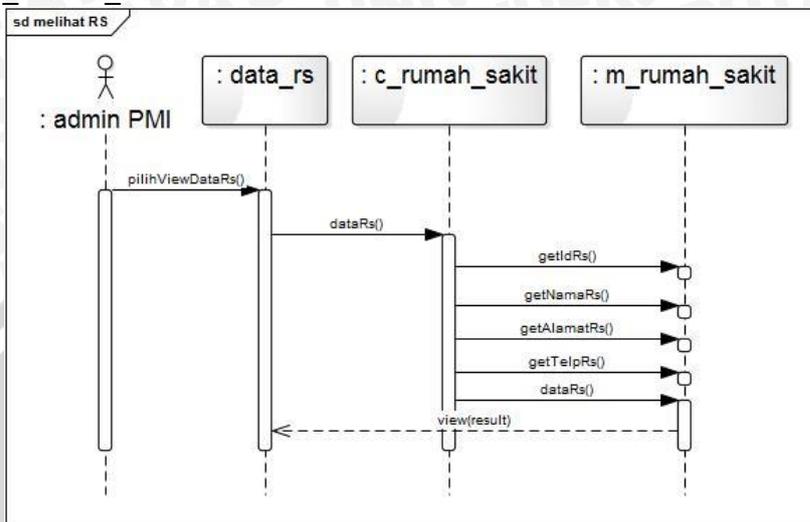
Gambar 5.12 merupakan diagram interaksi untuk proses memperbarui data Rumah Sakit. Aktor memilih untuk menampilkan halaman untuk mengisi data yang dibutuhkan, kemudian setelah data diisi objek `c_rumah_sakit` akan menerima fungsi `updateRS` yang digunakan untuk mengirimkan data tersebut ke objek `m_rumah_sakit`.



**Gambar 5.12 Sequence diagram memperbarui Rumah Sakit**

c. Melihat Rumah Sakit

Gambar 5.13 merupakan diagram interaksi untuk proses melihat data Rumah Sakit. Aktor memilih untuk menampilkan halaman data Rumah Sakit, kemudian objek `c_rumah_sakit` akan mendapatkan data dari objek `m_rumah_sakit`.

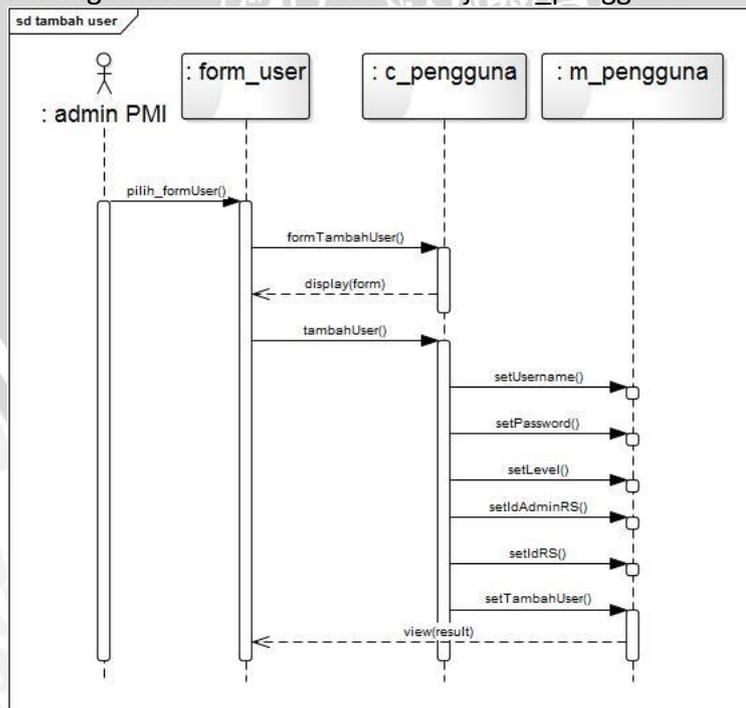


**Gambar 5.13 Sequence diagram melihat Rumah Sakit**

5. Mengelola user

a. Menambah user

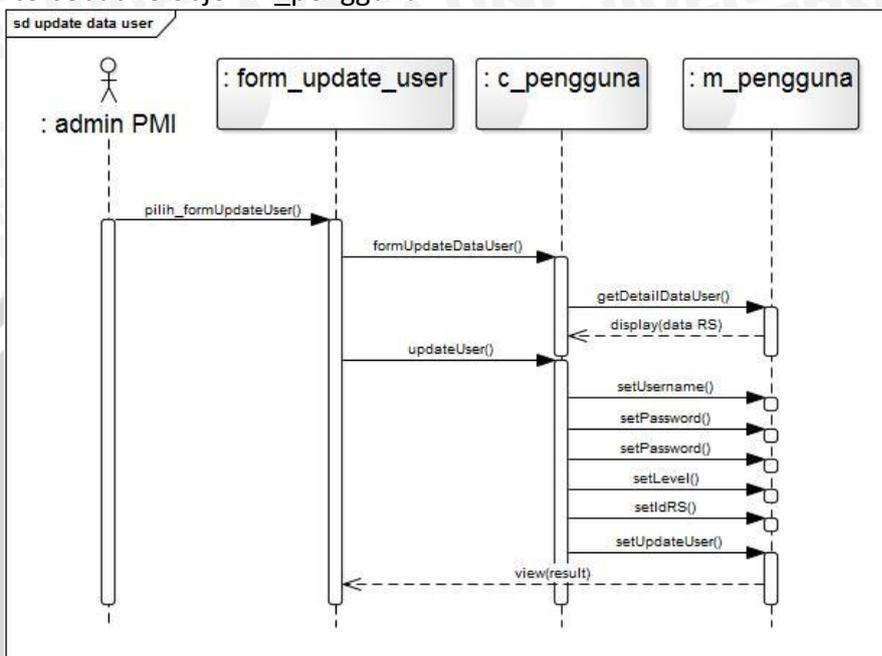
Gambar 5.14 merupakan diagram interaksi untuk proses menambah user yang dilakukan oleh PMI. Aktor memilih untuk menampilkan halaman untuk mengisi data yang dibutuhkan terkait user, kemudian setelah data diisi objek `c_pengguna` akan menerima fungsi `tambahUser` yang digunakan untuk mengirimkan data tersebut ke objek `m_pengguna`.



**Gambar 5.14 Sequence diagram menambah user**

b. Memperbarui user

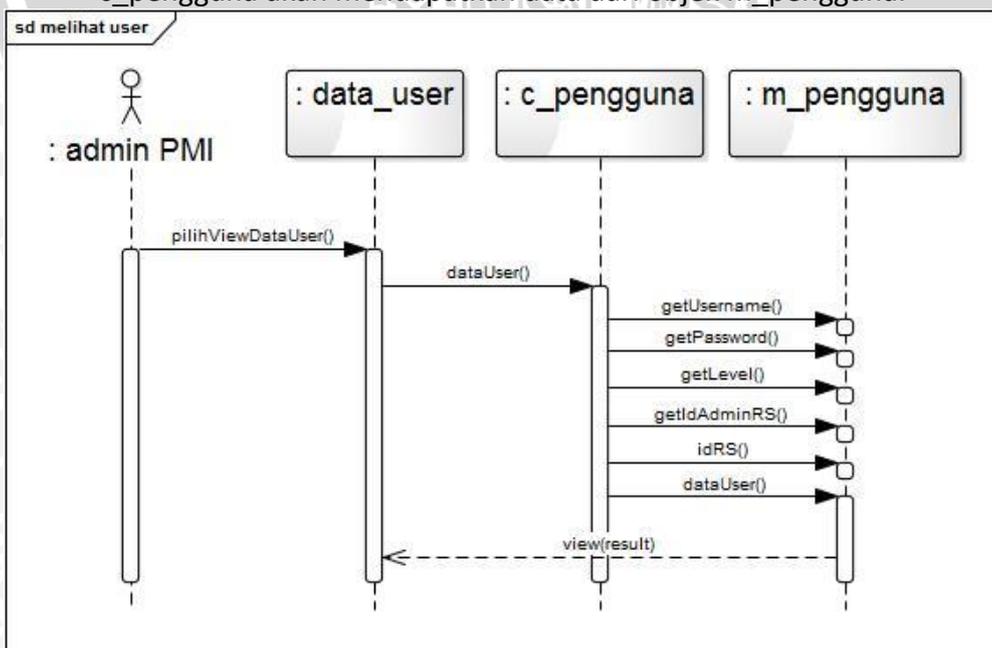
Gambar 5.15 merupakan diagram interaksi untuk proses memperbarui data user. Aktor memilih untuk menampilkan halaman untuk mengisi data yang dibutuhkan, kemudian setelah data diisi objek c\_pengguna akan menerima fungsi updateuser yang digunakan untuk mengirimkan data tersebut ke objek m\_pengguna.



Gambar 5.15 Sequence diagram memperbarui user

c. Melihat user

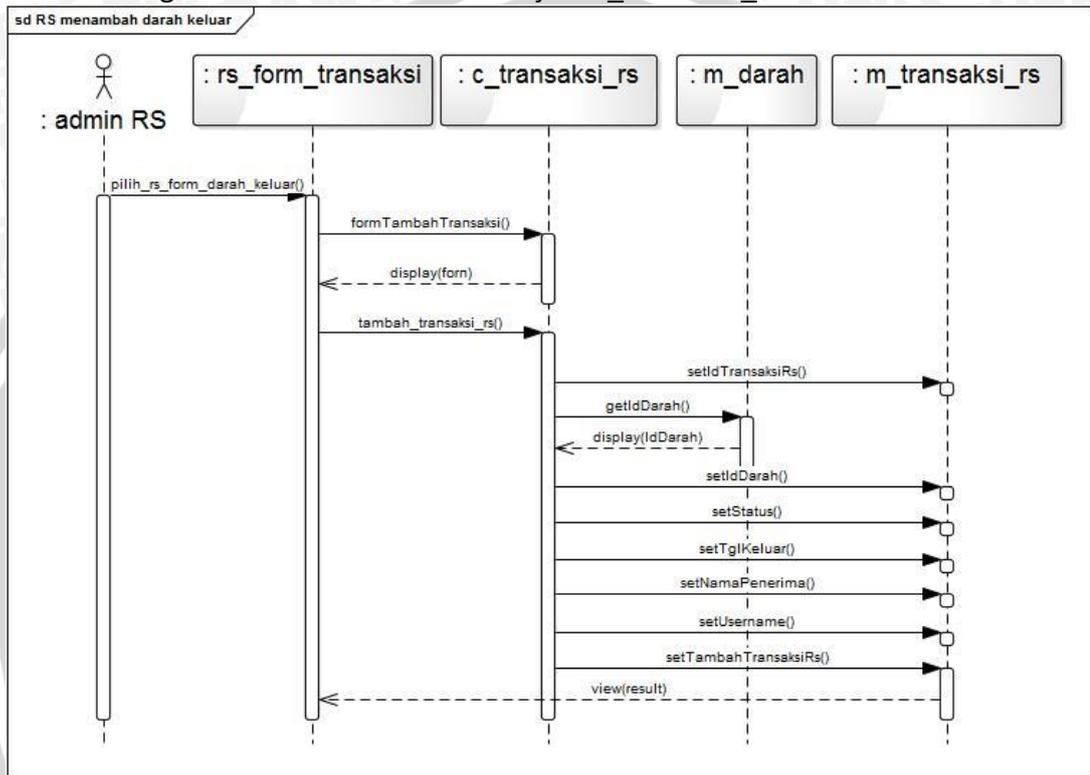
Gambar 5.16 merupakan diagram interaksi untuk proses melihat user. Aktor memilih untuk menampilkan halaman data user, kemudian objek c\_pengguna akan mendapatkan data dari objek m\_pengguna.



Gambar 5.16 Sequence diagram melihat user

- 6. Mengelola darah keluar (Rumah Sakit)
  - a. Menambah darah keluar

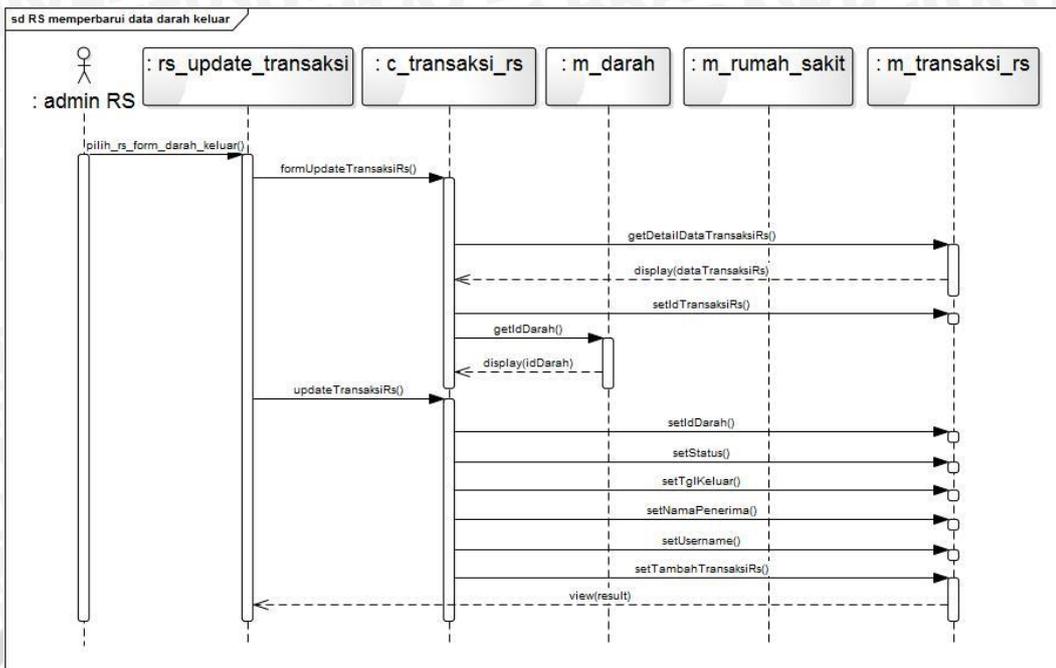
Gambar 5.17 merupakan diagram interaksi untuk proses menambah transaksi yang dilakukan oleh Rumah Sakit. Aktor memilih untuk menampilkan halaman untuk mengisi data yang dibutuhkan terkait transaksi Rumah Sakit, kemudian setelah data diisi objek `c_transaksi_rs` akan menerima fungsi `tambah_transaksi_rs` yang digunakan untuk mengirimkan data tersebut ke objek `m_transaksi_rs`.



**Gambar 5.17 Sequence diagram menambah darah keluar**

- b. Memperbarui darah keluar

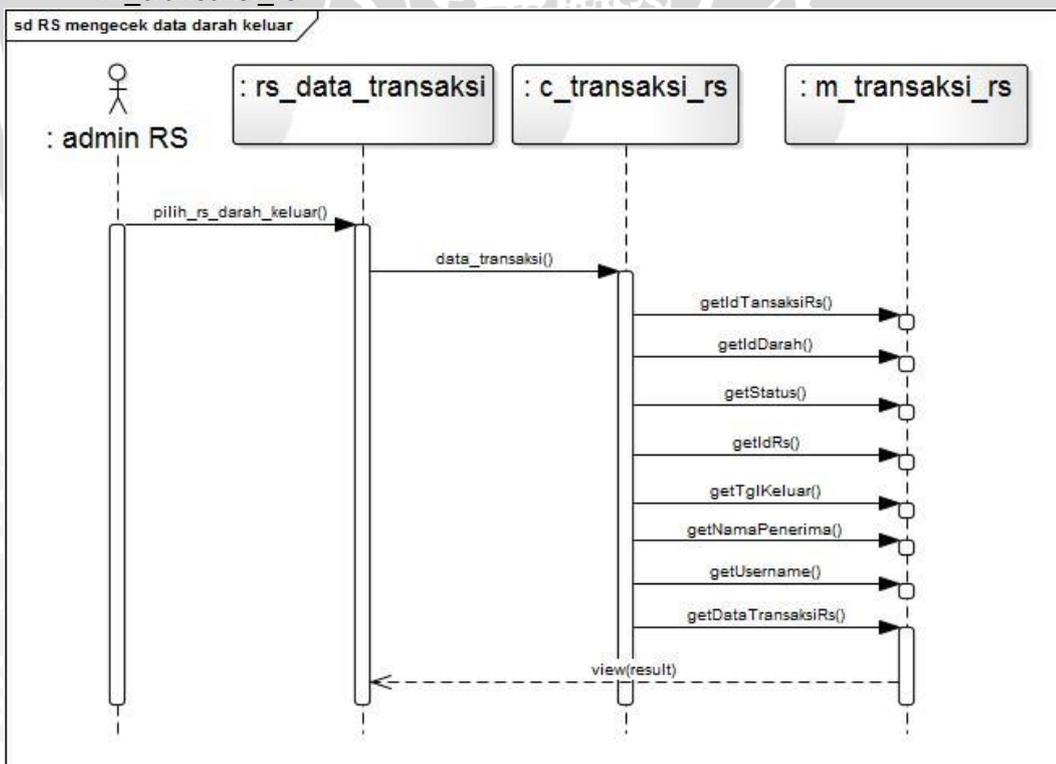
Gambar 5.18 merupakan diagram interaksi untuk proses memperbarui data transaksi yang dilakkan oleh Rumah Sakit. Aktor memilih untuk menampilkan halaman untuk mengisi data yang dibutuhkan, kemudian setelah data diisi objek `c_transaksi_rs` akan menerima fungsi `updateTransaksiRs` yang digunakan untuk mengirimkan data tersebut ke objek `m_transaksi_rs`.



Gambar 5.18 Sequence diagram memperbarui darah keluar

c. Melihat darah keluar

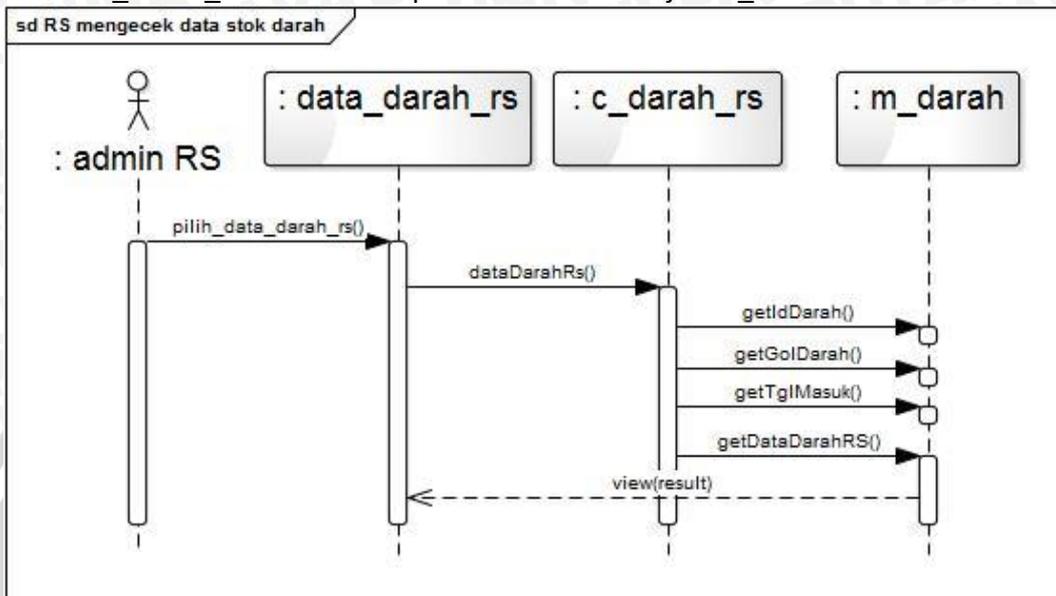
Gambar 5.19 merupakan diagram interaksi untuk proses melihat data transaksi yang dilakukan oleh Rumah Sakit. Aktor memilih untuk menampilkan halaman data darah keluar yang terjadi di Rumah Sakit, kemudian objek c\_transaksi\_rs akan mendapatkan data dari objek m\_transaksi\_rs.



Gambar 5.19 Sequence diagram melihat darah keluar

7. Mengecek data darah Rumah Sakit

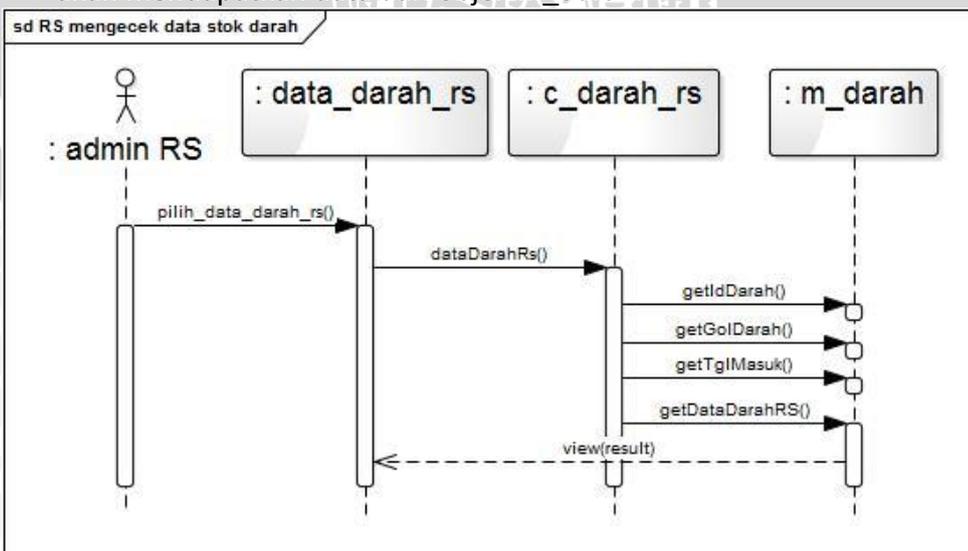
Gambar 5.20 merupakan diagram interaksi untuk proses Rumah Sakit melihat data stok darah yang ada di Rumah Sakit tersebut. Aktor memilih untuk menampilkan halaman data darah Rumah Sakit, kemudian objek `c_darah_rs` akan mendapatkan data dari objek `m_darah`.



**Gambar 5.20 Sequence diagram mengecek data darah Rumah Sakit**

8. Mengecek ketersediaan darah PMI dan Rumah Sakit

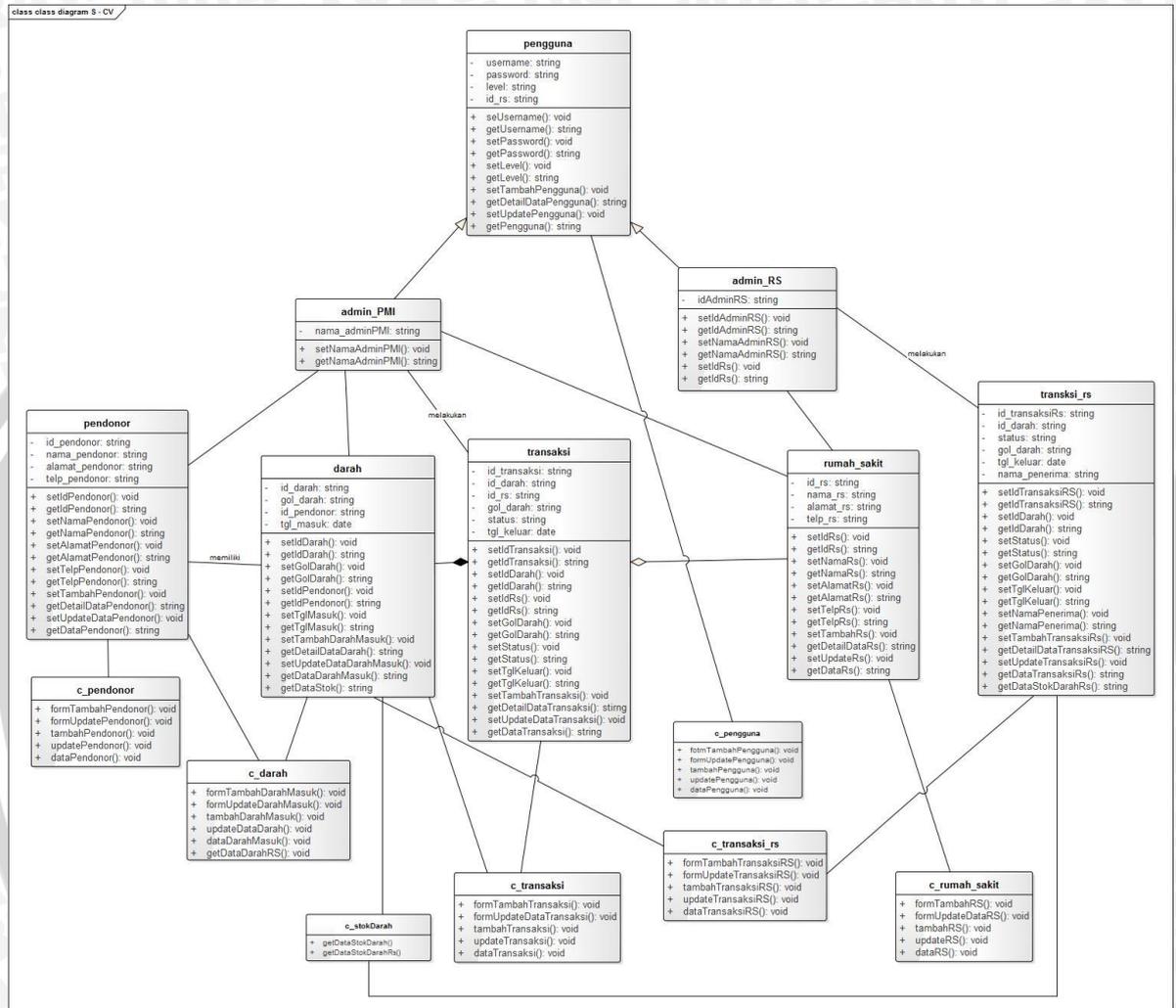
Gambar 5.21 merupakan diagram interaksi untuk proses melihat ketersediaan darah yang ada di PMI dan Rumah Sakit. Aktor memilih untuk menampilkan halaman ketersediaan data darah, kemudian objek `c_darah` akan mendapatkan data dari objek `m_darah`.



**Gambar 5.21 Sequence diagram mengecek ketersediaan darah PMI dan Rumah Sakit**

### 5.1.2 Diagram Kelas

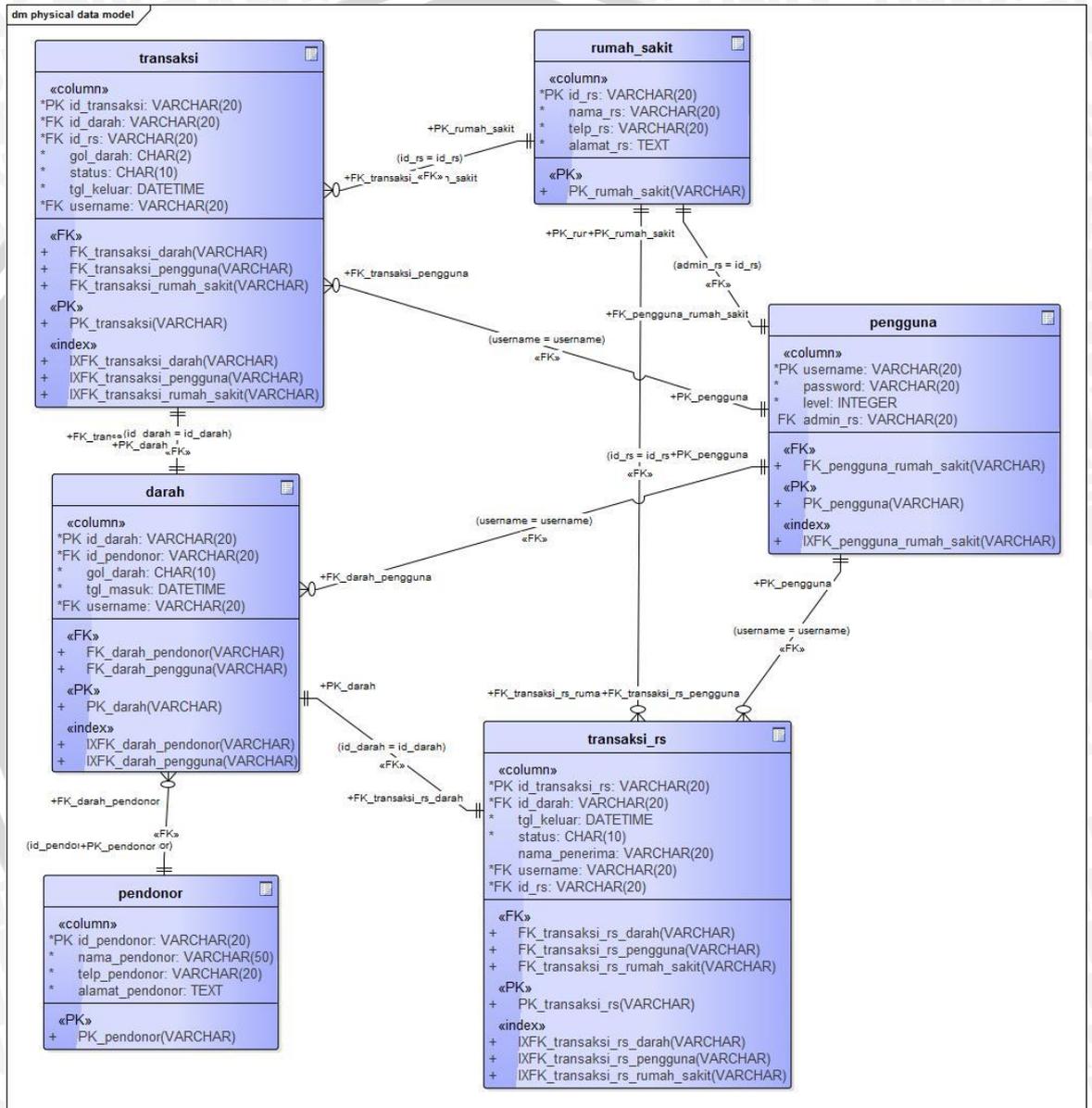
Diagram kelas atau *class diagram* digunakan untuk menggambarkan sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut dengan atribut dan metode atau operasi. Pada gambar 5.22 dijelaskan mengenai kelas diagram perancangan sistem.



Gambar 5.22 Kelas diagram

## 5.2 Perancangan basis data

Perancangan basis data digunakan sebagai gambaran mengenai bagaimana data yang diperlukan akan disimpan. Pada penelitian yang dilakukan perancangan basis data direpresentasikan dalam bentuk PDM (*Physical Data Model*). *Physical Data Model* menggambarkan hubungan antar data. Pada gambar 5.23 dijelaskan mengenai pemodelan tabel yang digunakan dalam perancangan sistem yang akan dikembangkan,



Gambar 5.23 Physical Data Model

Pada tabel 5.1 dijelaskan mengenai rincian pemodelan basis data setiap tabel atribut dan type data yang dimiliki.

**Tabel 5.1 Detail atribut perancangan basis data**

<b>Nama Tabel</b>	<b>Atribut</b>	<b>Tipe, Length</b>
pengguna	username	VARCHAR (20)
	password	VARCHAR (20)
	level	INTEGER
rumah_sakit	admin_rs	VARCHAR (20)
	id_rs	VARCHAR (20)
	nama_rs	VARCHAR (20)
	telp_rs	VARCHAR (20)
	alamat_rs	TEXT
	id_transaksi	VARCHAR (20)
	id_darah	VARCHAR (20)
transaksi	id_rs	VARCHAR (20)
	gol_darah	CHAR (2)
	status	CHAR (10)
	tgl_keluar	DATETIME
	username	VARCHAR (20)
darah	id_darah	VARCHAR (20)
	id_pendonor	VARCHAR (20)
	gol_darah	CHAR (10)
	tgl_masuk	DATETIME
pendonor	username	VARCHAR (20)
	id_pendonor	VARCHAR (20)
	nama_pendonor	VARCHAR (20)
	telp_pendonor	VARCHAR (20)
transaksi_rs	alamat_pendonor	TEXT
	id_transaksi_rs	VARCHAR (20)
	id_darah	VARCHAR (20)
	tgl_keluar	DATETIME
	status	CHAR (10)
	nama_penerima	VARCHAR (20)
	username	VARCHAR (20)
	id_rs	VARCHAR (20)

### 5.3 Perancangan antarmuka sistem

Perancangan antarmuka sistem ini digunakan untuk mempermudah dalam menjelaskan alur proses antarmuka dari sistem. Dalam sub bab ini diberikan beberapa contoh antarmuka sistem yang dirancang. Perancangan antarmuka pada sub bab ini merupakan beberapa contoh perancangan antarmuka beberapa fitur, untuk perancangan antarmuka lebih lengkap digambarkan pada lampiran A.

1. Halaman autentikasi

Gambar 5.24 merupakan antarmuka autentikasi. Berisikan kolom teks untuk memasukkan data berupa *username* dan *password*.



**Gambar 5.24 Antarmuka autentikasi**

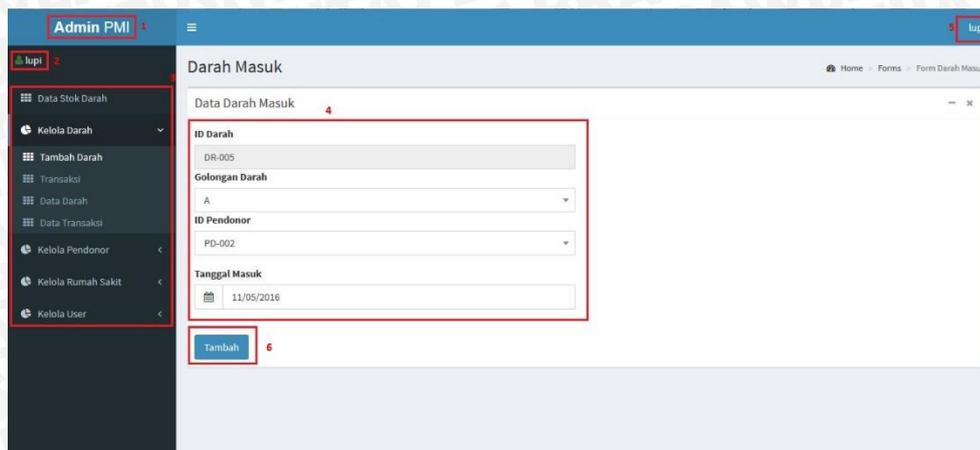
Tabel 5.2 merupakan keterangan antarmuka autentikasi.

**Tabel 5.2 Keterangan antarmuka autentikasi**

No	IdObject	Jenis	Nama	Keterangan
1	Autentikasi_Text1	Single text	Username	Menerima masukan username.
2	Autentikasi_Text2	Single text	Password	Menerima masukan password.
3	Autentikasi_Button1	Button	Sign Up	Button untuk melakukan submit data untuk autentikasi.

2. Halaman mengelola stok darah (menambah darah masuk)

Gambar 5.25 merupakan antarmuka untuk mengelola stok darah saat aktivitas menambah darah masuk.



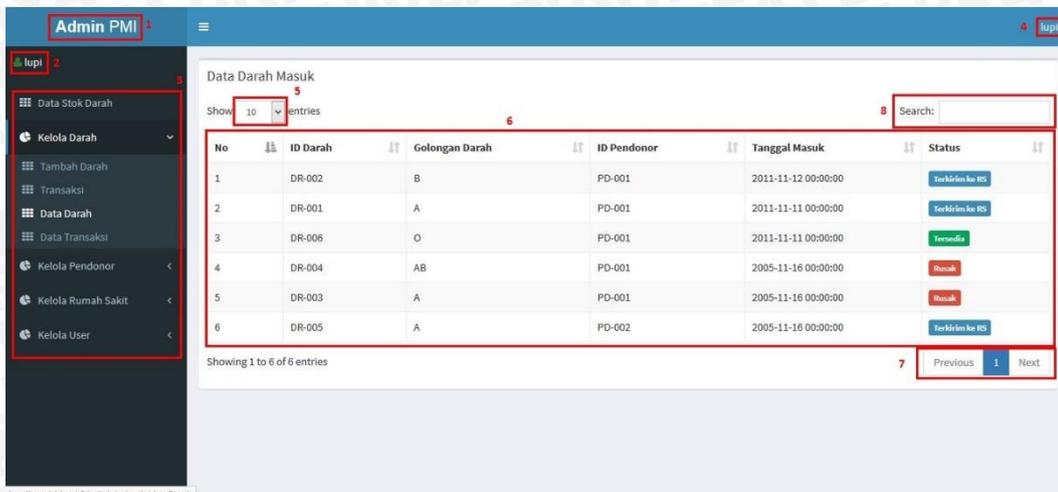
**Gambar 5.25** Antarmuka mengelola stok darah

Tabel 5.3 merupakan keterangan antarmuka mengelola stok darah saat aktivitas menambah darah masuk.

**Tabel 5.3** Keterangan antarmuka mengelola stok darah

No	IdObject	Jenis	Nama	Keterangan
1	DM_Button1	Button	Admin PMI	Menampilkan status saat berada di dalam sistem (admin PMI atau Admin Rumah Sakit).
2	DM_Button2	Button	lupi	Menampilkan nama profil sesuai dengan yang masuk kedalam sistem.
3	DM_Menu	Menu		Menampilkan menu pilihan untuk melakukan aksi.
4	DM_Kolom	Text, Combo Box		Menampilkan sebuah kolom yang harus diisi berupa ID darah, golongan darah, ID pendonor dan tanggal masuk.
5	DM_Button3	Button	lupi	Menampilkan nama profil sesuai dengan yang masuk kedalam sistem.
6	DM_button4	Button	Tambah	Tombol yang digunakan untuk melakukan submit data untuk menambah data darah masuk.

- Halaman menampilkan data darah masuk  
Gambar 5.26 merupakan antarmuka untuk menampilkan data darah masuk.



**Gambar 5.26 Antarmuka menampilkan data darah masuk**

Tabel 5.4 merupakan keterangan antarmuka untuk menampilkan data darah masuk.

**Tabel 5.4 Keterangan antarmuka menampilkan data darah masuk**

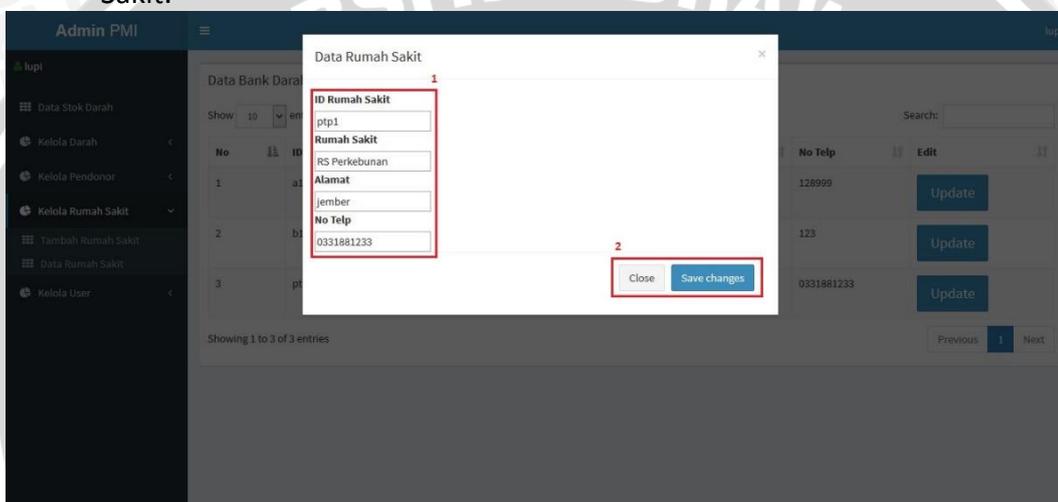
No	IdObject	Jenis	Nama	Keterangan
1	DDM_Button1	Button	Admin PMI	Menampilkan status saat berada di dalam sistem (admin PMI atau Admin Rumah Sakit).
2	DDM_Button2	Button	lupi	Menampilkan nama profil sesuai dengan yang masuk kedalam sistem.
3	DDM_Menu	Menu		Menampilkan menu pilihan untuk melakukan aksi.
4	DDM_Button3	Button	lupi	Menampilkan nama profil sesuai dengan yang masuk kedalam sistem.
5	DDM_Box	Combo Box	Show entries	Mengatur jumlah data yang muncul dalam satu halaman.
6	DDM_Tabel	Tabel		Menampilkan seluruh data darah (tersedia, rusak dan terkirim ke RS).

**Tabel 5.4 Keterangan antarmuka menampilkan data darah masuk (lanjutan)**

No	IdObject	Jenis	Nama	Keterangan
7	DDM_Button4	Button	Previous – Next	Digunakan untuk mengatur halaman sebelumnya maupun halaman selanjutnya.
8	DDM_Text	Single Text	Search	Untuk melakukan pencarian dengan menggunakan kata kunci.

4. Halaman memperbarui data Rumah Sakit

Gambar 5.27 merupakan antarmuka untuk memperbarui data Rumah Sakit.



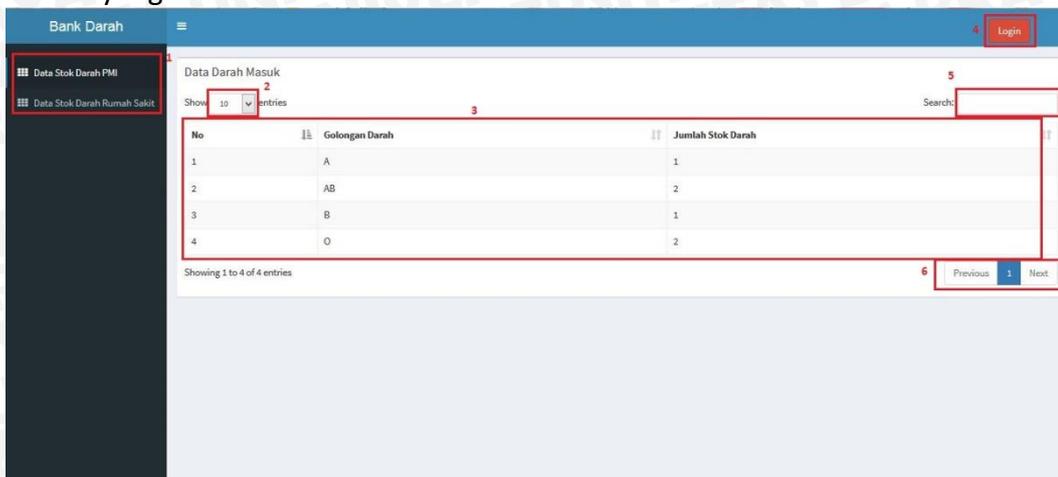
**Gambar 5.27 Antarmuka memperbarui data Rumah Sakit**

Tabel 5.5 merupakan keterangan antarmuka untuk memperbarui data Rumah Sakit.

**Tabel 5.5 Keterangan antarmuka memperbarui data Rumah Sakit**

No	IdObject	Jenis	Nama	Keterangan
1	URS_Text1	Text	ID Rumah Sakit, Rumah Sakit, Alamat, No. Telp	Menerima masukan yang digunakan untuk memperbarui data Rumah Sakit.
2	URS_Button1	Button	Close, save changes	Digunakan untuk submit data yang digunakan untuk memperbarui data Rumah Sakit.

5. Mengecek ketersediaan darah  
Gambar 5.28 merupakan antarmuka untuk mengecek ketersediaan darah yang ada di PMI.



**Gambar 5.28 Antarmuka mengecek ketersediaan darah PMI**

Tabel 5.6 merupakan keterangan untuk antarmuka mengecek ketersediaan darah di PMI.

**Tabel 5.6 Keterangan antarmuka mengecek ketersediaan darah PMI**

No	IdObject	Jenis	Nama	Keterangan
1	DPMI_Menu	Menu		Menampilkan menu pilihan untuk melakukan aksi.
2	DPMI_Box	Combo Box	Show entries	Mengatur jumlah data yang muncul dalam satu halaman.
3	DPMI_Tabel	Tabel		Menampilkan seluruh data darah beserta jumlah stok darah yang tersedia.
4	DPMI_Button1	Button	Login	Button yang digunakan untuk masuk ke halaman autentikasi
5	DPMI_Text	Single Text	Search	Untuk melakukan pencarian dengan menggunakan kata kunci.
6	DPMI_Button2	Button	Previous – Next	Digunakan untuk mengatur halaman sebelumnya maupun halaman selanjutnya.

## BAB 6 EVALUASI DAN ANALISIS

Bab ini membahas mengenai evaluasi dari analisis dan perancangan sistem informasi bank darah. Evaluasi dilakukan dengan 2 cara, yaitu tinjauan (review) yang digunakan untuk pengecekan keruntutan antara persyaratan dengan model perancangan (*traceability*) dan pengecekan kesesuaian antara alur yang disediakan sistem dengan alur yang dilakukan pengguna (*walkthrough*), cara yang kedua yaitu evaluasi prototipe dengan menggunakan kuisioner WEBUSE sebagai acuan untuk kategori navigasi dan antarmuka. Hasil evaluasi pengguna dapat dilihat pada lampiran B.

### 6.1 Tinjauan (review)

#### 6.1.1 Traceability

Tabel 6.1 merupakan beberapa pertanyaan untuk daftar pengecekan terkait keruntutan antara persyaratan hingga ke model perancangan.

**Tabel 6.1 Daftar pertanyaan traceability**

Traceability			
		✓	Keterangan
1	Apakah setiap persyaratan memiliki kode yang unik sehingga dapat digunakan untuk tujuan traceability?		
2	Apakah setiap persyaratan dapat dilacak ke dokumen atau pengguna?		
3	Apakah setiap persyaratan memiliki kemampuan <i>traceability</i> ke dalam kebutuhan dan model diagram?		
4	Apakah fitur pada Sistem Informasi Bank Darah sudah memenuhi seluruh kebutuhan pemangku kepentingan?		
5	Apakah persyaratan Sistem Informasi Bank Darah dapat diruntukan kedalam fitur?		

#### 6.1.2 Walkthrough

Tabel 6.2 merupakan daftar beberapa tugas yang akan dijalankan oleh calon pengguna sistem. Digunakan untuk mengecek antara alur yang disediakan sistem dengan alur yang dilakukan oleh pengguna.

**Tabel 6.2 Daftar tugas kesesuaian harapan pengguna**

No	Tugas	Tujuan	Kriteria Keberhasilan
1	Mengecek ketersediaan darah PMI dan Rumah Sakit	Mengidentifikasi kemudahan untuk melakukan pengecekan ketersediaan darah.	Menampilkan halaman ketersediaan darah PMI dan Rumah Sakit.
2	Mengelola stok darah (menambah data stok darah)	Mengidentifikasi kemudahan untuk melakukan menambah data stok darah atau menambah darah masuk.	Menambahkan data dan menyimpan data darah masuk.

**Tabel 6.2 Daftar tugas kesesuaian harapan pengguna (lanjutan)**

No	Tugas	Tujuan	Kriteria Keberhasilan
3	Mengelola data stok darah (menambah data darah keluar)	Mengidentifikasi kemudahan untuk melakukan menambah data darah keluar.	Menambahkan data darah keluar dan menyimpan data darah keluar.
4	Mengelola data Rumah Sakit (memperbarui data Rumah Sakit)	Mengidentifikasi kemudahan untuk melakukan memperbarui data Rumah Sakit.	Memperbarui data Rumah Sakit dan menyimpan data Rumah Sakit terbaru.
5	Mengelola data pendonor (menambah data pendonor)	Mengidentifikasi kemudahan untuk menambahkan data pendonor.	Menambahkan data dan menyimpan data pendonor.

Tabel 6.3 merupakan detail alur yang disediakan sistem sesuai dengan yang telah dispesifikasikan pada basic flow.

**Tabel 6.3 Detail alur sistem dan harapan pengguna**

<b>Use Case</b>	Mengecek ketersediaan darah PMI dan Rumah Sakit	
<b>Langkah-Langkah Pengguna</b>	<b>Langkah-Langkah Basic Flow</b>	<b>Ketersesuaian</b>
	<ol style="list-style-type: none"> <li>1. Aktor masyarakat membuka sistem.</li> <li>2. Aktor memilih untuk mengecek ketersediaan darah.</li> <li>3. Sistem menampilkan pilihan tempat ketersediaan darah.</li> <li>4. Aktor memilih tempat ketersediaan darah (PMI dan Rumah Sakit).</li> </ol>	
<b>Use Case</b>	Mengelola data stok darah (menambah data stok darah)	
<b>Langkah-Langkah Pengguna</b>	<b>Langkah-Langkah Basic Flow</b>	<b>Ketersesuaian</b>
	<ol style="list-style-type: none"> <li>1. Admin membuka sistem.</li> <li>2. Include use case autentikasi.</li> <li>3. Aktor masuk kedalam sistem dan mengakses untuk mengelola stok darah.</li> <li>4. Sistem menampilkan pilihan untuk melakukan mengelola stok darah berupa menambah data darah masuk, menambah data darah keluar dan menampilkan data darah yang memiliki fungsi memperbarui data darah.</li> <li>5. Admin memilih untuk menambah data darah masuk.</li> <li>6. Sistem meminta Admin untuk memasukkan data yang dibutuhkan terkait data darah masuk.</li> <li>7. Memilih simpan data.</li> </ol>	

**Tabel 6.3 Detail alur sistem dan harapan pengguna (lanjutan)**

Use Case	Mengelola data stok darah (menambah data darah keluar)	
Langkah-Langkah Pengguna	Langkah-Langkah Basic Flow	Ketersesuaian
	<ol style="list-style-type: none"> <li>1. Aktor membuka sistem.</li> <li>2. Include use case autentikasi.</li> <li>3. Aktor masuk kedalam sistem dan mengakses untuk mengelola stok darah.</li> <li>4. Sistem menampilkan halaman untuk melakukan mengelola stok darah berupa menambah data darah masuk, menambah data darah keluar dan menampilkan data darah yang memiliki fungsi memperbaiki data darah.</li> <li>5. Aktor memilih untuk menambah darah keluar.</li> <li>6. Sistem menampilkan halaman untuk mengisi data darah keluar.</li> <li>7. Aktor mengisi data yang diminta sistem terkait data darah keluar.</li> <li>8. Aktor memilih simpan data.</li> </ol>	
Use Case	Mengelola data Rumah Sakit (memperbarui data Rumah Sakit)	
Langkah-Langkah Pengguna	Langkah-Langkah Basic Flow	Ketersesuaian
	<ol style="list-style-type: none"> <li>1. Aktor membuka sistem.</li> <li>2. Include use case autentikasi.</li> <li>3. Aktor masuk kedalam sistem dan mengakses untuk mengelola data Rumah Sakit.</li> <li>4. Sistem menampilkan halaman untuk melakukan mengelola Rumah Sakit berupa menambah data Rumah Sakit dan menampilkan data Rumah Sakit yang memiliki fungsi memperbaiki Rumah Sakit.</li> <li>5. Aktor memilih untuk memperbaiki data Rumah Sakit.</li> <li>6. Aktor memilih data yang akan diperbarui.</li> <li>7. Sistem menampilkan detail data yang akan diperbarui.</li> <li>8. Aktor mengisi data Rumah Sakit terbaru sesuai yang dibutuhkan sistem.</li> <li>9. Aktor memilih simpan data.</li> </ol>	

**Tabel 6.3 Detail alur sistem dan harapan pengguna (lanjutan)**

<b>Use Case</b>	Mengelola data pendonor (menambah data pendonor)	
<b>Langkah-Langkah Pengguna</b>	<b>Langkah-Langkah Basic Flow</b>	<b>Ketersesuaian</b>
	<ol style="list-style-type: none"> <li>1. Aktor membuka sistem.</li> <li>2. Include use case autentikasi.</li> <li>3. Aktor masuk kedalam sistem dan mengakses untuk mengelola data pendonor.</li> <li>4. Sistem menampilkan halaman untuk melakukan mengelola stok pendonor berupa menambah data pendonor dan menampilkan data pendonor yang memiliki fungsi untuk memperbarui data pendonor.</li> <li>5. Aktor memilih untuk menambah data pendonor.</li> <li>6. Sistem menampilkan halaman untuk mengisi data pendonor.</li> <li>7. Aktor mengisi data terkait data pendonor. Aktor memilih simpan data.</li> </ol>	

Tabel 6.4 merupakan beberapa daftar pertanyaan terkait alur dari sistem dengan harapan alur pengguna.

**Tabel 6.4 Daftar pertanyaan harapan alur pengguna**

<b>Harapan Alur Pengguna</b>				
<b>Pertanyaan</b>	<b>Setuju</b>	<b>Netral</b>	<b>Tidak Setuju</b>	<b>Keterangan</b>
Respon sistem terhadap aksi yang dilakukan sudah sesuai dengan harapan				
Alur penggunaan sistem sudah sesuai dengan harapan pengguna				
<b>Pernyataan / Masukan dan Saran</b>				

## 6.2 Evaluasi prototipe

Tabel 6.5 merupakan daftar pertanyaan yang mengacu pada kuisioner WEBUSE untuk kategori navigasi dan antarmuka. Kuisioner tersebut diisi berdasarkan tugas yang telah dilakukan sebelumnya.

**Tabel 6.5 Daftar pertanyaan tinjauan navigasi dan antamuka pengguna**

Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
<b>Navigasi</b>				
Saya dapat dengan mudah mengetahui posisi saya pada sistem				
Sistem menyediakan navigasi menu dan <i>link</i> yang berguna untuk mendapatkan informasi				
Sistem mudah digunakan dengan menggunakan tombol kembali atau link				
Sistem tidak membuka banyak jendela browser ketika digunakan				
Peletakan menu atau link dapat dengan mudah dikenali				
Navigasi menu pada sistem sudah menunjukkan fungsi dengan jelas				
<b>Antarmuka pengguna</b>				
Perancangan antar muka sistem mudah untuk dipelajari				
Sistem tidak mengandung fitur yang membutuhkan scroll atau animasi yang berulang				
Sistem memiliki antar muka yang konsisten				

### 6.3 Hasil evaluasi

#### 6.3.1 Hasil tinjauan pengecekan traceability

Tabel 6.6 merupakan hasil tinjauan untuk pengecekan *traceability*. Pengecekan *traceability* dilakukan oleh pengembang atau yang melakukan analisis dan perancangan sistem.

**Tabel 6.6 Hasil tinjauan (traceability)**

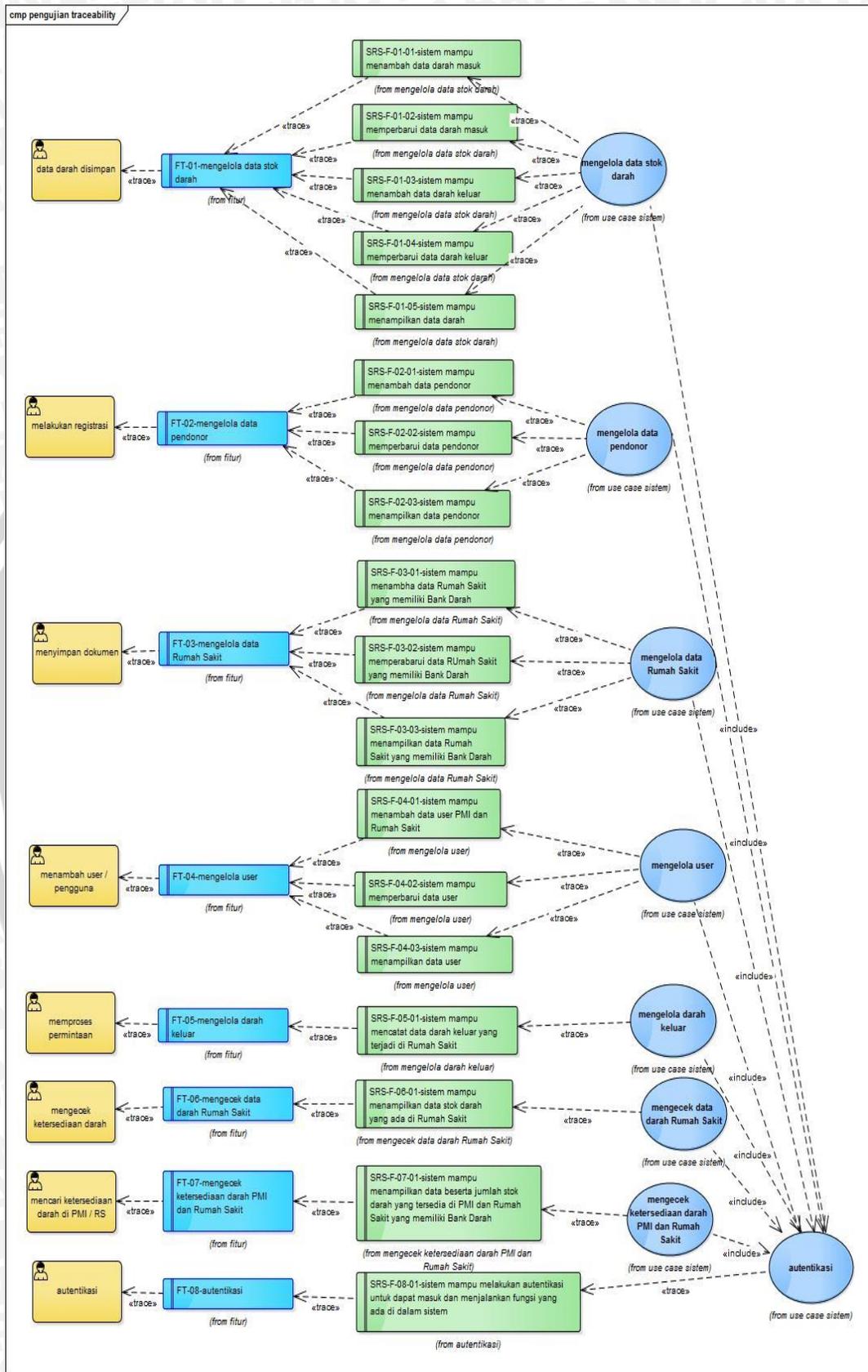
Traceability				
		✓	Keterangan	
1	Apakah setiap persyaratan memiliki kode yang unik sehingga dapat digunakan untuk tujuan traceability?	✓	Setiap persyaratan memiliki kode unik (contoh: SRS-F-XX). Begitu juga kode yang digunakan untuk use case (contoh: UC-XX).	
2	Apakah setiap persyaratan dapat dilacak ke dokumen atau pengguna?	✓	Persyaratan dapat dilacak keruntutannya, mulai dari persyaratan fungsional dalam tahapan analisis hingga diagram perancangan.	
3	Apakah setiap persyaratan memiliki kemampuan <i>traceability</i> ke dalam kebutuhan dan model diagram?	✓	Setiap persyaratan yang di definisikan sesuai dengan kebutuhan dan telah di definisikan dalam diagram perancangan.	

Tabel 6.6 Hasil tinjauan (traceability) lanjutan

Traceability			
		✓	Keterangan
4	Apakah fitur pada Sistem Informasi Bank Darah sudah memenuhi seluruh kebutuhan pemangku kepentingan?	✓	Setiap fitur telah memenuhi kebutuhan pemangku kepentingan.
5	Apakah persyaratan Sistem Informasi Bank Darah dapat dirunutkan kedalam fitur?	✓	Seluruh persyaratan telah dirunutkan ke dalam fitur pada aplikasi yang akan dibangun.

Gambar 6.1 merupakan peta *traceability* yang menggambarkan keruntutan mulai dari persyaratan sistem hingga ke model perancangan. Peta *traceability* digunakan untuk menelusuri apakah elemen-elemen model atau produk dari setiap aktivitas rekayasa persyaratan, analisis dan desain telah memiliki keterkaitan satu sama lain.





Gambar 5.1 Traceability



Tabel 6.7 merupakan tabel mengenai *traceability*, untuk memastikan tahap perancangan dapat dirunutkan ke tahap persyaratan.

Aktivitas BPMN	Fitur	Kode dasar persyaratan fungsional	Nama use case	Diagram aktivitas	Diagram sequence	Antarmuka
Data darah disimpan	FT-01	SRS-F-01	Mengelola data stok darah	Gambar 4.12 Gambar 4.13	Gambar 5.2 Gambar 5.3 Gambar 5.4 Gambar 5.5 Gambar 5.6 Gambar 5.7	Gambar 7.4 Gambar 7.5 Gambar 7.6 Gambar 7.7 Gambar 7.8 Gambar 7.9
Registrasi pendonor	FT-02	SRS-F-02	Mengelola data pendonor	Gambar 4.14	Gambar 5.8 Gambar 5.9 Gambar 5.10	Gambar 7.13 Gambar 7.14
Menyimpan dokumen	FT-03	SRS-F-03	Mengelola data Rumah Sakit	Gambar 4.15	Gambar 5.11 Gambar 5.12 Gambar 5.13	Gambar 7.10 Gambar 7.11 Gambar 7.12
Menambah user / pengguna	FT-04	SRS-F-04	Mengelola user	Gambar 4.16	Gambar 5.14 Gambar 5.15 Gambar 5.16	Gambar 7.18 Gambar 7.19
Memproses permintaan	FT-05	SRS-F-05	Mengelola darah keluar	Gambar 4.17	Gambar 5.17 Gambar 5.18 Gambar 5.19	Gambar 7.16 Gambar 7.17
Mengecek ketersediaan darah	FT-06	SRS-F-06	Mengecek data darah Rumah Sakit	Gambar 4.18	Gambar 5.20	Gambar 7.15
Mencari ketersediaan darah di PMI / Rumah Sakit	FT-07	SRS-F-07	Mengecek ketersediaan darah PMI dan Rumah Sakit	Gambar 4.19	Gambar 5.21	Gambar 7.1 Gambar 7.2
Autentikasi	FT-08	SRS-F-08	Autentikasi	Gambar 4.20	Gambar 5.1	Gambar 7.3

### 6.3.2 Hasil tinjauan walkthrough

Tabel 6.8 merupakan ringkasan mengenai kesesuaian antara alur yang disediakan sistem dengan alur yang dilakukan pengguna.

### 6.3.3 Hasil evaluasi prototipe

Sub bab ini menjelaskan terkait hasil evaluasi antarmuka yang dilakukan terhadap 5 responden. Tabel 6.9 merupakan hasil dari kuisisioner terkait navigasi dan antarmuka pengguna sesuai dengan tugas yang dilakukan.

## 6.4 Analisis hasil evaluasi

Pada sub bab ini dijelaskan mengenai analisis hasil evaluasi yang dilakukan berupa analisis dari tinjauan *traceability* dan *walkthrough* dan tinjauan prototipe dengan menggunakan WEBUSE.

### 6.4.1 Analisis hasil tinjauan

Hasil untuk evaluasi tinjauan menggunakan *traceability* menunjukkan bahwa seluruh persyaratan dapat dirunutkan sampai ke model perancangan, diantaranya dengan menggunakan kode yang unik sehingga keruntutan dapat dilacak. Untuk hasil dengan menggunakan *walkthrough* menunjukkan hasil bahwa alur yang disediakan sistem sesuai dengan harapan alur pengguna dengan adanya beberapa masukan dan saran yang diberikan oleh pengguna yaitu adalah alur untuk menambah data darah keluar harus dilakukan satu persatu untuk data darah yang dikeluarkan, saran yang diberikan agar data yang dimasukkan dapat lebih dari satu data darah yang dimasukkan dalam satu kali tranaksi yang dilakukan.

### 6.4.2 Analisis hasil prototipe

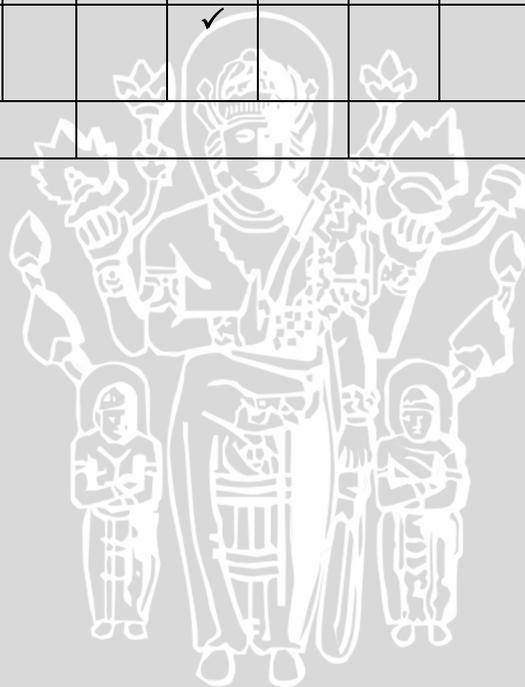
Analisis hasil dari evaluasi antarmuka yang dilakukan kepada 5 responden, hasil yang didapatkan sebagian besar responden menunjukkan setuju dan puas dengan antarmuka yang disediakan sistem. Namun terdapat beberapa yang harus diperbaiki yaitu penggunaan warna dan tanda link aktif untuk halaman yang sedang dibuka dan tampilan halaman awal kurang menarik, untuk pengembangan selanjutnya diharapkan untuk halaman terdapat informasi mengenai alur permintaan darah dan informasi terkait pelayanan pengambilan darah.

## 6.5 Analisis hasil evaluasi keseluruhan

Masukan dan saran dijelaskan pada tabel 6.10. Masukan dan saran dibuat berdasarkan kesimpulan keseluruhan dari pertanyaan kesesuaian alur pengguna dan pertanyaan antarmuka dan navigasi.

Tabel 6.8 Daftar hasil harapan alur pengguna

Harapan Alur Pengguna																
Pertanyaan	Responden 1			Responden 2			Responden 3			Responden 4			Responden 5			
	S	N	TS													
Respon sistem terhadap aksi yang dilakukan sudah sesuai dengan harapan	✓			✓			✓			✓			✓			
Alur penggunaan sistem sudah sesuai dengan harapan pengguna	✓				✓				✓	✓			✓			
Pernyataan / Masukan dan Saran																



Tabel 6.9 Daftar hasil tinjauan navigasi dan antarmuka pengguna

Navigasi																					
Responden 1					Responden 2				Responden 3				Responden 4				Responden 5				
Pertanyaan	S	N	TS	Ket	S	N	TS	Ket	S	N	TS	Ket	S	N	TS	Ket	S	N	TS	Ket	
Saya dapat dengan mudah mengetahui posisi saya pada sistem		✓			✓				✓				✓				✓				
Sistem menyediakan navigasi menu dan <i>link</i> yang berguna untuk mendapatkan informasi	✓				✓				✓				✓				✓				
Sistem mudah digunakan dengan menggunakan tombol kembali atau link	✓				✓						✓		✓					✓			
Sistem tidak membuka banyak jendela browser ketika digunakan	✓				✓				✓				✓				✓				
Peletakan menu atau link dapat dengan mudah dikenali	✓				✓					✓			✓						✓		
Navigasi menu pada sistem sudah menunjukkan fungsi dengan jelas	✓				✓					✓			✓						✓		
Antarmuka																					
Perancangan antar muka sistem mudah untuk dipelajari	✓				✓					✓			✓				✓				
Sistem tidak mengandung fitur yang membutuhkan scroll atau animasi yang berulang	✓				✓					✓			✓				✓				
Sistem memiliki antar muka yang konsisten	✓				✓					✓			✓				✓				

Tabel 6.10 Masukan dan saran

Masukan dan saran	
<b>Responden 1</b>	Sistem sudah sesuai dengan alur yang diharapkan pengguna. Untuk kriteria navigasi perlu ditambah letak menu aktif disamping untuk kemudahan pengguna. Untuk antarmuka pengguna desain yang diberikan sudah konsisten dan mudah untuk dipahami.
<b>Responden 2</b>	Untuk antarmuka sudah bagus, namun perlu diperbaiki untuk kejelasan teks atau font untuk diperjelas dan diperbesar. Untuk alur masih belum begitu jelas untuk melakukan transaksi. Mungkin bisa dilakukan untuk satu kali transaksi, bisa banyak data darah yang keluar. Untuk melakukan update data mungkin bisa diberi warna yang berbeda untuk data yang bisa diperbarui dan tidak bisa diperbarui.
<b>Responden 3</b>	Alur yang disediakan sistem sesuai dengan yang diharapkan, namun untuk penamaan menu atau fitur yang masih membingungkan, mungkin bisa lebih diperbaiki untuk penamaannya.
<b>Responden 4</b>	Alur yang disediakan sistem sudah sesuai dengan harapan pengguna.
<b>Responden 5</b>	Alur yang disediakan sistem sudah sesuai dengan harapan, namun perlu penjelasan atau menggunakan bahasa yang mudah dipahami dalam penamaan menu atau fitur.

## BAB 7 PENUTUP

Pada bab ini berisi kesimpulan dan saran dari penelitian yang dilakukan. Kesimpulan berisi analisis hasil pengujian untuk menjawab rumusan masalah. Saran ditujukan bagi pembaca yang ingin melanjutkan penelitian.

### 7.1 Kesimpulan

Berdasarkan hasil analisis, perancangan dan pengujian, maka diambil kesimpulan sebagai berikut:

1. Proses bisnis yang terjadi di perusahaan saat ini meliputi proses bisnis pencatatan data darah masuk dan keluar (menggunakan Microsoft Excel), pencatatan informasi data pendonor disimpan kedalam database, pencatatan data dan informasi Rumah Sakit yang memiliki Bank Darah dengan menggunakan dokumen MOU antara PMI dengan Rumah Sakit, proses untuk mendapatkan informasi tempat ketersediaan darah yaitu dengan masyarakat mencari dengan mendatangi langsung ke tempat tersebut. Proses bisnis usulan meliputi seluruh proses bisnis saat ini dengan perubahan aktivitas yang terjadi, yaitu pengelolaan data stok darah, pendonor dan Rumah Sakit saat ini dilakukan dengan menggunakan sistem, informasi mengenai ketersediaan darah di PMI dan Rumah Sakit.
2. Persyaratan sistem untuk perancangan Sistem Informasi Bank Darah yaitu berupa persyaratan fungsional. Persyaratan fungsional untuk PMI meliputi sistem mampu mengelola data stok darah, data pendonor, data Rumah Sakit, untuk persyaratan fungsional Rumah Sakit meliputi pencatatan transaksi Rumah Sakit.
3. Rancangan sistem yang dibuat mencakup rancangan kelas analisis, pengintegrasian kelas analisis, mekanisme analisis, package digram, kelas dan antarmuka pengguna sistem.
4. Berdasarkan hasil evaluasi, dapat ditarik kesimpulan secara keseluruhan dari evaluasi yang telah dilakukan yaitu:
  - a. Pada tinjauan untuk pengecekan traceability didapatkan hasil bahwa setiap persyaratan bisa dirunutkan sampai ke tahap perancangan.
  - b. Pada tinjauan yang dilakukan dengan menggunakan walkthrough pengguna telah melakukan tugas yang telah diberikan sesuai dengan alur dari basic flow dengan beberapa masukan dari pengguna diantaranya untuk alur dalam melakukan transaksi agar alur transaksi yang dilakukan dapat lebih dari satu data darah yang disimpan untuk satu kali transaksi,
  - c. Pada evaluasi prototipe untuk antarmuka pengguna dengan menggunakan kuisisioner WEBUSE untuk kategori antarmuka dan navigasi didapatkan hasil bahwa responden memilih setuju dari

pertanyaan yang telah diberikan dengan masukan yaitu untuk membedakan link aktif dan warna font.

## 7.2 Saran

Saran yang dapat diberikan untuk pengembangan aplikasi selanjutnya kedalam tahap implementasi selanjutnya antara lain:

1. Untuk pengembangan lebih lanjut, diharapkan analisis dan perancangan yang dilakukan oleh penulis dapat dijadikan acuan untuk tahap perancangan yang lebih detail sebelum memulai tahapan implementasi.
2. Untuk pengembangan lebih lanjut, setelah dilakukan implementasi sistem dapat dilakukan penelitian untuk menguji antara sistem yang lama dengan sistem yang baru.
3. Untuk pengembangan lebih lanjut, diharapkan dapat memperbaiki antarmuka dengan acuan saran dan masukan dari pengguna.



## BAB 8 DAFTAR PUSTAKA

- Agarwal, B.B., Tayal S.P. & Gupta, M., 2010. *Software Engineering & Testing*. [e-book] United States of America: Jones and Bartlett Publishers. Tersedia di: Google Books <<http://booksgoogle.com>> [Diakses 27 Juli 2016].
- Bitner, K dan Spence, I., 2002. *Use Case Modelling*. [e-book] U.S: Addison Wesley. Tersedia di: Google Books <<http://booksgoogle.com>> [Diakses 27 Juli 2016].
- Chiew, T.K., & Salim, S.S., 2003. WEBUSE: Website Usability Evaluation Tool, [online] Tersedia di: < <https://www.researchgate.net>> [Diakses 27 Juli 2016].
- Departemen Kesehatan RI, 2008. *Pedoman Pengelolaan Bank Darah Rumah Sakit (BDRS)*. [pdf]. Departemen Kesehatan RI. [Diakses 28 September 2015].
- Dix, A., Janet, F., Abowd, G.D., & Beale, R., 2004. *Human Computer Interaction Third Edition*. [e-book] England: Pearson. Tersedia di: Google Books <<http://booksgoogle.com>> [Diakses 27 Juli 2016].
- DSDM Consortium, 2016. *MoSCoW Prioritisation*. [online] Tersedia di: < <https://www.dsdm.org/content/moscow-prioritisation>> [Diakses 12 Juli 2016].
- Fitriah, R. 2012. *Analisis dan Perancangan Perpustakaan Digital dan Sistem Otomasi Perpustakaan SMA Negeri 3 Palembang*. S1. Universitas Bina Darma. Tersedia di: <<http://eprints.binadarma.ac.id>> [Diakses 27 Juli 2016].
- IBM Corporation., 2003. *PRJ270 Essentials of Rational Unified Process*. [e-book] United States of America: IBM. Tersedia di: <<http://ibm.com>> [Diakses 13 Maret 2016].
- IBM Corporation., 2004. *DEV475 Mastering Object Oriented Analysis and Design with UML 2.0 Volume 1*. [e-book] United States of America: IBM. Tersedia di: <<http://ibm.com>> [Diakses 13 Maret 2016].
- IBM Corporation., 2007. *Writing Good Use Case*. [e-book] United States of America: IBM. Tersedia di: <<http://ibm.com>> [Diakses 13 Maret 2016].
- Marnasusanto, N. 2014. *Analisa dan Perancangan Sistem Informasi dengan Penerapan Customer Relationship Management Service Automation pada Vitara Rent Cars Malang*. S1. Universitas Brawijaya.
- Naik, K., & Tripathy, P., 2008. *Software Testing and Quality Assurance*. [e-book] United States of America: A John Wiley & Sons. Tersedia di: <<https://www.amazon.com>> [Diakses 27 Juli 2016].
- Nugraha, N.F., Muqtadiroh, F.A. & Nisafani, A.S., 2014. *Analisis dan Perancangan Wiki Budaya Dalam Rangka Melestarikan Budaya Bangsa dan Kearifan Lokal Nusantara*. [online] Tersedia di: <<http://digilib.its.ac.id>> [Diakses 27 Juli 2016].
- OMG., 2013. *Business Process Model and Notation (BPMN)*. [e-book]. Tersedia di: <<http://www.omg.org>> [Diakses 27 Juli 2016].

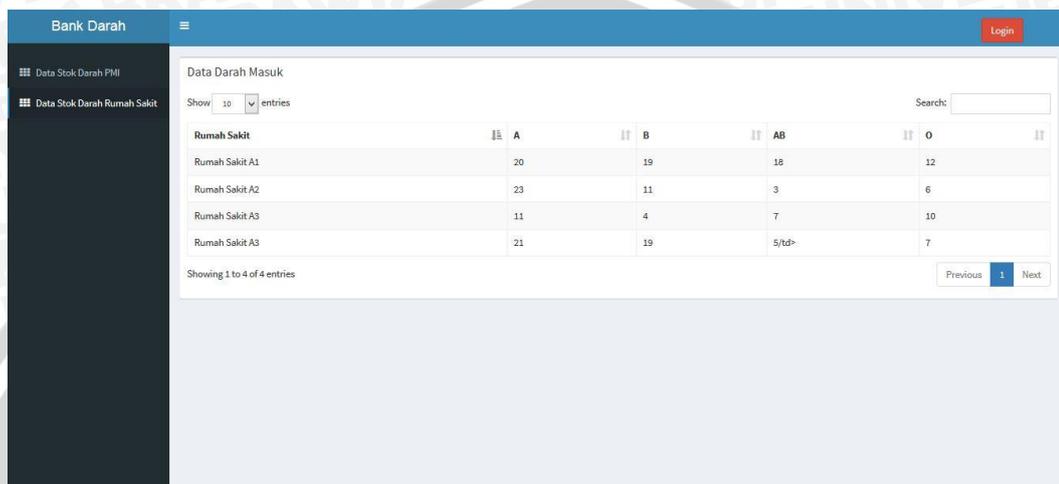
- O'Brien, J.A., & Marakas, G.M., 2014. *Management Information System*. Diterjemahkan dari Bahasa Inggris oleh Liza Nurbani Puspitasari dan Hirson Kurnia. 2014. Jakarta: Salemba Empat.
- Pressman, R.S., 2010. *Software Engineering: A Practitioner's Approach*. 7th ed. [e-book] Americas, New York: McGraw-Hill. Tersedia di: Google Books <<http://booksgoogle.com>> [Diakses 27 Juli 2016].
- Rosa A.S & Shalahuddin,M., 2014. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Informatika Bandung.
- Rumbaugh, J., Jacobson, I. & Booch, G., 2005. *The Unified Modeling Language User Guide*. Second Edition. [e-book] United States of America: Addison Wesley. Tersedia di: <<https://books.google.co.id>> [Diakses 27 Juli 2016].
- Sapta,S.A., 2009. *Kenali PMI*. [pdf]. Tersedia di: <<http://pmi.or.id>> [Diakses 02 Oktober 2015].
- Siahaan, D., 2012. *Analisa Kebutuhan Dalam Rekayasa Perangkat Lunak*. Andi Yogyakarta.
- Sommerville, I., 2001. *Software Engineering/Sixth Edition*. Diterjemahkan dari Bahasa Inggris oleh Dra. Yuhilza Hanum M.Eng. 2003. Jakarta: Erlangga.
- Utomo, K.B., 2010. Perancangan Sistem Informasi Bank Darah Hidup untuk Mempercepat Penyediaan Calon Penyumbang Darah dengan Ketepatan yang Tinggi (Studi di PMI Kota Samarinda), [online] Tersedia di: <<http://fmipa.unmul.ac.id>> [Diakses 27 Juli 2016].



## LAMPIRAN A HALAMAN ANTARMUKA SISTEM

### A.1 Mengecek ketersediaan darah Rumah Sakit

Gambar 7.1 merupakan halaman antarmuka untuk mengecek ketersediaan darah di Rumah Sakit.

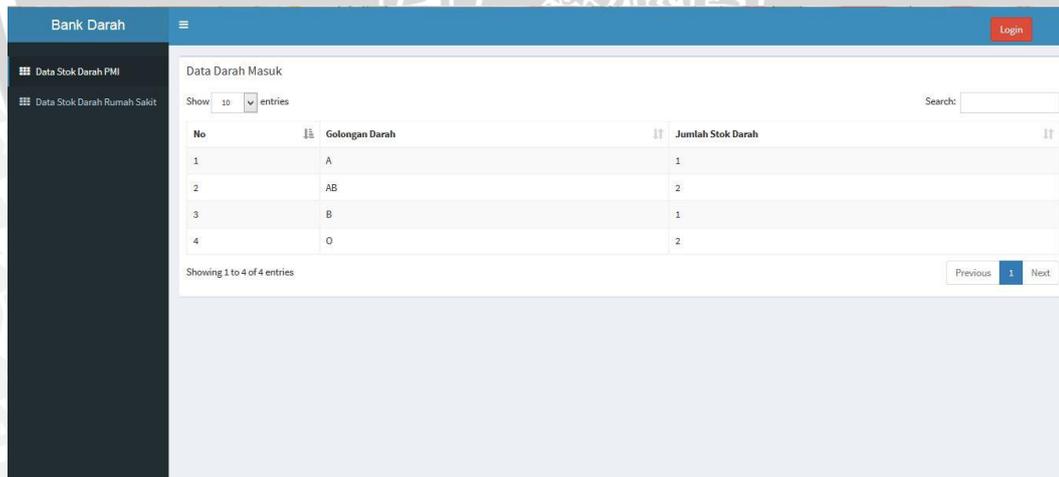


Rumah Sakit	A	B	AB	O
Rumah Sakit A1	20	19	18	12
Rumah Sakit A2	23	11	3	6
Rumah Sakit A3	11	4	7	10
Rumah Sakit A3	21	19	5	7

Gambar 7.1 Antarmuka mengecek ketersediaan darah Rumah Sakit

### A.2 Mengecek ketersediaan darah PMI

Gambar 7.2 merupakan halaman antarmuka untuk mengecek ketersediaan darah di PMI.

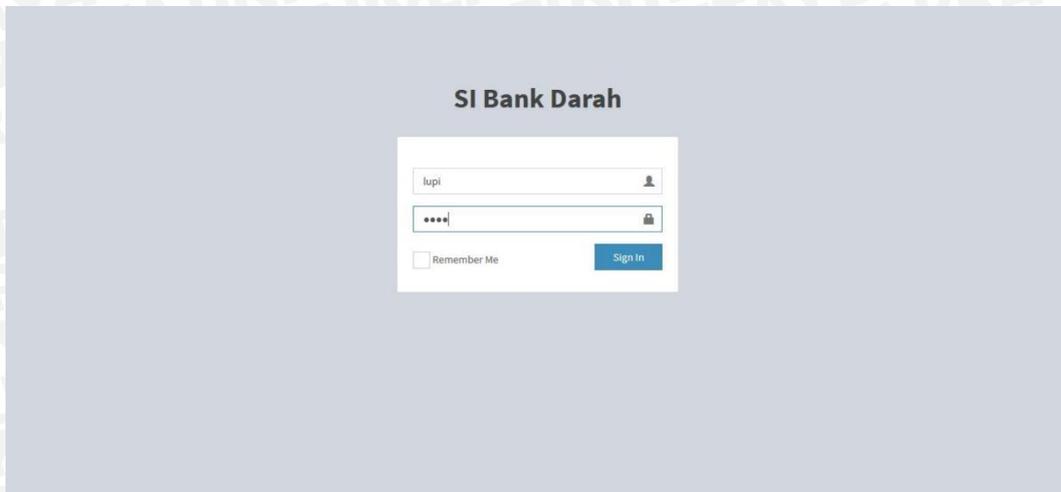


No	Golongan Darah	Jumlah Stok Darah
1	A	1
2	AB	2
3	B	1
4	O	2

Gambar 7.2 Antarmuka mengecek ketersediaan darah PMI

### A.3 Autentikasi

Gambar 7.3 merupakan halaman untuk antarmuka autentikasi.



**SI Bank Darah**

lupi

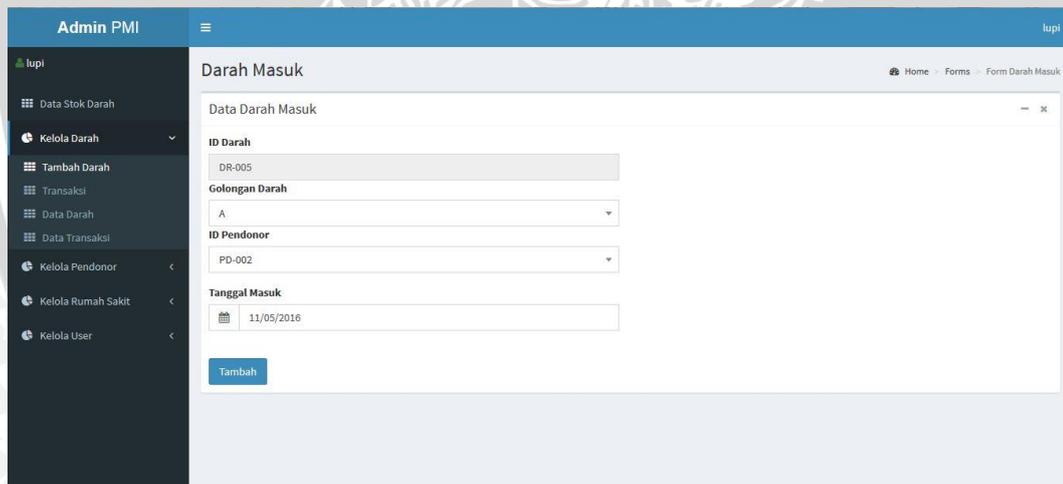
••••

Remember Me

Gambar 7.3 Antarmuka autentikasi

### A.4 Menambah darah masuk

Gambar 7.4 merupakan halaman antarmuka untuk PMI melakukan tambah darah masuk.



Admin PMI lupi

Data Stok Darah

Kelola Darah

Tambah Darah

Transaksi

Data Darah

Data Transaksi

Kelola Pendonor

Kelola Rumah Sakit

Kelola User

Darah Masuk

Data Darah Masuk

ID Darah

DR-005

Golongan Darah

A

ID Pendonor

PD-002

Tanggal Masuk

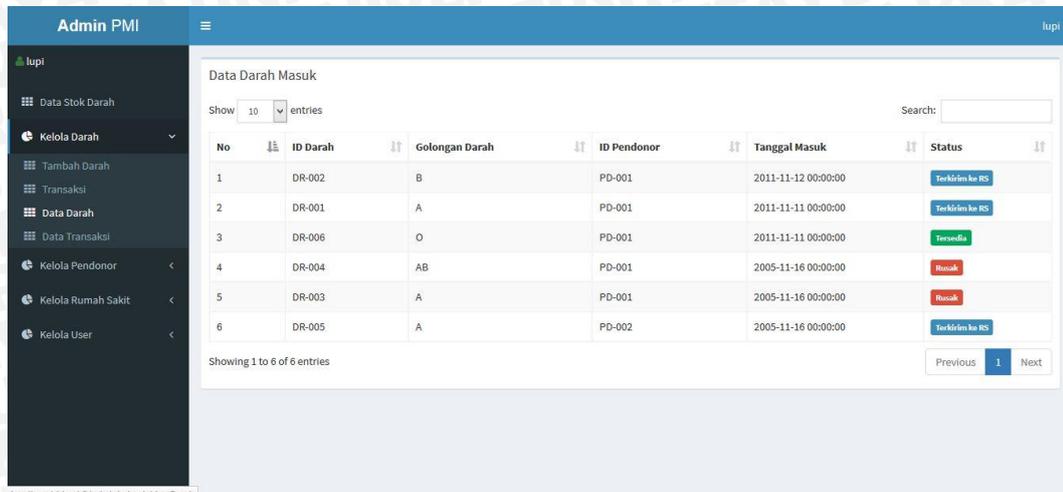
11/05/2016

Tambah

Gambar 7.4 Antarmuka menambah darah masuk

## A.5 Melihat data darah

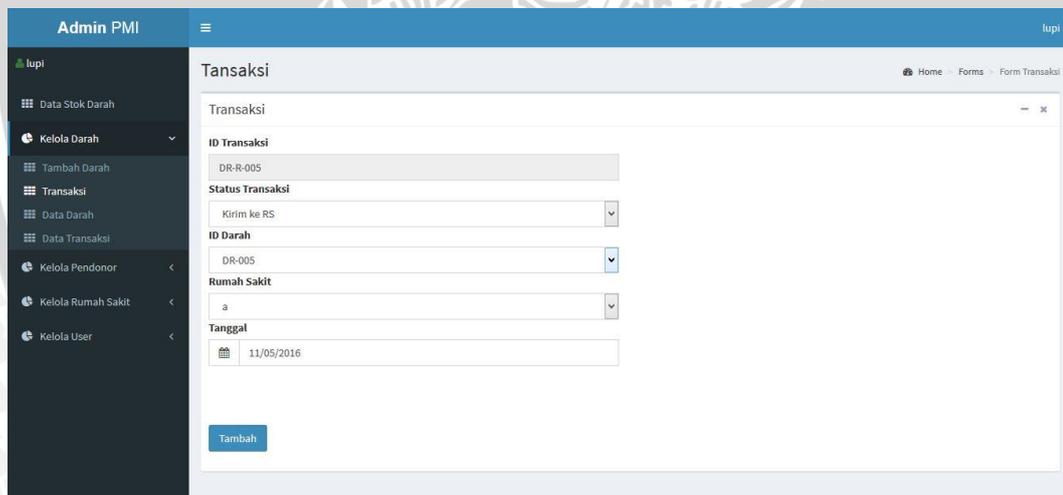
Gambar 7.5 merupakan halaman antarmuka PMI untuk melihat data darah.



Gambar 7.5 Antarmuka melihat data darah

## A.6 Menambah darah keluar atau transaksi ke Rumah Sakit

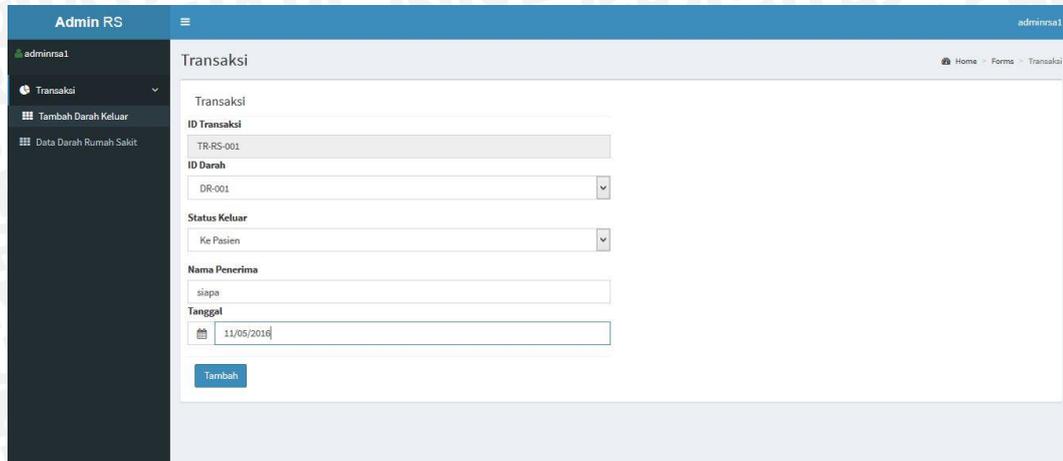
Gambar 7.6 merupakan halaman antarmuka untuk menambah darah keluar atau melakukan transaksi ke Rumah Sakit.



Gambar 7.6 Antarmuka transaksi ke Rumah Sakit

## A.7 Menambah darah keluar atau transaksi ke pasien

Gambar 7.7 merupakan halaman antarmuka untuk melakukan darah keluar atau transaksi ke pasien.

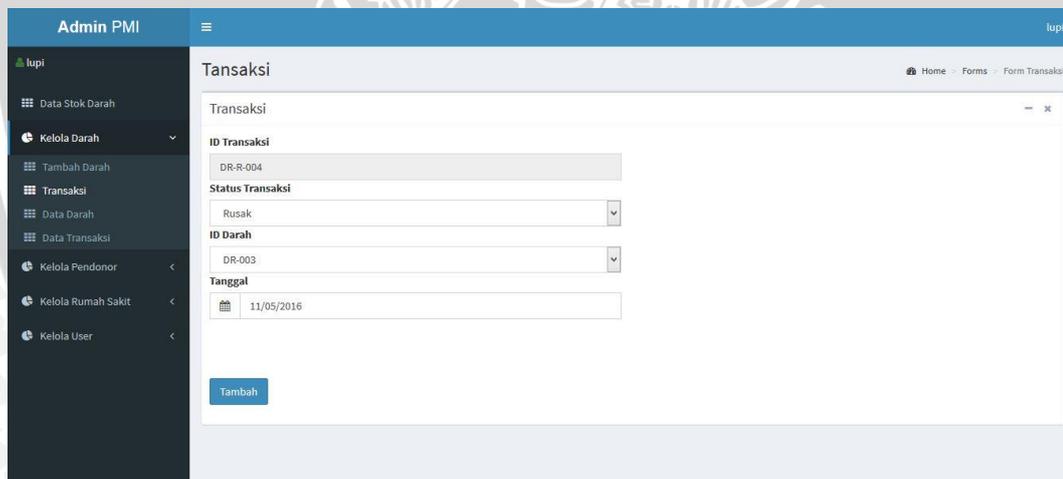


The screenshot shows a web application interface for 'Admin RS'. The main content area is titled 'Transaksi'. It contains a form with the following fields: 'ID Transaksi' (text input with value 'TR-RS-001'), 'ID Darah' (dropdown menu with value 'DR-001'), 'Status Keluar' (dropdown menu with value 'Ke Pasien'), 'Nama Penerima' (text input with value 'siapa'), and 'Tanggal' (calendar icon with value '11/05/2016'). A blue 'Tambah' button is located below the form. The left sidebar shows a menu with 'Transaksi' selected. The top right corner shows the user 'adminrs1'.

Gambar 7.7 Antarmuka transaksi ke pasien

## A.8 Menambah darah keluar atau transaksi untuk darah rusak

Gambar 7.8 merupakan halaman untuk melakukan darah keluar atau transaksi untuk darah rusak.

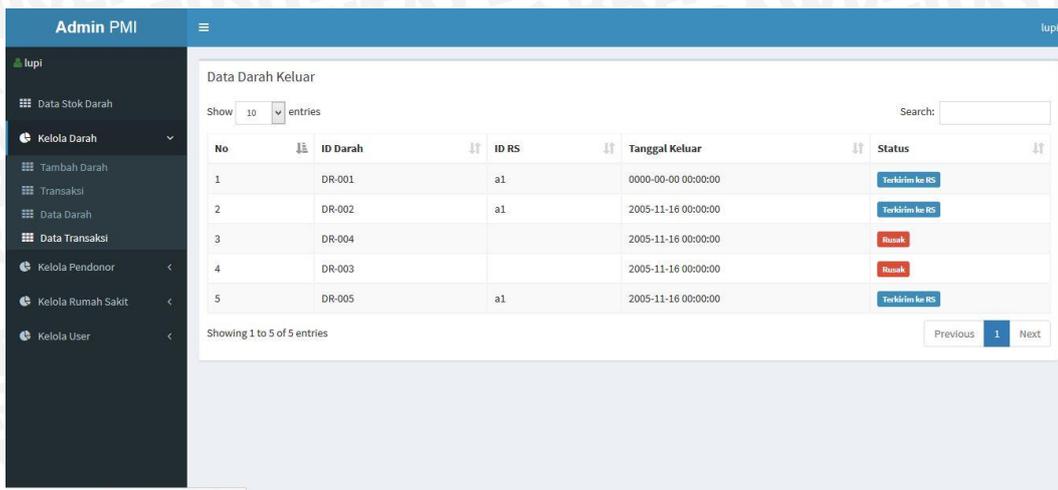


The screenshot shows a web application interface for 'Admin PMI'. The main content area is titled 'Tansaksi'. It contains a form with the following fields: 'ID Transaksi' (text input with value 'DR-R-004'), 'Status Transaksi' (dropdown menu with value 'Rusak'), 'ID Darah' (dropdown menu with value 'DR-003'), and 'Tanggal' (calendar icon with value '11/05/2016'). A blue 'Tambah' button is located below the form. The left sidebar shows a menu with 'Tansaksi' selected. The top right corner shows the user 'lupi'.

Gambar 7.8 Antarmuka darah rusak

## A.9 Melihat data darah keluar atau transaksi

Gambar 7.9 merupakan halaman untuk antarmuka melihat data darah keluar atau transaksi.

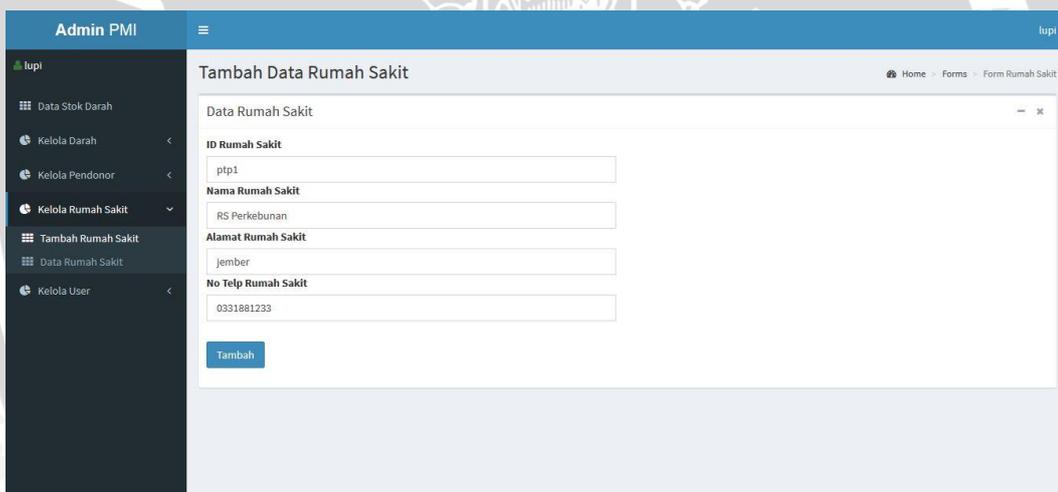


localhost/sidarah3/c\_keloladarahkeluar/datadarahkeluar

Gambar 7.9 Antarmuka melihat transaksi

## A.10 Menambah Rumah Sakit

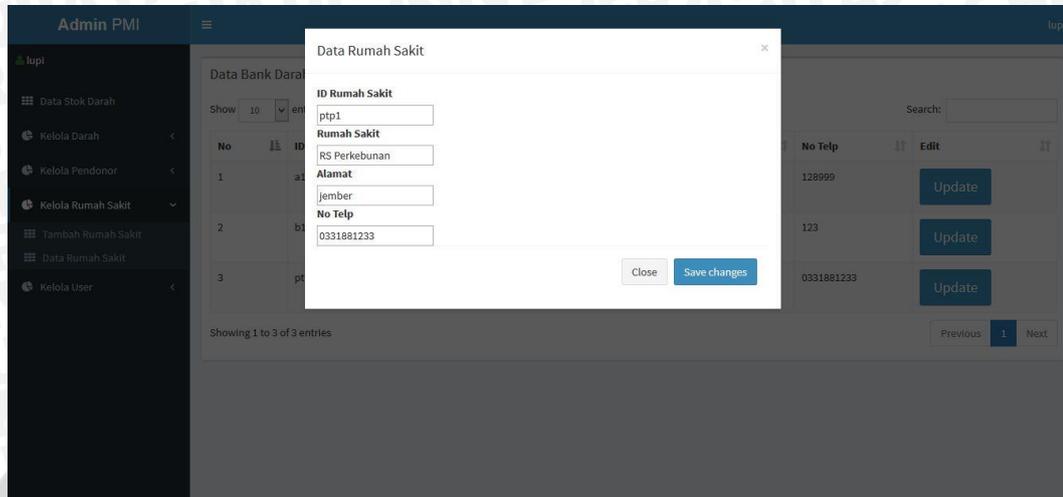
Gambar 7.10 merupakan halaman antarmuka untuk menambah data Rumah Sakit.



Gambar 7.10 Antarmuka menambah Rumah Sakit

## A.11 Memperbarui data Rumah Sakit

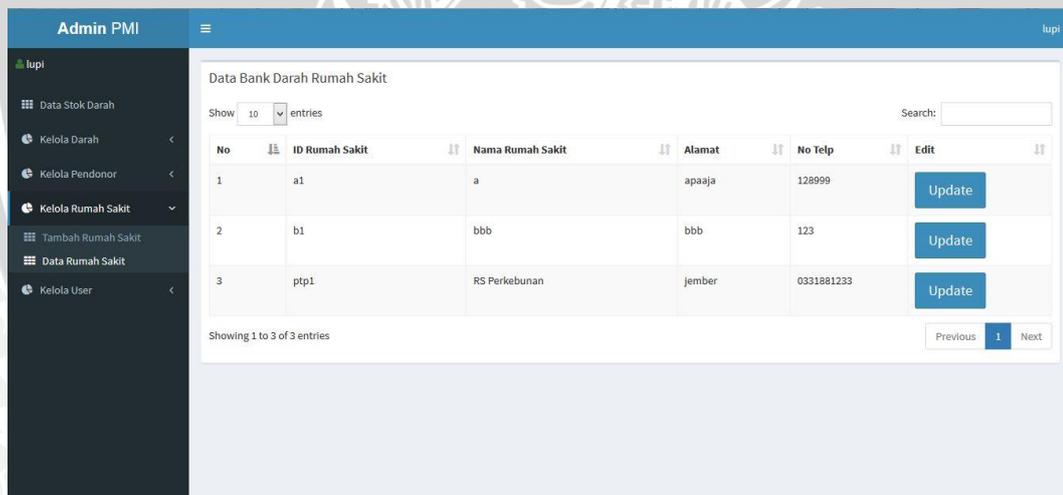
Gambar 7.11 merupakan halaman antarmuka untuk memperbarui data Rumah Sakit.



Gambar 7.11 Antarmuka memperbarui data Rumah Sakit

## A.12 Melihat data Rumah Sakit

Gambar 7.12 merupakan halaman antarmuka untuk melihat data Rumah Sakit.



Gambar 7.12 Antarmuka melihat data Rumah Sakit

### A.13 Menambah pendonor

Gambar 7.13 merupakan halaman antarmuka untuk menambah data pendonor.

The screenshot shows a web interface for adding donor data. The page title is 'Tambah Data Pendonor'. On the left is a dark sidebar with navigation items: 'Data Stok Darah', 'Kelola Darah', 'Kelola Pendonor', 'Tambah Pendonor', 'Data Pendonor', 'Kelola Rumah Sakit', and 'Kelola User'. The main content area contains a form with the following fields:

- ID Pendonor: PD-002
- Nama Pendonor: momo
- Alamat Pendonor: jember
- No Telp Pendonor: 085123321456

A blue 'Tambah' button is located at the bottom left of the form area.

Gambar 7.13 Antarmuka menambah pendonor

### A.14 Melihat data pendonor

Gambar 7.14 merupakan halaman antarmuka untuk melihat data pendonor.

The screenshot shows a web interface for viewing donor data. The page title is 'Data Pendonor'. The sidebar is identical to the previous screenshot. The main content area displays a table with the following data:

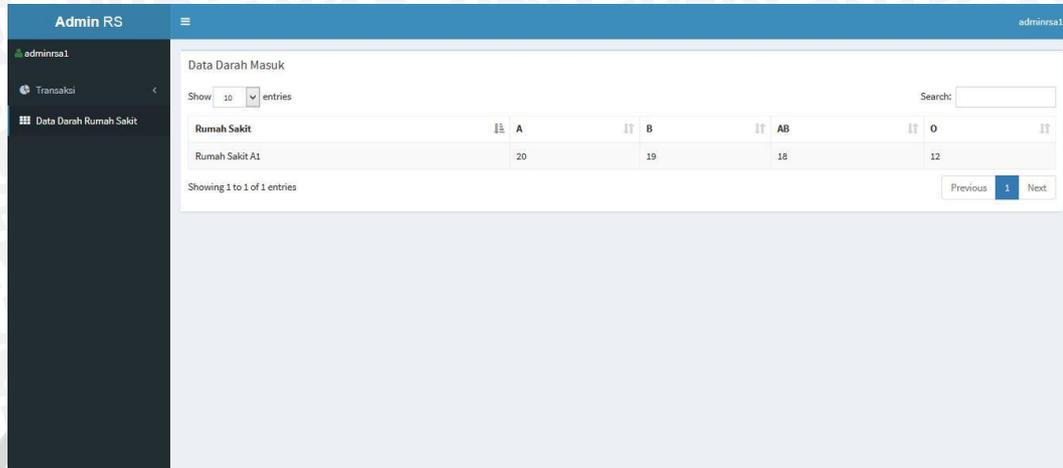
No	ID Pendonor	Nama Pendonor	Alamat Pendonor	No Telp
1	PD-001	aku	aku dimana	123
2	PD-002	momo	Jember	085123321456

Below the table, it says 'Showing 1 to 2 of 2 entries'. There are 'Previous', '1', and 'Next' buttons for pagination. A search bar is located at the top right of the table area.

Gambar 7.14 Antarmuka melihat data pendonor

### A.15 RS Mengecek ketersediaan darah Rumah Sakit

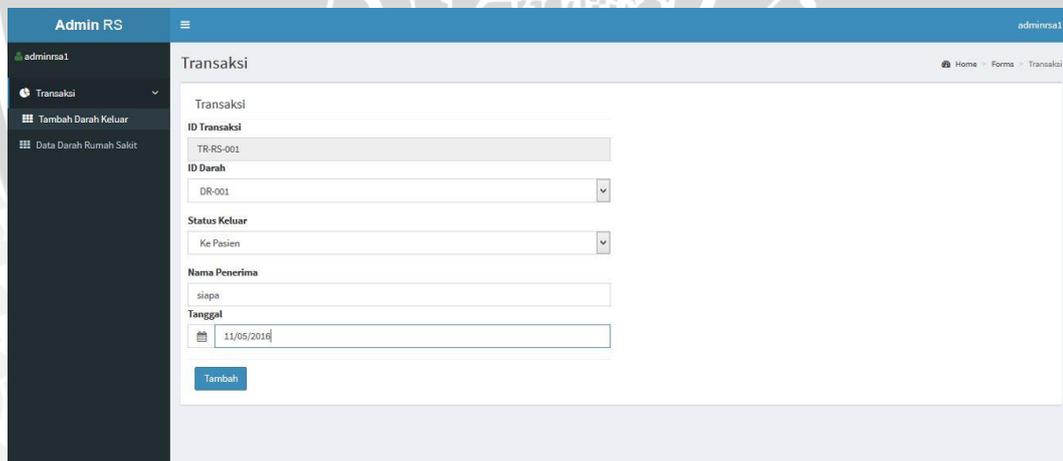
Gambar 7.15 merupakan halaman antarmuka yang digunakan oleh Rumah Sakit untuk mengecek ketersediaan Rumah Sakit yang ada di Rumah Sakit tersebut.



Gambar 7.15 Antarmuka Rumah Sakit mengecek ketersediaan darah Rumah Sakit

### A.16 RS menambah darah keluar atau transaksi ke pasien

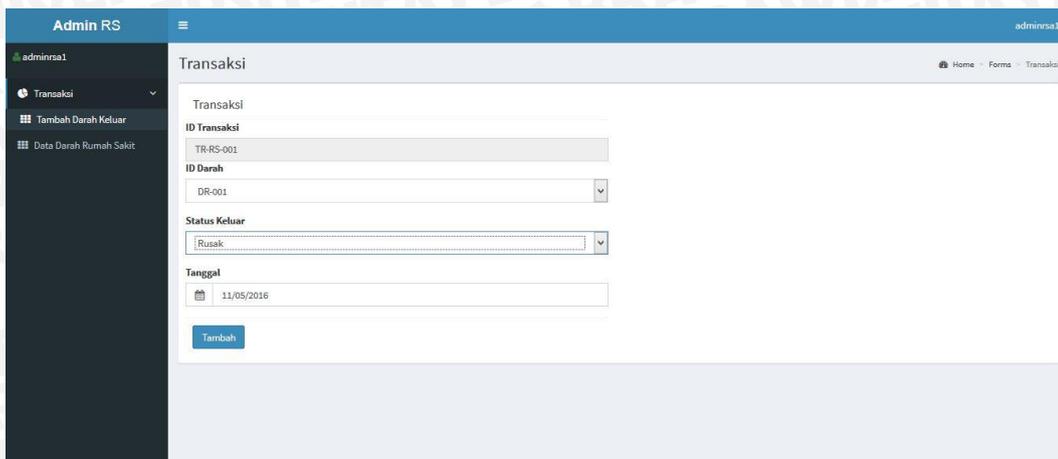
Gambar 7.16 merupakan halaman antarmuka yang digunakan Rumah Sakit untuk menambah darah keluar atau transaksi ke pasien.



Gambar 7.16 Antarmuka Rumah Sakit transaksi ke pasien

### A.17 RS menambah darah keluar atau transaksi darah rusak

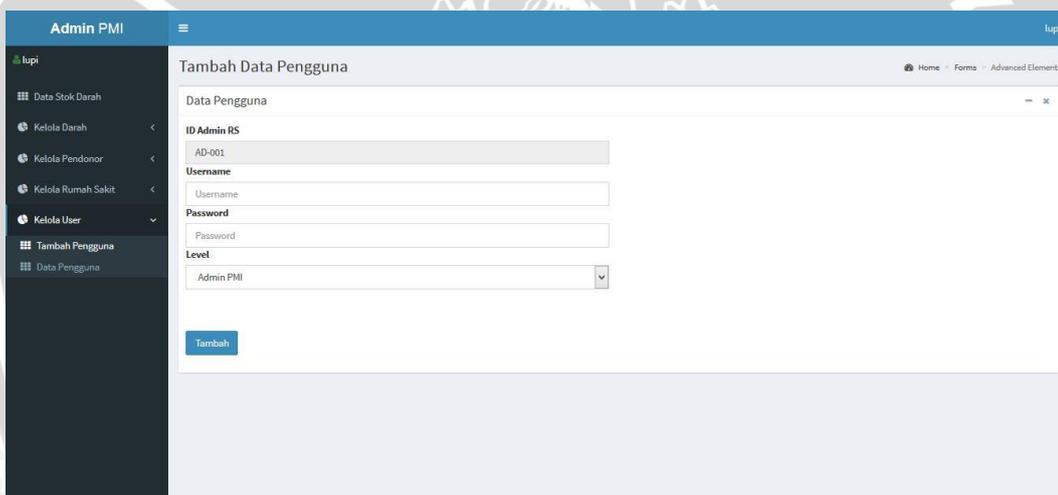
Gambar 7.17 merupakan halaman antarmuka yang digunakan oleh Rumah Sakit untuk menambah transaksi darah rusak.



Gambar 7.17 Antarmuka Rumah Sakit darah rusak

### A.18 Menambah user atau pengguna

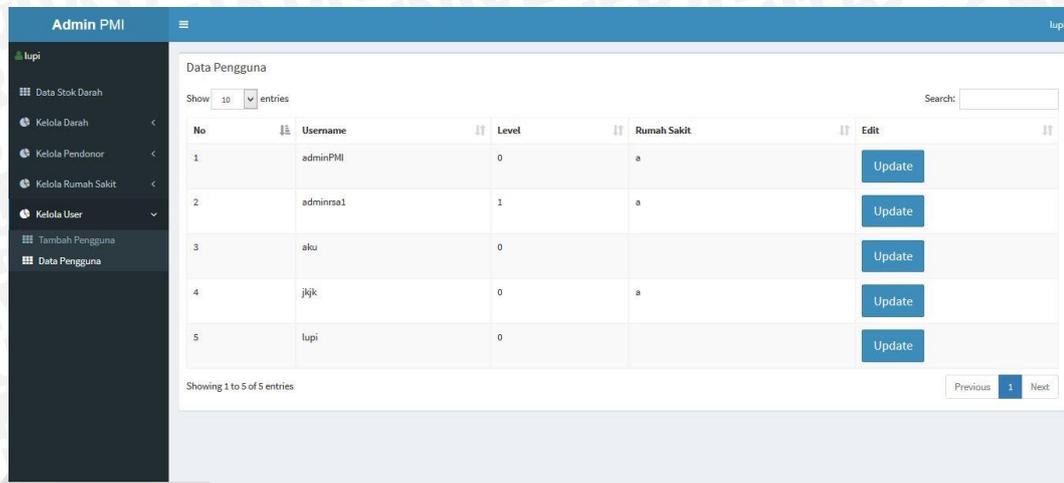
Gambar 7.18 merupakan halaman antarmuka yang digunakan oleh PMI untuk menambah user atau pengguna.



Gambar 7.18 Antarmuka menambah user atau pengguna

## A.19 Melihat data user atau pengguna

Gambar 7.19 merupakan halaman antarmuka yang digunakan oleh PMI untuk melihat data user atau pengguna.



The screenshot shows the 'Admin PMI' interface. On the left is a dark sidebar with navigation options: 'Data Stok Darah', 'Kelola Darah', 'Kelola Pendonor', 'Kelola Rumah Sakit', 'Kelola User', 'Tambah Pengguna', and 'Data Pengguna'. The main content area is titled 'Data Pengguna' and features a table with 5 entries. The table has columns for 'No', 'Username', 'Level', 'Rumah Sakit', and 'Edit'. Each row has an 'Update' button. Below the table, it says 'Showing 1 to 5 of 5 entries' and has 'Previous', '1', and 'Next' navigation links.

No	Username	Level	Rumah Sakit	Edit
1	adminPMI	0	a	<a href="#">Update</a>
2	adminra1	1	a	<a href="#">Update</a>
3	aku	0		<a href="#">Update</a>
4	jjjk	0	a	<a href="#">Update</a>
5	lupi	0		<a href="#">Update</a>

Gambar 7.19 Antarmuka melihat data user atau pengguna

## LAMPIRAN B HASIL EVALUASI PENGGUNA

### B.1 Hasil tinjauan prototipe oleh responden 1

Gambar 7.20 merupakan lembar hasil evaluasi prototipe pengguna terkait alur pengguna dengan *basic flow* yang dilakukan oleh responden 1.

**EVALUASI PENGGUNA**

Pengguna : Anita Kusuma N  
 Fitur : mengecek ketersediaan darah PMI dan RS, menambah data darah, menambah data darah keluar, memperbarui data RS, menambah data pendonor.

Pertanyaan	Jawaban
<b>Use case : Mengecek ketersediaan darah PMI dan Rumah Sakit</b>	
Apakah alur untuk mengecek ketersediaan darah yang ada di PMI dan Rumah Sakit sesuai dengan yang diharapkan?	ya, alur untuk mengecek ketersediaan darah di PMI dan RS sudah sesuai harapan
Apakah proses mengecek ketersediaan darah mudah dioperasikan?	ya, proses mengecek ketersediaan darah mudah dioperasikan, namun tampilan
Saran : <u>memperbaiki tampilan ikon getik terkait informasi mengenai cara atau prosedur untuk melakukan pemeriksaan darah di setiap RS.</u>	kurang menarik.
<b>Use case : Mengelola stok darah (menambah data darah)</b>	
Apakah alur menambah data darah sesuai dengan yang diharapkan?	alur untuk menambah data darah sudah sesuai dengan yang diharapkan
Apakah kolom/formulir menambah data darah sudah benar dan lengkap?	ya.
Apakah proses menambah data darah mudah dioperasikan?	proses untuk menambah data darah mudah dioperasikan.
Saran : <u>lebih diperbaiki untuk warna tampilan dan ukuran font beserta tanda link aktif</u> saat membuka menu.	
<b>Use Case : Mengelola stok darah (menambah data darah keluar)</b>	
Apakah alur menambah data darah keluar / transaksi sesuai dengan yang diharapkan?	alur untuk menambah data darah keluar / transaksi sesuai dengan yang diharapkan.
Apakah kolom/formulir menambah data darah keluar sudah benar dan lengkap?	ya
Apakah proses menambah data darah keluar mudah dioperasikan?	proses untuk menambah data darah keluar mudah dioperasikan.
Saran : -	
<b>Use Case : Mengelola data Rumah Sakit (memperbarui data Rumah Sakit)</b>	
Apakah alur untuk memperbarui data RS sesuai dengan yang diharapkan?	alur untuk memperbarui data RS sudah sesuai dengan yang diharapkan.
Apakah kolom/formulir memperbarui data Rumah Sakit sudah benar dan lengkap?	ya
Apakah proses memperbarui data Rumah Sakit mudah dioperasikan?	proses untuk memperbarui data RS mudah dioperasikan.
Saran : <u>untuk tampilan saat menampilkan detail data RS perlu diberi warna berbeda agar bisa mengetahui mana yg bisa diperbarui atau tidak.</u>	
<b>Use Case : Mengelola data pendonor (menambah data pendonor)</b>	
Apakah alur menambah data pendonor sesuai dengan yang diharapkan?	proses untuk menambah data pendonor sudah sesuai dengan yang diharapkan
Apakah kolom/formulir menambah data pendonor sudah benar dan lengkap?	ya
Apakah proses menambah data pendonor mudah dioperasikan?	proses menambah data pendonor mudah dioperasikan
Saran : -	

Gambar 7.20 Evaluasi pengguna responden 1

## B.2 Hasil tinjauan prototipe oleh responden 1

Gambar 7.21 merupakan lembar hasil evaluasi prototipe pengguna terkait navigasi dan antarmuka yang dilakukan oleh responden 1.

Navigasi				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Saya dapat dengan mudah mengetahui posisi saya pada sistem		✓		
Sistem menyediakan navigasi menu dan link yang berguna untuk mendapatkan informasi	✓			
Sistem mudah digunakan dengan menggunakan tombol kembali atau link	✓			
Sistem tidak membuka banyak jendela browser ketika digunakan	✓			
Peletakan menu atau link dapat dengan mudah dikenali	✓			
Navigasi menu pada sistem sudah menunjukkan fungsi dengan jelas	✓			
<b>Masukan dan Saran</b>	perlu ditampahkan tombol menu link aktif di samping untuk kemudahan pengguna.			

Antarmuka				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Perancangan antar muka sistem mudah untuk dipelajari	✓			
Sistem tidak mengandung fitur yang membutuhkan scroll atau animasi yang berulang	✓			
Sistem memiliki antar muka yang konsisten	✓			
<b>Masukan dan Saran</b>	desain antarmuka sudah konsisten dan mudah dipelajari.			

Harapan Alur Pengguna				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Respon sistem terhadap aksi yang dilakukan sudah sesuai dengan harapan	✓			
Alur penggunaan sistem sudah sesuai dengan harapan pengguna	✓			
<b>Pernyataan / Masukan dan Saran</b>	sistem sudah sesuai dengan alur yg diharapkan pengguna.			

Malang, 12 Mei 2016  
Reviewer

  
Ancha Kusuma.

Gambar 7.21 Evaluasi pengguna responden 1

### B.3 Hasil tinjauan prototipe oleh responden 2

Gambar 7.22 merupakan lembar hasil evaluasi prototipe pengguna terkait alur pengguna dengan *basic flow* yang dilakukan oleh responden 2.

2

**EVALUASI PENGGUNA**

**Pengguna** : Milla Angraeni  
**Fitur** : mengecek ketersediaan darah PMI dan RS, menambah data darah, menambah data darah keluar, memperbarui data RS, menambah data pendonor.

Pertanyaan	Jawaban
<b>Use case : Mengecek ketersediaan darah PMI dan Rumah Sakit</b>	
Apakah alur untuk mengecek ketersediaan darah yang ada di PMI dan Rumah Sakit sesuai dengan yang diharapkan?	Alur yang disediakan sesuai dengan harapan.
Apakah proses mengecek ketersediaan darah mudah dioperasikan?	Proses untuk mengecek ketersediaan darah dan RS mudah dijalankan.
<b>Saran</b> : Penggunaan warna dan ukuran font kurang jelas.	
<b>Use case : Mengelola stok darah (menambah data darah)</b>	
Apakah alur menambah data darah sesuai dengan yang diharapkan?	Alur untuk menambah darah masuk sudah sesuai dengan yang diharapkan.
Apakah kolom/formulir menambah data darah sudah benar dan lengkap?	Ya.
Apakah proses menambah data darah mudah dioperasikan?	Proses menambah darah mudah dijalankan.
<b>Saran</b> : Penggunaan warna dan ukuran font kurang jelas, sehingga sulit membedakan halaman yang aktif / tidak.	
<b>Use Case : Mengelola stok darah (menambah data darah keluar)</b>	
Apakah alur menambah data darah keluar / transaksi sesuai dengan yang diharapkan?	Alur untuk menambah data darah keluar / transaksi sesuai harapan.
Apakah kolom/formulir menambah data darah keluar sudah benar dan lengkap?	Ya
Apakah proses menambah data darah keluar mudah dioperasikan?	Proses untuk melakukan transaksi mudah dijalankan.
<b>Saran</b> : Untuk melakukan transaksi lebih baik jika satu kali transaksi banyak data darah ditransaksikan.	
<b>Use Case : Mengelola data Rumah Sakit (memperbarui data Rumah Sakit)</b>	
Apakah alur untuk memperbarui data RS sesuai dengan yang diharapkan?	alur untuk memperbarui data RS sudah sesuai dengan yang diharapkan.
Apakah kolom/formulir memperbarui data Rumah Sakit sudah benar dan lengkap?	Ya.
Apakah proses memperbarui data Rumah Sakit mudah dioperasikan?	proses untuk memperbarui data RS mudah dijalankan.
<b>Saran</b> : tombol untuk update terlalu besar, tampilan detail tidak dapat dibedakan.	
<b>Use Case : Mengelola data pendonor (menambah data pendonor)</b>	
Apakah alur menambah data pendonor sesuai dengan yang diharapkan?	alur sudah sesuai dengan yang diharapkan.
Apakah kolom/formulir menambah data pendonor sudah benar dan lengkap?	Ya
Apakah proses menambah data pendonor mudah dioperasikan?	Proses menambah data pendonor mudah dijalankan.
<b>Saran</b> : -	

Gambar 7.22 Evaluasi pengguna responden 2

## B.4 Hasil tinjauan prototipe oleh responden 2

Gambar 7.23 merupakan lembar hasil evaluasi prototipe pengguna terkait navigasi dan antarmuka yang dilakukan oleh responden 2.

Navigasi				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Saya dapat dengan mudah mengetahui posisi saya pada sistem	✓			
Sistem menyediakan navigasi menu dan link yang berguna untuk mendapatkan informasi	✓			
Sistem mudah digunakan dengan menggunakan tombol kembali atau link	✓			
Sistem tidak membuka banyak jendela browser ketika digunakan	✓			
Peletakan menu atau link dapat dengan mudah dikenali	✓			
Navigasi menu pada sistem sudah menunjukkan fungsi dengan jelas	✓			
<b>Masukan dan Saran</b>	tombol sudah kurang jelas, font warna untuk teks / font nya lebih besar			

Antarmuka				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Perancangan antar muka sistem mudah untuk dipelajari	✓			
Sistem tidak mengandung fitur yang membutuhkan scroll atau animasi yang berulang	✓			
Sistem memiliki antar muka yang konsisten	✓			
<b>Masukan dan Saran</b>	Tampilan antarmuka sudah bagus.			

Harapan Alur Pengguna				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Respon sistem terhadap aksi yang dilakukan sudah sesuai dengan harapan	✓			
Alur penggunaan sistem sudah sesuai dengan harapan pengguna		✓		
<b>Pernyataan / Masukan dan Saran</b>	sudah bagus, tetapi untuk alur masih belum begitu jelas seperti penambahan darah yang harus dilakukan satu persatu.			

Malang, 12 Mei 2016  
Reviewer  


Gambar 7.23 Evaluasi pengguna responden 2

## B.5 Hasil tinjauan prototipe oleh responden 3

Gambar 7.24 merupakan lembar hasil evaluasi prototipe pengguna terkait alur pengguna dengan *basic flow* yang dilakukan oleh responden 3.

3

**EVALUASI PENGGUNA**

**Pengguna** : *Pemata kharismawati*

**Fitur** : *mengecek ketersediaan darah PMI dan RS, menambah data darah, menambah data darah keluar, memperbarui data RS, menambah data pendonor.*

Pertanyaan	Jawaban
<b>Use case : Mengecek ketersediaan darah PMI dan Rumah Sakit</b>	
Apakah alur untuk mengecek ketersediaan darah yang ada di PMI dan Rumah Sakit sesuai dengan yang diharapkan?	Ya.
Apakah proses mengecek ketersediaan darah mudah dioperasikan?	Ya
<b>Saran :</b>	
<b>Use case : Mengelola stok darah (menambah data darah)</b>	
Apakah alur menambah data darah sesuai dengan yang diharapkan?	Ya
Apakah kolom/formulir menambah data darah sudah benar dan lengkap?	Ya
Apakah proses menambah data darah mudah dioperasikan?	Ya
<b>Saran :</b>	
<b>Use Case : Mengelola stok darah (menambah data darah keluar)</b>	
Apakah alur menambah data darah keluar / transaksi sesuai dengan yang diharapkan?	Ya
Apakah kolom/formulir menambah data darah keluar sudah benar dan lengkap?	Ya
Apakah proses menambah data darah keluar mudah dioperasikan?	Ya
<b>Saran :</b>	
<b>Use Case : Mengelola data Rumah Sakit (memperbarui data Rumah Sakit)</b>	
Apakah alur untuk memperbarui data RS sesuai dengan yang diharapkan?	Ya
Apakah kolom/formulir memperbarui data Rumah Sakit sudah benar dan lengkap?	Ya
Apakah proses memperbarui data Rumah Sakit mudah dioperasikan?	Ya
<b>Saran :</b>	
<b>Use Case : Mengelola data pendonor (menambah data pendonor)</b>	
Apakah alur menambah data pendonor sesuai dengan yang diharapkan?	Ya
Apakah kolom/formulir menambah data pendonor sudah benar dan lengkap?	Ya
Apakah proses menambah data pendonor mudah dioperasikan?	
<b>Saran : -</b>	

Gambar 7.24 Evaluasi pengguna responden 3

### B.6 Hasil tinjauan prototipe oleh responden 3

Gambar 7.25 merupakan lembar hasil evaluasi prototipe pengguna terkait navigasi dan antarmuka yang dilakukan oleh responden 3.

Navigasi				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Saya dapat dengan mudah mengetahui posisi saya pada sistem	✓			
Sistem menyediakan navigasi menu dan link yang berguna untuk mendapatkan informasi	✓			
Sistem mudah digunakan dengan menggunakan tombol kembali atau link			✓	
Sistem tidak membuka banyak jendela browser ketika digunakan	✓			
Peletakan menu atau link dapat dengan mudah dikenali		✓		
Navigasi menu pada sistem sudah menunjukkan fungsi dengan jelas		✓		
<b>Masukan dan Saran</b>	perintah pada menu kurang mudah dipahami			

Antarmuka				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Perancangan antar muka sistem mudah untuk dipelajari	✓			
Sistem tidak mengandung fitur yang membutuhkan scroll atau animasi yang berulang	✓			
Sistem memiliki antar muka yang konsisten	✓			
<b>Masukan dan Saran</b>	perintah pada menu kurang mudah dipahami			

Harapan Alur Pengguna				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Respon sistem terhadap aksi yang dilakukan sudah sesuai dengan harapan	✓			
Alur penggunaan sistem sudah sesuai dengan harapan pengguna			✓	
<b>Pernyataan / Masukan dan Saran</b>	perintah pada menu kurang mudah dipahami			

Malang, 12 Mei 2016  
Reviewer



Gambar 7.25 Evaluasi pengguna responden 3

## B.7 Hasil tinjauan prototipe oleh responden 4

Gambar 7.26 merupakan lembar hasil evaluasi prototipe pengguna terkait alur pengguna dengan *basic flow* yang dilakukan oleh responden 4.

4

**EVALUASI PENGGUNA**

**Pengguna** : Lana P

**Fitur** : mengecek ketersediaan darah PMI dan RS, menambah data darah, menambah data darah keluar, memperbarui data RS, menambah data pendonor.

Pertanyaan	Jawaban
<b>Use case : Mengecek ketersediaan darah PMI dan Rumah Sakit</b>	
Apakah alur untuk mengecek ketersediaan darah yang ada di PMI dan Rumah Sakit sesuai dengan yang diharapkan?	Ya
Apakah proses mengecek ketersediaan darah mudah dioperasikan?	Ya
<b>Saran :</b>	
<b>Use case : Mengelola stok darah (menambah data darah)</b>	
Apakah alur menambah data darah sesuai dengan yang diharapkan?	Ya
Apakah kolom/formulir menambah data darah sudah benar dan lengkap?	Ya
Apakah proses menambah data darah mudah dioperasikan?	Ya
<b>Saran :</b>	
<b>Use Case : Mengelola stok darah (menambah data darah keluar)</b>	
Apakah alur menambah data darah keluar / transaksi sesuai dengan yang diharapkan?	Ya
Apakah kolom/formulir menambah data darah keluar sudah benar dan lengkap?	Ya
Apakah proses menambah data darah keluar mudah dioperasikan?	Ya
<b>Saran : Menperjelas alur transaksi</b>	
<b>Use Case : Mengelola data Rumah Sakit (memperbarui data Rumah Sakit)</b>	
Apakah alur untuk memperbarui data RS sesuai dengan yang diharapkan?	Ya
Apakah kolom/formulir memperbarui data Rumah Sakit sudah benar dan lengkap?	Ya
Apakah proses memperbarui data Rumah Sakit mudah dioperasikan?	Ya
<b>Saran :</b>	
<b>Use Case : Mengelola data pendonor (menambah data pendonor)</b>	
Apakah alur menambah data pendonor sesuai dengan yang diharapkan?	Ya
Apakah kolom/formulir menambah data pendonor sudah benar dan lengkap?	Ya
Apakah proses menambah data pendonor mudah dioperasikan?	Ya
<b>Saran : -</b>	

Gambar 7.26 Evaluasi pengguna responden 4

## B.8 Hasil tinjauan prototipe oleh responden 4

Gambar 7.27 merupakan lembar hasil evaluasi prototipe pengguna terkait navigasi dan antarmuka yang dilakukan oleh responden 4.

Navigasi				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Saya dapat dengan mudah mengetahui posisi saya pada sistem	✓			
Sistem menyediakan navigasi menu dan <i>link</i> yang berguna untuk mendapatkan informasi	✓			
Sistem mudah digunakan dengan menggunakan tombol kembali atau link	✓			
Sistem tidak membuka banyak jendela browser ketika digunakan	✓			
Peletakan menu atau link dapat dengan mudah dikenali	✓			
Navigasi menu pada sistem sudah menunjukkan fungsi dengan jelas	✓			
<b>Masukan dan Saran</b>				

Antarmuka				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Perancangan antar muka sistem mudah untuk dipelajari	✓			
Sistem tidak mengandung fitur yang membutuhkan scroll atau animasi yang berulang	✓			
Sistem memiliki antar muka yang konsisten	✓			
<b>Masukan dan Saran</b>				

Harapan Alur Pengguna				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Respon sistem terhadap aksi yang dilakukan sudah sesuai dengan harapan	✓			
Alur penggunaan sistem sudah sesuai dengan harapan pengguna	✓			
<b>Pernyataan / Masukan dan Saran</b>	alir yang disediakan sistem sudah sesuai harapan pengguna.			

Malang, 12 Mei 2016  
Reviewer



Gambar 7.27 Evaluasi pengguna responden 4

## B.9 Hasil tinjauan prototipe oleh responden 5

Gambar 7.28 merupakan lembar hasil evaluasi prototipe pengguna terkait alur pengguna dengan *basic flow* yang dilakukan oleh responden 5.

5

**EVALUASI PENGGUNA**

**Pengguna** : Novi...  
**Fitur** : mengecek ketersediaan darah PMI dan RS, menambah data darah, menambah data darah keluar, memperbarui data RS, menambah data pendonor.

Pertanyaan	Jawaban
<b>Use case : Mengecek ketersediaan darah PMI dan Rumah Sakit</b>	
Apakah alur untuk mengecek ketersediaan darah yang ada di PMI dan Rumah Sakit sesuai dengan yang diharapkan?	Ya
Apakah proses mengecek ketersediaan darah mudah dioperasikan?	Ya
<b>Saran :</b>	
<b>Use case : Mengelola stok darah (menambah data darah)</b>	
Apakah alur menambah data darah sesuai dengan yang diharapkan?	Ya
Apakah kolom/formulir menambah data darah sudah benar dan lengkap?	Ya
Apakah proses menambah data darah mudah dioperasikan?	Ya
<b>Saran :</b>	
<b>Use Case : Mengelola stok darah (menambah data darah keluar)</b>	
Apakah alur menambah data darah keluar / transaksi sesuai dengan yang diharapkan?	Ya
Apakah kolom/formulir menambah data darah keluar sudah benar dan lengkap?	Ya
Apakah proses menambah data darah keluar mudah dioperasikan?	Ya
<b>Saran :</b>	
<b>Use Case : Mengelola data Rumah Sakit (memperbarui data Rumah Sakit)</b>	
Apakah alur untuk memperbarui data RS sesuai dengan yang diharapkan?	Ya
Apakah kolom/formulir memperbarui data Rumah Sakit sudah benar dan lengkap?	Ya
Apakah proses memperbarui data Rumah Sakit mudah dioperasikan?	Ya
<b>Saran :</b>	
<b>Use Case : Mengelola data pendonor (menambah data pendonor)</b>	
Apakah alur menambah data pendonor sesuai dengan yang diharapkan?	Ya
Apakah kolom/formulir menambah data pendonor sudah benar dan lengkap?	Ya
Apakah proses menambah data pendonor mudah dioperasikan?	Ya
<b>Saran : -</b>	

Gambar 7.28 Evaluasi pengguna responden 5

## B.10 Hasil tinjauan prototipe oleh responden 5

Gambar 7.29 merupakan lembar hasil evaluasi prototipe pengguna terkait navigasi dan antarmuka yang dilakukan oleh responden 5.

Navigasi				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Saya dapat dengan mudah mengetahui posisi saya pada sistem	✓			
Sistem menyediakan navigasi menu dan link yang berguna untuk mendapatkan informasi	✓			
Sistem mudah digunakan dengan menggunakan tombol kembali atau link		✓		
Sistem tidak membuka banyak jendela browser ketika digunakan	✓			
Peletakan menu atau link dapat dengan mudah dikenali		✓		
Navigasi menu pada sistem sudah menunjukkan fungsi dengan jelas		✓		
<b>Masukan dan Saran</b> —				

Antarmuka				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Perancangan antar muka sistem mudah untuk dipelajari	✓			
Sistem tidak mengandung fitur yang membutuhkan scroll atau animasi yang berulang	✓			
Sistem memiliki antar muka yang konsisten	✓			
<b>Masukan dan Saran</b> —				

Harapan Alur Pengguna				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Respon sistem terhadap aksi yang dilakukan sudah sesuai dengan harapan	✓			
Alur penggunaan sistem sudah sesuai dengan harapan pengguna	✓			
<b>Pernyataan / Masukan dan Saran</b>				

Alur sudah sesuai harapan, menurut saya gunakan bahasa yang mudah dipahami untuk penamaan fitur. Malang, 12 Mei 2016

Reviewer



Gambar 7.29 Evaluasi pengguna responden 5