



**SISTEM INFORMASI PERAMALAN JUMLAH
PERMINTAAN DARAH MENGGUNAKAN METODE
EXPONENTIAL SMOOTHING
(STUDI KASUS : UTD PMI KABUPATEN JEMBER)**

SKRIPSI

Oleh

Wenny Hardiyanti Pratiwi

NIM 132410101091

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2017



**SISTEM INFORMASI PERAMALAN JUMLAH
PERMINTAAN DARAH MENGGUNAKAN METODE
EXPONENTIAL SMOOTHING
(STUDI KASUS : UTD PMI KABUPATEN JEMBER)**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Pendidikan Sarjana (S1) Program Studi Sistem Informasi Universitas Jember dan mencapai gelar Sarjana Komputer

Oleh

Wenny Hardiyanti Pratiwi

NIM 132410101091

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2017

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi;
2. Ibunda Ita Kustatik dan Ayahanda Zainal Abidin tersayang yang selalu memberikan dukungan do'a, kasih sayang, motivasi yang luar biasa;
3. Adik Dita Yusrilia Madini tercinta yang selalu memberi dukungan;
4. Teman-teman satu angkatan intention yang meberikan dukungan dan motivasi, serta membantu dalam pembuatan skripsi ini;
5. Sahabatku bersama doa dan bantuannya;
6. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi;
7. Almamater program studi sistem informasi Universitas Jember.

MOTTO

“Sesungguhnya bersama kesulitan itu ada kemudahan¹”



¹ Qs. Al-Insyirah (94:6).

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Wenny Hardiyanti Pratiwi

NIM : 132410101091

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Informasi Peramalan Jumlah Permintaan menggunakan Metode *Exponential Smoothing* (Studi Kasus : UTD PMI Kabupaten Jember)”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember,

Yang menyatakan,

Wenny Hardiyanti Pratiwi

NIM 132410101091

SKRIPSI

**SISTEM INFORMASI PERAMALAN JUMLAH PERMINTAAN DARAH
MENGUNAKAN METODE *EXPONENTIAL SMOOTHING*
(STUDI KASUS : UTD PMI KABUPATEN JEMBER)**

Oleh

Wenny Hardiyanti Pratiwi

NIM 132410101091

Pembimbing :

Dosen Pembimbing Utama : Anang Andrianto., S.T., M.T.

Dosen Pembimbing Pendamping : Diah Ayu Retnani W, ST.,M.Eng

PENGESAHAN PEMBIMBING

Skripsi berjudul “Sistem Informasi Peramalan Jumlah Permintaan menggunakan Metode *Exponential Smoothing* (Studi Kasus : UTD PMI Kabupaten Jember)” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh :

Pembimbing I,

Pembimbing II,

Anang Andrianto.,S.T.,M.T
NIP 196909281993021001

Diah Ayu Retnani W, ST.,M.Eng
NIP 198603052014042001

PENGESAHAN PENGUJI

Skripsi berjudul “Sistem Informasi Peramalan Jumlah Permintaan menggunakan Metode *Exponential Smoothing* (Studi Kasus : UTD PMI Kabupaten Jember)” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim penguji:

Penguji I,

Penguji II,

Nelly Oktavia A, S.Si., MT.

Fahrobby Adnan, S.Kom., M.MSI

NIP. 198410242009122008

NIP. 198706192014041001

Mengesahkan

Ketua Program Studi,

Prof. Drs. Slamin, M.Comp.Sc., Ph.D

NIP 196704201992011001

RINGKASAN

Sistem Informasi Peramalan Jumlah Permintaan menggunakan Metode *Exponential Smoothing* (Studi Kasus : UTD PMI Kabupaten Jember); Wenny Hardiyanti Pratiwi, 132410101091; 2017: 206 halaman; Progam Studi Sistem Informasi Universitas Jember.

UTD sebagai penyelenggara pelayanan darah dituntut untuk selalu memenuhi kebutuhan darah yang ada di Kabupaten Jember, terlebih lagi mengingat kebutuhan darah ini menyangkut kesehatan atau bahkan nyawa seseorang. Upaya pemenuhan kebutuhan darah tersebut dilakukan dengan cara mengontrol ketersediaan darah di UTD PMI itu sendiri. Untuk membantu mengontrol ketersediaan darah, UTD PMI Kabupaten Jember melakukan perencanaan kebutuhan darah. Perencanaan kebutuhan darah ini memerlukan teknik peramalan untuk memperkirakan berapa jumlah permintaan darah pada periode berikutnya. Salah satu metode peramalan adalah metode *exponential smoothing*.

Pada penelitian ini dibangun sistem informasi peramalan jumlah permintaan darah dengan menggunakan metode *exponential smoothing*, sehingga membantu dalam proses perencanaan kebutuhan darah. Perbandingan MAPE setiap metode *exponential smoothing* akan digunakan dalam penentuan metode mana yang tepat untuk diterapkan dalam perhitungan peramalan jumlah permintaan darah. Sistem ini telah di uji dengan membandingkan perhitungan manual dan sistem, dan menghasilkan metode serta hasil peramalan yang sama. Dataset yang digunakan meramalkan jumlah permintaan darah pada tanggal 14 Maret 2017 dan menghasilkan metode serta perhitungan yang sesuai pola data selama tanggal 7 – 13 Maret 2017.

PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul Sistem Informasi Peramalan Jumlah Permintaan menggunakan Metode *Exponential Smoothing* (Studi Kasus : UTD PMI Kabupaten Jember). Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

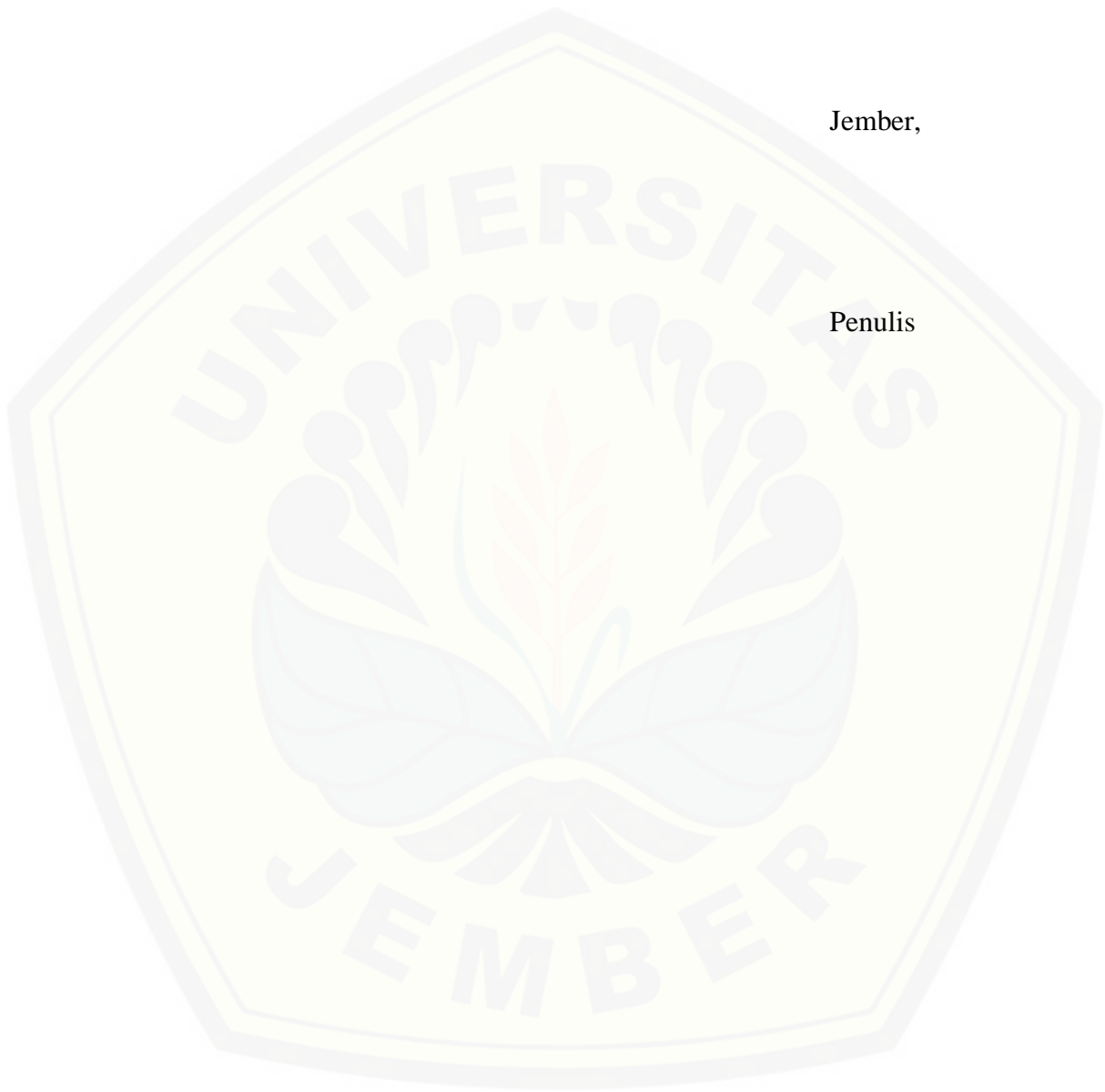
Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamin, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Anang Andrianto, S.T., M.T., selaku Dosen Pembimbing Utama dan Diah Ayu Retnani W, ST.,M.Eng, selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Muhammad Arief Hidayat S.Kom., M.Kom., sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di program studi sistem informasi;
5. Ibunda Ita Kustatik dan Ayahanda Zainal Abidin tersayang yang selalu memberikan dukungan do'a, kasih sayang, motivasi yang luar biasa;
6. Adik Dita Yusrilia Madini tercinta yang selalu memberi dukungan;
7. UTD PMI Kabupaten Jember sebagai lokasi penelitian yang telah meluangkan waktu yang membantu dalam melancarkan skripsi ini;
8. Keluarga besar INTENTION angkatan 2013 yang telah menjadi keluarga selama menempuh pendidikan S1;
9. Semua pihak yang tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember,

Penulis



DAFTAR ISI

PERSEMBAHAN	iii
MOTTO.....	iv
PERNYATAAN.....	v
SKRIPSI.....	vi
PENGESAHAN PEMBIMBING.....	vii
PENGESAHAN PENGUJI.....	viii
RINGKASAN	ix
PRAKATA.....	x
DAFTAR ISI.....	xii
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xvi
DAFTAR LAMPIRAN	xviii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan dan Manfaat.....	3
1.4. Batasan Masalah.....	4
1.5. Sitematika Penulisan	5
BAB 2 TINJAUAN PUSTAKA	7
2.1 Penelitian Terdahulu	7
2.2 Unit Transfusi Darah (UTD).....	8
2.3 Peramalan (Forecasting).....	9
2.4 Pola Data	10
2.5 Metode <i>Exponential Smoothing</i>	11
2.6 <i>Mean Absolute Percentage Error</i> (MAPE).....	15
BAB 3 METODOLOGI PENELITIAN	17
3.1 Jenis Penelitian.....	17
3.2 Waktu dan Tempat Penelitian	17

3.3 Tahapan Penelitian	17
BAB 4 PENGEMBANGAN SISTEM.....	22
4.1 Analisa Kebutuhan	22
4.2 Desain Sistem	24
4.3 Implementasi	41
4.4 Pengujian	48
BAB 5 HASIL DAN PEMBAHASAN	51
5.1 Sistem Informasi Peramalan Jumlah Permintaan Darah	51
5.2 Implementasi Metode Exponential Smoothing	69
5.3 Tingkat Kebenaran Peramalan	77
5.4 Perbandingan MAPE dalam Menentukan Metode yang Tepat	83
5.5 Pengujian Perhitungan Sistem Menggunakan Metode <i>Exponential Smoothing</i>	91
BAB 6 PENUTUP	93
6.1 Kesimpulan.....	93
6.2 Saran.....	94
DAFTAR PUSTAKA	95
LAMPIRAN	96
Lampiran A. <i>Scenario</i>	96
Lampiran B. Activity Diagram.....	137
Lampiran C. Sequence Diagram.....	154
Lampiran D. Kode Program.....	166
Lampiran E. Pengujian White Box.....	183
Lampiran F. Pengujian Black Box	186

DAFTAR TABEL

Tabel 4.10 Kode Program function updateRamal2 **Error! Bookmark not defined.**

Tabel 5.1 Hasil Peramalan dengan Perhitungan Manual Metode Single Exponential Smoothing 69

Tabel 5.2 Contoh perhitungan peramalan dengan metode single exponential smoothing 71

Tabel 5.3 Hasil Peramalan dengan Perhitungan Manual Metode Double Exponential Smoothing 71

Tabel 5.4 Contoh perhitungan peramalan dengan metode double exponential smoothing 72

Tabel 5.5 Hasil Peramalan dengan Perhitungan Manual Metode Triple Exponential Smoothing 74

Tabel 5.6 Contoh perhitungan peramalan dengan metode triple exponential smoothing 75

Tabel 5.7 PE peramalan single exponential smoothing selama bulan Maret 201777

Tabel 5.8 contoh perhitungan manual PE single exponential smoothing 78

Tabel 5.9 PE peramalan double exponential smoothing selama bulan Maret 2017 79

Tabel 5.10 contoh perhitungan manual PE double exponential smoothing 80

Tabel 5.11 PE peramalan triple exponential smoothing selama bulan Maret 2017 81

Tabel 5.12 contoh perhitungan manual PE triple exponential smoothing 82

Tabel 5.13 Perbandingan MAPE 84

Tabel A.0.1 Scenario Login Perawat 96

Tabel A.0.2 Scenario Login Admin Rumah Sakit 97

Tabel A.0.3 Scenario Login Admin Pelayanan UTD PMI 98

Tabel A.0.4 Scenario Login Admin Penjamin Mutu UTD PMI 99

Tabel A.0.5 Scenario Mengelola Data Permintaan Darah 100

Tabel A.0.6 Scenario Cetak Label Sample Darah 106

Tabel A.0.7 Scenario Mengelola Data Dokter 109

Tabel A.0.8 Scenario Mengelola Data Perawat	111
Tabel A.0.9 Scenario Mengelola Data Ruang Rumah Sakit.....	113
Tabel A.0.10 Scenario Melihat Data Permintaan Darah.....	115
Tabel A.0.11 Scenario Melihat Stok Darah oleh Admin Pelayanan UTD.....	117
Tabel A.0.12 Scenario Melihat Stok Darah oleh Admin Penjamin Mutu UTD .	117
Tabel A.0.13 Scenario Melihat Stok Darah oleh Perawat	118
Tabel A.0.14 Scenario Melihat Stok Darah oleh Admin Rumah Sakit	118
Tabel A.0.15 Scenario Melihat Stok Darah oleh User Umum.....	119
Tabel A.0.16 Scenario Validasi Data Permintaan Darah.....	120
Tabel A.0.17 Scenario Mengelola Data Pendoror.....	122
Tabel A.0.18 Scenario Menambah Data Transaksi Donor	126
Tabel A.0.19 Scenario Menambah Data Rumah Sakit	129
Tabel A.0.20 Scenario Melihat Data Transaksi Donor dan Transaksi Permintaan Darah.....	131
Tabel A.0.21 Scenario Verifikasi Kualitas Darah Donor	135

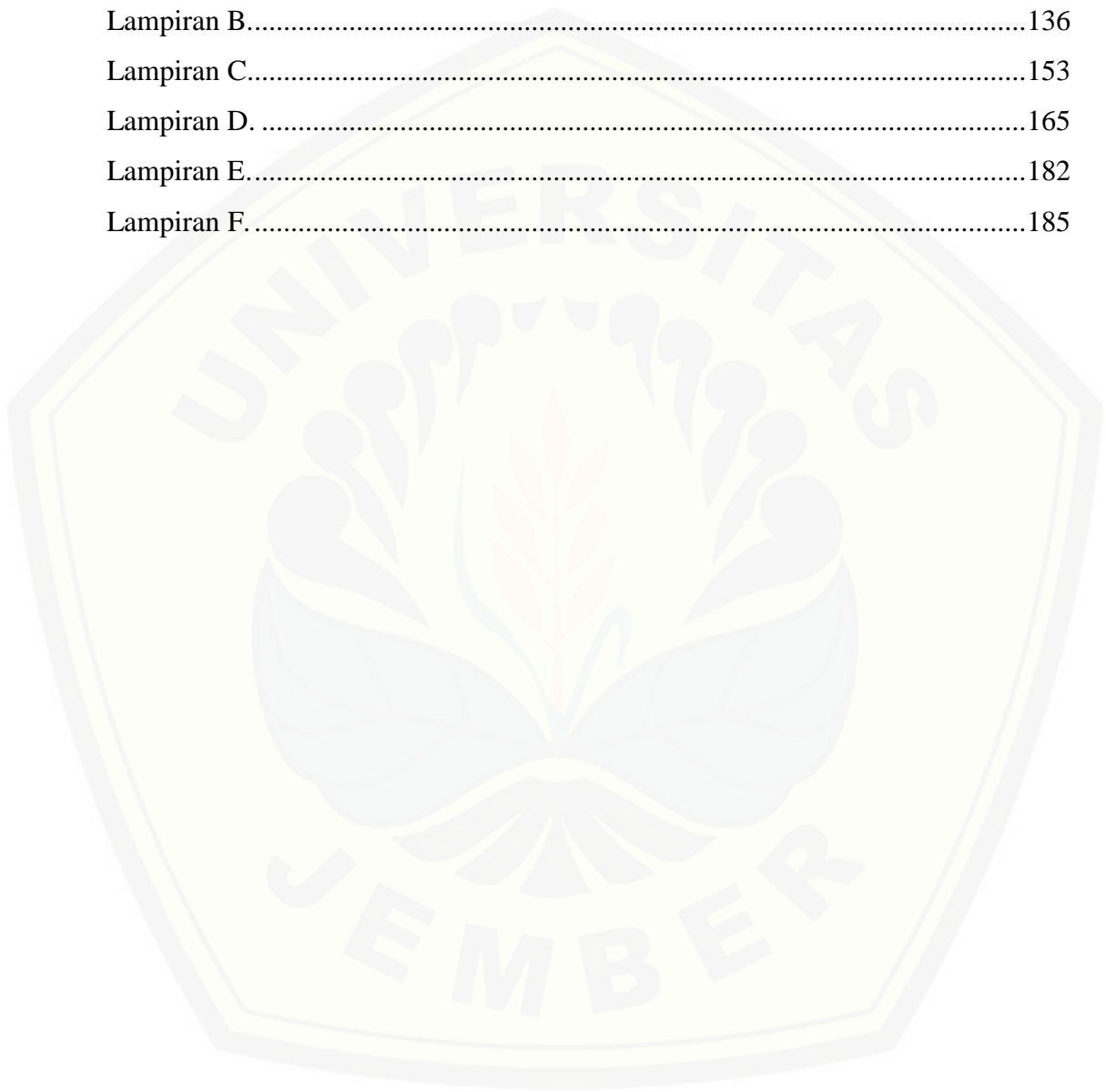
DAFTAR GAMBAR

Gambar 2.1 Pola Data Horizontal	10
Gambar 2.2 Pola Data Musiman	11
Gambar 2.3 Pola Data Trend.....	11
Gambar 3.1 Tahapan Pengembangan Sistem Model Waterfall	18
Gambar 4.1 Flowchart pemilihan metode dan perhitungan exponential smoothing	23
Gambar 4.2 Bussiness Process	25
Gambar 4.3 Usecase Diagram.....	26
Gambar 5.1 Beranda.....	51
Gambar 5.2 Halaman Login	52
Gambar 5.3Tampilan menu permintaan darah oleh perawat	52
Gambar 5.4 Detail tombol action	53
Gambar 5.5 Tampilan detail data permintaan oleh perawat	53
Gambar 5.6 Tampilan edit data permintaan oleh perawat	54
Gambar 5.7 Label Sample Darah dalam PDF	54
Gambar 5.8 Tampilan formulir permintaan darah	55
Gambar 5.9 Tampilan menu dokter oleh admin rumah sakit.....	56
Gambar 5.10 Form tambah data dokter.....	57
Gambar 5.11 Tampilan menu perawat	57
Gambar 5.12 Form tambah perawat.....	58
Gambar 5.13 Tampilan menu ruangan.....	58
Gambar 5.14 Form tambah data ruangan rumah sakit	59
Gambar 5.15 Tampilan menu permintaan darah oleh admin rumah sakit	59
Gambar 5.16 Tabel stok darah	60
Gambar 5.17 Tabel daftar permintaan darah belum divalidasi	61
Gambar 5.18 Halaman Validasi Permintaan	62
Gambar 5.19 Tampilan menu donor darah	63
Gambar 5.20 Form tambah data pendonor.....	63

Gambar 5.21 Detail data pendonor	64
Gambar 5.22 Tampilan riwayat donor dan form tambah data donor	65
Gambar 5.23 Tampilan menu rumah sakit	66
Gambar 5.24 Form data tambah rumah sakit	66
Gambar 5.25 Tabel data transaksi donor	67
Gambar 5.26 Tabel data transaksi permintaan darah	67
Gambar 5.27 Tampilan menu kualitas darah	68
Gambar 5.28 Tampilan menu ramalan permintaan	68
Gambar 5.29 Hasil Peramalan	69
Gambar 5.30 Grafik permintaan darah 1-7 Maret 2017	85
Gambar 5.31 Grafik permintaan darah 2-8 Maret 2017	86
Gambar 5.32 Grafik permintaan darah 3-9 Maret 2017	87
Gambar 5.33 Grafik permintaan darah 4-10 Maret 2017	88
Gambar 5.34 Grafik permintaan darah 5-11 Maret 2017	89
Gambar 5.35 Grafik permintaan darah 6-12 Maret 2017	90
Gambar 5.36 Grafik permintaan darah 7-13 Maret 2017	91
Gambar 5.37 Hasil perhitungan peramalan sistem	92

DAFTAR LAMPIRAN

	Halaman
Lampiran A.	95
Lampiran B.....	136
Lampiran C.....	153
Lampiran D.	165
Lampiran E.....	182
Lampiran F.....	185



BAB 1 PENDAHULUAN

Bab ini menjelaskan hal-hal yang melatarbelakangi penelitian dan pembuatan sistem informasi peramalan jumlah permintaan darah. Adapun yang akan dijelaskan antara lain adalah latar belakang penelitian, perumusan masalah, tujuan dan manfaat penelitian, batasan masalah, serta sistematika penulisan.

1.1. Latar Belakang

Unit Transfusi Darah yang selanjutnya disingkat UTD adalah fasilitas pelayanan kesehatan yang menyelenggarakan pelayanan donor darah, penyediaan darah, dan pendistribusian darah di Indonesia (Peraturan Pemerintah Republik Indonesia No. 7 Tahun 2011 tentang Pelayanan Darah, 2011). Sebanyak 417 UTD telah tersebar di seluruh kota/kabupaten yang ada di Indonesia (Kementrian Kesehatan, 2014), salah satunya di Kabupaten Jember. Semua UTD tersebut dikelola oleh Palang Merah Indonesia dan pemerintah masing-masing daerah.

Peran UTD sebagai penyelenggara pelayanan darah menuntut unit kesehatan tersebut untuk selalu memenuhi kebutuhan darah yang ada di Kabupaten Jember, terlebih lagi mengingat kebutuhan darah ini menyangkut kesehatan atau bahkan nyawa seseorang. Upaya pemenuhan kebutuhan darah tersebut dilakukan dengan cara mengontrol ketersediaan darah di UTD PMI itu sendiri, namun pada kenyataannya pihak UTD PMI Kabupaten Jember menemui beberapa hambatan dalam mengontrol ketersediaan darah. Diantaranya adalah sifat darah yang mudah rusak dan tidak dapat diproduksi di pabrik, serta ketersediaan darah yang bergantung pada pendonor yang mendonorkan darahnya secara sukarela.

Hambatan-hambatan yang telah disebutkan di atas mengharuskan UTD PMI Kabupaten Jember untuk melakukan perencanaan kebutuhan darah dan pengerahan serta pelestarian pendonor darah. Perencanaan kebutuhan darah merupakan hal yang sangat penting untuk dilakukan karena dengan perencanaan tersebut pihak UTD mampu memperkirakan berapa jumlah darah yang dibutuhkan dan berapa jumlah donor darah yang harus diperoleh untuk memenuhi kebutuhan

darah tersebut, sehingga UTD PMI Kabupaten Jember dapat memunculkan strategi-strategi baru dalam pengerahan serta pelestarian pendonor darah.

Pembuatan rencana kebutuhan darah memerlukan suatu teknik yang mampu memperkirakan berapa jumlah darah yang dibutuhkan. Salah satu teknik yang dapat membantu adalah teknik peramalan. Peramalan atau *forecasting* dapat didefinisikan sebagai suatu proses memperkirakan apa yang akan terjadi di masa yang akan datang dengan melihat pola-pola yang terbentuk dari fakta-fakta yang sudah ada sebelumnya.

Penelitian ini menerapkan peramalan deret waktu atau *time series* dengan menggunakan data masa lalu untuk menunjukkan bagaimana pola permintaan produk darah di UTD PMI Kabupaten Jember. Metode *time series* yang digunakan adalah metode *exponential smoothing*. Metode ini dipilih karena memiliki ketepatan yang baik untuk peramalan jangka pendek (MJ, 2014), sehingga sesuai diterapkan dalam penelitian ini dimana akan memperkirakan atau meramalkan jumlah permintaan darah di UTD PMI Kabupaten Jember dalam kurun waktu satu hari. Selain itu menurut (Makridakis, S dkk, 1983), metode *exponential smoothing* memberikan bobot yang lebih besar terhadap data baru dan bobot yang lebih kecil pada data yang lebih lama sehingga besaran data baru akan lebih diprioritaskan dalam peramalan daripada data yang lebih lama.

Metode *exponential smoothing* dapat dibagi menjadi tiga metode yang memiliki kriteria pola data berbeda-beda, yaitu *single exponential smoothing* yang mengikuti pola data stationer, *double exponential smoothing* yang mengikuti pola data *trend*, dan *triple exponential smoothing* yang mengikuti pola data musiman (Nazim & Aftanorhan, 2014). Ketiga metode *exponential smoothing* tersebut diterapkan dalam penelitian ini dengan tujuan untuk mengantisipasi perubahan pola runtut waktu data permintaan darah pada UTD PMI Kabupaten Jember sehingga sistem informasi peramalan jumlah permintaan darah yang akan dibangun melalui penelitian ini mampu melakukan peramalan dengan tingkat kebenaran yang tinggi.

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas, dapat dikemukakan permasalahan yang muncul dalam penelitian ini antara lain adalah:

1. Bagaimana melakukan peramalan jumlah permintaan darah pada UTD PMI Kabupaten Jember menggunakan metode *exponential smoothing*?
2. Bagaimana merancang dan membangun sebuah sistem informasi peramalan jumlah permintaan darah dengan menggunakan metode *exponential smoothing* pada UTD PMI Kabupaten Jember?
3. Bagaimana memilih metode *exponential smoothing* yang cocok dengan pola data jumlah permintaan darah di UTD PMI Kabupaten Jember?

1.3. Tujuan dan Manfaat

Berikut merupakan tujuan yang ingin dicapai dan manfaat yang ingin diperoleh dari penelitian ini :

1.3.1 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini antara lain :

1. Menerapkan metode *exponential smoothing* dalam peramalan jumlah permintaan darah di UTD PMI Kabupaten Jember.
2. Merancang dan membangun sebuah sistem informasi peramalan permintaan produk darah di UTD PMI Kabupaten Jember dengan menggunakan metode *exponential smoothing*.
3. Menerapkan metode *exponential smoothing* dalam sistem informasi peramalan permintaan darah di UTD PMI Kabupaten Jember sehingga sistem mampu memilih metode *exponential smoothing* yang tepat sesuai pola data yang ada.

1.3.2 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Manfaat akademis
 - a. Penelitian ini diharapkan mampu memberikan informasi kepada pembaca dan khususnya kepada Program Studi Sistem Informasi

Universitas Jember tentang hal-hal yang berkaitan dengan judul penelitian yang dilakukan.

- b. Penelitian ini diharapkan mampu menjadi referensi yang menjelaskan bagaimana keterkaitan setiap metode dalam *exponential smoothing* dengan kebutuhan pola data setiap metode tersebut.
2. Manfaat bagi peneliti
 - a. Mengetahui bagaimana proses penerapan metode *exponential smoothing* pada peramalan jumlah permintaan darah di UTD PMI Kabupaten Jember.
 - b. Mengetahui bagaimana pengaruh pola data pada perhitungan metode *exponential smoothing*.
 - c. Mengetahui bagaimana cara membantu pihak UTD PMI Kabupaten Jember dalam membuat perencanaan kebutuhan darah berdasarkan peramalan jumlah permintaan darah.
 3. Manfaat bagi objek penelitian
 - a. Membantu UTD PMI Kabupaten Jember dalam meramalkan jumlah permintaan darah berdasarkan data yang sudah ada.
 - b. Membantu UTD PMI Kabupaten Jember dalam melakukan perencanaan kebutuhan darah berdasarkan hasil peramalan sehingga UTD PMI Kabupaten Jember dapat memunculkan strategi baru dalam pengerahan serta pelestarian pendonor darah.

1.4. Batasan Masalah

Batasan masalah ditetapkan agar tidak terjadi penyimpangan dalam proses penelitian dan pembangunan sistem informasi peramalan jumlah permintaan darah menggunakan metode *exponential smoothing*. Adapun batasan-batasan masalah tersebut antara lain :

1. Objek yang digunakan pada penelitian ini adalah UTD PMI Kabupaten Jember.
2. Sistem yang dibuat dapat mengelola data transaksi donor dan transaksi permintaan darah yang ada di UTD PMI Kabupaten Jember.

3. Penentuan nilai peramalan jumlah permintaan darah di UTD PMI Kabupaten Jember dihitung menggunakan metode *single exponential smoothing*, *Brown's double exponential smoothing*, *Brown's triple exponential smoothing*.
4. Sistem mampu meramalkan jumlah permintaan darah tanpa membedakan golongan darah pada UTD PMI Kabupaten Jember setiap harinya.

1.5. Sitematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

1. **Pendahuluan**
Bab pendahuluan merupakan langkah awal dari penyusunan tugas akhir yang membahas latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan.
2. **Tinjauan Pustaka**
Bab tinjauan pustaka merupakan bab yang menjelaskan teori-teori yang melandasi penelitian, tinjauan pustaka, dan studi terdahulu yang menjadi acuan dalam penelitian.
3. **Metodologi Penelitian**
Bab metodologi penelitian menyajikan pembahasan tentang tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem yang digunakan dalam penelitian.
4. **Pengembangan Sistem**
Bab pengembangan sistem adalah bab yang berisi proses pengembangan sistem yang dibangun dalam penelitian. Adapun tahapan-tahapan pengembangan sistem ini didasarkan pada metodologi penelitian yang telah ditulis pada bab sebelumnya.
5. **Hasil dan Pembahasan**
Bab hasil dan pembahasan adalah bab yang menjelaskan secara rinci pemecahan masalah yang ada pada rumusan masalah yang telah dijelaskan pada bab pendahuluan.

6. Penutup

Bab penutup merupakan kesimpulan penelitian yang dilakukan dan saran untuk penelitian selanjutnya.



BAB 2 TINJAUAN PUSTAKA

Bab ini menguraikan teori serta konsep yang menjadi kerangka pemikiran dalam penelitian. Teori dan konsep tersebut didapatkan dari penelitian terdahulu serta kajian pustaka terkait penelitian.

2.1 Penelitian Terdahulu

Penelitian terdahulu berjudul "Implementasi Metode *Weighted Moving Average* dalam Prediksi Jumlah Stok Darah pada Palang Merah Indonesia Jember" yang dilakukan oleh Rizky Fauziah mahasiswi Program Studi Sistem Informasi Universitas Jember, telah menghasilkan suatu sistem peramalan stok darah PMI untuk menentukan stok darah perbulan. Penelitian tersebut menggunakan metode *weighted moving average* (Fauziah, 2016). Kelemahan metode WMA adalah metode ini hanya cocok diterapkan apabila pola data menunjukkan suatu *trend*. Hal ini berbeda dengan metode *exponential smoothing* yang dapat mengikuti pola data stationer, *trend*, maupun pola data musiman sehingga perubahan pola data yang terjadi pada data permintaan darah dapat diatasi.

Penelitian lain mengenai metode *exponential smoothing* juga dilakukan oleh Ni Putu Yuli Sukmarani, Stasiswaty, dan Rahmat Ramadhan, mahasiswa Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari pada penelitiannya yang berjudul "Penerapan Metode *Exponential Smoothing* pada Peramalan Penjualan dalam Penentuan Kuantitas Produksi Roti (Studi Kasus Perusahaan Roti Dhiba Kendari)". Penelitian ini telah menghasilkan sebuah aplikasi penerapan metode *exponential smoothing* dalam meramalkan penjualan untuk penentuan kuantitas produksi roti. Jurnal penelitian ini menyebutkan bahwa metode *exponential smoothing* cocok untuk diterapkan pada studi kasus yang memiliki data yang cenderung mengalami fluktuasi atau data yang tingkat kenaikan dan penurunan permintaan yang tidak menentu (Ni Putu Yuli Sukmarani, 2016).

2.2 Unit Transfusi Darah (UTD)

Unit Transfusi Darah merupakan fasilitas pelayanan kesehatan yang menyelenggarakan pendonor darah, penyediaan darah, dan pendistribusian darah (Menteri Kesehatan, 2014). Pengorganisasian UTD sebagaimana telah diatur dalam Peraturan Menteri Kesehatan Republik Indonesia nomor 83 tahun 2014 memiliki struktur yang terdiri dari :

1. Kepala UTD yang bertugas dan bertanggung jawab untuk menetapkan kebijakan teknis dan rencana kerja UTD, menentukan pola dan tata cara kerja, memimpin pelaksanaan kegiatan teknis UTD, melaksanakan pengawasan, pengendalian, dan evaluasi kegiatan UTD, serta melakukan koordinasi teknis dengan lintas sektor.
2. Penanggung jawab teknis pelayanan yang bertugas dan bertanggung jawab untuk melaksanakan kebijakan teknis dan rencana UTD, melaksanakan pola dan tata cara kerja pelayanan darah, melaksanakan pengawasan, pengendalian, dan evaluasi kegiatan pelayanan darah, serta melakukan koordinasi teknis pelayanan.
3. Penanggung jawab administrasi yang bertugas dan bertanggung jawab untuk melaksanakan kegiatan teknis administrasi, melaksanakan fungsi koordinasi, dan melaksanakan pengawasan, pengendalian serta evaluasi administrasi.
4. Penanggung jawab mutu yang bertugas dan bertanggung jawab untuk melaksanakan kebijakan teknis dan rencana kerja pengendalian mutu, melaksanakan pola dan tata cara kerja, melaksanakan kegiatan pengawasan, pengendalian mutu, evaluasi kegiatan mutu, dan melakukan koordinasi teknis pengendalian mutu.

Salah satu fungsi Unit Transfusi Darah adalah menyelenggarakan pendistribusian darah yang akan dilakukan apabila terdapat permintaan kebutuhan darah. Adapun prosedur permintaan darah ini sebagaimana telah dijelaskan pada (UTD PMI Kabupaten Jember, 2017) antara lain :

1. Permintaan darah harus dari Rumah Sakit atau klinik-klinik kesehatan

2. Formulir permintaan rangkap 5 harus terbawa lengkap pada saat permintaan darah ke UDD PMI Kabupaten Jember.
3. Setiap form permintaan darah yang masuk ke UDD PMI Kabupaten Jember harus diisi lengkap data, dan catatan medis yang bersangkutan, jenis produk komponen, jumlah kantong yang dibutuhkan.
4. Formulir permintaan darah harus disertai sampel menggunakan tabung reaksi tanpa menggunakan anti pembeku
5. Data sample darah tertera identitas pasien. Data tersebut harus sama dengan identitas yang tertera dalam formulir permohonan darah.

Adapun beberapa kondisi permintaan darah akan diolah terkait dengan sample darah pada poin nomor 5, yaitu adanya kesalahan tulis data sample darah, adanya keraguan terhadap identitas atau kualitas, dan sampel darah yang tidak disertai label.

2.3 Peramalan (Forecasting)

Peramalan berasal dari kata ramalan yang artinya adalah suatu situasi atau kondisi yang diperkirakan akan terjadi pada masa yang akan datang, sedangkan peramalan adalah bentuk kegiatannya. Peramalan merupakan kegiatan memperkirakan keadaan di masa yang akan datang melalui pengujian keadaan dimasa lalu.

Segala sesuatu serba tidak pasti dan sukar diperkirakan secara tepat dalam kehidupan sosial, oleh karena itu diperlukan peramalan. Peramalan yang dibuat diharapkan mampu memberikan nilai yang bisa meminimumkan kesalahan meramal (*forecast error*) yang biasanya diukur dengan *mean square error*, *mean absolute error*, dan sebagainya (Nazim & Aftanorhan, 2014).

Metode peramalan adalah cara untuk memperkirakan secara kuantitatif apa yang akan terjadi pada masa yang akan datang dengan dasar data yang relevan pada masa lalu. Baik tidaknya suatu peramalan yang disusun selain ditentukan oleh metode yang digunakan, juga ditentukan oleh baik tidaknya informasi yang digunakan. Selama informasi yang digunakan tidak dapat menyakinkan untuk

mendapat hasil yang bagus, hasil peramalan yang disusun juga akan sukar dipercaya ketepatannya. Keberhasilan dari suatu peramalan sangat ditentukan oleh:

- a. Pengetahuan teknik tentang pengumpulan informasi (data) masa lalu, dapat ataupun informasi tersebut bersifat kuantitatif.
- b. Teknik dan metode yang tepat dan sesuai dengan pola data yang dikumpulkan.

2.4 Pola Data

Ada beberapa pola data yang harus diperhatikan dalam peramalan, yaitu:

2.4.1 Pola Data Horizontal

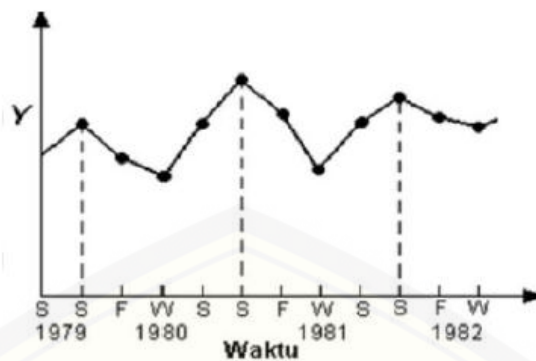


Gambar 2.1 Pola Data Horizontal

Pola ini terjadi jika terdapat data yang berfluktuasi disekitar nilai rata-rata yang konstan (Makridakis, 1999). Suatu produk yang suatu produk yang penjualannya tidak meningkat atau menurun selama waktu tertentu termasuk jenis pola ini. Adapun grafik pola data stationer dapat dilihat pada gambar 2.1 diatas.

2.4.2 Pola Data Musiman

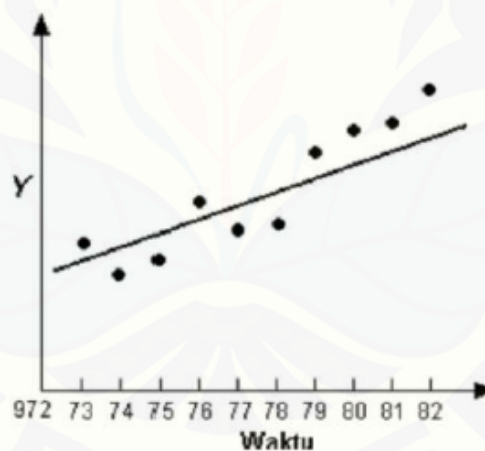
Pola data ini terjadi jika terdapat suatu deret data yang dipengaruhi oleh faktor musiman (misalnya kuartal tahun tertentu, bulanan, atau hari-hari pada minggu tertentu). Penjualan dari produk seperti minuman ringan, es krim, dan bahan bakar pemanas ruang semuanya menunjukkan jenis pola ini. Grafik pola data musiman dapat dilihat pada gambar 2.2.



Gambar 2.2 Pola Data Musiman

2.4.3 Pola Data Trend

Pola data trend terjadi jika terdapat kenaikan atau penurunan sekuler jangka panjang dalam data. Contoh : penjualan banyak perusahaan, GNP dan berbagai indikator bisnis atau ekonomi lainnya. Grafik pola data *trend* dapat dilihat pada gambar 2.4.



Gambar 2.3 Pola Data Trend

2.5 Metode *Exponential Smoothing*

Smoothing adalah mengambil rata-rata dari nilai pada beberapa periode untuk menaksir nilai pada suatu periode. *Exponential smoothing* adalah suatu peramalan rata-rata bergerak yang melakukan pembobotan menurun secara *exponential* terhadap nilai-nilai observasi yang lebih tua (Makridakis, 1993:79).

Metode *exponential smoothing* merupakan pengembangan dari metode *moving average*. Adapun metode *exponential smoothing* dibagi menjadi *single*

exponential smoothing, double exponential smoothing-Brown's one parameter linear method, double exponential smoothing-Holt's two parameter method, triple exponential smoothing-Brown's one parameter quadratic method, dan triple exponential smoothing-Winter's three parameter trend and seasonality method (Strasheim, 1992).

2.5.1 *Single Exponential Smoothing / SES*

Metode ini digunakan ketika data runtut waktu mengikuti pola data horizontal (Nazim & Aftanorhan, 2014). Bentuk umum yang digunakan untuk menghitung ramalan adalah:

$$\hat{Y}_{t+1} = \alpha Y_t + (1 - \alpha) \hat{Y}_t \quad \dots\dots\dots \text{Persamaan (2.1)}$$

Dimana:

- \hat{Y}_{t+1} = nilai ramalan untuk periode berikutnya
- α = konstanta pemulusan
- Y_t = data baru atau nilai Y yg sebenarnya pada periode t
- \hat{Y}_t = nilai pemulusan yang lama atau rata-rata pemulusan hingga periode t-1

2.5.2 *Double Exponential Smoothing / DES*

Metode ini digunakan dalam peramalan data runtut waktu yang mengikuti suatu *trend* (Nazim & Aftanorhan, 2014). Pada pengembangannya metode ini terbagi menjadi dua metode yang memiliki persamaan berbeda, yaitu metode *Brown's one parameter linear method* yang dikembangkan oleh Brown, dan *Holt's two parameter method* yang dikembangkan oleh Holt.

a. *Brown's one parameter linear method*

Dasar pemikiran metode pemulusan eksponensial linear dari Brown adalah serupa dengan rata-rata bergerak linear. Perbedaan antara nilai pemulusan tunggal dan ganda dapat ditambahkan dengan nilai pemulusan tunggal dan disesuaikan untuk trend (Noeryanti, Oktafiani, & Andriyani, 2012). Persamaan yang dipakai dalam implementasi pemulusan eksponensial linear satu-parameter dari Brown adalah sebagai berikut :

$$A_t = \alpha Y_t + (1 - \alpha)A_{t-1} \dots\dots\dots \text{Persamaan (2.2)}$$

$$A'_t = \alpha A_t + (1 - \alpha)A'_{t-1} \dots\dots\dots \text{Persamaan (2.3)}$$

$$a_t = 2A_t - A'_t \dots\dots\dots \text{Persamaan (2.4)}$$

$$b_t = \frac{\alpha}{1 - \alpha}(A_t - A'_t) \dots\dots\dots \text{Persamaan (2.5)}$$

Persamaan yang digunakan untuk membuat peramalan pada periode p yang akan datang adalah:

$$\hat{Y}_{t+p} = a_t + b_t p \dots\dots\dots \text{Persamaan (2.6)}$$

Dimana :

- A_t = nilai pemulusan eksponensial
- A'_t = nilai pemulusan eksponensial ganda
- α = konstanta pemulusan
- a_t = perbedaan antara nilai-nilai pemulusan eksponensial
- b_t = faktor penyesuai tambahan = pengukuran slope suatu kurva
- Y_t = nilai aktual pada periode t
- p = jumlah periode ke depan yang akan diramalkan

b. *Holt's two parameter method*

Metode pemulusan eksponensial linear dari Holt, pada prinsipnya adalah serupa dengan Brown kecuali bahwa Holt tidak menggunakan rumus pemulusan berganda secara langsung. Sebagai gantinya, Holt memuluskan nilai trend dengan parameter yang berbeda dari parameter yang digunakan pada deret yang asli (Noeryanti, Oktafiani, & Andriyani, 2012). Ramalan dari pemulusan eksponensial linear Holt didapat dengan menggunakan dua konstanta pemulusan (dengan nilai antara 0 dan 1) dan tiga persamaan :

$$S_t = \alpha X_t + (1 - \alpha)(S_{t-1} + b_{t-1}) \dots\dots\dots \text{Persamaan (2.7)}$$

$$b_t = \gamma(S_t - S_{t-1}) + (1 - \gamma)b_{t-1} \dots\dots\dots \text{Persamaan (2.8)}$$

$$F_{t+m} = S_t + b_t(m) \dots\dots\dots \text{Persamaan (2.9)}$$

Selain pada jumlah parameter yang digunakan, perbedaan antara *double exponential smoothing* oleh Brown dan Holt terletak pada kriteria datanya. Pola data yang diikuti kedua metode ini sama-sama memiliki pola *trend*, namun untuk data yang mengalami *trend* secara naik turun menggunakan metode Brown dan untuk data yang secara kontinyu mengalami *trend* naik atau *trend* turun menggunakan metode Holt (Inayah, 2010). Data permintaan darah pada UTD PMI Kabupaten Jember memiliki trend yang naik turun, oleh karena itu metode *double exponential smoothing* yang digunakan dalam penelitian ini adalah metode *Brown's one parameter linear method*.

2.5.3 Triple Exponential Smoothing / TES

Metode *triple exponential smoothing* dapat dijadikan cara untuk meramalkan data yang mengandung faktor musiman (Makridakis, 1999). Berdasarkan (Strasheim, 1992) metode ini dibagi menjadi dua metode yang memiliki persamaan berbeda yaitu *triple exponential smoothing-Brown's one parameter quadratic method* yang dikembangkan oleh Brown, dan *triple exponential smoothing-Winter's three parameter trend and seasonality method* yang dikembangkan oleh Winter.

a. *Brown's one parameter quadratic method*

Sesuai namanya, metode ini memiliki satu parameter yaitu parameter α . Metode ini lebih cocok digunakan untuk membuat *forecast* data yang berfluktuasi atau mengalami gelombang pasang surut (Arifianto, Nohe, & Goejantoro, 2013). Bentuk umum yang digunakan untuk menghitung ramalan adalah :

$$A_t = \alpha Y_t + (1 - \alpha) A_{t-1} \dots\dots\dots \text{Persamaan (2.10)}$$

$$A'_t = \alpha A_t + (1 - \alpha) A'_{t-1} \dots\dots\dots \text{Persamaan (2.11)}$$

$$A''_t = \alpha A_t + (1 - \alpha) A''_{t-1} \dots\dots\dots \text{Persamaan (2.12)}$$

$$at = (3At) - (3A't) + A''t \dots\dots\dots \text{Persamaan (2.13)}$$

$$bt = \alpha / ((2(1-\alpha))((6-(5\alpha)At)-(10-(8\alpha)A't)+(4-(3\alpha)A''t))) \dots\dots\dots \text{Persamaan (2.14)}$$

$$ct = \alpha^2 / ((1-\alpha)^2 (At-(2A't)+A''t)) \dots\dots\dots \text{Persamaan (2.15)}$$

Persamaan yang digunakan untuk membuat peramalan pada periode yang akan datang adalah:

$$Fx = at + bt + (0.5ct) \dots\dots\dots \text{Persamaan (2.16)}$$

b. *Winter's three parameter trend and seasonality method*

Menurut Makridakis, Wheelright, dan McGee (1999), metode *Winter's* merupakan metode yang dapat menangani faktor-faktor musiman dan *trend* secara langsung. Metode ini didasarkan atas tiga persamaan pemulusan dengan tiga parameter, yaitu satu untuk unsur keseluruhan, satu untuk *trend*, dan satu untuk musiman (Arifianto, Nohe, & Goejantoro, 2013).

• Pemulusan Keseluruhan

$$S_t = \alpha \frac{X_t}{I_{t-L}} + (1 - \alpha)(S_{t-1} + b_{t-1}) \dots\dots\dots \text{Persamaan (2.17)}$$

• Pemulusan Trend

$$b_t = \gamma(S_t - S_{t-1}) + (1 - \gamma)b_{t-1} \dots\dots\dots \text{Persamaan (2.18)}$$

• Pemulusan Musiman

$$I_t = \beta \frac{X_t}{S_t} + (1 - \beta)I_{t-L} \dots\dots\dots \text{Persamaan (2.19)}$$

• Ramalan

$$F_{t+m} = (S_t + b_t m)I_{t-L+m} \dots\dots\dots \text{Persamaan (2.20)}$$

Data permintaan darah pada UTD PMI Kabupaten Jember memiliki fluktuasi atau gelombang yang mengalami pasang surut atau dengan kata lain tidak memiliki panjang periode musim yang pasti. Oleh karena itu penelitian ini memilih menerapkan metode Brown daripada metode Winter yang menghitung pemulusan musiman dan *trend* sekaligus.

2.6 *Mean Absolute Percentage Error (MAPE)*

Metode peramalan yang baik adalah metode peramalan yang memberikan tingkat kesalahan yang paling kecil. Tingkat kesalahan merupakan selisih antara nilai aktual dengan nilai peramalan. Kesalahan dalam peramalan dapat disebabkan karena nilai peramalan terlalu kecil atau terlalu besar disbanding nilai aktual yang ada. Nilai kesalahan tidak dapat secara langsung dijumlahkan atau dirata-rata karena kesalahan ada yang bertanda positif dan ada yang bertanda negatif. Nilai

kesalahan atau error harus dimutlakkan atau dikuadratkan terlebih dahulu untuk menghilangkan tanda negatifnya (Suliyanto, 2008).

Teknik perhitungan *Mean Absolute Percentage Error* (MAPE) dilakukan dengan mencari persentase nilai rata-rata kesalahan. Untuk menggunakan teknik ini, langkah-langkahnya adalah sebagai berikut (Jogiyanto, 2005):

1. Menghitung persentase kesalahan peramalan (PE) dengan rumus pada persamaan 2.14.
2. Membagi jumlah nilai absolut kesalahan dengan jumlah pengamatan seperti pada persamaan 2.15.

$$PE = \sum \frac{|X_t - F_t|}{X_t} \times 100\% \dots\dots\dots \text{Persamaan (2.14)}$$

$$MAPE = \frac{PE}{n} \dots\dots\dots \text{Persamaan (2.15)}$$

Dimana :

n = banyaknya periode

X_t = Nilai aktual pada periode-t

F_t = Nilai peramalan pada periode-t

BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan tentang sekumpulan metode-metode yang digunakan dalam penelitian. Selain itu bab ini juga menjelaskan langkah dan prosedur yang dilakukan dalam pengumpulan data atau informasi guna memecahkan permasalahan dalam penelitian.

3.1 Jenis Penelitian

Penelitian ini menggunakan dua jenis penelitian, yaitu penelitian kualitatif dan penelitian kuantitatif. Penelitian jenis kualitatif digunakan dalam proses analisa studi literatur yang berhubungan dengan sistem informasi peramalan jumlah permintaan darah menggunakan metode *exponential smoothing* pada UTD PMI Kabupaten Jember. Sedangkan penelitian jenis kuantitatif digunakan pada proses pengolahan data dalam bentuk angka yang bertujuan untuk mendapatkan parameter dan perhitungan. Parameter dan perhitungan ini kemudian digunakan untuk mengembangkan sistem.

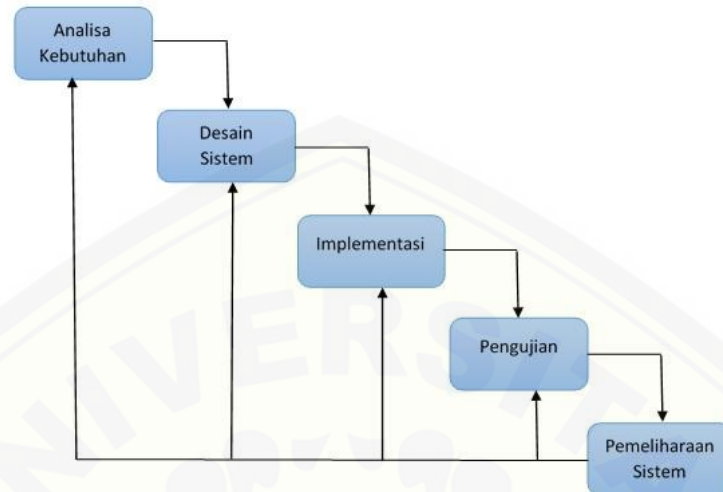
3.2 Waktu dan Tempat Penelitian

Studi kasus penelitian ini bertempat di UTD PMI Kabupaten Jember. Waktu penelitian dilakukan selama lima bulan, dimulai bulan November 2016 sampai dengan bulan April 2016.

3.3 Tahapan Penelitian

Tahapan penelitian ini dilakukan sesuai dengan metode SDLC (*System Development Life Cycle*) model *waterfall*. Model *waterfall* dipilih karena model ini memiliki tahapan pengembangan sistem yang dilakukan secara terurut, mulai dari pengumpulan kebutuhan, desain sistem, implementasi sistem, *testing* dan *maintenance*, sehingga model ini mampu menghasilkan sistem yang terstruktur dengan baik. Selain itu, model ini baik diterapkan dalam pengembangan sistem berskala kecil sehingga cocok dengan sistem informasi berskala kecil yang nantinya dibangun melalui penelitian ini.

Adapun model waterfall menurut (Basuki, 2006) memiliki tahapan-tahapan seperti yang dapat dilihat pada Gambar 3.1 berikut:



Gambar 3.1 Tahapan Pengembangan Sistem Model *Waterfall*

3.3.1 Analisa Kebutuhan

Analisa kebutuhan merupakan tahap awal dalam penelitian. Tahap ini terbagi lagi menjadi dua tahap, tahap yang pertama adalah tahap pengumpulan data dan tahap yang kedua adalah tahap pengolahan data.

3.3.1.1 Tahap Pengumpulan Data

Tahap pengumpulan data bertujuan untuk memperoleh data atau informasi yang dibutuhkan dalam mencapai tujuan penelitian. Pengumpulan data pada penelitian ini dilakukan melalui beberapa cara yaitu:

- a. Studi literatur, jurnal, serta mencari informasi menggunakan media internet tentang darah, UTD PMI, dan metode peramalan khususnya metode *exponential smoothing*.
- b. Wawancara dengan pegawai UTD PMI Kabupaten Jember untuk mengetahui prosedur donor dan permintaan darah, serta permasalahan terkait permintaan darah yang ada disana.

Adapun data dan informasi yang didapatkan dari tahap ini antara lain data jumlah permintaan darah per hari selama bulan Maret 2017 tanpa membedakan golongan darah, prosedur donor darah, prosedur permintaan darah, dan metode *exponential smoothing*. Data jumlah permintaan darah diambil per hari karena

sesuai dengan batasan masalah penelitian ini yang meramalkan jumlah permintaan darah perhari.

3.3.1.2 Tahap Pengolahan Data

Pengolahan data dilakukan untuk menentukan kebutuhan sistem informasi peramalan jumlah permintaan darah berdasarkan data dan prosedur-prosedur yang telah didapatkan dari studi literatur dan wawancara yang dilakukan sebelumnya. Selain itu data jumlah permintaan darah akan diolah untuk mendapatkan hasil peramalan dengan menggunakan metode *exponential smoothing*.

3.3.2 Desain Sistem

Pembuatan desain sistem pada penelitian ini menggunakan *Unified Modeling Language (UML)* yang dirancang dengan konsep *Object-Oriented Design (OOD)*. Pemodelan UML yang digunakan adalah sebagai berikut :

1. *Business Proccess*

Business Proccess merupakan diagram yang menggambarkan proses yang lengkap, diantaranya adalah input, event yang mendorong terjadinya proses dan *goal* yang dituju.

2. *Use Case Diagaran*

Use case adalah model yang menggambarkan fungsi atau tugas yang dilakukan oleh user.

3. *Scenario*

Scenario Diagram berfungsi untuk menjelaskan alur sistem dari fitur yang ada di *job specification* dan *job description* yang ada pada *Use Case Diagram*.

4. *Activity Diagram*

Activity Diagram digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi. *Activity diagram* mempunyai fungsi yang sama dengan *scenario* namun diimplementasikan dalam diagram alir.

5. *Sequence Diagram*

Sequence diagram menggambarkan aliran logika interaksi antar objek yang mengindikasikan komunikasi antar obyek di dalam sistem yang disusun pada suatu urutan atau rangkaian waktu.

6. *Class Diagram*

Class diagram menggambarkan struktur dan deskripsi class serta hubungan antar class yang dibuat untuk memudahkan proses pengkodean.

7. *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) merupakan gambaran struktur basisdata yang dibangun.

3.3.3 Implementasi

Tahapan ini merupakan proses pengimplementasian desain sistem yang telah dibuat kedalam penulisan kode program *HTML*, *CSS*, dan *PHP* untuk website. Selain itu dalam tahap implementasi ini juga dilakukan manajemen basis data menggunakan *DBMS MySQL*.

3.3.4 Pengujian

Setelah tahap implementasi selesai, tahap selanjutnya adalah pengujian terhadap sistem yang telah dibangun. Penelitian ini melakukan pengujian sistem dengan dua cara yaitu:

1. *White Box Testing*

White box testing merupakan pengujian perangkat lunak yang dilakukan dengan cara melihat modul yang telah dibuat dengan program yang ada. Pengujian ini menggunakan *Cyclomatic Complexity* (CC) untuk mengukur tingkat kompleksitas program dengan cara menelusuri jalur-jalur yang ada melalui *source codenya*. Rumus CC dapat dilihat pada persamaan 3.1.

$$CC = (Edge - Node) + 2 \dots\dots\dots (persamaan 3.1)$$

2. *Black Box Testing*

Berbeda dengan *white box testing*, *black box testing* merupakan pengujian perangkat lunak yang dilakukan oleh *user* dengan cara memeriksa fungsionalitas dari sistem yang berkaitan dengan struktur internal atau kerja saat sistem dijalankan.

3.3.5 Pemeliharaan Sistem

Tahap pemeliharaan merupakan tahap dimana sistem ini akan langsung di implementasikan di UTD PMI Kabupaten Jember. Tujuan dari tahapan implementasi agar pihak UTD menguasai cara kerja serta apa yang diperoleh dari sistem.



BAB 4 PENGEMBANGAN SISTEM

Bab ini menjelaskan tahapan pengembangan sistem informasi peramalan jumlah permintaan darah menggunakan metode *exponential smoothing*. Tahapan-tahapan perancangan dilaksanakan berdasarkan model *waterfall* seperti yang telah dijelaskan pada bab tiga.

4.1 Analisa Kebutuhan

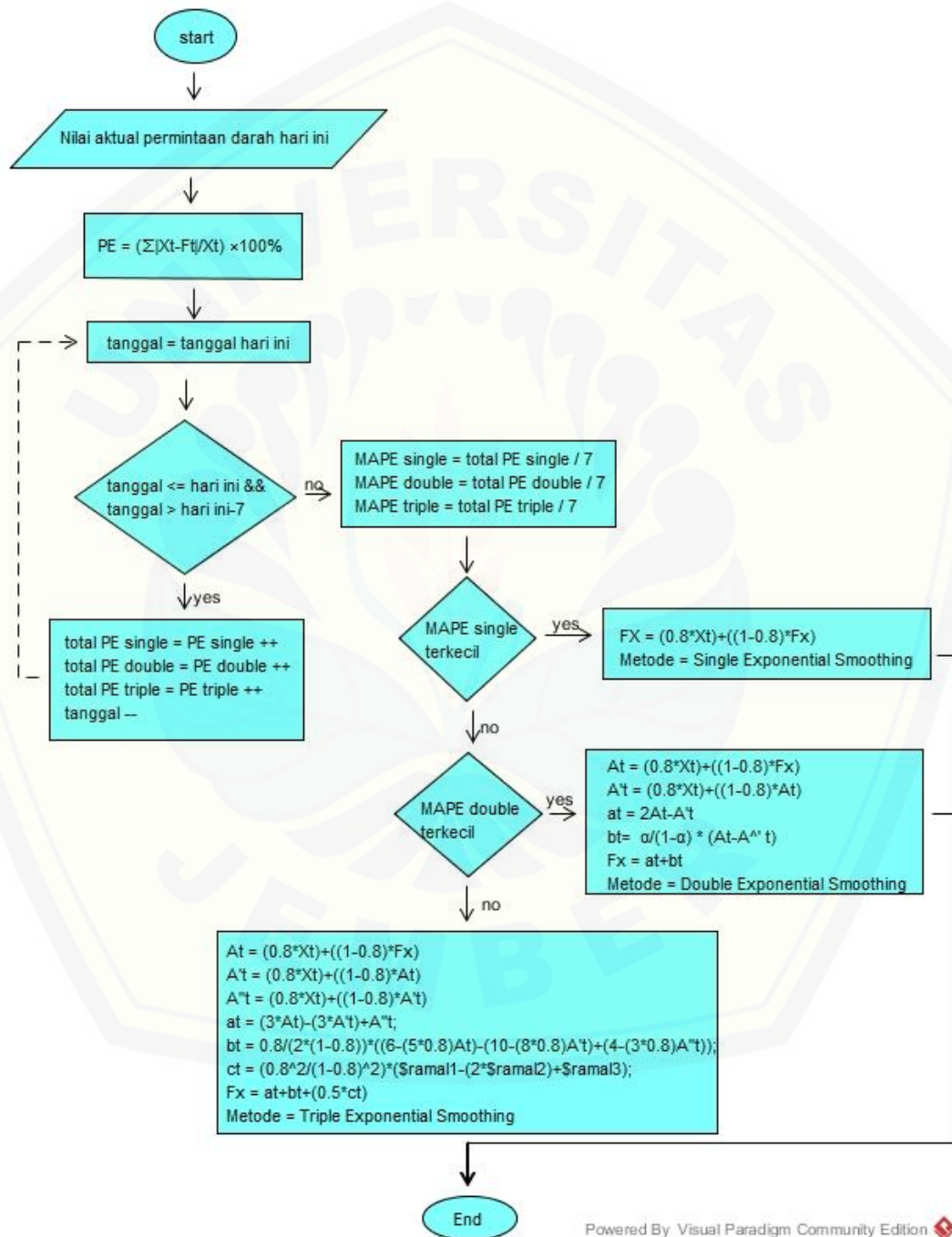
Tahap ini menggunakan hasil dari pengumpulan dan analisa data yang telah didapatkan sebagai dasar penentuan kebutuhan sistem informasi peramalan jumlah permintaan darah sebagaimana telah dijelaskan pada sub bab 3.3. Kebutuhan-kebutuhan sistem tersebut didefinisikan dalam kebutuhan fungsional dan kebutuhan non-fungsional yang menjabarkan hal apa saja yang mampu dilakukan oleh sistem.

4.1.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang menggambarkan proses apa saja yang mampu dilakukan oleh sistem. Kebutuhan fungsional dari sistem informasi peramalan jumlah permintaan darah antara lain :

1. Sistem mampu mengelola data *user* (*view, insert, edit*)
2. Sistem mampu menampilkan stok darah terbaru
3. Sistem mampu mengelola data transaksi permintaan (*view, insert, edit*)
4. Sistem mampu mengelola data dokter (*view, insert, edit*)
5. Sistem mampu mengelola data rumah sakit (*view, insert, edit*)
6. Sistem mampu mengelola data ruang rumah sakit (*view, insert, edit*)
7. Sistem mampu mengelola data pendonor (*view, insert, edit*)
8. Sistem mampu mengelola data transaksi donor (*view, insert*)
9. Sistem mampu menentukan metode *exponential smoothing* mana yang cocok digunakan untuk peramalan hari selanjutnya sesuai dengan data permintaan selama tujuh hari kebelakang
10. Sistem mampu mengelola data ramalan permintaan darah (*view, insert*)

Terkait dengan kebutuhan fungsional sistem poin sembilan, berikut *flowchart* dalam penentuan metode *exponential smoothing* mana yang cocok digunakan untuk peramalan beserta perhitungan hasil peramalan permintaan darah pada Gambar 4.1



Gambar 4.1 *Flowchart* pemilihan metode dan perhitungan *exponential smoothing*

4.1.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang tidak berkaitan dengan fungsi sistem tetapi diperlukan untuk mendukung aktivitas sistem dalam memenuhi kebutuhan fungsionalnya. Kebutuhan non-fungsional dari sistem informasi peramalan jumlah permintaan darah yaitu sistem menggunakan *username* dan *password* sebagai autentifikasi akses terhadap sistem.

4.2 Desain Sistem

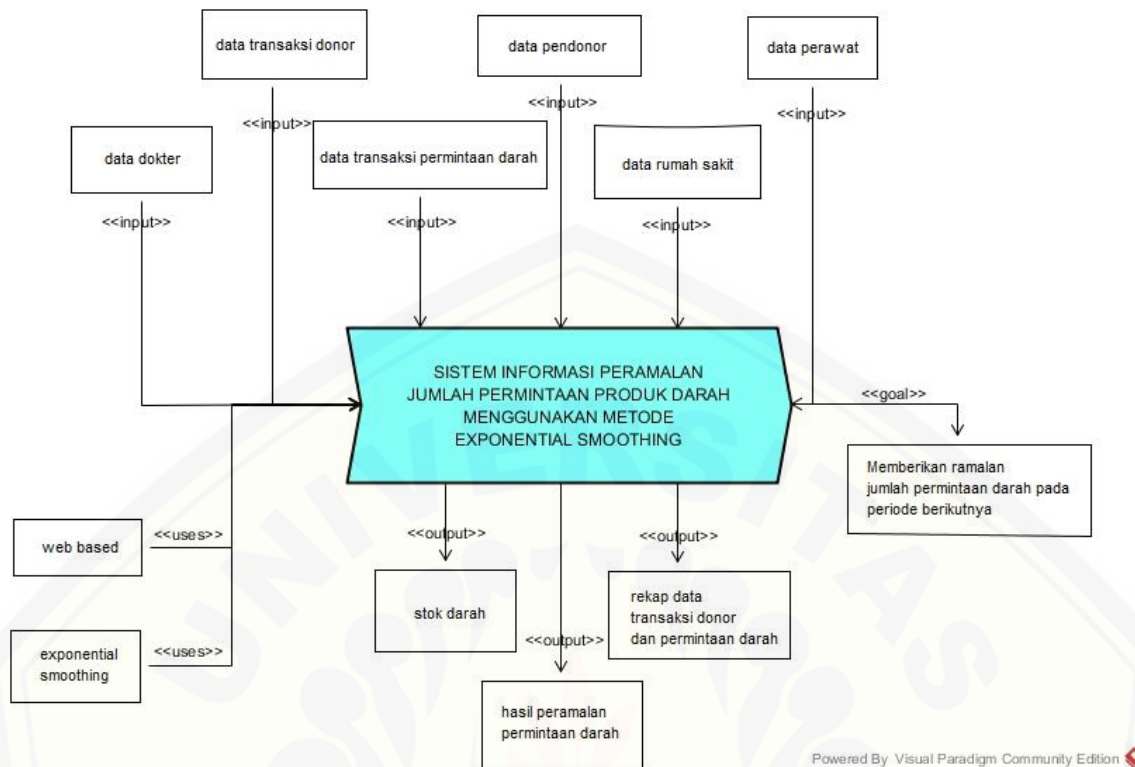
Desain sistem ini dibuat berdasarkan analisa kebutuhan sistem yang telah ditetapkan pada tahap sebelumnya. Seperti yang telah dijelaskan pada bab tiga, desain sistem pada penelitian ini meliputi *business process*, *use case diagram*, *scenario*, *sequence diagram*, *activity diagram*, *class diagram*, dan *Entity Relationship Diagram (ERD)*.

4.2.1 *Bussiness Process*

Business Process merupakan diagram yang menggambarkan proses dalam sistem secara lengkap berdasarkan kebutuhan fungsional dan non-fungsional sistem. Diagram ini meliputi :

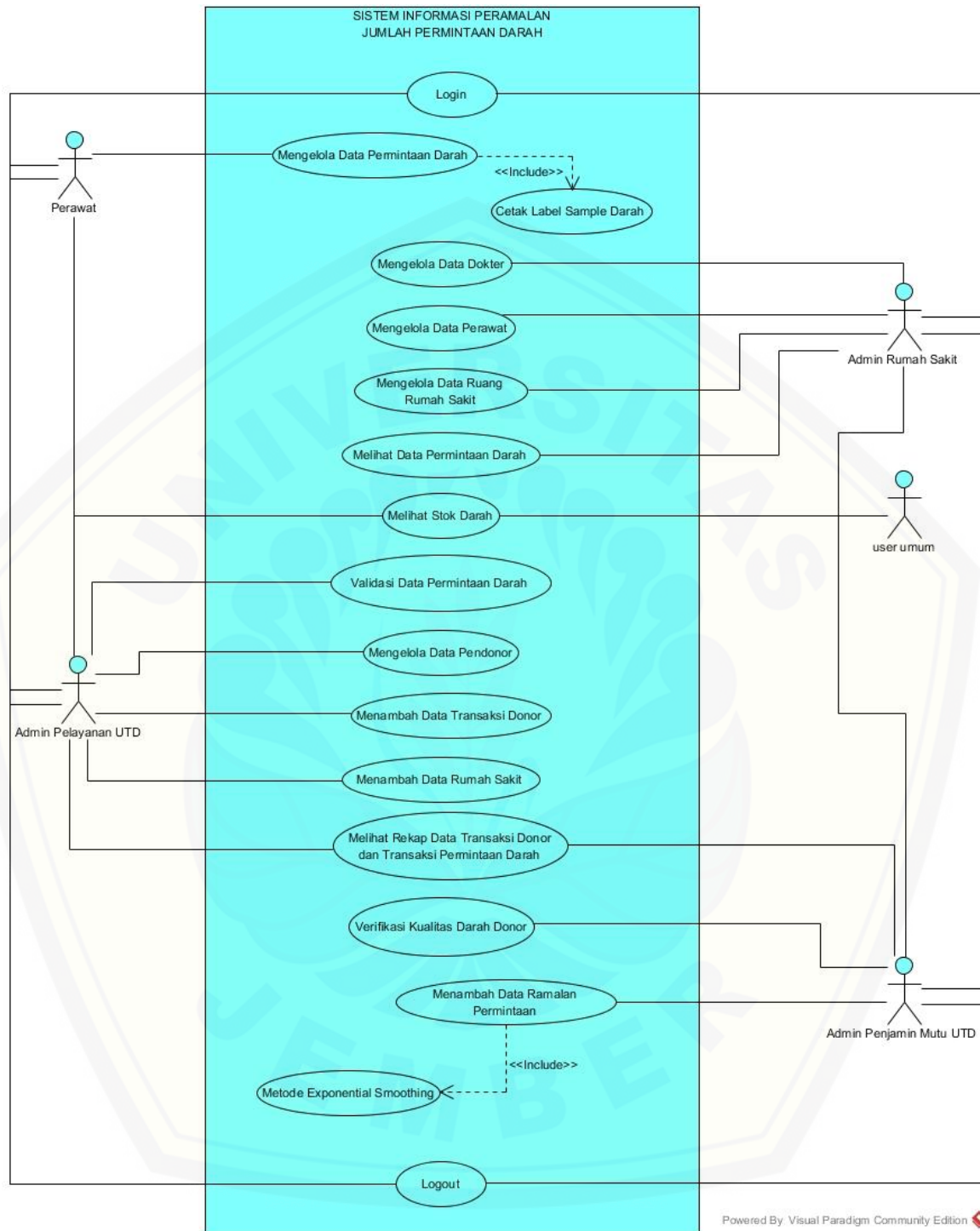
1. Data masukan sistem (*input*) yang merupakan data masukan yang dibutuhkan sistem untuk memenuhi kebutuhan fungsional dan non-fungsionalnya.
2. Data keluaran sistem (*output*) yang merupakan data yang dihasilkan sistem dari data masukan dan proses pengolahan sistem untuk dapat memenuhi kebutuhan fungsional dan non-fungsional sistem.
3. Tujuan sistem (*goal*) merupakan tujuan dari sistem informasi peramalan jumlah permintaan darah itu sendiri, yaitu memberikan hasil peramalan jumlah permintaan darah.
4. Media yang digunakan (*uses*) merupakan bentuk sistem dan metode yang diterapkan dalam sistem.

Business Process sistem informasi peramalan jumlah permintaan darah dapat dilihat pada Gambar 4.2.

Gambar 4.2 *Bussiness Process*

4.2.2 *Usecase Diagram*

Tahapan analisis kebutuhan sistem menghasilkan siapa saja aktor yang terlibat dalam sistem informasi peramalan jumlah permintaan darah. *Usecase diagram* merupakan gambaran interaksi antara aktor-aktor tersebut dengan fitur sistem informasi peramalan jumlah permintaan darah yang dibuat untuk memenuhi kebutuhan fungsional dan non-fungsional sistem. Selain menyajikan *usecase diagram* dari sistem informasi peramalan jumlah permintaan darah pada Gambar 4.3, sub bab ini akan memberikan penjelasan tentang setiap aktor dan *usecase* yang ada.



Gambar 4.3 Usecase Diagram

1. Definisi Aktor

Berikut merupakan penjelasan aktor atau pengguna pada sistem informasi peramalan jumlah permintaan darah. Terdapat lima actor yang dijelaskan pada tabel 4.1 berikut ini.

Tabel 4.1. Definisi Aktor

No.	Aktor	Deskripsi
1.	Perawat	Aktor perawat memiliki hak akses untuk mengelola data permintaan darah kepada UTD sesuai kebutuhan pasien yang ditetapkan oleh dokter penanggung jawab pasien.
2.	Admin Rumah Sakit	Aktor admin rumah sakit merupakan pemegang hak akses penuh dalam pengelolaan rumah sakit. Admin rumah sakit dapat mengelola data perawat dan data dokter yang bekerja pada rumah sakit tersebut, serta mengelola data ruang yang terdapat pada rumah sakit. Selain itu, admin rumah sakit juga dapat melihat data permintaan darah yang telah dibuat oleh perawat rumah sakit tersebut.
3.	Admin Pelayanan UTD	Aktor admin pelayanan UTD memiliki hak akses yang meliputi seluruh pelayanan UTD PMI, baik itu pengelolaan data donor darah, maupun data permintaan darah yang diajukan rumah sakit. Selain itu admin pelayanan UTD juga mampu mengelola data rumah sakit mitra UTD.
4.	Admin Penjamin Mutu UTD	Aktor penjamin mutu UTD merupakan penanggung jawab kualitas darah donor yang masuk serta memiliki hak akses untuk melakukan peramalan permintaan darah yang bertujuan untuk membantu perencanaan pencarian donor agar mampu memenuhi permintaan darah yang ada.
5.	User umum	Aktor <i>user</i> merupakan pengguna umum dari sistem informasi peramalan jumlah permintaan darah. Aktor ini tidak dapat melakukan login dan hanya memiliki hak akses untuk melihat beranda dari system berbasis <i>web</i> ini.

2. Definisi *Usecase*

Definisi *usecase* menjelaskan setiap *usecase* dalam sistem informasi peramalan jumlah permintaan darah. Definisi *usecase* ini dapat dilihat pada tabel 4.2.

Tabel 4.2. Definisi *Usecase*

No.	<i>Usecase</i>	Deskripsi
1.	<i>Login</i>	Menggambarkan proses autentifikasi user untuk masuk ke dalam sistem
2.	Mengelola data permintaan darah	Menggambarkan proses menambah, mengubah, dan melihat data permintaan darah.
3.	Cetak label <i>sample</i> darah	Menggambarkan proses cetak label yang hanya dapat dilakukan saat data permintaan darah telah ditambahkan
4.	Mengelola data dokter	Menggambarkan proses menambah, mengubah, dan melihat data dokter.
5.	Mengelola data perawat	Menggambarkan proses menambah, mengubah, dan melihat data perawat.
6.	Mengelola data ruang rumah sakit	Menggambarkan proses menambah, mengubah, dan melihat data ruang yang ada pada rumah sakit.
7.	Melihat data permintaan darah	Menggambarkan proses melihat data permintaan darah yang telah ditambahkan oleh perawat rumah sakit.
8.	Melihat stok darah	Menggambarkan proses melihat stok darah pada UTD PMI yang dapat dilihat oleh semua aktor.
9.	Validasi data permintaan darah	Menggambarkan proses persetujuan permintaan darah yang diajukan rumah sakit.
10.	Mengelola data pendonor	Menggambarkan proses menambah, mengubah, dan melihat data pendonor.
11.	Menambah data transaksi darah	Menggambarkan proses penambahan data darah masuk atau data donor darah.
12.	Menambahkan data rumah sakit	Menggambarkan proses penambahan data rumah sakit mitra UTD PMI.
13.	Melihat rekap data transaksi donor dan transaksi permintaan darah	Menggambarkan proses melihat data darah masuk (donor darah) dan data darah keluar (permintaan darah).
14.	Verifikasi kualitas darah donor	Menggambarkan proses verifikasi kualitas darah donor untuk memastikan bahwa darah yang masuk layak untuk di transfusikan pada pasien yang membutuhkan.
15.	Menambah data ramalan permintaan	Menggambarkan proses penambahan data peramalan permintaan darah.

No.	Usecase	Deskripsi
16.	Logout	Menggambarkan proses <i>user</i> keluar dari sistem.

4.2.3 Scenario

Scenario merupakan penjelasan alur sistem sesuai dengan yang telah digambarkan dalam *Usecase Diagram*. *Scenario* sistem meliputi penjelasan alur berjalannya sistem dalam *scenario* normal dan alternatif.

1. Scenario Login

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif *login* dijelaskan pada lampiran A.

2. Scenario Mengelola Data Permintaan Darah

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif mengelola data permintaan darah dijelaskan pada lampiran A.

3. Scenario Cetak Label *Sample* Darah

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif cetak label *sample* darah dijelaskan pada lampiran A.

4. Scenario Mengelola Data Dokter

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif mengelola data dokter dijelaskan pada lampiran A.

5. Scenario Mengelola Data Perawat

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif mengelola data perawat dijelaskan pada lampiran A.

6. Scenario Mengelola Data Ruang Rumah Sakit

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif mengelola data ruang rumah sakit dijelaskan pada lampiran A.

7. *Scenario* Melihat Data Permintaan Darah

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif melihat data permintaan darah dijelaskan pada lampiran A.

8. *Scenario* Melihat Stok Darah

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif melihat stok darah dijelaskan pada lampiran A.

9. *Scenario* Validasi Data Permintaan Darah

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif validasi data permintaan darah dijelaskan pada lampiran A.

10. *Scenario* Mengelola Data Pendorong

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif mengelola data pendorong dijelaskan pada lampiran A.

11. *Scenario* Menambah Data Transaksi Donor

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif menambah data transaksi donor dijelaskan pada lampiran A.

12. *Scenario* Menambah Data Rumah Sakit

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif menambah data rumah sakit dijelaskan pada lampiran A.

13. *Scenario* Melihat Data Transaksi Donor dan Transaksi Permintaan Darah

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif melihat data transaksi donor dan transaksi permintaan darah dijelaskan pada lampiran A.

14. *Scenario* Verifikasi Kualitas Darah Donor

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif verifikasi kualitas darah donor dijelaskan pada lampiran A.

15. *Scenario* Menambah Data Ramalan Permintaan

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif menambah data ramalan permintaan dijelaskan pada tabel 4.3 dibawah ini.

Tabel 4.3 *Scenario* Menambahkan Data Ramalan Permintaan

Nama Use Case	Menambah Data Ramalan Permintaan Darah
Aktor	Admin Penjamin Mutu UTD
Deskripsi Singkat	Admin penjamin mutu UTD akan menambah data ramalan permintaan darah
Prekondisi	Aktor telah melakukan login dan sistem menampilkan halaman <i>dashboard</i> admin penjamin mutu UTD
Prakondisi	Aktor telah menambah data ramalan permintaan dan sistem menampilkan halaman ramalan permintaan darah
Flow of Events	
Skenario Normal : Menambah Data Ramalan Permintaan Darah	
Aksi Aktor	Reaksi
1. Klik menu ramalan permintaan	
	2. Menampilkan halaman ramalan permintaan yang meliputi grafik permintaan setiap hari selama satu minggu. Halaman ini juga menampilkan tabel rekap permalan yang pernah dilakukan. Tabel ini menyajikan: <ul style="list-style-type: none"> • tanggal peramalan • nilai aktual • metode yang digunakan • nilai ramalan

	<ul style="list-style-type: none"> • MAPE atau tingkat kesalahan pada peramalan <p>Di bawah grafik permintaan ditampilkan tombol peramalan untuk melakukan peramalan jumlah permintaan darah hari selanjutnya.</p>
<p>3. Klik tombol peramalan</p>	
	<p>4. Perhitungan peramalan menggunakan metode <i>exponential smoothing</i> dan pemilihan metode mana yang tepat dengan membandingkan rata-rata persen kesalahan peramalan selama tujuh hari. Adapun rumus <i>exponential smoothing</i> antara lain:</p> <ul style="list-style-type: none"> • <i>Single Exponential Smoothing</i> $\hat{Y}_{t+1} = \alpha Y_t + (1-\alpha)\hat{Y}_t$ • <i>Double Exponential Smoothing</i> $A_t = \alpha Y_t + (1-\alpha)A_{t-1}$ $A'_t = \alpha A_t + (1-\alpha)A'_{t-1}$ $a_t = 2A_t - A'_t$ $b_t = \frac{\alpha}{1-\alpha}(A_t - A'_t)$ $\hat{Y}_{t+p} = a_t + b_t p$ • <i>Triple Exponential Smoothing</i> $S'_t = \alpha X_t + (1-\alpha)S'_{t-1}$ $S''_t = \alpha S'_t + (1-\alpha)S''_{t-1}$ $S'''_t = \alpha S''_t + (1-\alpha)S'''_{t-1}$ $a_t = 3S'_t - 3S''_t + S'''_t$ $b_t = a/2(1-a)(6-5.a)S'_t - (10-8.a)S''_t + (4-3.a)S'''_t$ $c_t = a^2/(1-a)^2(S'_t - 2S''_t + S'''_t)$ $F_{t+m} = a_t + b_t(1) + \frac{1}{2}c_t(1)$
	<p>5. Menyimpan ke <i>database</i></p>
	<p>6. Menampilkan hasil peramalan yang meliputi metode peramalan yang</p>

	digunakan serta berapa nilai permalan permintaan darah pada periode selanjutnya. Pada halaman ini juga ditampilkan tombol kembali yang digunakan untuk kembali ke halaman ramalan permintaan
7. Klik tombol kembali	
	<p>8. Menampilkan halaman ramalan permintaan yang meliputi grafik permintaan setiah hari selama satu minggu. Halaman ini juga menampilkan tabel rekap permalan yang pernah dilakukan. Tabel ini menyajikan:</p> <ul style="list-style-type: none"> • tanggal peramalan • nilai aktual • metode yang digunakan • nilai ramalan • MAPE atau tingkat kesalahan pada peramalan <p>Di bawah grafik permintaan ditampilkan tombol peramalan untuk melakukan peramalan jumlah permintaan darah hari selanjutnya.</p>

4.2.4 Activity Diagram

Activity Diagram merupakan penggambaran alur kerja system yang dijelaskan pada *scenario* sistem.

1 *Activity Diagram Login*

Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram login* dapat dilihat pada lampiran B.

2 *Activity Diagram Mengelola Data Permintaan Darah*

Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram mengelola data permintaan darah* dapat dilihat pada lampiran B.

3. *Activity Diagram Cetak Label Sample Darah*

Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram* cetak label *sample* darah dapat dilihat pada lampiran B.

4. *Activity Diagram* Mengelola Data Dokter

Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram* mengelola data dokter dapat dilihat pada lampiran B.

5. *Activity Diagram* Mengelola Data Perawat

Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram* mengelola data perawat dapat dilihat pada lampiran B.

6. *Activity Diagram* Mengelola Data Ruang Rumah Sakit

Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram* mengelola data ruang rumah sakit dapat dilihat pada lampiran B.

7. *Activity Diagram* Melihat Data Permintaan Darah

Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram* melihat data permintaan darah dapat dilihat pada lampiran B.

8. *Activity Diagram* Melihat Stok Darah

Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram* melihat stok darah dapat dilihat pada lampiran B.

9. *Activity Diagram* Validasi Data Permintaan Darah

Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram* validasi data permintaan darah dapat dilihat pada lampiran B.

10. *Activity Diagram* Mengelola Data Pendorong

Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram* mengelola data pendonor dapat dilihat pada lampiran B.

11. *Activity Diagram* Menambah Data Transaksi Donor

Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram* menambah data transaksi donor dapat dilihat pada lampiran B.

12. *Activity Diagram* Menambah Data Rumah Sakit

Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram* menambah data rumah sakit dapat dilihat pada lampiran B.

13. *Activity Diagram* Melihat Data Transaksi Donor dan Transaksi Permintaan Darah

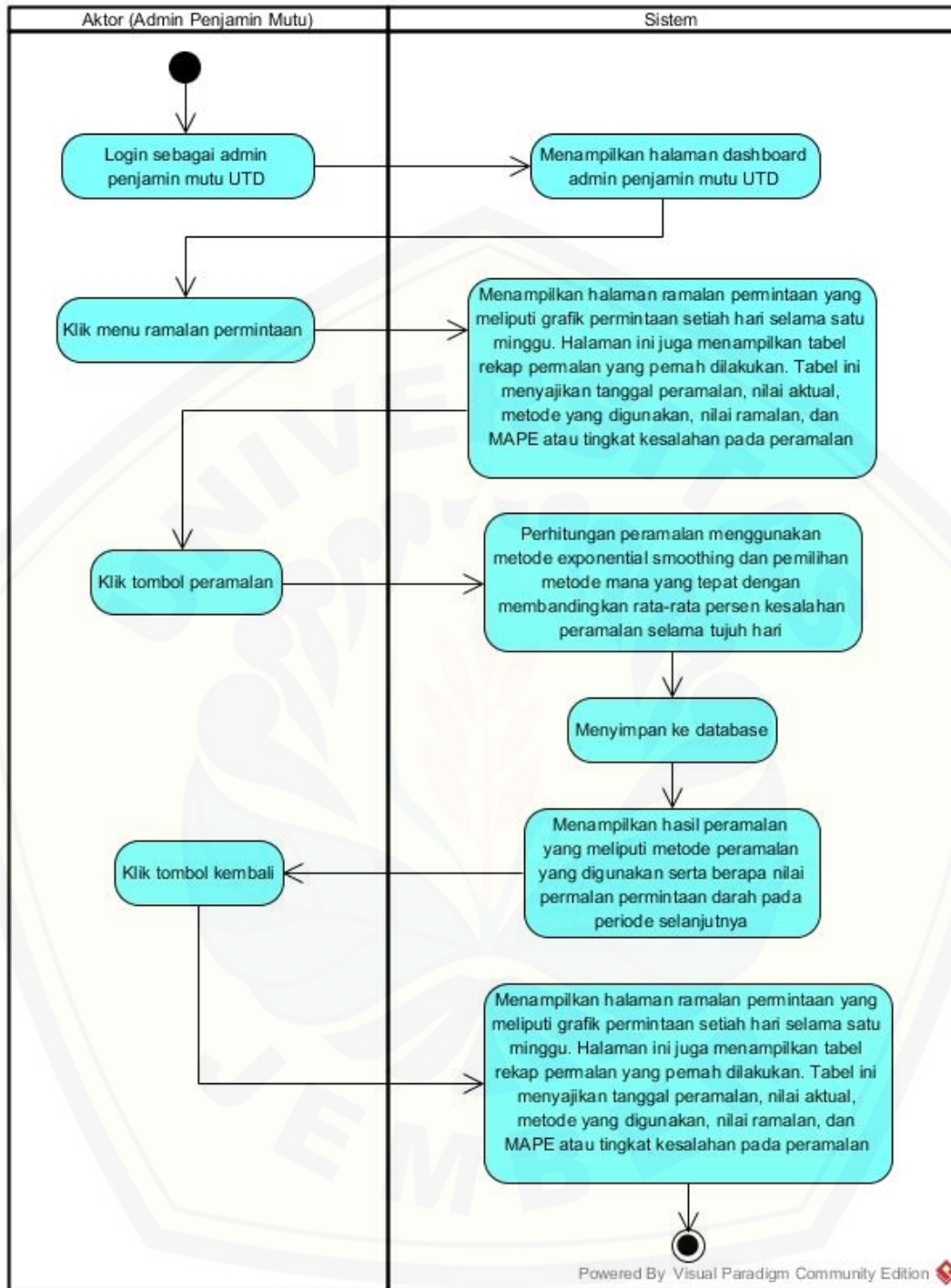
Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram* melihat data transaksi donor dan transaksi permintaan darah dapat dilihat pada lampiran B.

14. *Activity Diagram* Verifikasi Kualitas Darah Donor

Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram* verifikasi kualitas darah donor dapat dilihat pada lampiran B.

15. *Activity Diagram* Menambah Data Ramalan Permintaan

Gambaran alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam *activity diagram* menambah data ramalan permintaan dijelaskan pada gambar 4.4 dibawah ini.



Gambar 4.4 Activity Diagram Menambah Data Ramalan Permintaan

4.2.5 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar obyek di dalam sistem. *Sequence* diagram digunakan untuk memodelkan aliran logika dalam sistem dengan cara *visual*.

1. *Sequence Diagram Login*
Sequence diagram login dapat dilihat pada lampiran C.
2. *Sequence Diagram Mengelola Data Permintaan Darah*
Sequence diagram mengelola data permintaan darah dapat dilihat pada lampiran C.
3. *Sequence Diagram Cetak Label Sample Darah*
Sequence diagram cetak label *sample* darah dapat dilihat pada lampiran C.
4. *Sequence Diagram Mengelola Data Dokter*
Sequence diagram mengelola data dokter dapat dilihat pada lampiran C.
5. *Sequence Diagram Mengelola Data Perawat*
Sequence diagram mengelola data perawat dapat dilihat pada lampiran C.
6. *Sequence Diagram Mengelola Data Ruang Rumah Sakit*
Sequence diagram mengelola data ruang rumah sakit dapat dilihat pada lampiran C.
7. *Sequence Diagram Melihat Data Permintaan Darah*
Sequence diagram melihat data permintaan darah dapat dilihat pada lampiran C.
8. *Sequence Diagram Melihat Stok Darah*
Sequence diagram melihat stok darah dapat dilihat pada lampiran C.
9. *Sequence Diagram Validasi Data Permintaan Darah*
Sequence diagram validasi data permintaan darah dapat dilihat pada lampiranC
10. *Sequence Diagram Mengelola Data Pendorong*
Sequence diagram mengelola data pendonor dapat dilihat pada lampiran C
11. *Sequence Diagram Menambah Data Transaksi Donor*
Sequence diagram menambah data transaksi donor dapat dilihat pada lampiran C.

12. *Sequence Diagram* Menambah Data Rumah Sakit

Sequence diagram menambah data rumah sakit dapat dilihat pada lampiran C.

13. *Sequence Diagram* Melihat Data Transaksi Donor dan Transaksi Permintaan Darah

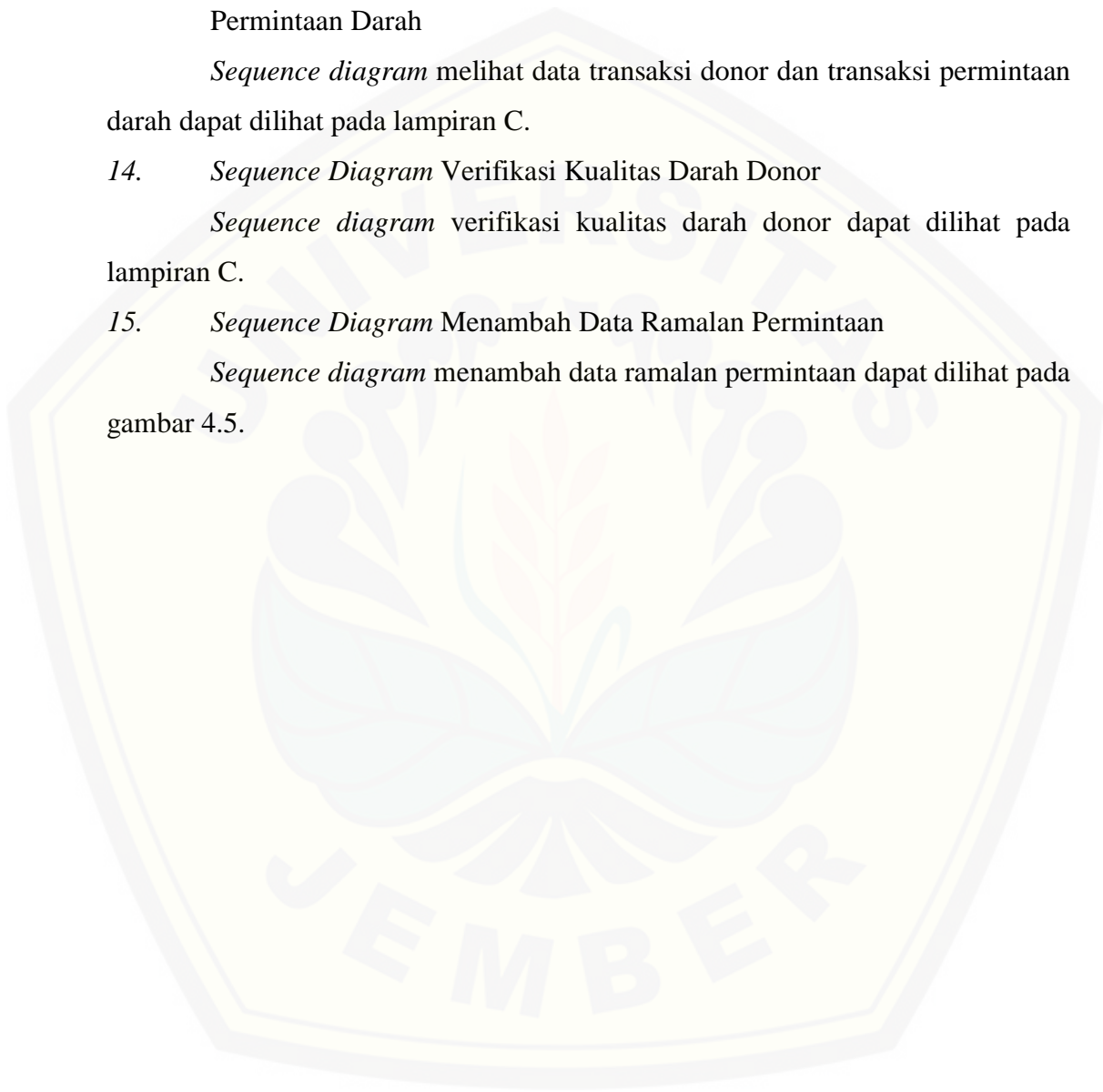
Sequence diagram melihat data transaksi donor dan transaksi permintaan darah dapat dilihat pada lampiran C.

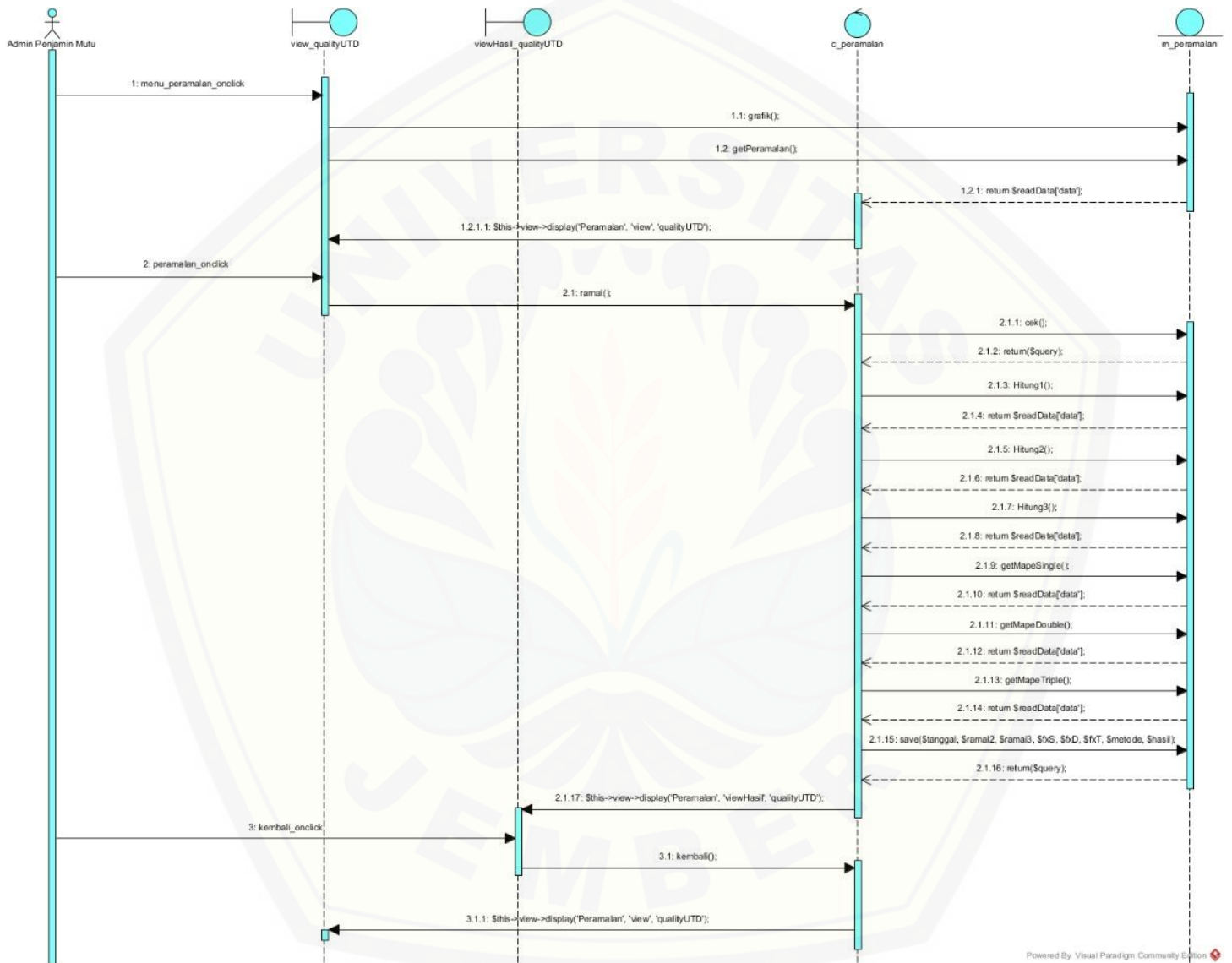
14. *Sequence Diagram* Verifikasi Kualitas Darah Donor

Sequence diagram verifikasi kualitas darah donor dapat dilihat pada lampiran C.

15. *Sequence Diagram* Menambah Data Ramalan Permintaan

Sequence diagram menambah data ramalan permintaan dapat dilihat pada gambar 4.5.

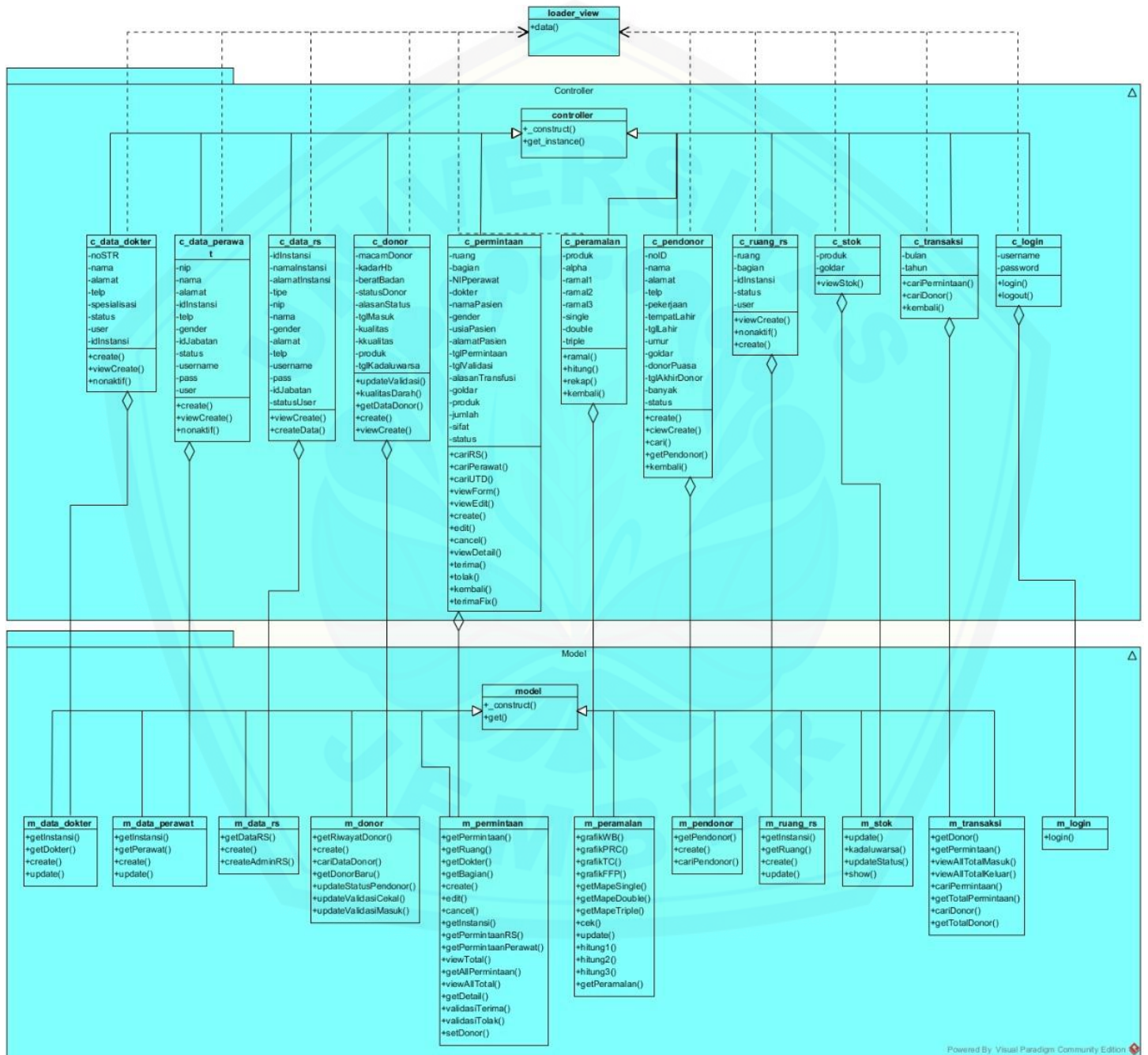




Gambar 4.5. Sequence Diagram Menambah Data Ramalan Permintaan

4.2.6 Class Diagram

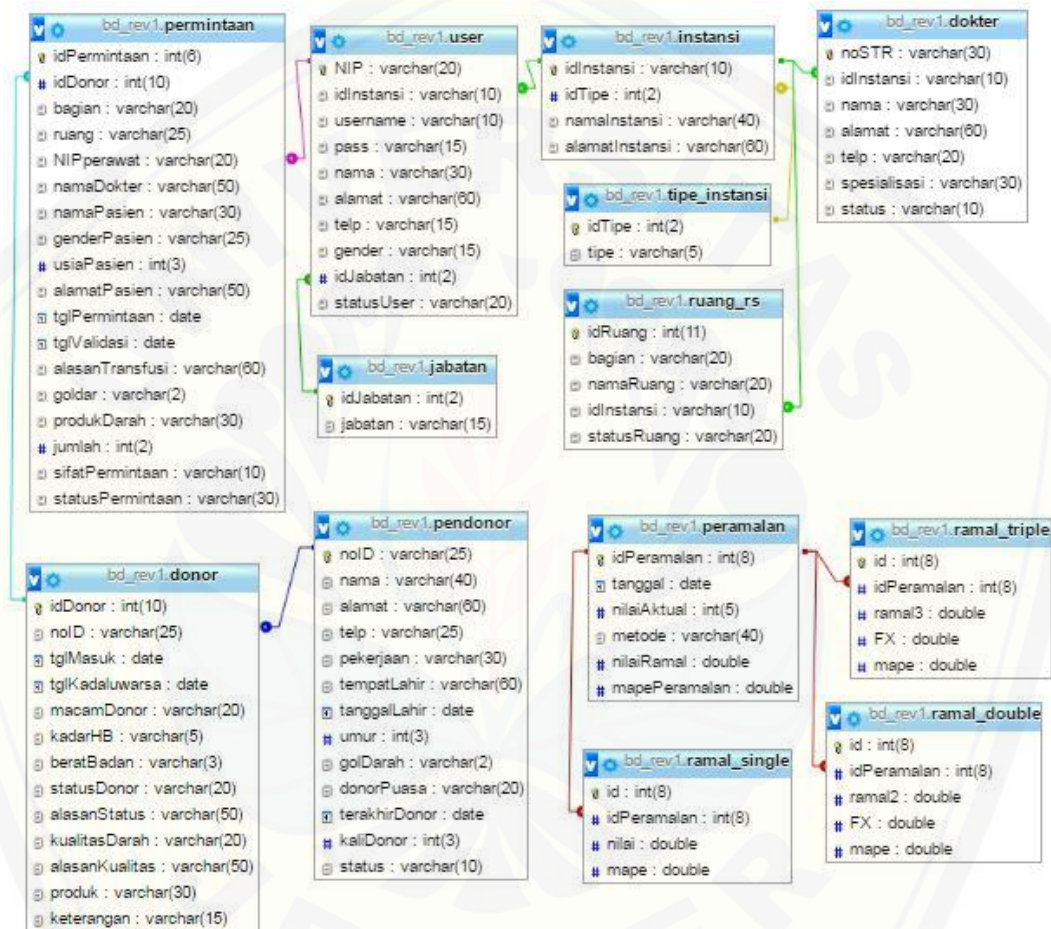
Class Diagram menggambarkan hubungan antarkelas yang ada dalam suatu sistem. Class Diagram sistem informasi peramalan jumlah permintaan darah menggunakan metode *exponential smoothing* dapat dilihat pada Gambar 4.6.



Gambar 4.6. Class Diagram

4.2.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) menjelaskan hubungan antar data dalam *database* yang digunakan dalam pembangunan sistem. ERD sistem informasi peramalan jumlah permintaan darah menggunakan metode *exponential smoothing* dapat dilihat pada Gambar 4.7.



Gambar 4.7. Entity Relationship Diagram (ERD)

4.3 Implementasi

Tahap implementasi merupakan tahapan penulisan kode program sesuai desain yang telah dibuat sebelumnya. Setiap fitur yang dimiliki oleh sistem informasi peramalan jumlah permintaan darah menggunakan metode *exponential smoothing* ditulis dalam tiga bagian kelas, meliputi kelas *view*, *controller*, dan *model*.

4.3.1 Kode program *Login*

Kode program *login* terletak pada kelas *login*, *c_login*, dan *m_login*. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.2 Kode program Mengelola Data Permintaan Darah

Kode program mengelola data permintaan darah terletak pada kelas *create_perawat*, *edit_perawat*, *view_perawat*, *viewDetail_perawat*, *c_permintaan*, dan *m_permintaan*. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.3 Kode program Cetak Label *Sample* Darah

Kode program cetak label *sample* darah terletak pada kelas *viewDetail_perawat*, *c_permintaan*, dan *m_permintaan*. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.4 Kode program Mengelola Data Dokter

Kode program mengelola data dokter terletak pada kelas *create_rs*, *view_rs*, *c_data_dokter*, dan *m_data_dokter*. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.5 Kode program Mengelola Data Perawat

Kode program mengelola data perawat terletak pada kelas *create_rs*, *view_rs*, *c_data_perawat*, dan *m_data_perawat*. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.6 Kode program Mengelola Data Ruang Rumah Sakit

Kode program mengelola data ruang rumah sakit terletak pada kelas *create_rs*, *view_rs*, *c_ruang_rs*, dan *m_ruang_rs*. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.7 Kode program Melihat Data Permintaan Darah

Kode program melihat data permintaan darah terletak pada kelas *view_rs*, *c_permintaan*, dan *m_permintaan*. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.8 Kode program Melihat Stok Darah

Kode program melihat stok darah terletak pada kelas *view*, *c_stok*, dan *m_stok*. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.9 Kode program Validasi Data Permintaan Darah

Kode program validasi data permintaan darah terletak pada kelas `viewDetail_pelayananUTD`, `c_permintaan`, dan `m_permintaan`. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.10 Kode program Mengelola Data Pendoror

Kode program mengelola data pendonor terletak pada kelas `create_pelayananUTD`, `view_pelayananUTD`, `viewDetail_pelayananUTD`, `c_pendoror`, dan `m_pendoror`. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.11 Kode program Menambah Data Transaksi Donor

Kode program menambah data transaksi donor terletak pada kelas `create_pelayananUTD`, `viewDetail_pelayananUTD`, `c_donor`, dan `m_donor`. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.12 Kode program Menambah Data Rumah Sakit

Kode program menambah data rumah sakit terletak pada kelas `create_pelayananUTD`, `view_pelayananUTD`, `c_data_rs`, dan `m_data_rs`. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.13 Kode program Melihat Data Transaksi Donor dan Transaksi Permintaan Darah

Kode program melihat data transaksi donor dan transaksi permintaan darah terletak pada kelas `view_qualityUTD`, `c_transaksi`, dan `m_transaksi`. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.14 Kode program Verifikasi Kualitas Darah Donor

Kode program verifikasi kualitas darah donor terletak pada kelas `validasi_qualityUTD`, `c_donor`, dan `m_donor`. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.15 Kode program Menambah Data Ramalan Permintaan

Kode program menambah data ramalan permintaan terletak pada kelas `view_qualityUTD`, `viewHasil_qualityUTD`, `c_peramalan`, dan `m_peramalan`. Penulisan kode program untuk fungsi peramalan pada *class controller* yang

merupakan urutan perhitungan peramalan terdapat pada *function* ramal() yang dapat dilihat pada tabel 4.4. Sebelum melakukan perhitungan peramalan, sistem melakukan pengecekan jumlah permintaan aktual pada hari ini dan melakukan *update* jumlah permintaan aktual serta nilai MAPE pada *database* dimana peramalan jumlah permintaan untuk hari selanjutnya dilakukan. Pengecekan ini dilakukan pada *function* cek() yang dapat dilihat pada tabel 4.5. Setelah melakukan pengecekan dan *update* sistem melakukan perhitungan nilai peramalan menggunakan setiap metode. Perhitungan ini dilakukan pada *function* Hitung1() yang mencari nilai peramalan dengan *single exponential smoothing* pada tabel 4.6, *function* Hitung2() yang mencari nilai peramalan dengan *double exponential smoothing* pada tabel 4.7, dan *function* Hitung3() yang mencari nilai peramalan dengan *triple exponential smoothing* pada tabel 4.8. Langkah selanjutnya adalah mengambil nilai MAPE tujuh hari sebelum tanggal yang diramalkan, yang dapat dilihat pada *function* getRamalSingle() tabel 4.9, *function* getRamalSingle() tabel 4.10, *function* getRamalSingle() tabel 4.11. Setelah nilai MAPE diketahui, nilai tersebut dibandingkan seperti yang dapat dilihat pada tabel 4.4 baris 13 sampai 28.

Tabel 4.4 Kode Program function ramal

1	function ramal() {
2	\$cek = \$this->Model->cek();
3	\$fx1 = \$this->Model->Hitung1();
4	\$rm2 = \$this->Model->Hitung2();
5	\$fx2 = \$rm2['ramalDouble'];
6	\$ramal2 = \$rm2['ramal2'];
7	\$rm3 = \$this->Model->Hitung3();
8	\$fx3 = \$rm3['ramalTriple'];
9	\$ramal3 = \$rm3['ramal3'];
10	\$single = \$this->Model->getMapeSingle();
11	\$double = \$this->Model->getMapeDouble();
12	\$triple = \$this->Model->getMapeTriple();
13	if (\$single < \$double && \$single < \$triple) {
14	\$this->view->metode = "Single Exponential Smoothing";
15	\$metode = "single";
16	\$this->view->ramalan = \$fx1;
17	\$this->view->display('Peramalan', 'viewHasil', 'qualityUTD');
18	} elseif (\$double < \$single && \$double < \$triple) {
19	\$this->view->metode = "Double Exponential Smoothing";
20	\$metode = "double";

```

21     $this->view->ramalan = $fx2;
22     $this->view->display('Peramalan', 'viewHasil', 'qualityUTD');
23 } elseif ($triple < $single && $triple < $double) {
24     $this->view->metode = "Triple Exponential Smoothing";
25     $metode = "triple";
26     $this->view->ramalan = $fx3;
27     $this->view->display('Peramalan', 'viewHasil', 'qualityUTD');
28 }
29 $idPeramalan = $this->Model->insertPeramalan($metode, $this->view-
    >ramalan);
30 if ($idPeramalan != 0) {
31     $this->Model->insert1($idPeramalan, $fx1);
32     $this->Model->insert2($idPeramalan, $ramal2, $fx2);
33     $this->Model->insert3($idPeramalan, $ramal3, $fx3);
34 }
35 }

```

Tabel 4.5 Kode Program function cek

```

1 function cek() {
2     $today = '2017-03-13';
3     $readData = $this->db->selectWhere("SELECT * FROM peramalan
        WHERE tanggal = :tanggal", array('tanggal' => $today));
4     foreach ($readData['data'] as $key => $value) {
5         $id = $value['idPeramalan'];
6         $jumlah = $value['nilaiAktual'];
7         $ramalan = $value['nilaiRamal'];
8         if (is_null($jumlah)) {
9             $tot = $this->db->selectWhere("SELECT count(*) as jum FROM
                `permintaan` WHERE statusPermintaan = 'terima' and
                tglValidasi = :tglValidasi", array('tglValidasi' => $today));
10            $saktual = $tot['data'][0]['jum'];
11            $spe = (abs($saktual-$ramalan)/$saktual)*100;
12            $this->db->update("UPDATE `peramalan` SET `nilaiAktual`=
                :nilaiAktual, `PEperamalan` = :PEperamalan WHERE
                idPeramalan = :idPeramalan", array('nilaiAktual' => $saktual,
                'PEperamalan' => $spe, 'idPeramalan' => $id));
13            $this->updateRamal1($id, $saktual);
14            $this->updateRamal2($id, $saktual);
15            $this->updateRamal3($id, $saktual);
16        }
17    }
18 }

```

Tabel 4.6 Kode Program function Hitung1

```

1 function Hitung1() {
2     $readData = $this->db->selectWhere("SELECT * FROM ramal_single r
      join peramalan p on r.idPeramalan = p.idPeramalan WHERE
      tanggal = date_add('2017-03-23',interval -1 day)");
3     foreach ($readData['data'] as $key => $value) {
4         $aktual = $value['nilaiAktual'];
5         $ramal_ = $value['nilai'];
6         $ramalSingle = (0.8*$aktual)+((1-0.8)*$ramal_);
7         return $ramalSingle;
8     }
9 }

```

Tabel 4.7 Kode Program function Hitung2

```

1 function Hitung2() {
2     $a = $this->db->selectWhere("SELECT * FROM ramal_single r join
      peramalan p on r.idPeramalan = p.idPeramalan WHERE tanggal =
      date_add('2017-03-23',interval -1 day)");
3     foreach ($a['data'] as $key => $value) {
4         $aktual = $value['nilaiAktual'];
5         $ramalS_ = $value['nilai'];
6         $ramalSingle = (0.8*$aktual)+((1-0.8)*$ramalS_);
7         return $ramalSingle;
8     }
9     $b = $this->db->selectWhere("SELECT * FROM ramal_double r
      join peramalan p on r.idPeramalan = p.idPeramalan WHERE
      tanggal = date_add('2017-03-23',interval -1 day)");
10    foreach ($b['data'] as $key => $value) {
11        $ramalD_ = $value['ramal2'];
12        $ramal2 = (0.8 * $ramalSingle)+((1-0.8)*$ramalD_);
13        $sat = (2*$ramalSingle)-$ramal2;
14        $bt = (0.8/(1-0.8))*($ramalSingle-$ramal2);
15        $ramalDouble = abs($sat + $bt);
16        return $ramalDouble;
17    }
18 }

```

Tabel 4.8 Kode Program function Hitung3

```

1 function Hitung3() {
2     $a = $this->db->selectWhere("SELECT * FROM ramal_single r join
      peramalan p on r.idPeramalan = p.idPeramalan WHERE tanggal =
      date_add('2017-03-23',interval -1 day)");
3     foreach ($a['data'] as $key => $value) {
4         $aktual = $value['nilaiAktual'];
5         $ramalS_ = $value['nilai'];

```



```

6      $ramalSingle = (0.8*$aktual)+((1-0.8)*$ramalS_);
7      return $ramalSingle;
8      $b = $this->db->selectWhere("SELECT * FROM ramal_double r
      join peramalan p on r.idPeramalan = p.idPeramalan WHERE
      tanggal = date_add('2017-03-23',interval -1 day)");
9      foreach ($b['data'] as $key => $value) {
10         $ramalD_ = $value['ramal2'];
11         $ramal2 = (0.8 * $ramalSingle)+((1-0.8)*$ramalD_);
12         $sat = (2*$ramalSingle)-$ramal2;
13         $bt = (0.8/(1-0.8))*($ramalSingle-$ramal2);
14         $ramalDouble = abs($sat + $bt);
15         $c = $this->db->selectWhere("SELECT * FROM ramal_triple r
      join peramalan p on r.idPeramalan = p.idPeramalan WHERE
      tanggal = date_add('2017-03-23',interval -1 day)");
16         foreach ($c['data'] as $key => $value) {
17             $ramalT_ = $value['ramal3'];
18             $ramal3 = (0.8 * $ramalDouble)+((1-0.8)*$ramalT_);
19             $sat = (3*$ramalSingle)-(3*$ramalDouble)+$ramal3;
20             $bt = 0.8/(2*(1-0.8))*((6-(5*0.8)*$ramalSingle)-(10-
      (8*0.8)*$ramalDouble)+(4-(3*0.8)*$ramal3));
21             $pangkat1 = pow(0.8, 2);
22             $pangkat2 = pow((1-0.8), 2);
23             $ct = ($pangkat1/$pangkat2)*($ramalSingle-
      (2*$ramalDouble)+$ramal3);
24             $ramalTriple = $sat+$bt+(0.5*$ct);
25             return $ramalTriple;
26         }
27     }
28 }
29 }

```

Tabel 4.9 Kode Program function getMapeSingle

```

1 function getMapeSingle () {
2     $readData = $this->db->selectWhere("SELECT SUM(mape) AS jumlah
      FROM ramal_single r join peramalan p on r.idPeramalan =
      p.idPeramalan WHERE tanggal <= '2017-03-22' && tanggal >=
      date_add('2017-03-22',interval -7 day)");
3     foreach ($readData['data'] as $key => $value) {
4         $mape = $value['jumlah']/7;
5         return $mape;
6     }
7 }

```

Tabel 4.10 Kode Program function getMapeDouble

1	function getMapeDouble () {
2	\$readData = \$this->db->selectWhere("SELECT SUM(mape) AS jumlah FROM ramal_double r join peramalan p on r.idPeramalan = p.idPeramalan WHERE tanggal <= '2017-03-22' && tanggal >= date_add('2017-03-22',interval -7 day)");
3	foreach (\$readData['data'] as \$key => \$value) {
4	\$mape = \$value['jumlah']/7;
5	return \$mape;
6	}
7	}

Tabel 4.11 Kode Program function getMapeTriple

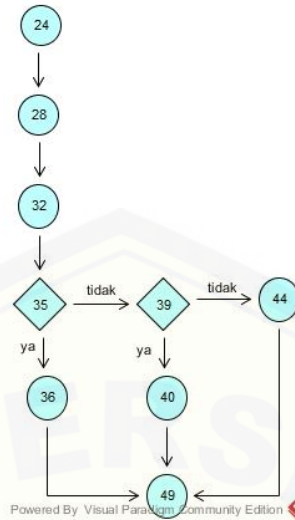
1	function getMapeTriple () {
2	\$readData = \$this->db->selectWhere("SELECT SUM(mape) AS jumlah FROM ramal_triple r join peramalan p on r.idPeramalan = p.idPeramalan WHERE tanggal <= '2017-03-22' && tanggal >= date_add('2017-03-22',interval -7 day)");
3	foreach (\$readData['data'] as \$key => \$value) {
4	\$mape = \$value['jumlah']/7;
5	return \$mape;
6	}
7	}

4.4 Pengujian

Tahap pengujian dilakukan untuk mengevaluasi sistem informasi yang telah selesai diimplementasikan. Seperti yang telah dijelaskan pada bab tiga, tahapan ini dilakukan dengan dua metode, yaitu metode pengujian *whitebox* dan metode pengujian *blackbox*.

4.4.1 Pengujian *White Box*

Pengujian *white box* pada sistem informasi peramalan jumlah permintaan darah menggunakan metode *exponential smoothing* ini dilakukan dengan cara menggambar diagram alir, menghitung kompleksitas siklomatiknya (CC), dan membuat tabel pengujian *test case*. Pengujian kompleksitas siklomatik pada fitur menambah data peramalan permintaan dapat dilihat pada Gambar 4.8 tabel pengujian *test case* dapat dilihat pada Tabel 4.15. Sedangkan untuk fitur yang lain dapat dilihat pada lampiran E.



Gambar 4.8 Diagram Alir *function* ramal

$$CC = E - N + 2 = 10 - 9 + 2 = 3$$

Tabel 4.13 *Test Case function* ramal

<i>Test case 1</i>	Jika metode terpilih adalah <i>single exponential smoothing</i>
Target yang diharapkan	Menampilkan hasil perhitungan peramalan jumlah permintaan menggunakan metode <i>single exponential smoothing</i>
Hasil pengujian	Benar
<i>Path/Jalur</i>	24 – 28 – 32 – 35 – 36 – 49
<i>Test case 2</i>	Jika metode terpilih adalah <i>double exponential smoothing</i>
Target yang diharapkan	Menampilkan hasil perhitungan peramalan jumlah permintaan menggunakan metode <i>double exponential smoothing</i>
Hasil pengujian	Benar
<i>Path/Jalur</i>	24 – 28 – 32 – 35 – 39 – 40 – 49
<i>Test case 3</i>	Jika metode terpilih adalah <i>triple exponential smoothing</i>

Target yang diharapkan	Menampilkan hasil perhitungan peramalan jumlah permintaan menggunakan metode <i>triple exponential smoothing</i>
Hasil pengujian	Benar
Path/Jalur	24 – 28 – 32 – 35 – 39 – 44 – 49

4.4.2 Pengujian *Black Box*

Pengujian *black box* untuk fitur menambah data peramalan permintaan dapat dilihat pada Tabel 4.16. Sedangkan untuk pengujian fitur yang lain dapat dilihat pada lampiran F.

Tabel 4.14 Pengujian Black Box Menambah Data Peramalan Permintaan

No.	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Tidak
1.	Menambah data peramalan permintaan	Klik tombol peramalan	Menampilkan hasil peramalan permintaan hari selanjutnya dan metode apa yang tepat untuk menghasilkan peramalan tersebut	V	

4.4.3 Pengujian Tingkat Kebenaran Peramalan

Pengujian ini akan dijelaskan pada sub bab 5.3. Tujuan dari pengujian ini adalah untuk mengetahui apakah metode *exponential smoothing* mampu memberikan nilai peramalan yang mendekati nilai aktual permintaan darah pada UTD PMI Kabupaten Jember. Pengujian ini dihitung berdasarkan rumus *Percentage Error* (PE) dan MAPE yang rumusnya dapat dilihat pada persamaan 2.14 dan 2.15.

BAB 6 PENUTUP

Bab ini berisi kesimpulan dan saran dari penelitian yang telah dilakukan. Kesimpulan dan saran ini diharapkan mampu menjadi acuan untuk melakukan penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Peramalan jumlah permintaan darah pada UTD PMI Kabupaten Jember menggunakan metode *exponential smoothing* khususnya metode *single exponential smoothing*, *Brown's double exponential smoothing*, dan *Brown's triple exponential smoothing* dilakukan dengan menggunakan nilai $\alpha = 0.8$ dan data permintaan aktual serta nilai peramalan pada satu hari sebelum tanggal peramalan.
2. Perancangan dan pembangunan sistem informasi peramalan jumlah permintaan darah dengan menggunakan metode *exponential smoothing* pada UTD PMI Kabupaten Jember ini menggunakan metode SDLC model *waterfall*. Adapun hambatan dalam pembangunan sistem menggunakan model *waterfall* ini adalah proses analisis data yang membutuhkan waktu yang lama. Hal tersebut dikarenakan tahap analisis data merupakan langkah awal yang menentukan tahapan selanjutnya dalam proses pembangunan. Analisis data yang kurang matang akan menimbulkan kesalahan pada tahap-tahap selanjutnya dan dalam model *waterfall* ini, apabila terdapat kesalahan kita tidak bisa langsung memperbaiki kesalahan pada tahap kesalahan itu terjadi, melainkan harus mengulang dari tahap awal. Namun karena sistem informasi peramalan jumlah permintaan darah ini tergolong sistem berskala kecil, maka hambatan tersebut tidak begitu menjadi masalah besar.
3. Pemilihan metode *exponential smoothing* yang cocok dengan pola data jumlah permintaan darah di UTD PMI Kabupaten Jember dilakukan dengan cara membandingkan nilai MAPE peramalan jumlah permintaan

darah interval satu minggu sebelum tanggal peramalan. Cara ini mampu memilih metode *exponential smoothing* yang sesuai dengan pola data permintaan darah yang ada dengan tingkat kebenaran peramalan mencapai 83.53%.

6.2 Saran

Saran yang ditujukan untuk memberikan masukan yang lebih baik yaitu :

1. Sistem informasi peramalan jumlah permintaan darah diharapkan pada pengembang selanjutnya memberikan fitur yang lebih lengkap hingga mencakup pengelolaan setiap produk dan jenis darah yang ada di UTD PMI Kabupaten Jember.
2. Pengembang selanjutnya diharapkan mampu membangun sistem peramalan permintaan darah yang dapat memberikan hasil peramalan untuk setiap produk atau jenis darah yang ada di UTD PMI Kabupaten Jember.
3. Proses peramalan jumlah permintaan darah dilakukan dengan metode peramalan yang berbeda atau mengkombinasikan metode *exponential smoothing* dengan metode lainnya.

DAFTAR PUSTAKA

- Basuki, A. (2006). *Pengembangan Perangkat Lunak*. Surabaya: PENS-ITS.
- Fauziah, R. (2016). *Implementasi Metode Weighted Moving Average dalam Prediksi Jumlah Stok Darah pada Palang Merah Indonesia Jember*. Jember: Program Studi Sistem Informasi Universitas Jember.
- Kementrian Kesehatan, R. I. (2014). *Situasi Donor Darah di Indonesia*. Jakarta Selatan.
- Makridakis, S dkk. (1983). *Forecasting, Method and Applications*. Canada: John Wiley and Sons Inc.
- Menteri Kesehatan. (2014). *Peraturan Menteri Kesehatan 83 Tahun 2014 . Peraturan Menteri Kesehatan 83 Tahun 2014 tentang Unit Transfusi Darah, Bank Darah Rumah Sakit, dan Jejaring Pelayanan Transfusi Darah*. MJ, N. (2014, Maret 22). *EXPONENTIAL SMOOTHING*. Retrieved from <http://mjnurul.blogspot.co.id/2014/03/exponential-smoothing.html>
- Nazim, Ahmad ; Afthanorhan, Asyraf ;. (2014). A comparison between single exponential smoothing (SES), double exponential smoothing (DES), holt's (brown) and adaptive response rate exponential smoothing (ARRES) techniques in forecasting Malaysia population. *Global Journal of Mathematical Analysis*.
- Ni Putu Yuli Sukmarani, S. R. (2016). Penerapan Metode Exponential Smoothing pada Peramalan Penjualan dalam Penentuan Kuantitas Produksi Roti (Studi Kasus Perusahaan Roti Dhiba Kendari. *semanTIK, Vol.2*.
- UTD PMI Kabupaten Jember. (2017). *Prosedur Permintaan Darah*. Retrieved from http://donordarahsehat.com/index.php?option=com_content&view=article&id=63&Itemid=73

LAMPIRAN

Lampiran A. Scenario

A1 Scenario Login

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif *login* dijelaskan pada tabel A.1 sampai tabel A.4.

Tabel A.0.1 Scenario Login Perawat

Nama Use Case	<i>Login</i>
Aktor	Perawat
Deskripsi Singkat	Perawat akan mengakses sistem
Prekondisi	Aktor telah membuka halaman <i>wesite</i>
Prakondisi	Aktor berhasil <i>login</i>
Flow of Events	
Skenario Normal : Login	
Aksi Aktor	Reaksi
1. Menginputkan <i>username</i> dan <i>password</i>	
2. Klik tombol <i>login</i>	
	3. Mengecek ke <i>database</i>
	4. Menampilkan halaman <i>dashboard</i> perawat
Skenario Alternatif : Username dan Password Salah	
2a. Klik tombol <i>login</i>	
	3a. Menampilkan pesan " <i>login gagal</i> "
Skenario Alternatif : Kolom Username dan Password Kosong	
2b. Klik tombol <i>login</i>	

	3b. Menampilkan pesan dibawah kolom yang kosong “ <i>please fill out this field</i> ”
--	---

Tabel A.0.2 *Scenario Login Admin Rumah Sakit*

Nama Use Case	<i>Login</i>
Aktor	Admin Rumah Sakit
Deskripsi Singkat	Admin Rumah Sakit akan mengakses sistem
Prekondisi	Aktor telah membuka halaman <i>website</i>
Prakondisi	Aktor berhasil <i>login</i>
Flow of Events	
Skenario Normal : <i>Login</i>	
Aksi Aktor	Reaksi
1. Membuka halaman website	
	2. Menampilkan halaman beranda website
3. Memilih menu Login	
	4. Menampilkan form <i>login</i>
5. Menginputkan username dan password	
6. Klik tombol login	
	7. Mengecek ke database
	8. Menampilkan halaman dashboard Admin Rumah Sakit
Skenario Alternatif : <i>Username dan Password Salah</i>	

6a. Klik tombol login	
	7a. Menampilkan pesan “login gagal”
Skenario Alternatif : Kolom <i>Username</i> dan <i>Password</i> Kosong	
6b. Klik tombol login	
	7b. Menampilkan pesan dibawah kolom yang kosong “Please fill out this field”

Tabel A.0.3 *Scenario Login* Admin Pelayanan UTD PMI

Nama Use Case	<i>Login</i>
Aktor	Admin Pelayanan UTD PMI
Deskripsi Singkat	Admin Pelayanan UTD PMI akan mengakses sistem
Prekondisi	Aktor telah membuka halaman <i>website</i>
Prakondisi	Aktor berhasil <i>login</i>
Flow of Events	
Skenario Normal : <i>Login</i>	
Aksi Aktor	Reaksi
1. Membuka halaman website	
	2. Menampilkan halaman beranda website
3. Memilih menu Login	
	4. Menampilkan form <i>login</i>
5. Menginputkan username dan password	
6. Klik tombol login	

	7. Mengecek ke database
	8. Menampilkan halaman dashboard Admin Pelayanan UTD PMI
Skenario Alternatif : Username dan Password Salah	
6a. Klik tombol login	
	7a. Menampilkan pesan “login gagal”
Skenario Alternatif : Kolom Username dan Password Kosong	
6b. Klik tombol login	
	7b. Menampilkan pesan dibawah kolom yang kosong “Please fill out this field”

Tabel A.0.4 *Scenario Login* Admin Penjamin Mutu UTD PMI

Nama Use Case	<i>Login</i>
Aktor	Admin Penjamin Mutu UTD PMI
Deskripsi Singkat	Admin Penjamin Mutu UTD PMI akan mengakses sistem
Prekondisi	Aktor telah membuka halaman <i>website</i>
Prakondisi	Aktor berhasil <i>login</i>
Flow of Events	
Skenario Normal : Login	
Aksi Aktor	Reaksi
1. Membuka halaman website	
	2. Menampilkan halaman beranda website

3. Memilih menu Login	
	4. Menampilkan form <i>login</i>
5. Menginputkan username dan password	
6. Klik tombol login	
	7. Mengecek ke database
	8. Menampilkan halaman dashboard Admin Penjamin Mutu UTD PMI
Skenario Alternatif : <i>Username dan Password Salah</i>	
6a. Klik tombol login	
	7a. Menampilkan pesan “login gagal”
Skenario Alternatif : Kolom <i>Username dan Password Kosong</i>	
6b. Klik tombol login	
	7b. Menampilkan pesan dibawah kolom yang kosong “Please fill out this field”

A2 *Scenario* Mengelola Data Permintaan Darah

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif mengelola data permintaan darah dijelaskan pada tabel A.5.

Tabel A.0.5 *Scenario* Mengelola Data Permintaan Darah

Nama Use Case	Mengelola Data Permintaan Darah
Aktor	Perawat

Deskripsi Singkat	Perawat akan mengelola data permintaan darah
Prekondisi	Data permintaan darah
Prakondisi	<ol style="list-style-type: none"> 1. Data permintaan darah berhasil ditambahkan 2. Data permintaan darah berhasil dibatalkan 3. Detail data permintaan darah ditampilkan 4. Data permintaan darah berhasil diubah
<i>Flow of Events</i>	
Skenario Normal : Menambah Data Permintaan Darah	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai perawat	
	2. Menampilkan halaman <i>dashboard</i> perawat
3. Klik menu permintaan darah	
	<p>4. Menampilkan table data permintaan darah yang pernah dibuat oleh perawat. Tabel ini menampilkan :</p> <ul style="list-style-type: none"> • nama pasien • nama bagian perawatan • nama ruang tempat pasien dirawat • status permintaan (terima, tolak atau kosong apabila belum di validasi oleh UTD) • tombol dropdown action yang memuat 3 tombol aksi yaitu detail untuk melihat detail permintaan, edit untuk mengubah data permintaan, dan cancel untuk membatalkan permintaan <p>Halaman ini juga menampilkan tombol isi formulir yang digunakan apabila perawat akan menambahkan data permintaan darah, serta kolom pencarian</p>

	berdasarkan nama pasien di atas bagian atas tabel.
5. Klik tombol isi formulir	
	6. Menampilkan halaman form data permintaan darah
7. Mengisi form data permintaan darah yang meliputi nama pasien, jenis kelamin pasien, ruang tempat pasien dirawat, nama dokter yang menangani pasien, usia pasien, alamat pasien, alasan transfusi, sifat permintaan darah, golongan darah, produk darah, dan jumlah darah yang dibutuhkan	
8. Klik tombol <i>submit</i>	
	9. Menyimpan ke <i>database</i>
	<p>10. Menampilkan table data permintaan darah yang pernah dibuat oleh perawat. Tabel ini menampilkan :</p> <ul style="list-style-type: none"> • nama pasien • nama bagian perawatan • nama ruang tempat pasien dirawat • status permintaan (terima, tolak atau kosong apabila belum di validasi oleh UTD) • tombol dropdown action yang memuat 3 tombol aksi yaitu detail untuk melihat detail permintaan, edit untuk mengubah data permintaan, dan cancel untuk membatalkan permintaan <p>Halaman ini juga menampilkan tombol isi formulir yang digunakan apabila perawat akan menambahkan data permintaan darah, serta kolom pencarian</p>

	berdasarkan nama pasien di atas bagian atas tabel.
Skenario Alternatif : Kolom Pada Form Ada yang Kosong	
8a. Klik tombol <i>submit</i>	
	9a. Menampilkan pesan dibawah kolom yang kosong “please fill this field”
Skenario Normal : Melihat Data Permintaan Darah	
5b. Isi kolom pencarian dengan nama pasien yang ingin dicari	
6b. Klik cari	
	7b. Menampilkan table data permintaan darah pasien yang dicari. Tabel ini menampilkan : <ul style="list-style-type: none"> • nama pasien • nama bagian perawatan • nama ruang tempat pasien dirawat • status permintaan (terima, tolak atau kosong apabila belum di validasi oleh UTD) • tombol dropdown action yang memuat 3 tombol aksi yaitu detail untuk melihat detail permintaan, edit untuk mengubah data permintaan, dan cancel untuk membatalkan permintaan
8b. Klik detail pada dropdown action	
	9b. Menampilkan detail data permintaan darah berbentuk form yang terdiri dari : <ul style="list-style-type: none"> • id permintaan • nama bagian perawatan • nama ruang tempat pasien dirawat • nama dokter yang menangani pasien • nama pasien • jenis kelamin pasien

	<ul style="list-style-type: none"> • usia pasien • alamat pasien • tanggal permintaan • tanggal validasi UTD • alasan transfuse • golongan darah • produk darah • jumlah permintaan • sifat permintaan • status permintaan <p>Halaman ini juga menampilkan tombol cetak label yang digunakan untuk mencetak label sample darah yang akan dibawa ke UTD saat pengambilan darah, dan tombol kembali untuk kembali ke halaman permintaan darah.</p>
Skenario Normal : Membatalkan Permintaan Darah	
8c. Klik cancel pada dropdown <i>action</i>	
	9c. Mengubah status permintaan pada <i>database</i> menjadi “dibatalkan”
	10c. Menampilkan pesan “Permintaan Darah Berhasil Dibatalkan”
	<p>11c. Menampilkan table data permintaan darah yang pernah dibuat oleh perawat. Tabel ini menampilkan :</p> <ul style="list-style-type: none"> • nama pasien • nama bagian perawatan • nama ruang tempat pasien dirawat • status permintaan (terima, tolak atau kosong apabila belum di validasi oleh UTD) • tombol dropdown action yang memuat 3 tombol aksi yaitu detail untuk melihat detail permintaan, edit untuk mengubah data permintaan,

	<p>dan cancel untuk membatalkan permintaan</p> <p>Halaman ini juga menampilkan tombol isi formulir yang digunakan apabila perawat akan menambahkan data permintaan darah, serta kolom pencarian berdasarkan nama pasien di atas bagian atas tabel.</p>
Skenario Normal : Merubah Data Permintaan Darah	
8d. Klik edit pada dropdown <i>action</i>	
	9d. Menampilkan form isian permintaan darah yang telah terisi dengan data sebelumnya
10d. Mengubah data permintaan darah yang meliputi nama pasien, jenis kelamin pasien, ruang tempat pasien dirawat, nama dokter yang menangani pasien, usia pasien, alamat pasien, alasan transfusi, sifat permintaan darah, golongan darah, produk darah, dan jumlah darah yang dibutuhkan	
11d. Klik tombol <i>submit</i>	
	12d. Menyimpan ke <i>database</i>
	<p>13d. Menampilkan table data permintaan darah yang pernah dibuat oleh perawat. Tabel ini menampilkan :</p> <ul style="list-style-type: none"> • nama pasien • nama bagian perawatan • nama ruang tempat pasien dirawat • status permintaan (terima, tolak atau kosong apabila belum di validasi oleh UTD) • tombol dropdown action yang memuat 3 tombol aksi yaitu detail untuk melihat detail

	<p>permintaan, edit untuk mengubah data permintaan, dan cancel untuk membatalkan permintaan</p> <p>Halaman ini juga menampilkan tombol isi formulir yang digunakan apabila perawat akan menambahkan data permintaan darah, serta kolom pencarian berdasarkan nama pasien di atas bagian atas tabel.</p>
--	---

A3 Scenario Cetak Label *Sample* Darah

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif cetak label *sample* darah dijelaskan pada lampiran A.6.

Tabel A.0.6 Scenario Cetak Label *Sample* Darah

Nama Use Case	Cetak label sample darah
Aktor	Perawat
Deskripsi Singkat	Perawat akan mencetak label sample darah
Prekondisi	Label sample darah belum dicetak
Prakondisi	Label berhasil dicetak
Flow of Events	
Skenario Normal : Mencetak Label Sample Darah	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai perawat	
	2. Menampilkan halaman <i>dashboard</i> perawat

3. Klik menu permintaan darah	
	<p>4. Menampilkan table data permintaan darah yang pernah dibuat oleh perawat. Tabel ini menampilkan :</p> <ul style="list-style-type: none"> • nama pasien • nama bagian perawatan • nama ruang tempat pasien dirawat • status permintaan (terima, tolak atau kosong apabila belum di validasi oleh UTD) • tombol dropdown action yang memuat 3 tombol aksi yaitu detail untuk melihat detail permintaan, edit untuk mengubah data permintaan, dan cancel untuk membatalkan permintaan <p>Halaman ini juga menampilkan tombol isi formulir yang digunakan apabila perawat akan menambahkan data permintaan darah, serta kolom pencarian berdasarkan nama pasien di atas bagian atas tabel.</p>
5. Isi kolom pencarian dengan nama pasien yang ingin dicari	
6. Klik cari	
	<p>7. Menampilkan table data permintaan darah pasien yang dicari. Tabel ini menampilkan :</p> <ul style="list-style-type: none"> • nama pasien • nama bagian perawatan • nama ruang tempat pasien dirawat • status permintaan (terima, tolak atau kosong apabila belum di validasi oleh UTD)

	<ul style="list-style-type: none"> tombol dropdown action yang memuat 3 tombol aksi yaitu detail untuk melihat detail permintaan, edit untuk mengubah data permintaan, dan cancel untuk membatalkan permintaan
8. Klik detail pada dropdown action	
	<p>9. Menampilkan detail data permintaan darah berbentuk form yang terdiri dari :</p> <ul style="list-style-type: none"> id permintaan nama bagian perawatan nama ruang tempat pasien dirawat nama dokter yang menangani pasien nama pasien jenis kelamin pasien usia pasien alamat pasien tanggal permintaan tanggal validasi UTD alasan transfuse golongan darah produk darah jumlah permintaan sifat permintaan status permintaan <p>Halaman ini juga menampilkan tombol cetak label yang digunakan untuk mencetak label sample darah yang akan dibawa ke UTD saat pengambilan darah, dan tombol kembali untuk kembali ke halaman permintaan darah.</p>
10. Klik cetak label	
	11. Mengunduh label sample darah dalam bentuk pdf yang kemudian dicetak

A4 *Scenario* Mengelola Data Dokter

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif mengelola data dokter dijelaskan pada lampiran A.7.

Tabel A.0.7 *Scenario* Mengelola Data Dokter

Nama Use Case	Mengelola Data Dokter
Aktor	Admin Rumah Sakit
Deskripsi Singkat	Admin Rumah Sakit akan mengelola data dokter
Prekondisi	Data dokter
Prakondisi	1. Data dokter ditampilkan 2. Status dokter berhasil diubah 3. Data dokter berhasil ditambahkan
Flow of Events	
Skenario Normal : Melihat Data Dokter	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai admin rumah sakit	
	2. Menampilkan halaman <i>dashboard</i> admin rumah sakit
3. Klik menu dokter	
	4. Menampilkan halaman data dokter meliputi tabel data dokter yang bekerja pada rumah sakit. Tabel ini menampilkan : <ul style="list-style-type: none"> • nomor Surat Tanda Registrasi dokter • nama dokter • alamat dokter • nomor telepon dokter • spesialisasi dokter • tombol “non-aktif” untuk menonaktifkan dokter

	Halaman ini juga menampilkan tombol “tambah dokter” yang digunakan saat admin rumah sakit akan menambah data dokter baru
Skenario Normal : Merubah Status Dokter	
5a. Klik tombol non-aktif	
	6a. Mengubah status dokter pada <i>database</i> menjadi non-aktif
	7a. Menampilkan pesan “Status Dokter Non-aktif”
Skenario Normal : Menambah Data Dokter	
5a. Klik tombol tambah dokter	
	6a. Menampilkan halaman form data dokter
7a. Mengisi form data dokter yang meliputi nomor Surat Tanda Registrasi, nama dokter, alamat dokter, nomor telepon dokter, dan spesialisasi dokter	
8a. Klik tombol <i>submit</i>	
	9a. Menyimpan ke <i>database</i>
	<p>10a. Menampilkan halaman data dokter meliputi tabel data dokter yang bekerja pada rumah sakit. Tabel ini menampilkan :</p> <ul style="list-style-type: none"> • nomor Surat Tanda Registrasi dokter • nama dokter • alamat dokter • nomor telepon dokter • spesialisasi dokter • tombol “non-aktif” untuk menonaktifkan dokter <p>Halaman ini juga menampilkan tombol “tambah dokter” yang digunakan saat admin rumah sakit akan menambah data dokter baru</p>

Skenario Alternatif : Kolom Pada Form Ada yang Kosong	
8c. Klik tombol <i>submit</i>	
	9c. Menampilkan pesan dibawah kolom yang kosong " <i>please fill out this field</i> "

A5 Scenario Mengelola Data Perawat

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif mengelola data perawat dijelaskan pada lampiran A.8.

Tabel A.0.8 Scenario Mengelola Data Perawat

Nama Use Case	Mengelola Data Perawat
Aktor	Admin Rumah Sakit
Deskripsi Singkat	Admin Rumah Sakit akan mengelola data perawat
Prekondisi	Data perawat
Prakondisi	1. Data perawat ditampilkan 2. Status perawat berhasil diubah 3. Data perawat berhasil ditambahkan
Flow of Events	
Skenario Normal : Melihat Data Perawat	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai admin rumah sakit	
	2. Menampilkan halaman <i>dashboard</i> admin rumah sakit
3. Klik menu perawat	
	4. Menampilkan halaman data perawat meliputi tabel data dokter yang bekerja pada rumah sakit. Tabel ini menampilkan : <ul style="list-style-type: none"> • NIP perawat • nama perawat

	<ul style="list-style-type: none"> • alamat perawat • nomor telepon perawat • jenis kelamin perawat • tombol “non-aktif” untuk menonaktifkan perawat <p>Halaman ini juga menampilkan tombol “tambah perawat” yang digunakan saat admin rumah sakit akan menambah data dokter baru</p>
Skenario Normal : Mengubah Status Perawat	
5a. Klik tombol non-aktif	
	6a. Mengubah status perawat pada <i>database</i> menjadi non-aktif
	7a. Menampilkan pesan “Status Perawat Non-aktif”
Skenario Normal : Menambah Data Perawat	
5a. Klik tombol tambah perawat	
	6a. Menampilkan halaman form data perawat
7a. Mengisi form data perawat yang meliputi NIP perawat, nama perawat, alamat perawat, nomor telepon perawat, jenis kelamin perawat, username akun perawat, dan <i>password</i> aku perawat	
8a. Klik tombol <i>submit</i>	
	9a. Menyimpan ke <i>database</i>
	10a. Menampilkan halaman data perawat meliputi tabel data dokter yang bekerja pada rumah sakit. Tabel ini menampilkan : <ul style="list-style-type: none"> • NIP perawat • nama perawat • alamat perawat • nomor telepon perawat • jenis kelamin perawat

	<ul style="list-style-type: none"> tombol “non-aktif” untuk menonaktifkan perawat <p>Halaman ini juga menampilkan tombol “tambah perawat” yang digunakan saat admin rumah sakit akan menambah data dokter baru</p>
Skenario Alternatif : Kolom Pada Form Ada yang Kosong	
8c. Klik tombol <i>submit</i>	
	9c. Menampilkan pesan dibawah kolom yang kosong “ <i>please fill out this field</i> ”

A6 Scenario Mengelola Data Ruang Rumah Sakit

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif mengelola data ruang rumah sakit dijelaskan pada lampiran A.9.

Tabel A.0.9 Scenario Mengelola Data Ruang Rumah Sakit

Nama Use Case	Mengelola Data Ruang Rumah Sakit
Aktor	Admin Rumah Sakit
Deskripsi Singkat	Admin rumah sakit akan mengelola data ruang rumah sakit
Prekondisi	Data ruang rumah sakit
Prakondisi	<ol style="list-style-type: none"> Data ruang rumah sakit berhasil ditambahkan Status ruang rumah sakit berhasil dibatalkan diubah
Flow of Events	
Skenario Normal : Menambah Data Ruang Rumah Sakit	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai admin rumah sakit	
	2. Menampilkan halaman <i>dashboard</i> admin rumah sakit

3. Klik menu ruangan	
	<p>4. Menampilkan table data ruangan yang ada di rumah sakit. Tabel ini menampilkan :</p> <ul style="list-style-type: none"> • nama ruangan • nama bagian ruangan • status ruangan (masih dipakai atau tidak) • tombol kosongkan untuk merubah status ruangan menjadi tidak dipakai <p>Halaman ini juga menampilkan tombol tambah ruangan yang digunakan apabila admin rumah sakit akan menambahkan data ruang ruamah sakit</p>
5. Klik tombol tambah ruangan	
	6. Menampilkan halaman form data ruang rumah sakit
7. Mengisi form data ruang rumah sakit yang meliputi nama ruangan, dan bagian.	
8. Klik tombol <i>submit</i>	
	9. Menyimpan ke <i>database</i>
	<p>10. Menampilkan table data ruangan yang ada di rumah sakit. Tabel ini menampilkan :</p> <ul style="list-style-type: none"> • nama ruangan • nama bagian ruangan • status ruangan (masih dipakai atau tidak) • tombol kosongkan untuk merubah status ruangan menjadi tidak dipakai <p>Halaman ini juga menampilkan tombol tambah ruangan yang digunakan apabila admin rumah sakit akan menambahkan data ruang ruamah sakit</p>

Skenario Alternatif : Kolom Pada Form Ada yang Kosong	
8b. Klik tombol <i>submit</i>	
	9b. Menampilkan pesan dibawah kolom yang kosong “ <i>please fill out this field</i> ”
Skenario Normal : Mengubah Status Data Ruang Rumah Sakit	
5a. Klik tombol kosongkan	
	6a. Mengubah status ruang pada <i>database</i> menjadi tidak terpakai
	7a. Menampilkan pesan “Ruangan Dihapus”

A7 Scenario Melihat Data Permintaan Darah

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif melihat data permintaan darah dijelaskan pada lampiran A.10.

Tabel A.0.10 Scenario Melihat Data Permintaan Darah

Nama Use Case	Melihat Data Permintaan Darah
Aktor	Admin Rumah Sakit
Deskripsi Singkat	Admin rumah sakit akan melihat data permintaan darah
Prekondisi	Rekap data permintaan darah
Prakondisi	Detail data permintaan darah yang dibutuhkan ditampilkan
<i>Flow of Events</i>	
Skenario Normal : Melihat Data Permintaan Darah	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai admin rumah sakit	
	2. Menampilkan halaman <i>dashboard</i> admin rumah sakit
3. Klik menu permintaan darah	

	<p>4. Menampilkan tabel daftar permintaan darah yang dilakukan pada hari itu. Tabel ini meliputi :</p> <ul style="list-style-type: none"> • nama pasien • nama bagian perawatan • nama ruang tempat pasien dirawat • sifat permintaan (emergency atau biasa) • tombol view untuk menampilkan detail permintaan
<p>5. Masukkan tanggal pada kolom pencarian</p>	
<p>6. Klik tombol cari</p>	
	<p>7. Menampilkan tabel daftar permintaan darah yang dilakukan pada tanggal yang dicari. Tabel ini meliputi :</p> <ul style="list-style-type: none"> • nama pasien • nama bagian perawatan • nama ruang tempat pasien dirawat • sifat permintaan (emergency atau biasa) • tombol view untuk menampilkan detail permintaan
<p>8. Klik tombol ber-<i>icon</i> kaca pembesar pada data yang ingin dilihat</p>	
	<p>9. Menampilkan detail data permintaan darah berbentuk form yang terdiri dari :</p> <ul style="list-style-type: none"> • id permintaan • nama bagian perawatan • nama ruang tempat pasien dirawat • nama pasien • jenis kelamin pasien • usia pasien

	<ul style="list-style-type: none"> • alamat pasien • tanggal permintaan • tanggal validasi UTD • alasan transfuse • golongan darah • produk darah • jumlah permintaan • sifat permintaan • status permintaan
--	---

A8 Scenario Melihat Stok Darah

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif melihat stok darah dijelaskan pada lampiran A.11 sampai A.15

Tabel A.0.11 Scenario Melihat Stok Darah oleh Admin Pelayanan UTD

Nama Use Case	Melihat Data Stok Darah
Aktor	Admin Pelayanan UTD
Deskripsi Singkat	Admin pelayanan UTD akan melihat data stok darah
Prekondisi	Data stok darah belum ditampilkan
Prakondisi	Data stok darah
Flow of Events	
Skenario Normal : Melihat Data Stok Darah	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai admin pelayanan UTD	
	2. Menampilkan halaman <i>dashboard</i> admin pelayanan UTD yang meliputi tabel data stok darah

Tabel A.0.12 Scenario Melihat Stok Darah oleh Admin Penjamin Mutu UTD

Nama Use Case	Melihat Data Stok Darah
----------------------	-------------------------

Aktor	Admin Penjamin Mutu UTD
Deskripsi Singkat	Admin penjamin mutu UTD akan melihat data stok darah
Prekondisi	Data stok darah belum ditampilkan
Prakondisi	Data stok darah
<i>Flow of Events</i>	
Skenario Normal : Melihat Data Stok Darah	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai admin penjamin mutu UTD	
	2. Menampilkan halaman <i>dashboard</i> admin penjamin mutu UTD yang meliputi tabel data stok darah

Tabel A.0.13 *Scenario* Melihat Stok Darah oleh Perawat

Nama Use Case	Melihat Data Stok Darah
Aktor	Perawat
Deskripsi Singkat	Perawat akan melihat data stok darah
Prekondisi	Data stok darah belum ditampilkan
Prakondisi	Data stok darah
<i>Flow of Events</i>	
Skenario Normal : Melihat Data Stok Darah	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai perawat	
	2. Menampilkan halaman <i>dashboard</i> perawat yang meliputi tabel data stok darah

Tabel A.0.14 *Scenario* Melihat Stok Darah oleh Admin Rumah Sakit

Nama Use Case	Melihat Data Stok Darah
----------------------	-------------------------

Aktor	Admin Rumah Sakit
Deskripsi Singkat	Admin rumah sakit akan melihat data stok darah
Prekondisi	Data stok darah belum ditampilkan
Prakondisi	Data stok darah
<i>Flow of Events</i>	
Skenario Normal : Melihat Data Stok Darah	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai admin rumah sakit	
	2. Menampilkan halaman <i>dashboard</i> admin rumah sakit yang meliputi tabel data stok darah

Tabel A.0.15 Scenario Melihat Stok Darah oleh User Umum

Nama Use Case	Melihat Data Stok Darah
Aktor	User Umum
Deskripsi Singkat	User umum akan melihat data stok darah
Prekondisi	Data stok darah belum ditampilkan
Prakondisi	Data stok darah
<i>Flow of Events</i>	
Skenario Normal : Melihat Data Stok Darah	
Aksi Aktor	Reaksi
1. Membuka halaman <i>website</i>	
	2. Menampilkan halaman beranda <i>website</i> yang meliputi tabel data stok darah

A9 *Scenario* Validasi Data Permintaan Darah

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif validasi data permintaan darah dijelaskan pada lampiran A.16

Tabel A.0.16 *Scenario* Validasi Data Permintaan Darah

Nama Use Case	Validasi Data Permintaan Darah
Aktor	Admin Pelayanan UTD
Deskripsi Singkat	Admin pelayanan UTD akan melakukan validasi data permintaan darah
Prekondisi	Data permintaan darah belum divalidasi
Prakondisi	<ol style="list-style-type: none"> 1. Data permintaan darah disetujui 2. Data permintaan darah ditolak
Flow of Events	
Skenario Normal : Menerima Data Permintaan Darah	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai admin pelayanan UTD	
	2. Menampilkan halaman <i>dashboard</i> admin pelayanan UTD
3. Klik menu permintaan darah	
	<p>4. Menampilkan halaman data permintaan darah yang belum divalidasi meliputi tabel data permintaan darah yang belum divalidas. Tabel ini meliputi:</p> <ul style="list-style-type: none"> • tanggal permintaan • nama pasien • nama rumah sakit • jumlah • tombol view untuk melihat detail permintaan <p>Halaman ini juga menampilkan kolom</p>

	pencarian berdasarkan tanggal permintaan
5. Klik tombol ber- <i>icon</i> kaca pembesar	
	<p>6. Menampilkan detail permintaan darah yang terdiri dari :</p> <ul style="list-style-type: none"> • id permintaan • nama rumah sakit • nama bagian perawatan • nama ruang tempat pasien dirawat • nama dokter yang menangani pasien • nama pasien • jenis kelamin pasien • usia pasien • alamat pasien • tanggal permintaan • tanggal validasi UTD • alasan transfuse • golongan darah • produk darah • jumlah permintaan • sifat permintaan • status permintaan
7. Klik tombol terima	
	8. Menyimpan ke <i>database</i>
	<p>9. Menampilkan halaman data permintaan darah yang belum divalidasi meliputi tabel data permintaan darah yang belum divalidas. Tabel ini meliputi:</p> <ul style="list-style-type: none"> • tanggal permintaan • nama pasien • nama rumah sakit • jumlah • tombol view untuk melihat detail permintaan <p>Halaman ini juga menampilkan kolom</p>

	pencarian berdasarkan tanggal permintaan
Skenario Normal : Menolak Data Permintaan Darah	
7a. Klik tombol tolak	
	8a. Menyimpan ke <i>database</i>
	<p>9a. Menampilkan halaman data permintaan darah yang belum divalidasi meliputi tabel data permintaan darah yang belum divalidas. Tabel ini meliputi:</p> <ul style="list-style-type: none"> • tanggal permintaan • nama pasien • nama rumah sakit • jumlah • tombol view untuk melihat detail permintaan <p>Halaman ini juga menampilkan kolom pencarian berdasarkan tanggal permintaan</p>

A10 Scenario Mengelola Data Pendonor

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif mengelola data pendonor dijelaskan pada lampiran A.17.

Tabel A.0.17 Scenario Mengelola Data Pendonor

Nama Use Case	Mengelola Data Pendonor
Aktor	Admin Pelayanan UTD
Deskripsi Singkat	Admin Pelayanan UTD akan mengelola data pendonor
Prekondisi	Data pendonor
Prakondisi	1. Data pendonor berhasil ditambahkan 2. Data pendonor berhasil ditampilkan
<i>Flow of Events</i>	
Skenario Normal : Menambah Data Pendonor	
Aksi Aktor	Reaksi

1. <i>Login</i> sebagai admin pelayanan UTD	
	2. Menampilkan halaman <i>dashboard</i> admin rumah sakit
3. Klik menu donor darah	
	<p>4. Menampilkan table data pendonor yang menampilkan :</p> <ul style="list-style-type: none"> • nomor identitas • nama pendonor • golongan darah • status pendonor (cekal atau tidak) • tombol view untuk menampilkan detail permintaan <p>Halaman ini juga menampilkan tombol tambah pendonor yang digunakan apabila admin akan menambahkan data pendonor</p>
5. Klik tombol tambah pendonor	
	6. Menampilkan halaman form data pendonor
7. Mengisi form data pendonor yang meliputi nomor identitas, nama, alamat, telepon, pekerjaan, tempat lahir, tanggal lahir, golongan darah, kesediaan untuk mendonor saat puasa, tanggal terakhir kali donor, dan banyak donor yang telah dilakukan	
8. Klik tombol <i>submit</i>	
	9. Menyimpan ke <i>database</i>
	10. Menampilkan table data pendonor yang menampilkan :

	<ul style="list-style-type: none"> • nomor identitas • nama pendonor • golongan darah • status pendonor (cekal atau tidak) • tombol view untuk menampilkan detail permintaan <p>Halaman ini juga menampilkan tombol tambah pendonor yang digunakan apabila admin akan menambahkan data pendonor</p>
Skenario Alternatif : Kolom Pada Form Ada yang Kosong	
8a. Klik tombol <i>submit</i>	
	9a. Menampilkan pesan dibawah kolom yang kosong “ <i>please fill out this field</i> ”
Skenario Normal : Mencari Data Pendonor	
5b. Masukkan nama pendonor pada kolom pencarian yang ada di atas table data pendonor	
6b. Klik cari	
	<p>7b. Menampilkan table data pendonor yang dicari dimana menampilkan :</p> <ul style="list-style-type: none"> • nomor identitas • nama pendonor • golongan darah • status pendonor (cekal atau tidak) • tombol view untuk menampilkan detail permintaan <p>Halaman ini juga menampilkan tombol tambah pendonor yang digunakan apabila admin akan menambahkan data pendonor</p>
8b. Klik tombol ber- <i>icon</i> kaca pembesar	

	<p>9b. Menampilkan detail data pendonor berbentuk form yang terdiri dari :</p> <ul style="list-style-type: none"> • nomor identitas • nama pendonor • alamat pendonor • nomor telepon pendonor • pekerjaan pendonor • tempat lahir pendonor • tanggal lahir pendonor • usia pendonor • golongan darah pendonor • kesediaan donor saat puasa • tanggal terakhir donor • banyak donor yang telah dilakukan • status pendonor <p>Halaman ini juga menampilkan tombol donor darah untuk melakukan donor darah dan tombol kembali untuk menampilkan halaman sebelumnya</p>
10b. Klik tombol kembali	
	<p>11b. Menampilkan table data pendonor yang menampilkan :</p> <ul style="list-style-type: none"> • nomor identitas • nama pendonor • golongan darah • status pendonor (cekal atau tidak) • tombol view untuk menampilkan detail permintaan <p>Halaman ini juga menampilkan tombol tambah pendonor yang digunakan apabila admin akan menambahkan data pendonor.</p>

A.11 *Scenario* Menambah Data Transaksi Donor

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif menambah data transaksi donor dijelaskan pada lampiran A.18.

Tabel A.0.18 *Scenario* Menambah Data Transaksi Donor

Nama Use Case	Menambah Data Transaksi Donor
Aktor	Admin Pelayanan UTD
Deskripsi Singkat	Admin Pelayanan UTD akan menambahkan data transaksi donor
Prekondisi	Data transaksi donor
Prakondisi	Data transaksi donor berhasil ditambahkan
Flow of Events	
Skenario Normal : Menambah Data Transaksi Donor	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai admin rumah sakit	
	2. Menampilkan halaman <i>dashboard</i> admin rumah sakit
3. Klik menu donor darah	
4. Masukkan nama pendonor pada kolom pencarian yang ada di atas table data pendonor	
5. Klik cari	
	6. Menampilkan tabel data pendonor yang dicari. Table data pendonor ini menampilkan : <ul style="list-style-type: none"> • nomor identitas • nama pendonor • golongan darah

	<ul style="list-style-type: none"> • status pendonor (cekal atau tidak) • tombol view untuk menampilkan detail permintaan <p>Halaman ini juga menampilkan tombol tambah pendonor yang digunakan apabila admin akan menambahkan data pendonor</p>
7. Klik tombol ber- <i>icon</i> kaca pembesar pada data yang ingin dilihat	
	<p>8. Menampilkan detail data pendonor berbentuk form yang terdiri dari :</p> <ul style="list-style-type: none"> • nomor identitas • nama pendonor • alamat pendonor • nomor telepon pendonor • pekerjaan pendonor • tempat lahir pendonor • tanggal lahir pendonor • usia pendonor • golongan darah pendonor • kesediaan donor saat puasa • tanggal terakhir donor • banyak donor yang telah dilakukan • status pendonor <p>Halaman ini juga menampilkan tombol donor darah untuk melakukan donor darah dan tombol kembali untuk menampilkan halaman sebelumnya</p>
9. Klik tombol donor darah	
	<p>10. Menampilkan tabel riwayat donor yang menampilkan tanggal, kada Hb, berat badan, status donor, dan keterangan.</p>

	Halaman ini juga menampilkan form data donor darah
11. Mengisi form data donor yang meliputi macam donor, kadar HB pendonor, berat badan pendonor, status donor, dan alasan status	
12. Klik tombol <i>submit</i>	
	13. Menyimpan ke <i>database</i>
	<p>14. Menampilkan tabel data pendonor yang dicari. Table data pendonor ini menampilkan :</p> <ul style="list-style-type: none"> • nomor identitas • nama pendonor • golongan darah • status pendonor (cekal atau tidak) • tombol view untuk menampilkan detail permintaan <p>Halaman ini juga menampilkan tombol tambah pendonor yang digunakan apabila admin akan menambahkan data pendonor</p>
Skenario Normal : Batal Menambahkan Data Permintaan Darah	
12a. Kik tombol kembali	
	<p>13a. Menampilkan tabel data pendonor yang dicari. Table data pendonor ini menampilkan :</p> <ul style="list-style-type: none"> • nomor identitas • nama pendonor • golongan darah • status pendonor (cekal atau tidak) • tombol view untuk menampilkan detail permintaan <p>Halaman ini juga menampilkan tombol tambah pendonor yang digunakan apabila admin akan menambahkan data pendonor</p>

Skenario Alternatif : Kolom Pada Form Ada yang Kosong	
12b. Klik tombol <i>submit</i>	
	13b. Menampilkan pesan dibawah kolom yang kosong “ <i>please fill out this field</i> ”

A.12 Scenario Menambah Data Rumah Sakit

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif menambah data rumah sakit dijelaskan pada lampiran A.19

Tabel A.0.19 Scenario Menambah Data Rumah Sakit

Nama Use Case	Menambahkan Data Rumah Sakit
Aktor	Admin Pelayanan UTD
Deskripsi Singkat	Admin pelayanan UTD akan menambah data rumah sakit
Prekondisi	Data rumah sakit
Prakondisi	Data rumah sakit berhasil ditambahkan
Flow of Events	
Skenario Normal : Menambah Data Rumah Sakit	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai admin pelayanan UTD	
	2. Menampilkan halaman <i>dashboard</i> admin pelayanan UTD
3. Klik menu rumah sakit	
	4. Menampilkan table data rumah sakit yang meliputi : <ul style="list-style-type: none"> • nomor register rumah sakit • nama rumah sakit • alamat rumah sakit Halaman ini juga menampilkan tombol tambah untuk

	menambahkan data rumah sakit baru
5. Klik tombol tambah	
	6. Menampilkan halaman form data rumah sakit
7. Mengisi form data rumah sakit yang meliputi nomor registrasi rumah sakit, nama rumah sakit, dan alamat rumah sakit	
8. Klik tombol <i>submit</i>	
	9. Menyimpan ke <i>database</i>
	10. Menampilkan table data rumah sakit yang meliputi : <ul style="list-style-type: none"> • nomor register rumah sakit • nama rumah sakit • alamat rumah sakit Halaman ini juga menampilkan tombol tambah untuk menambahkan data rumah sakit baru
Skenario Alternatif : Kolom Pada Form Ada yang Kosong	
8a. Klik tombol <i>submit</i>	
	9a. Menampilkan pesan dibawah kolom yang kosong " <i>please fill out this field</i> "

A.13 *Scenario* Melihat Data Transaksi Donor dan Transaksi Permintaan Darah
 Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif melihat data transaksi donor dan transaksi permintaan darah dijelaskan pada lampiran A.20.

Tabel A.0.20 *Scenario* Melihat Data Transaksi Donor dan Transaksi Permintaan Darah

Nama Use Case	Melihat Rekap Data Transaksi Donor dan Transaksi Permintaan Darah
Aktor	Admin Penjamin Mutu UTD
Deskripsi Singkat	Admin pelayanan UTD akan melihat data transaksi donor dan transaksi permintaan darah
Prekondisi	Data transaksi donor dan transaksi permintaan darah belum ditampilkan
Prakondisi	<ol style="list-style-type: none"> 1. Rekap data transaksi donor berhasil ditampilkan 2. Rekap data transaksi permintaan darah berhasil ditampilkan
Flow of Events	
Skenario Normal : Melihat Data Transaksi Donor	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai admin pelayanan UTD	
	2. Menampilkan halaman <i>dashboard</i> admin pelayanan UTD
3. Klik menu transaksi darah	
	<p>4. Menampilkan halaman transaksi darah yang memiliki dua tabel yaitu tabel transaksi donor dan tabel transaksi permintaan darah. Tabel transaksi donor terdiri dari :</p> <ul style="list-style-type: none"> • tanggal donor • nama pendonor • golongan darah • tombol view untuk melihat detail <p>Sedangkan tabel transaksi permintaan darah terdiri dari :</p> <ul style="list-style-type: none"> • tanggal permintaan

	<ul style="list-style-type: none"> • nama pasien • nama rumah sakit • jumlah yang diminta • tombol view untuk melihat detail <p>Halaman ini juga memiliki kolom pencarian berdasarkan tanggal pada masing-masing tabel.</p> <p>Dibawah tabel akan di berikan jumlah transaksi donor per hari dan jumlah transaksi permintaan darah per hari pula.</p>
5. Isikan tanggal pada kolom pencarian.	
6. Klik tombol cari	
	7. Menampilkan data transaksi donor yang dicari
8. Klik tombol ber- <i>icon</i> kaca pembesar pada data yang akan dilihat	
	<p>9. Menampilkan detail data transaksi donor dalam bentuk form yang menyajikan :</p> <ul style="list-style-type: none"> • nomor identitas pendonor • nama pendonor • alamat pendonor • nomor telepon pendonor • tanggal darah masuk • macam donor • kualitas darah • keterangan kualitas darah • produk darah
10. Klik tombol kembali	
	11. Menampilkan halaman transaksi darah yang memiliki dua tabel yaitu tabel transaksi

	<p>donor dan tabel transaksi permintaan darah. Tabel transaksi donor terdiri dari :</p> <ul style="list-style-type: none"> • tanggal donor • nama pendonor • golongan darah • tombol view untuk melihat detail <p>Sedangkan tabel transaksi permintaan darah terdiri dari :</p> <ul style="list-style-type: none"> • tanggal permintaan • nama pasien • nama rumah sakit • jumlah yang diminta • tombol view untuk melihat detail <p>Halaman ini juga memiliki kolom pencarian berdasarkan tanggal pada masing-masing tabel.</p> <p>Dibawah tabel akan di berikan jumlah transaksi donor per hari dan jumlah transaksi permintaan darah per hari pula.</p>
Skenario Normal : Melihat Data Transaksi Permintaan Darah	
5a. Isikan tanggal pada kolom pencarian.	
6a. Klik tombol cari	
	7a. Menampilkan data transaksi permintaan darah yang dicari
8a. Klik tombol ber- <i>icon</i> kaca pembesar pada data yang akan dilihat	
	<p>9a. Menampilkan detail data transaksi permintaan darah dalam bentuk form yang menyajikan:</p> <ul style="list-style-type: none"> • id permintaan • nama rumah sakit

	<ul style="list-style-type: none"> • nama bagian perawatan • nama ruang tempat pasien dirawat • nama pasien • jenis kelamin pasien • usia pasien • alamat pasien • tanggal permintaan • tanggal validasi UTD • alasan transfuse • golongan darah • produk darah • jumlah permintaan • sifat permintaan • status permintaan • id donor
10a. Klik tombol kembali	
	<p>11 a. Menampilkan halaman transaksi darah yang memiliki dua tabel yaitu tabel transaksi donor dan tabel transaksi permintaan darah. Tabel transaksi donor terdiri dari :</p> <ul style="list-style-type: none"> • tanggal donor • nama pendonor • golongan darah • tombol view untuk melihat detail <p>Sedangkan tabel transaksi permintaan darah terdiri dari :</p> <ul style="list-style-type: none"> • tanggal permintaan • nama pasien • nama rumah sakit • jumlah yang diminta • tombol view untuk melihat detail <p>Halaman ini juga memiliki kolom pencarian berdasarkan tanggal pada masing-masing tabel.</p>

	Dibawah tabel akan di berikan jumlah transaksi donor per hari dan jumlah transaksi permintaan darah per hari pula.
--	--

A.14 Scenario Verifikasi Kualitas Darah Donor

Penjelasan alur aksi aktor dan reaksi sistem informasi peramalan jumlah permintaan darah dalam skenario normal dan skenario alternatif verifikasi kualitas darah donor dijelaskan pada lampiran A.21.

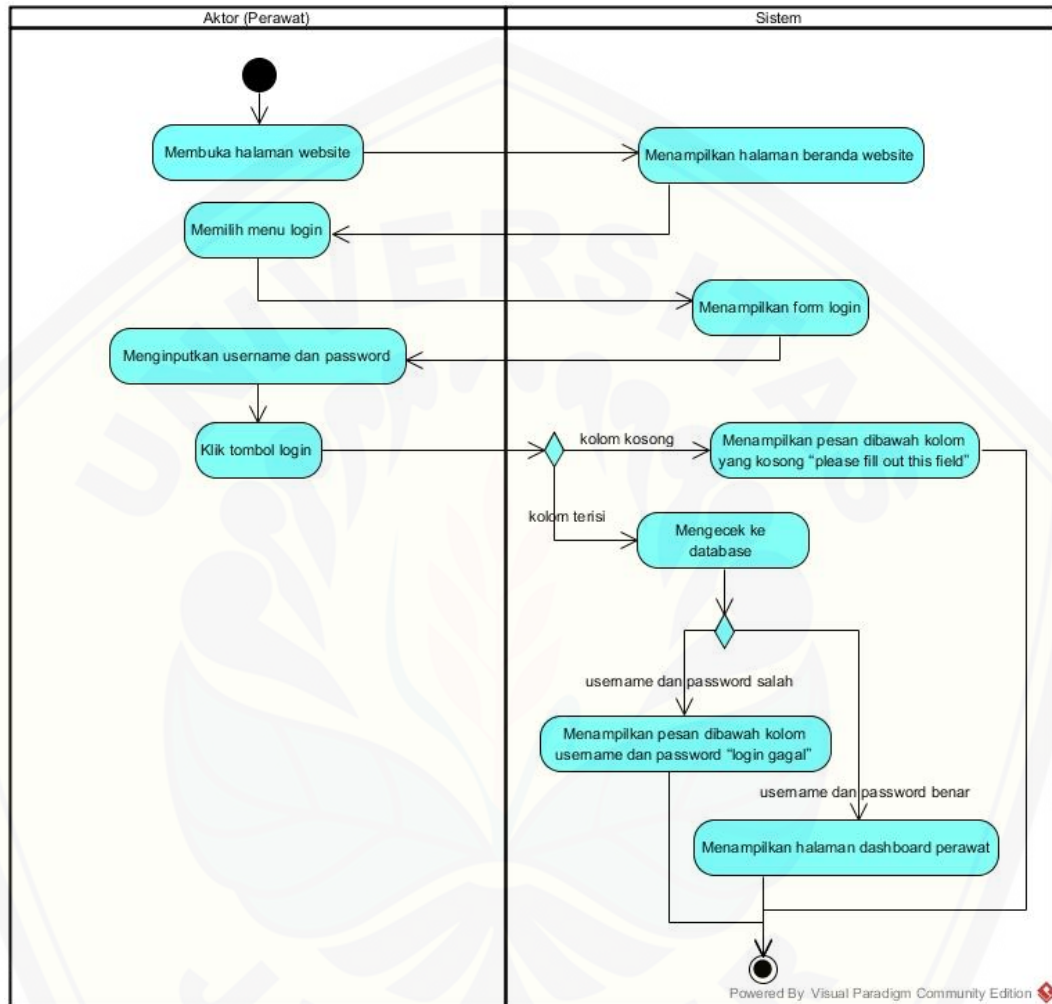
Tabel A.0.21 Scenario Verifikasi Kualitas Darah Donor

Nama Use Case	Verifikasi Kualitas Darah Donor
Aktor	Admin Penjamin Mutu UTD
Deskripsi Singkat	Admin penjamin mutu UTD akan melakukan validasi kualitas darah donor
Prekondisi	Data donor
Prakondisi	Data donor telah diverifikasi kualitasnya
Flow of Events	
Skenario Normal : Verifikasi Kualitas Darah Donor	
Aksi Aktor	Reaksi
1. <i>Login</i> sebagai admin penjamin mutu UTD	
	2. Menampilkan halaman <i>dashboard</i> admin penjamin mutu UTD
3. Klik menu kualitas darah	
	4. Menampilkan halaman kualitas darah yang memiliki tabel daftar donor yang belum di validasi. Tabel donor terdiri dari : <ul style="list-style-type: none"> • tanggal donor • nama pendonor • golongan darah

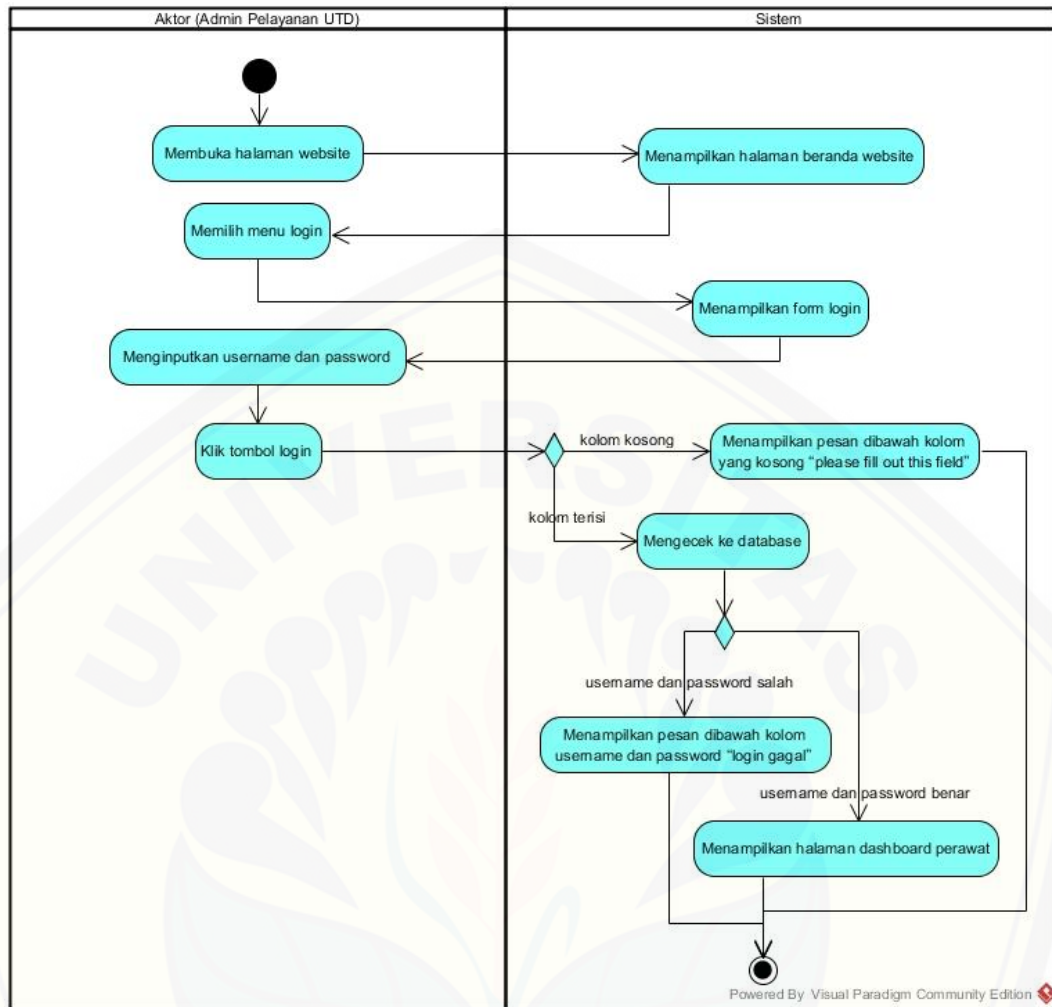
	<ul style="list-style-type: none"> • berat badan • kadar Hb • tombol verifikasi untuk menuju ke halaman verifikasi
5. Klik tombol verifikasi pada data yang akan di verifikasi	
	6. Menampilkan halaman form verifikasi kualitas darah donor
7. Mengisi form data donor yang meliputi kualitas darah, keterangan kualitas darah, dan darah tersebut akan diolah menjadi produk darah apa	
8. Klik tombol <i>submit</i>	
	9. Menyimpan ke <i>database</i>
	<p>10. Menampilkan halaman kualitas darah yang memiliki tabel daftar donor yang belum di validasi. Tabel donor terdiri dari :</p> <ul style="list-style-type: none"> • tanggal donor • nama pendonor • golongan darah • berat badan • kadar Hb • tombol verifikasi untuk menuju ke halaman verifikasi

Lampiran B. Activity Diagram

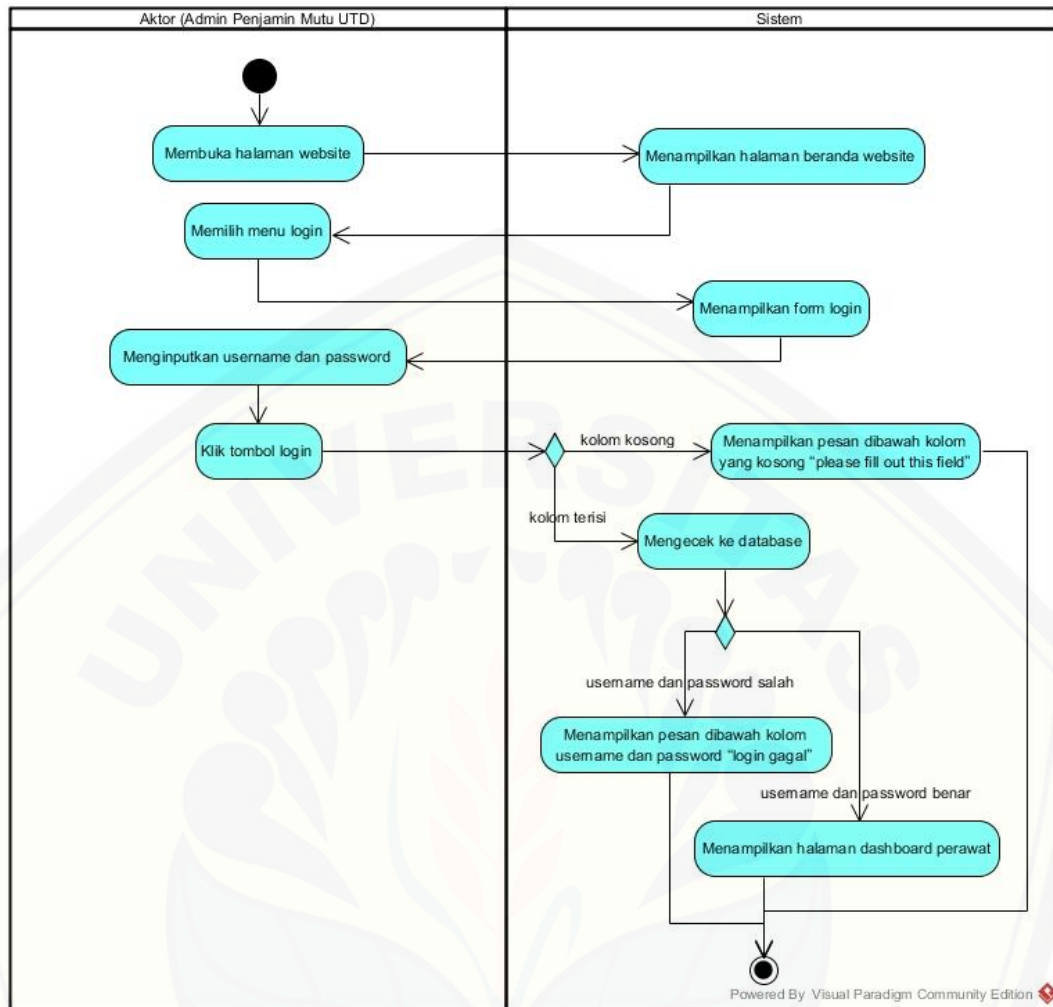
B1 Activity Diagram Login



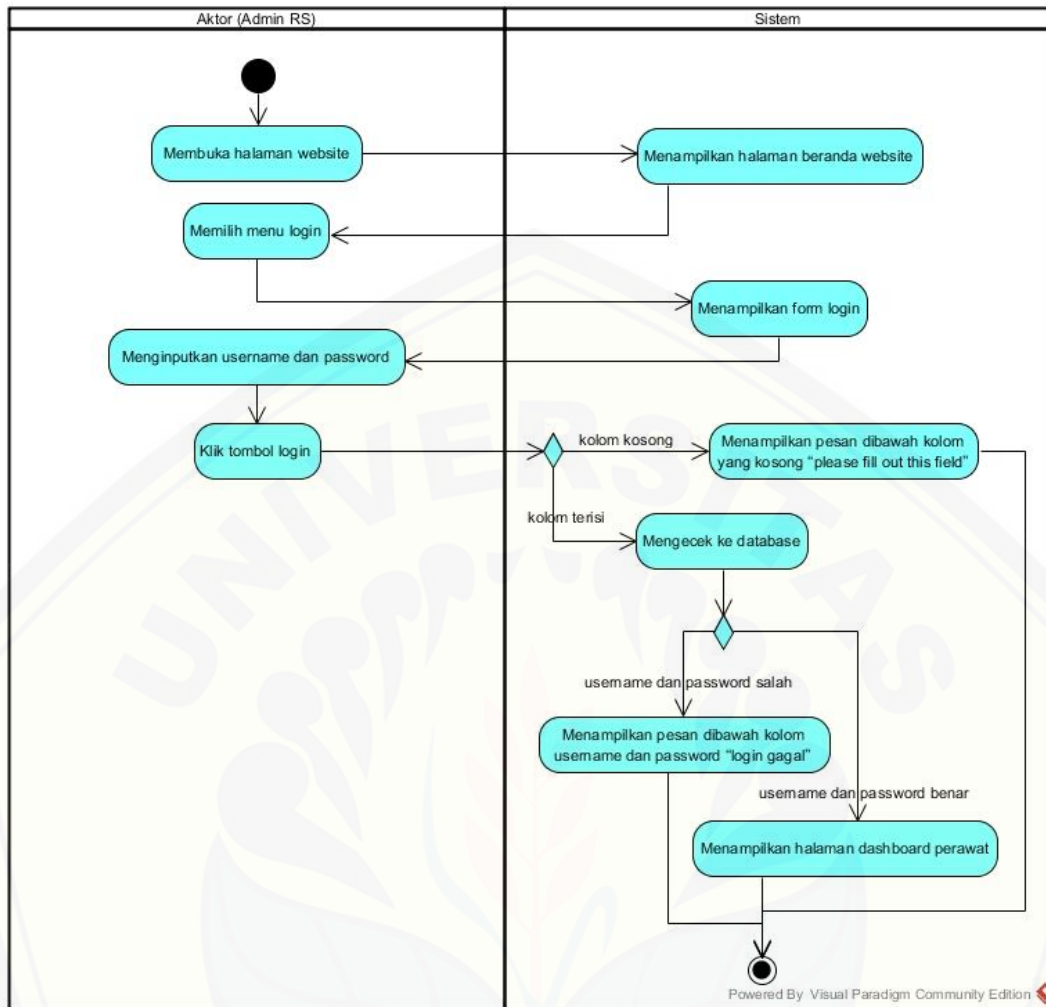
Gambar B. 1 Activity Diagram Login (Perawat)



Gambar B. 2 Activity Diagram Login (Admin Pelayanan UTD)

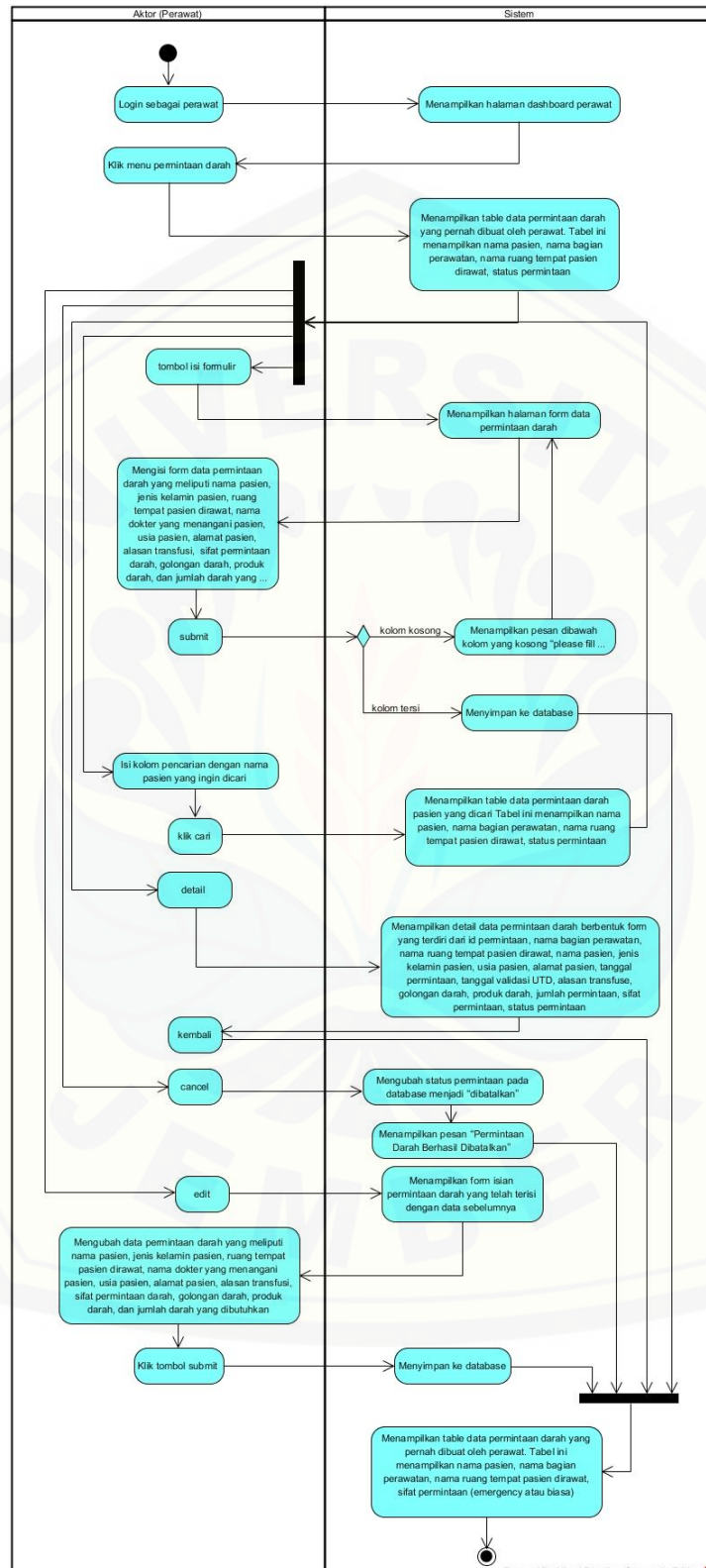


Gambar B. 3 Activity Diagram Login (Admin Penjamin Mutu UTD)



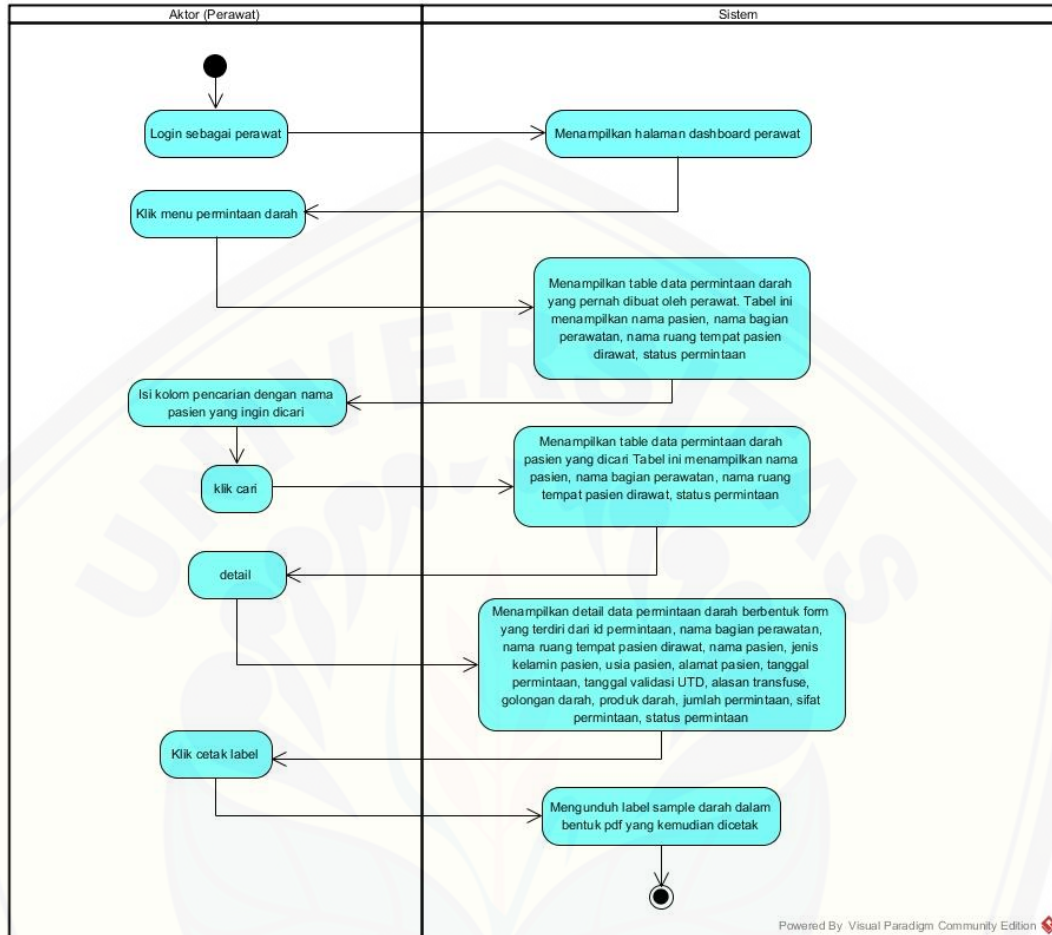
Gambar B. 4 Activity Diagram Login (Admin Rumah Sakit)

B2 Activity Diagram Mengelola Data Permintaan Darah



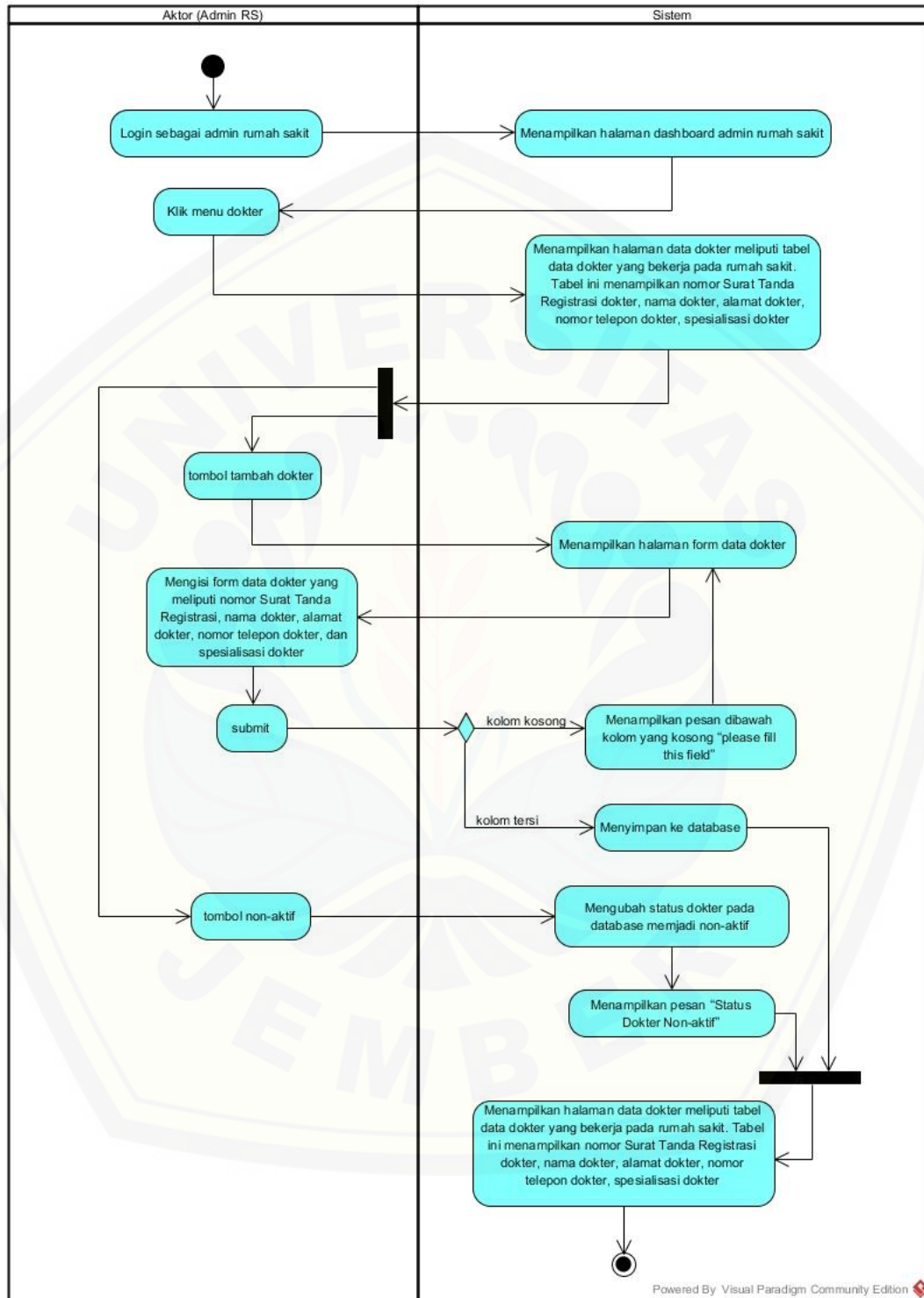
Gambar B. 5 Activity Diagram Mengelola Data Permintaan Darah

B3 Activity Diagram Cetak Label Sample Darah



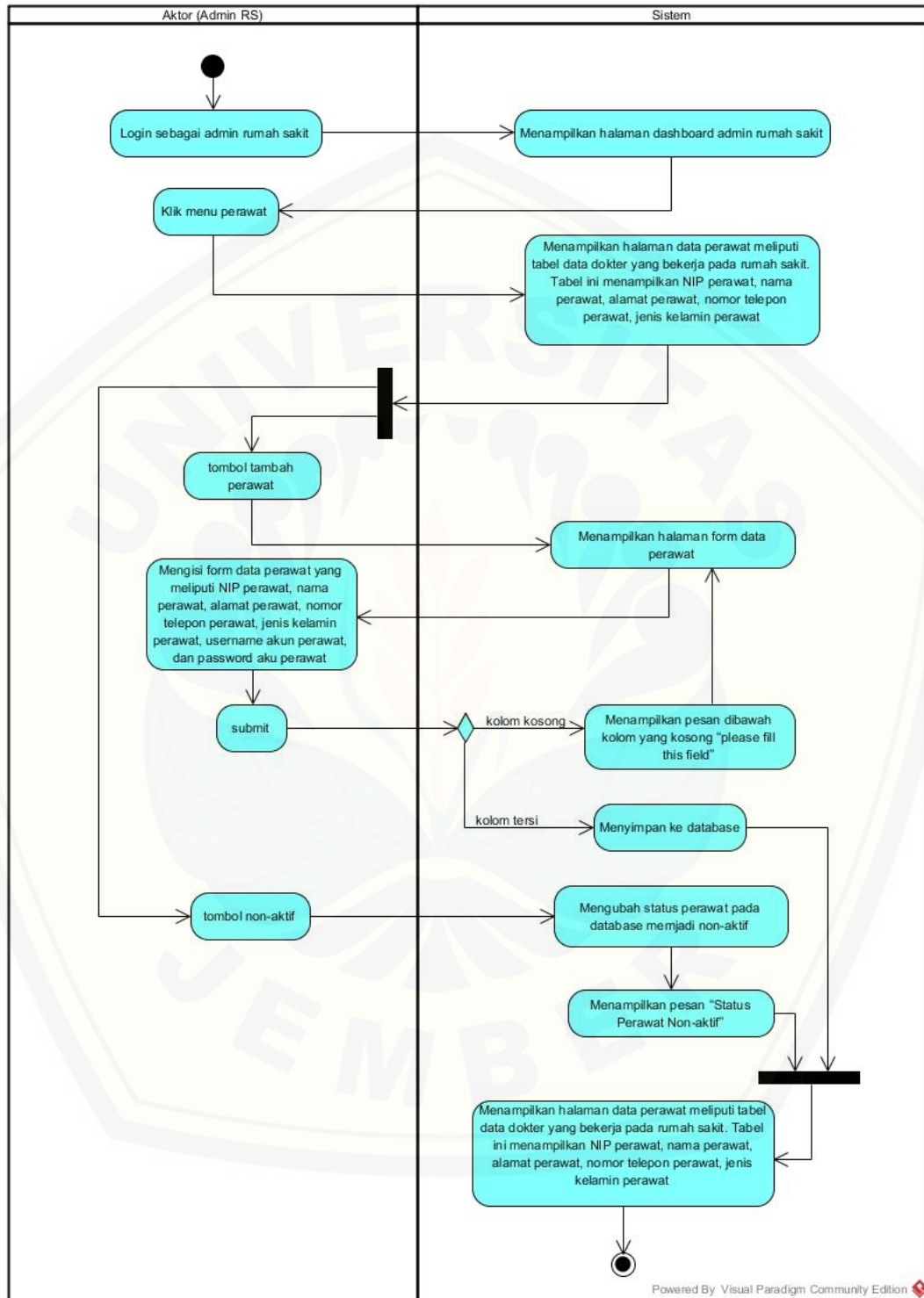
Gambar B. 6 Activity Diagram Cetak Label Sample Darah

B4 Activity Diagram Mengelola Data Dokter



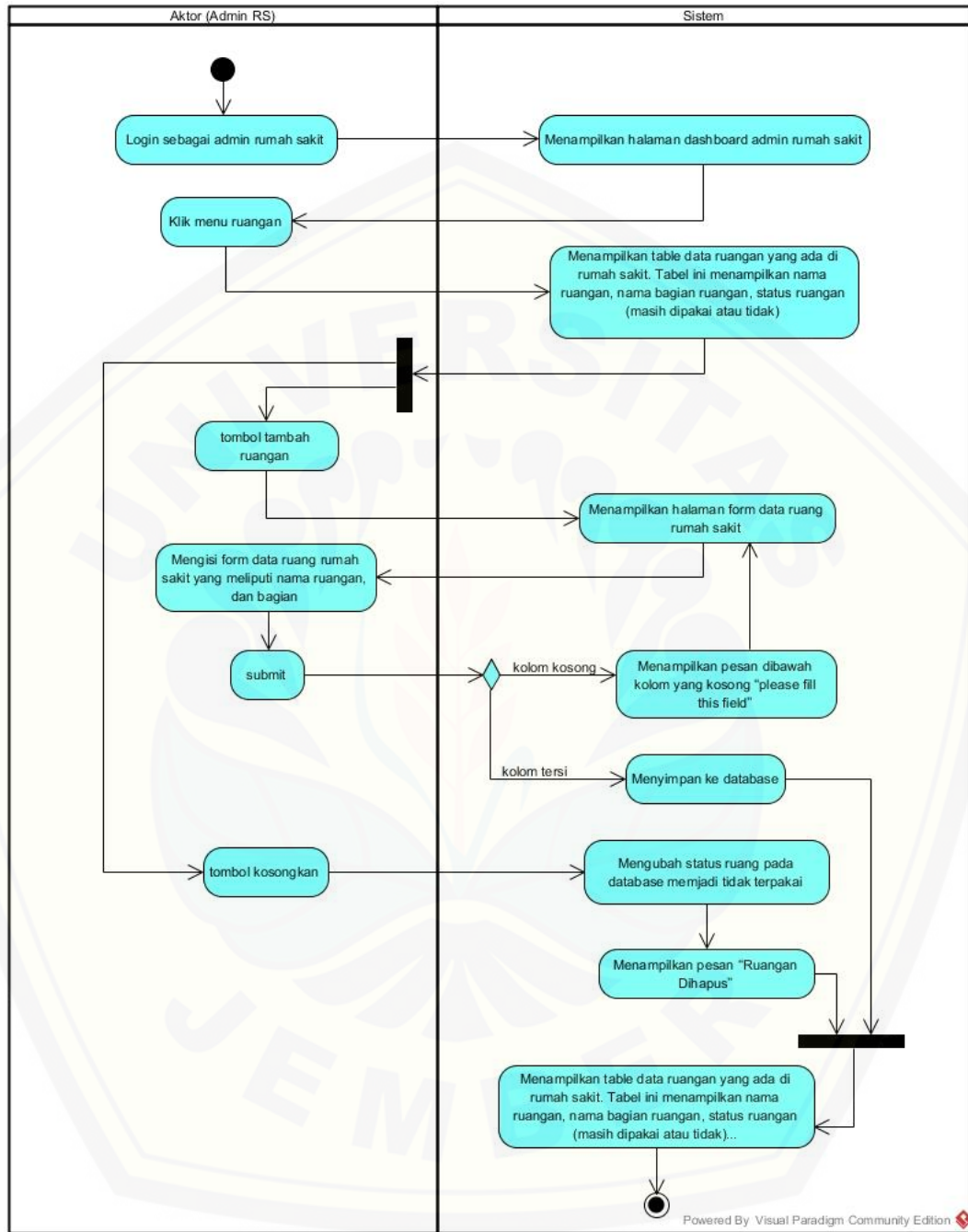
Gambar B. 7 Activity Diagram Mengelola Data Dokter

B5 Activity Diagram Mengelola Data Perawat



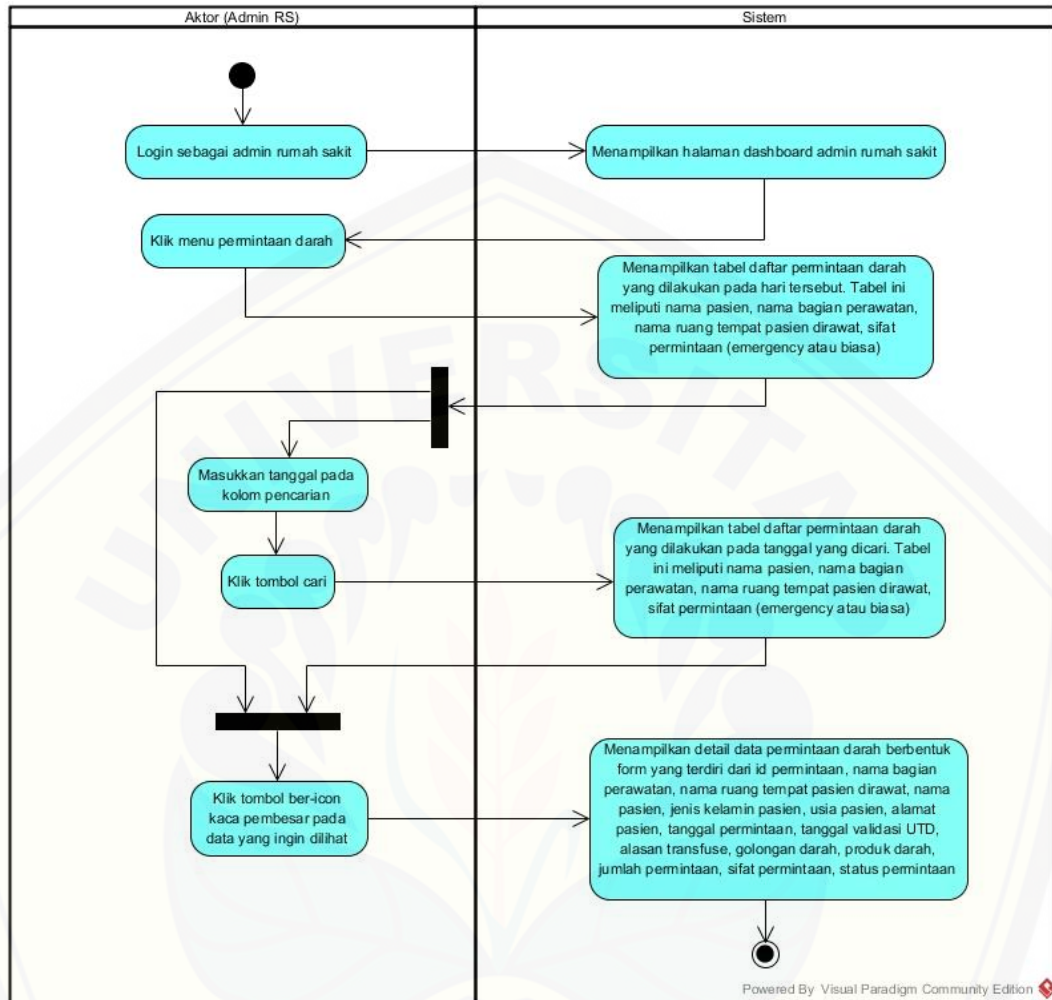
Gambar B. 8 Activity Diagram Mengelola Data Perawat

B6 Activity Diagram Mengelola Data Ruang Rumah Sakit



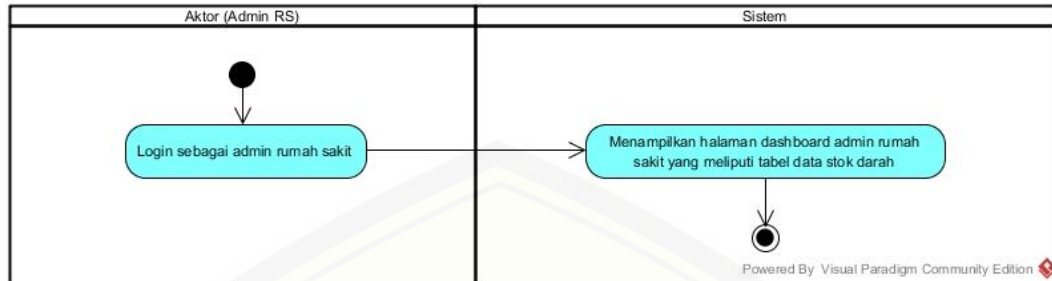
Gambar B. 9 Activity Diagram Mengelola Data Ruang Rumah Sakit

B7 Activity Diagram Melihat Data Permintaan Darah

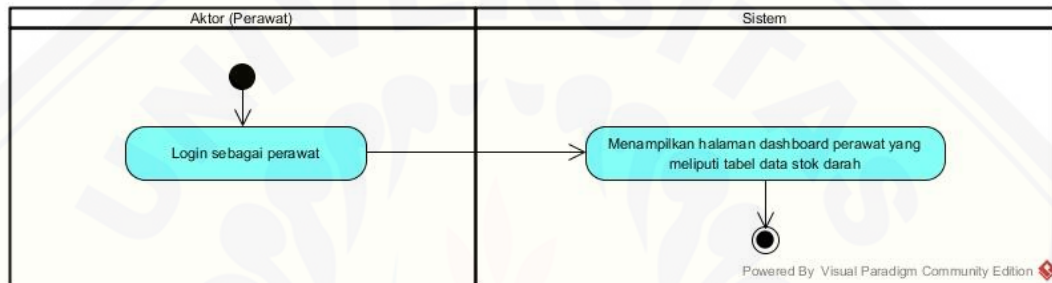


Gambar B. 10 Activity Diagram Melihat Data Permintaan Darah

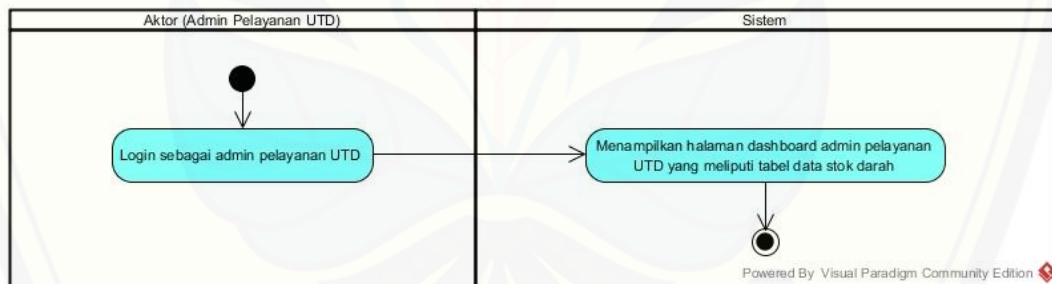
B8 Activity Diagram Melihat Stok Darah



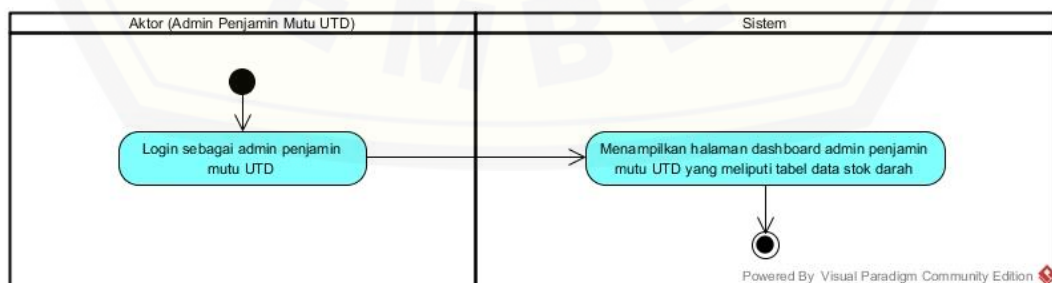
Gambar B. 11 Activity Diagram Melihat Stok Darah (Admin Rumah Sakit)



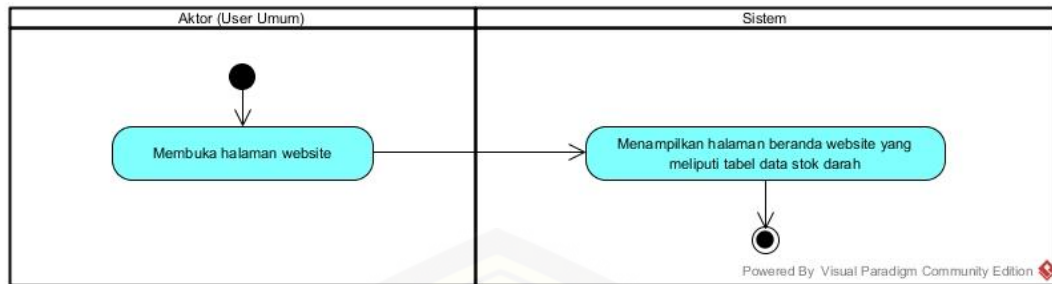
Gambar B. 12 Activity Diagram Melihat Stok Darah (Perawat)



Gambar B. 13 Activity Diagram Melihat Stok Darah (Admin Pelayanan UTD)

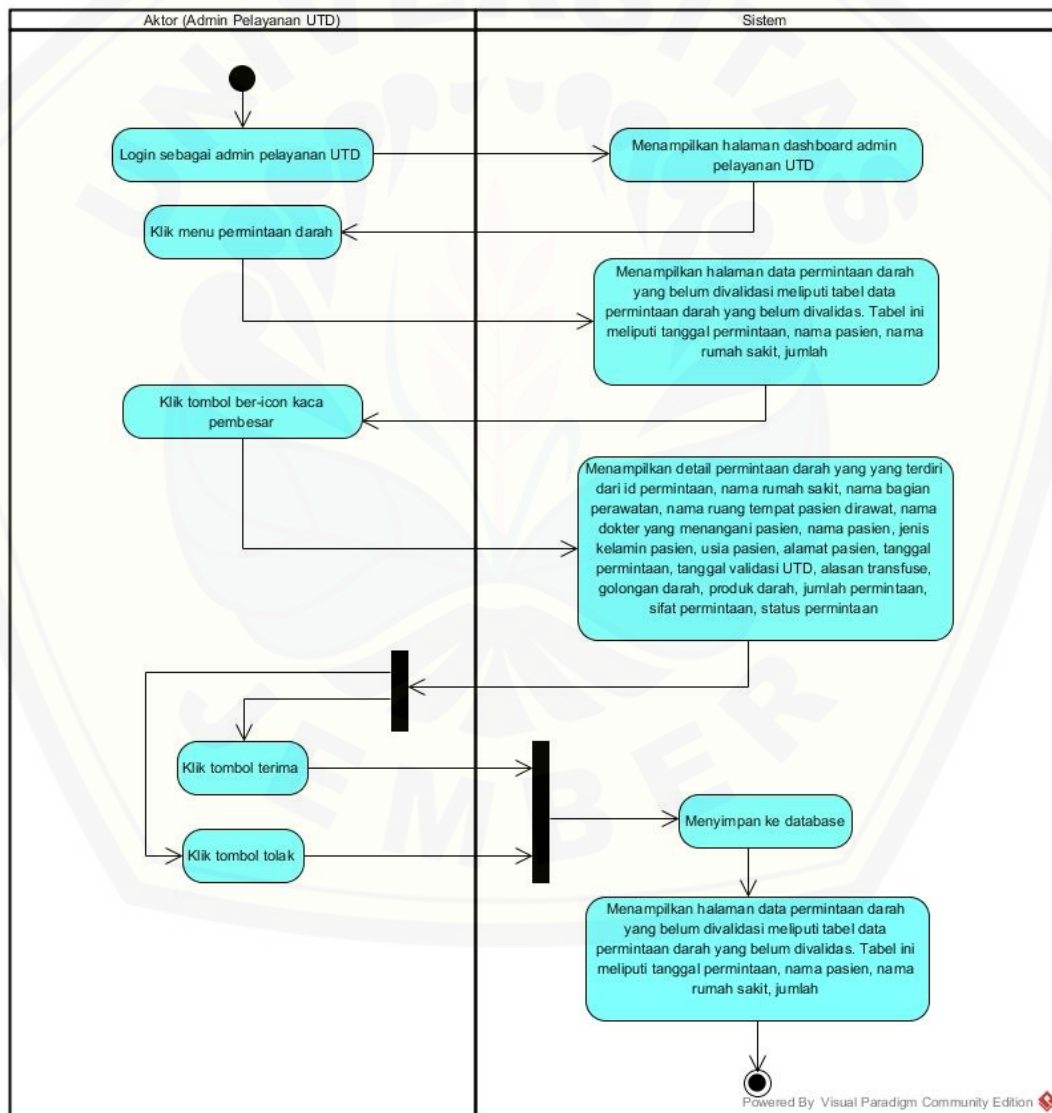


Gambar B. 14 Activity Diagram Melihat Stok Darah (Admin Penjamin Mutu UTD)



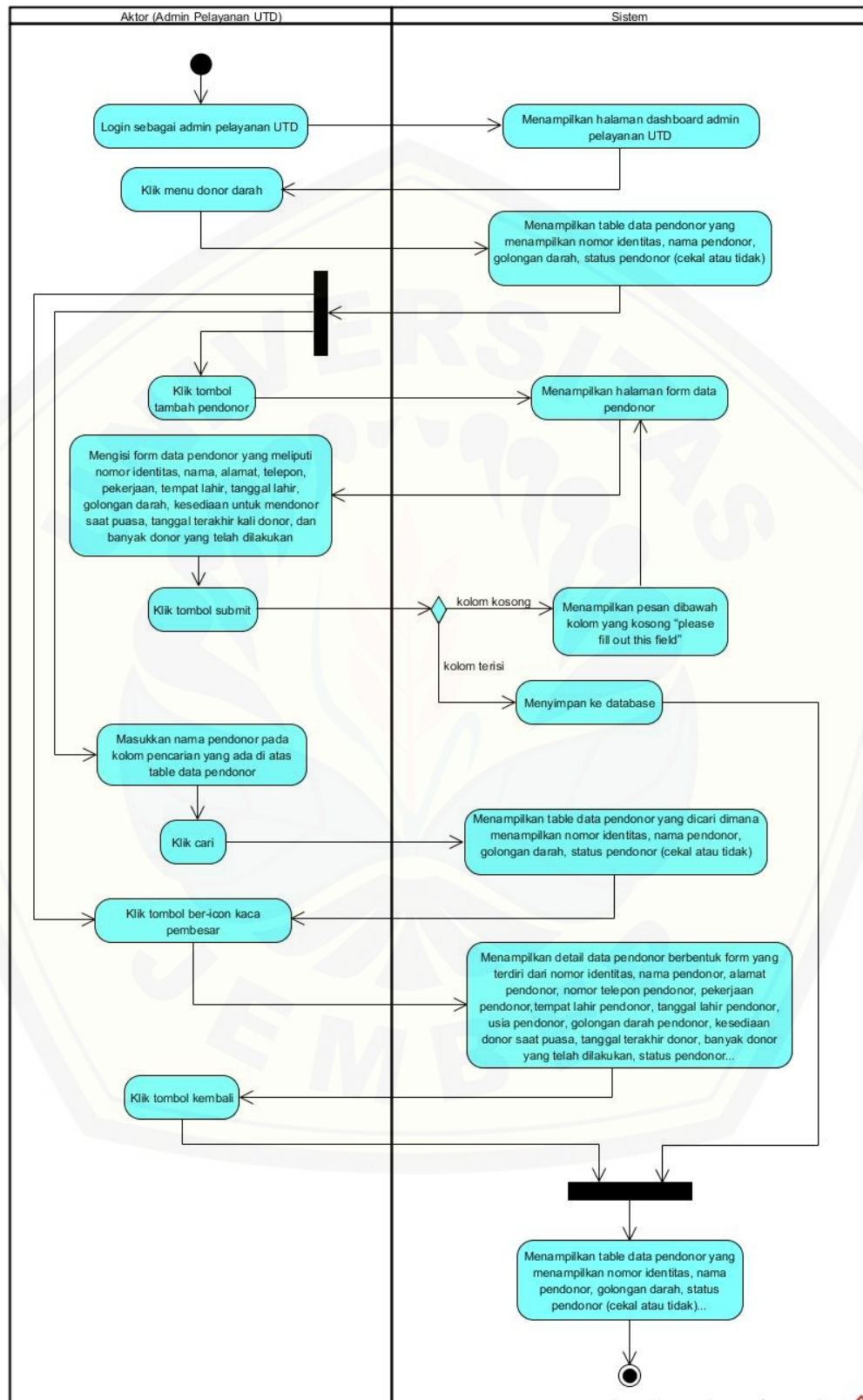
Gambar B. 15 Activity Diagram Melihat Stok Darah (User Umum)

B9 Activity Diagram Validasi Data Permintaan Darah



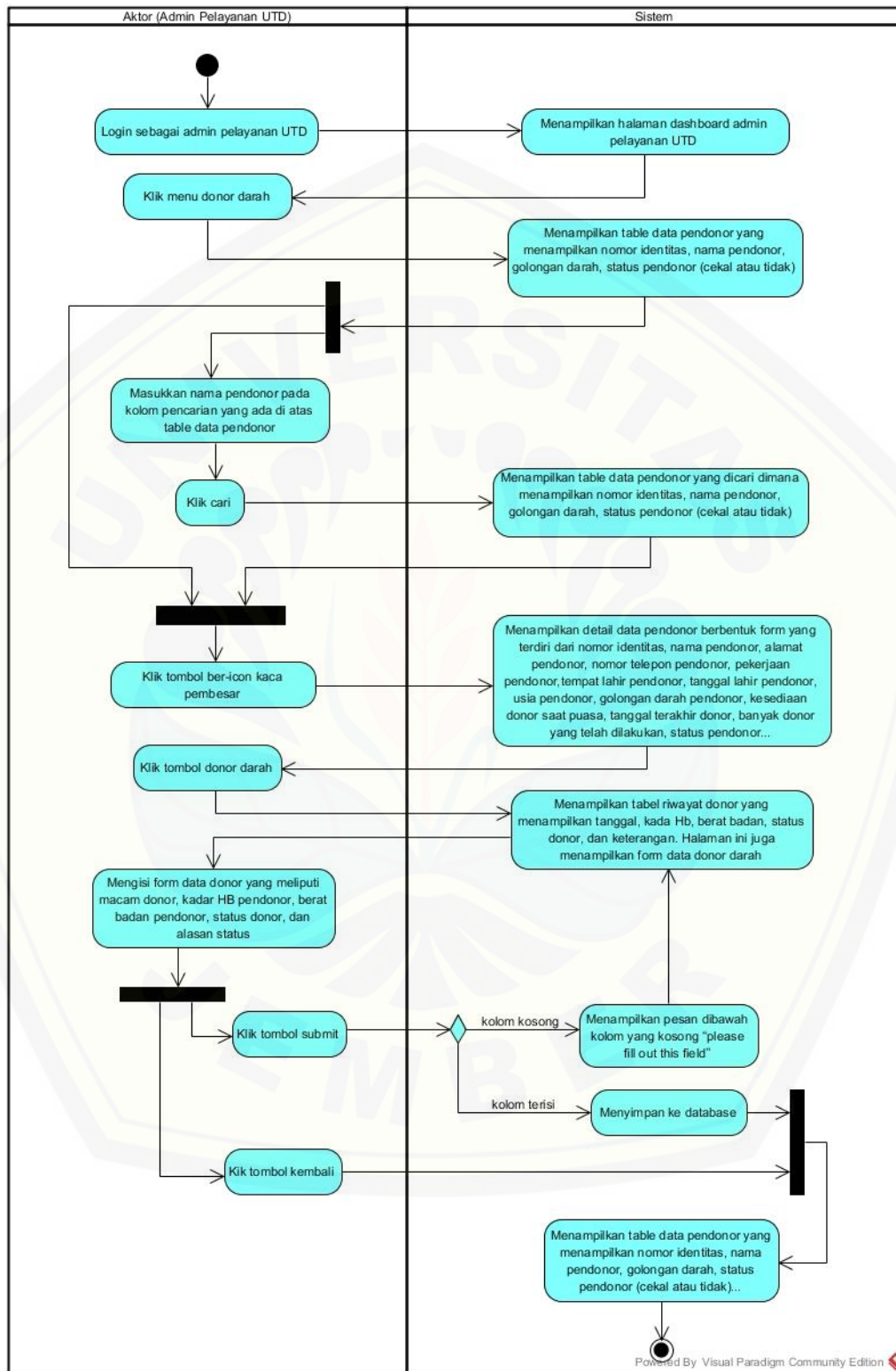
Gambar B. 16 Activity Diagram Validasi Data Permintaan Darah

B10 Activity Diagram Mengelola Data Pendoron



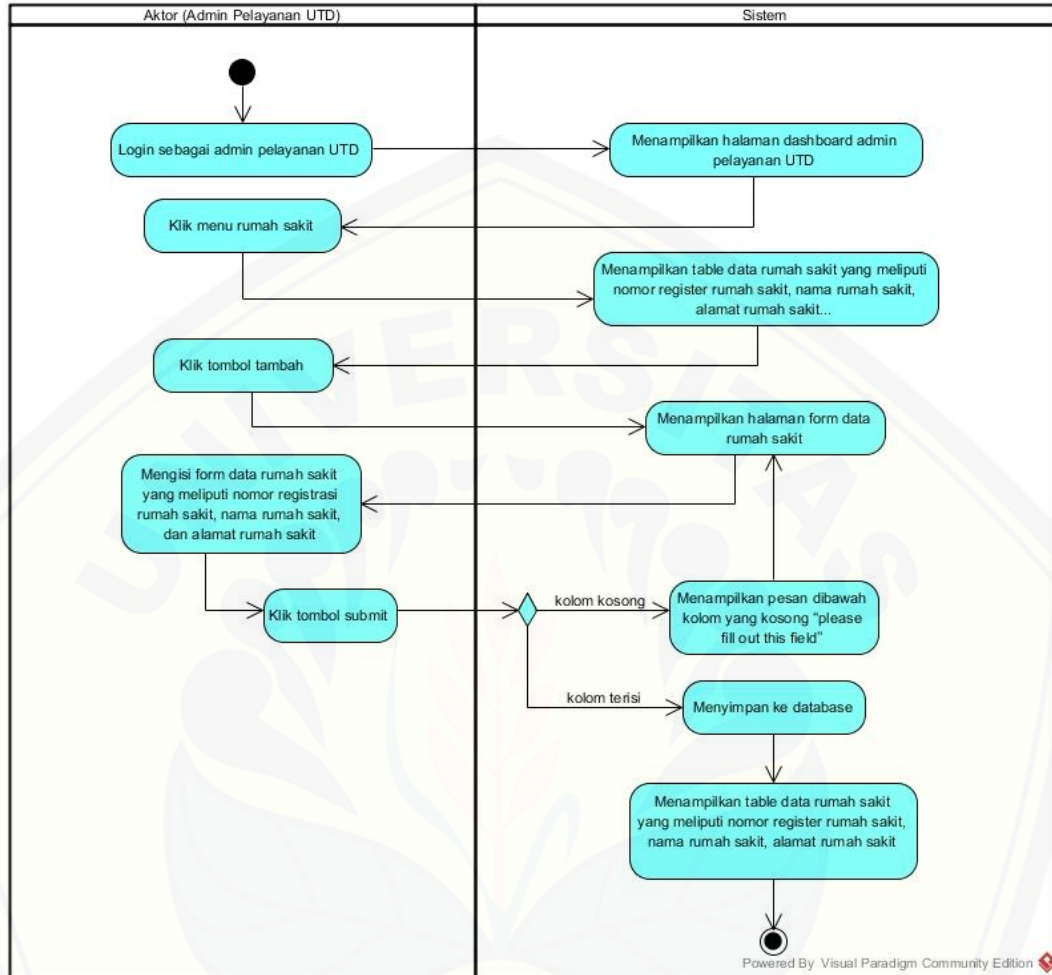
Gambar B. 17 Activity Diagram Mengelola Data Pendoron

B11 Activity Diagram Menambah Data Transaksi Donor



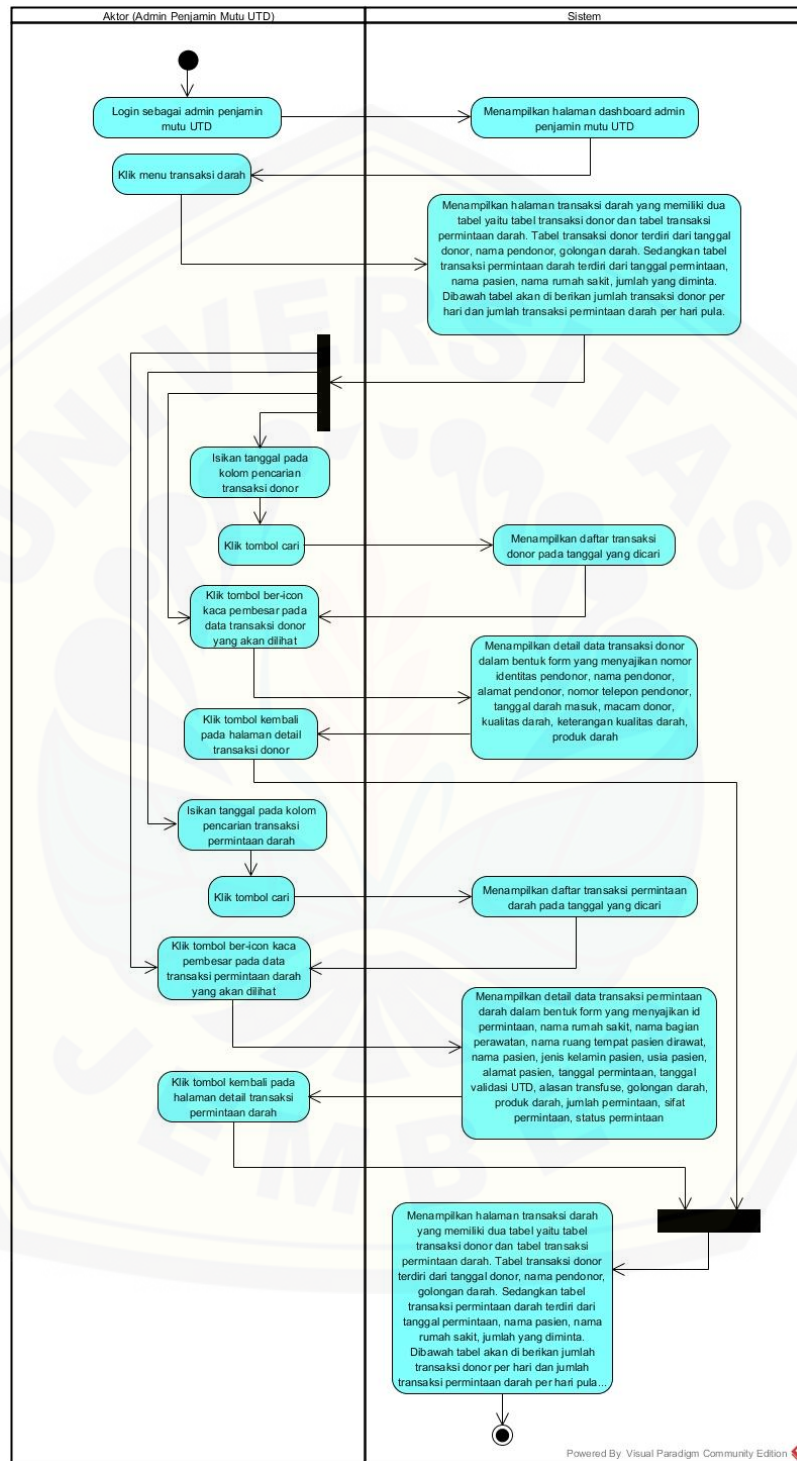
Gambar B. 18 Activity Diagram Menambah Data Transaksi Donor

B12 Activity Diagram Menambah Data Rumah Sakit



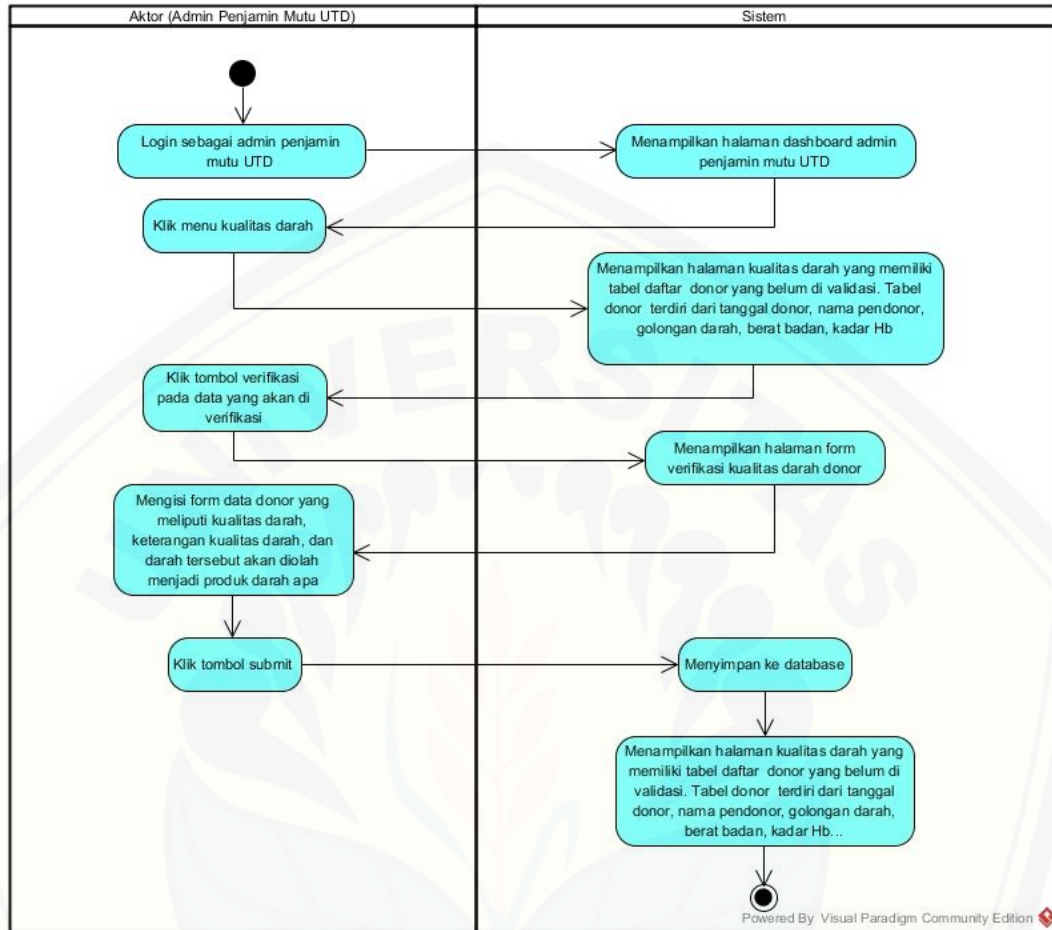
Gambar B. 19 Activity Diagram Menambah Data Rumah Sakit

B13 Activity Diagram Melihat Data Transaksi Donor dan Transaksi Permintaan Darah



Gambar B. 20 Activity Diagram Melihat Data Transaksi Donor dan Transaksi Permintaan Darah

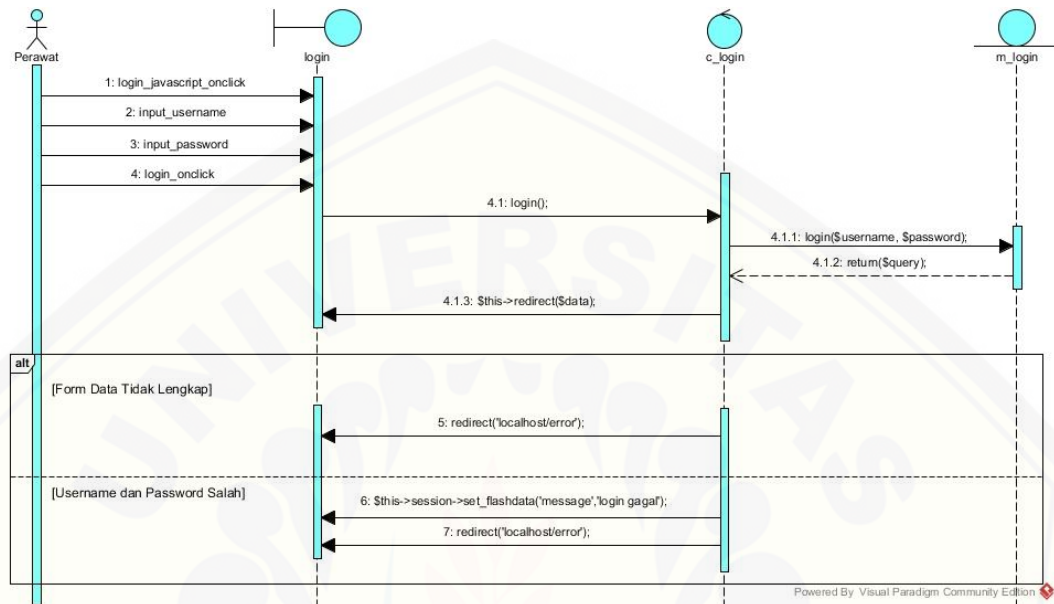
B14 Activity Diagram Verifikasi Kualitas Darah Donor



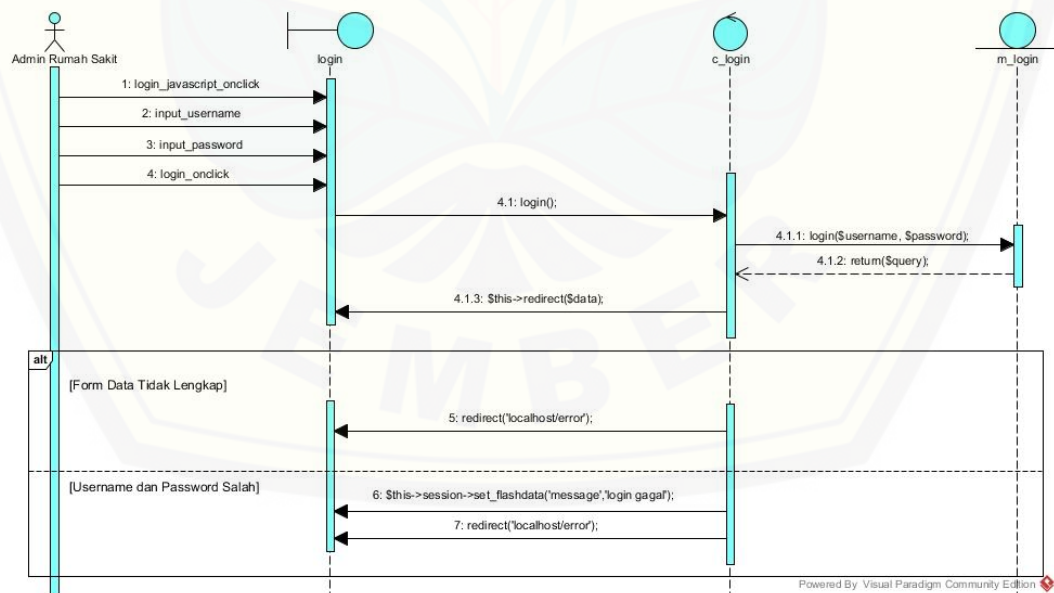
Gambar B. 21 Activity Diagram Verifikasi Kualitas Darah Donor

Lampiran C. Sequence Diagram

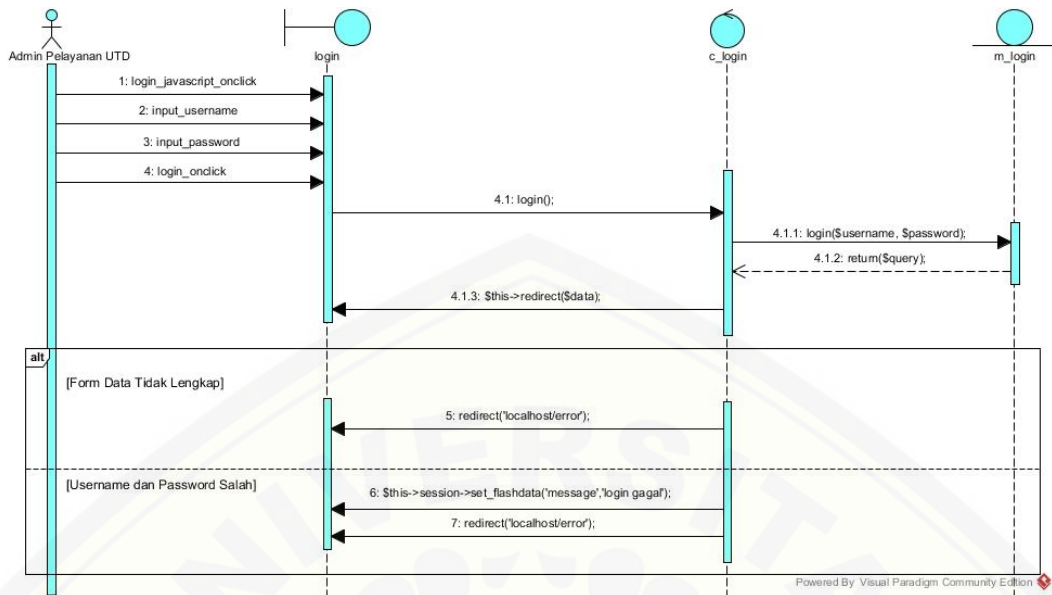
C1 Sequence Diagram Login



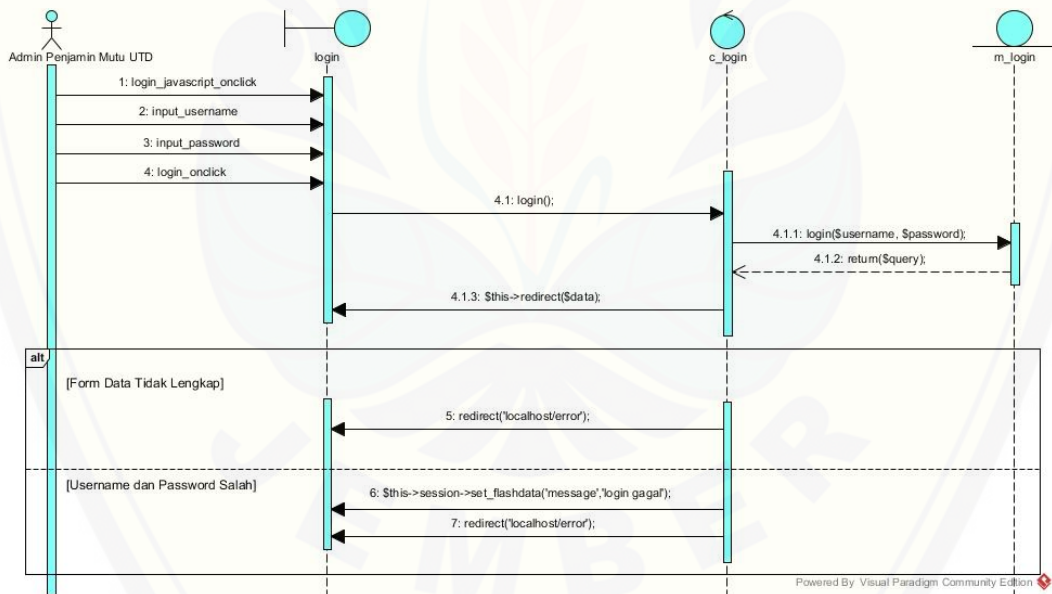
Gambar C. 1 Sequence Diagram Login (Perawat)



Gambar C. 2 Sequence Diagram Login (Admin Rumah Sakit)

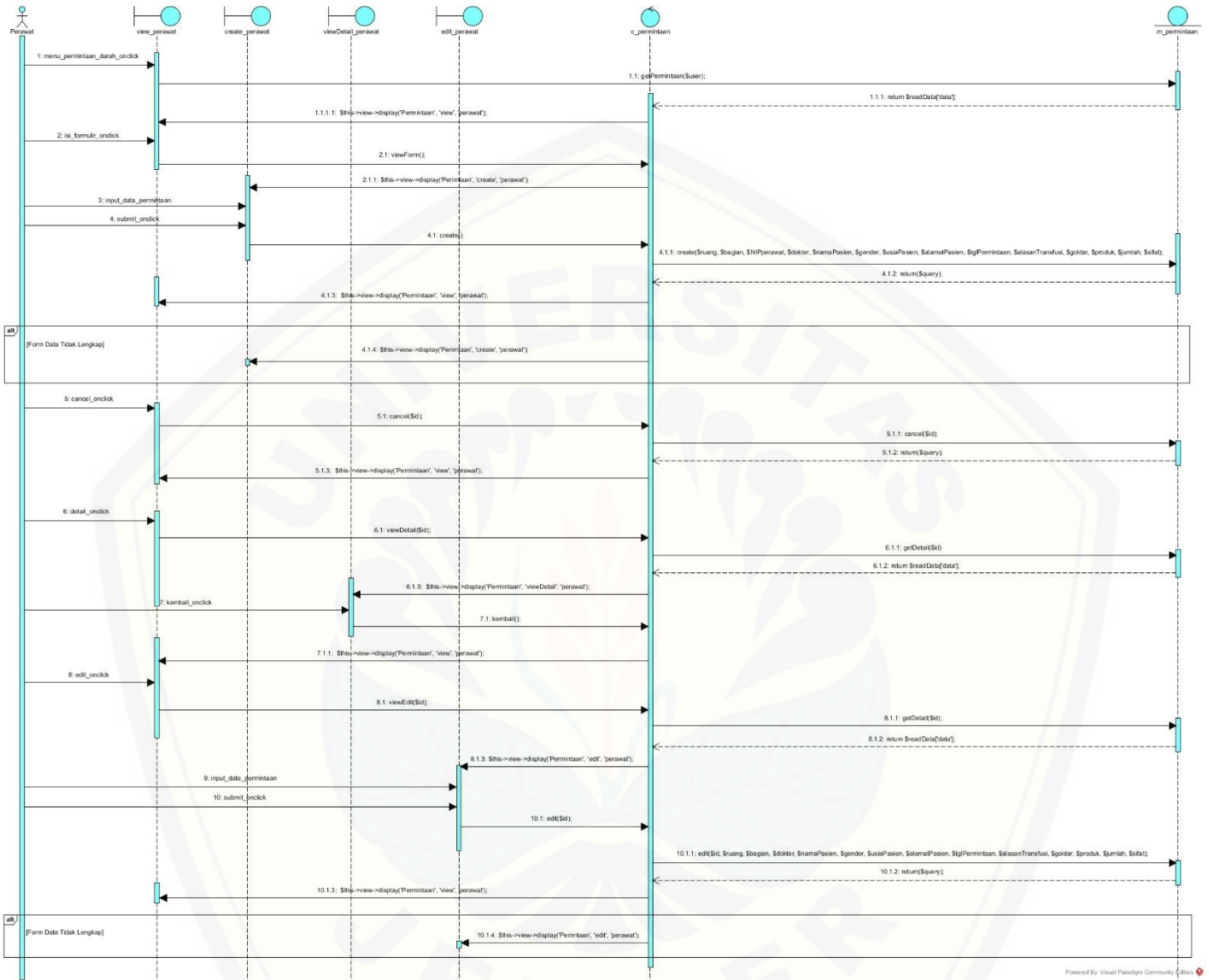


Gambar C. 3 Sequence Diagram Login (Admin Pelayanan UTD)



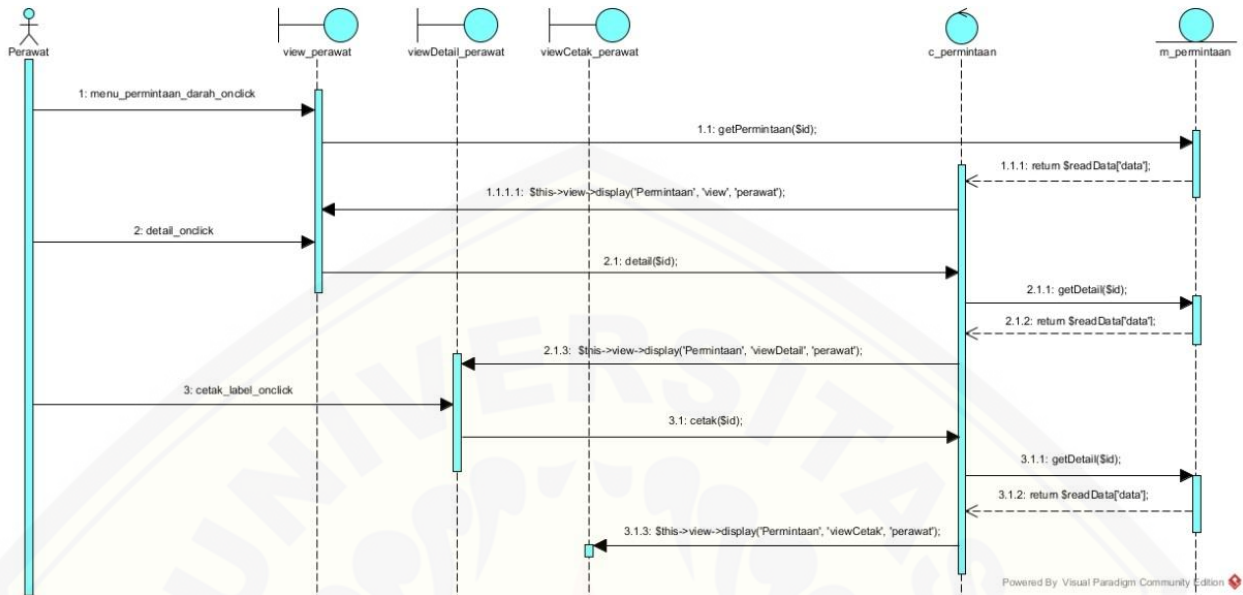
Gambar C. 4 Sequence Diagram Login (Admin Penjamin Mutu)

C2 Sequence Diagram Mengelola Data Permintaan Darah



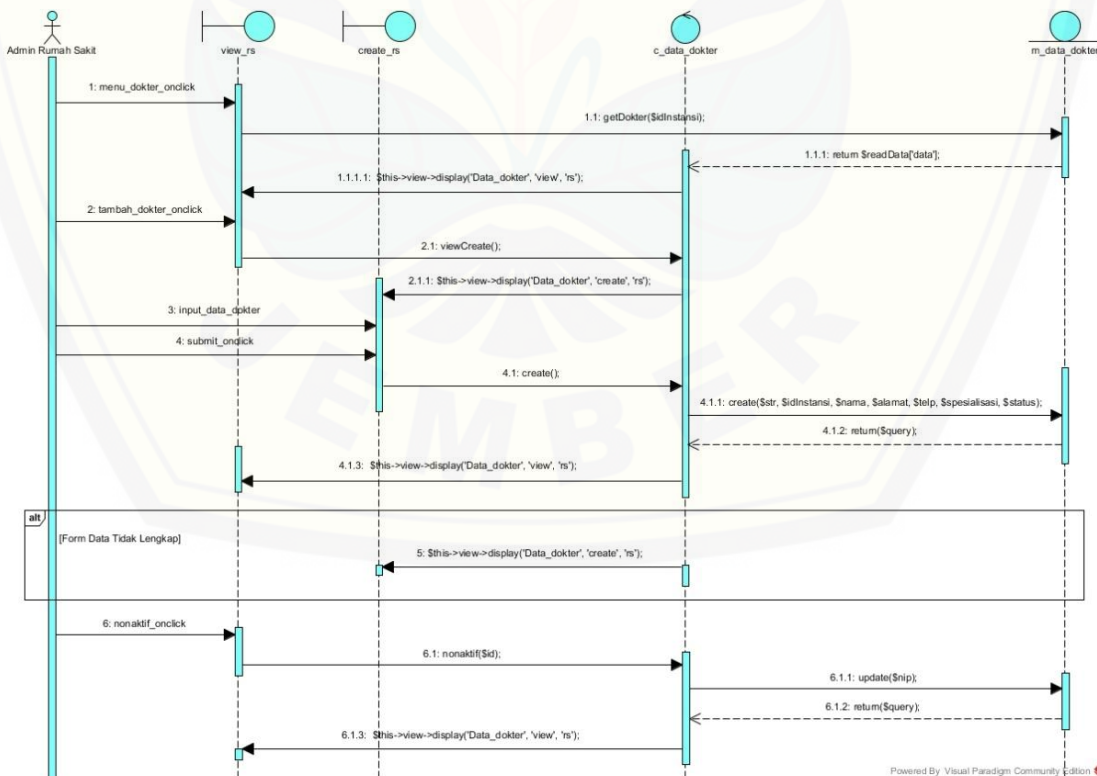
Gambar C. 5 Sequence Diagram Mengelola Data Permintaan Darah

C3 Sequence Diagram Cetak Label Sample Darah



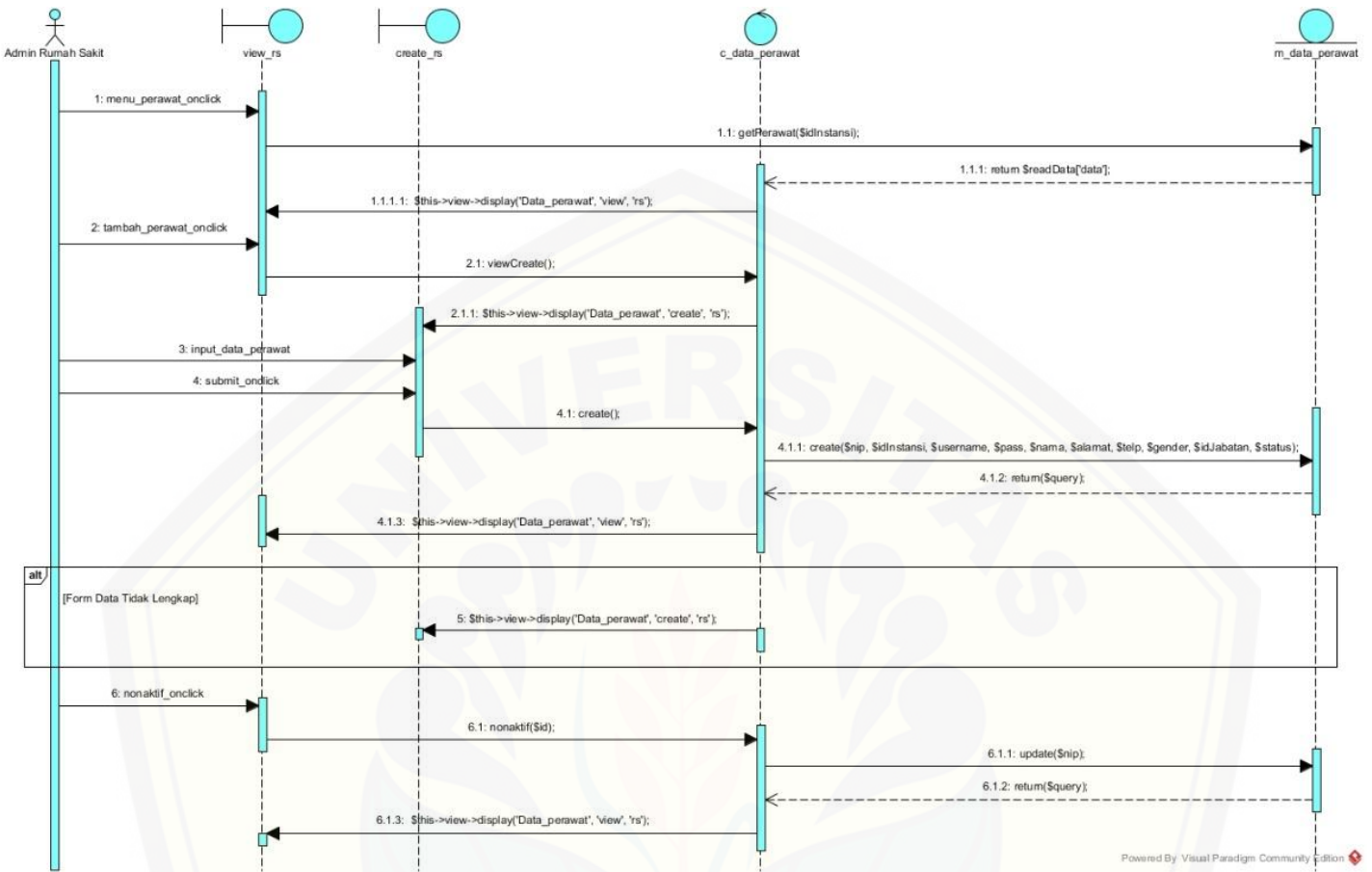
Gambar C. 6 Sequence Diagram Cetak Label Sample Darah

C4 Sequence Diagram Mengelola Data Dokter



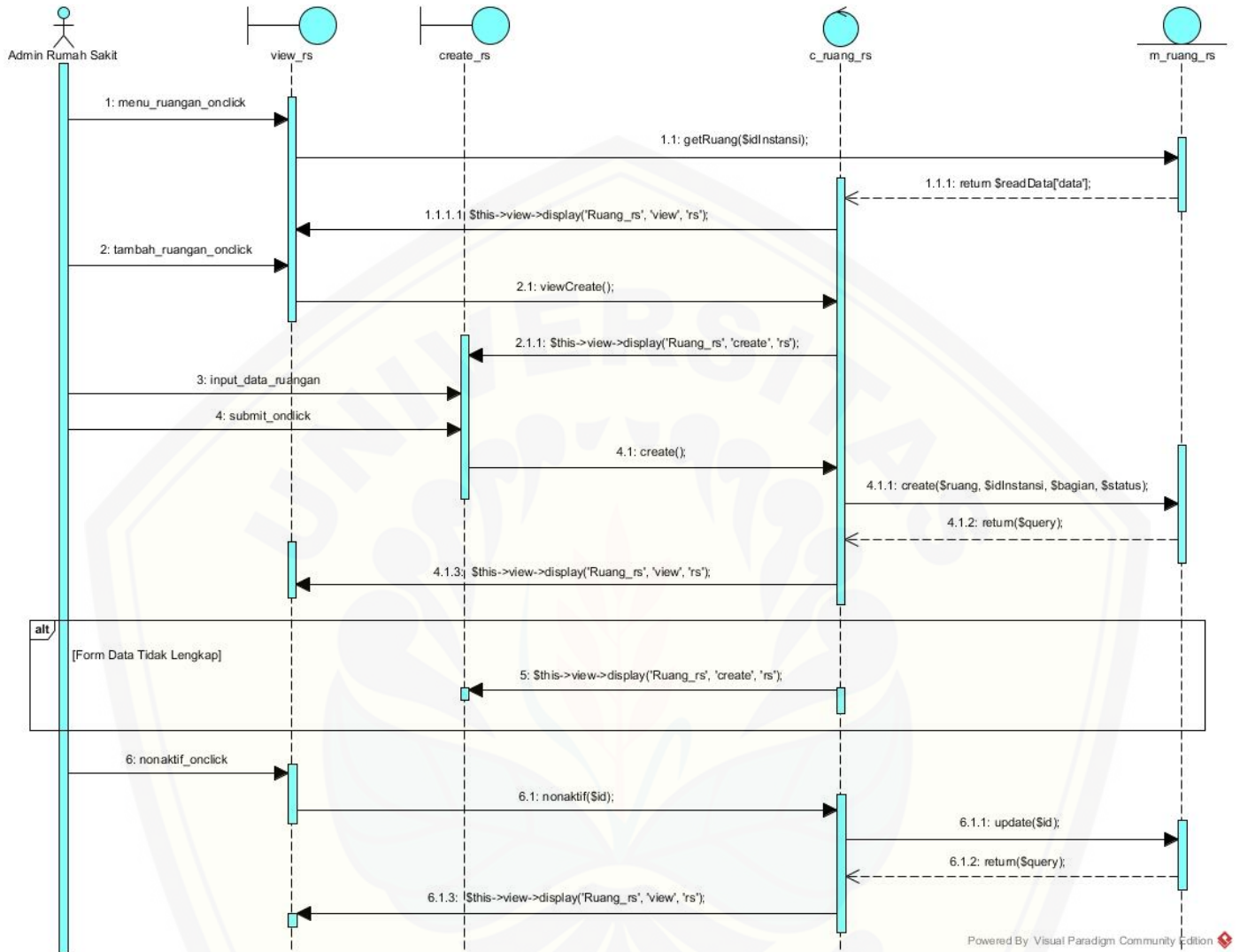
Gambar C. 7 Sequence Diagram Mengelola Data Dokter

C5 Sequence Diagram Mengelola Data Perawat



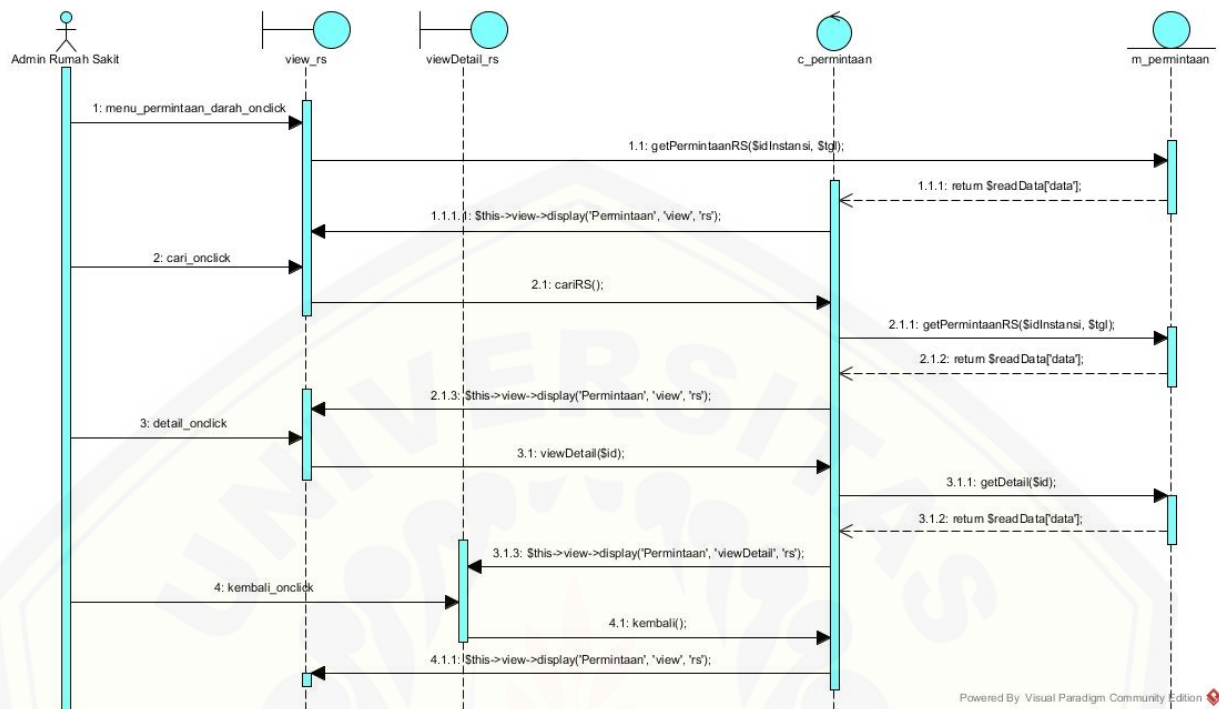
Gambar C. 8 Sequence Diagram Mengelola Data Perawat

C6 Sequence Diagram Mengelola Data Ruang Rumah Sakit



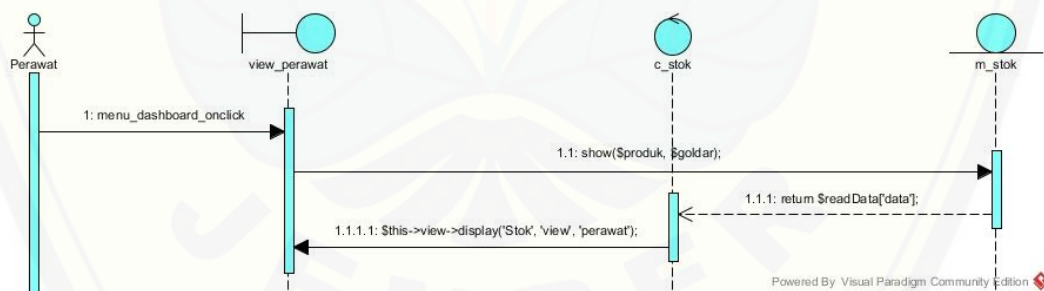
Gambar C. 9 Sequence Diagram Mengelola Data Ruang Rumah Sakit

C7 Sequence Diagram Melihat Data Permintaan Darah



Gambar C. 10 Sequence Diagram Melihat Data Permintaan Darah

C8 Sequence Diagram Melihat Stok Darah



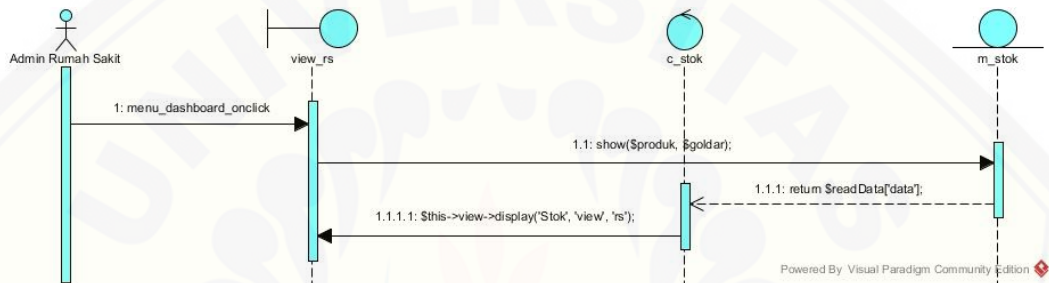
Gambar C. 11 Sequence Diagram Melihat Stok Darah (Perawat)



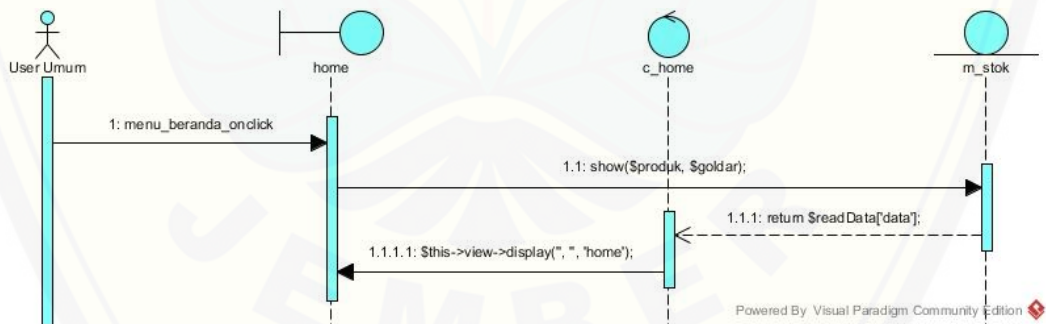
Gambar C. 12 Sequence Diagram Melihat Stok Darah (Admin Pelayanan UTD)



Gambar C. 13 Sequence Diagram Melihat Stok Darah (Admin Penjamin Mutu)

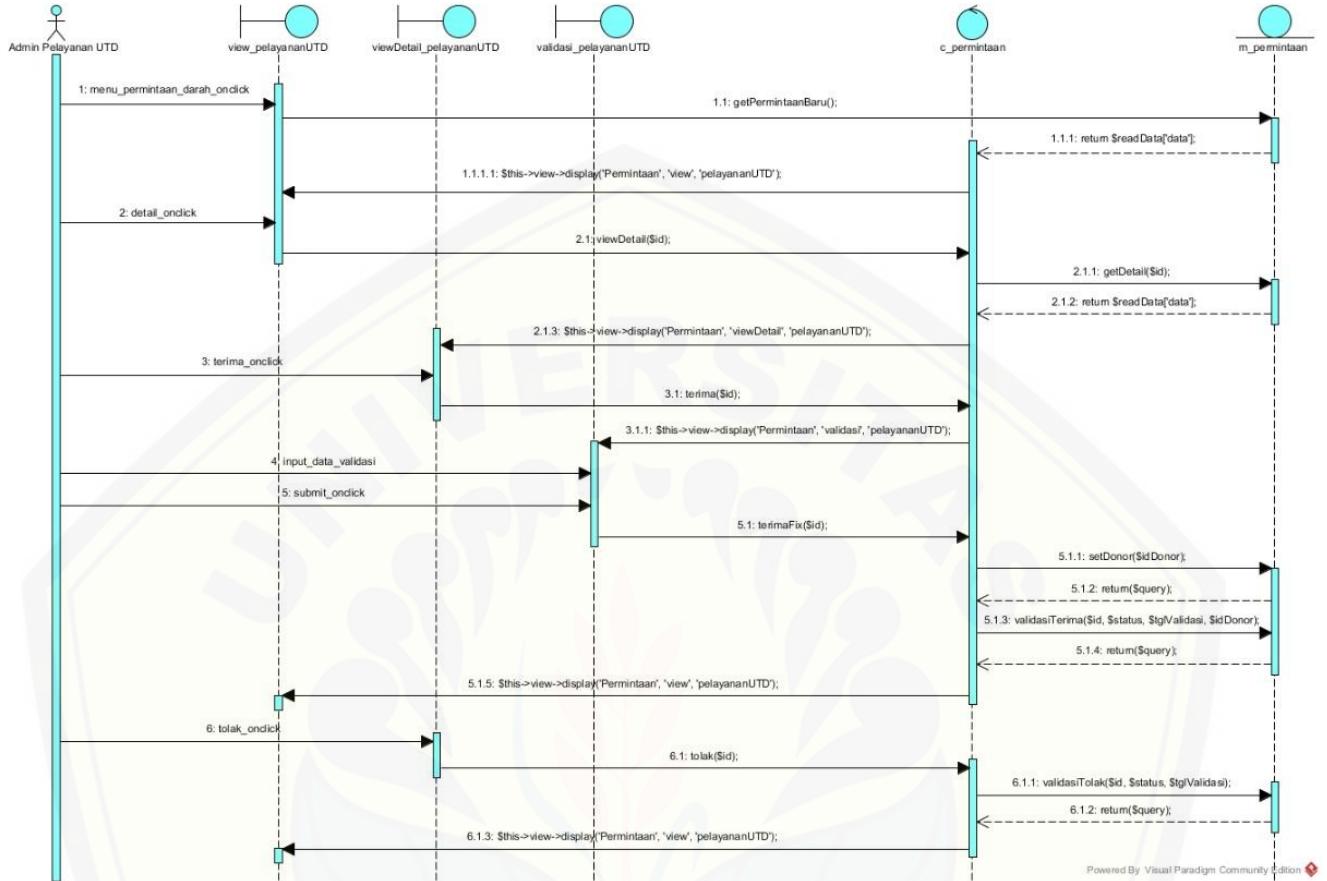


Gambar C. 14 Sequence Diagram Melihat Stok Darah (Admin Rumah Sakit)



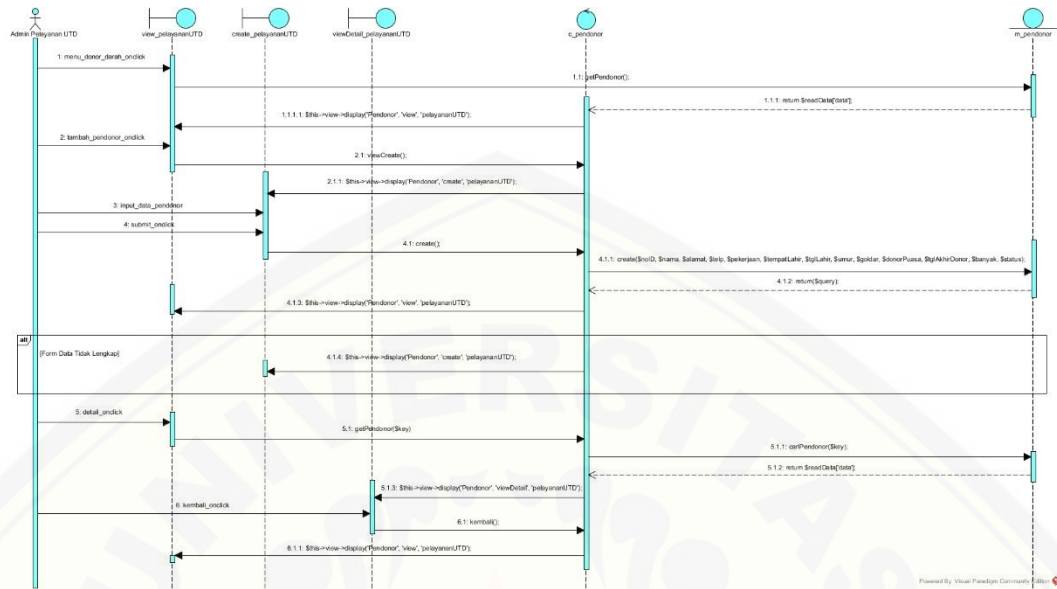
Gambar C. 15 Sequence Diagram Melihat Stok Darah (User Umum)

C9 Sequence Diagram Validasi Data Permintaan Darah



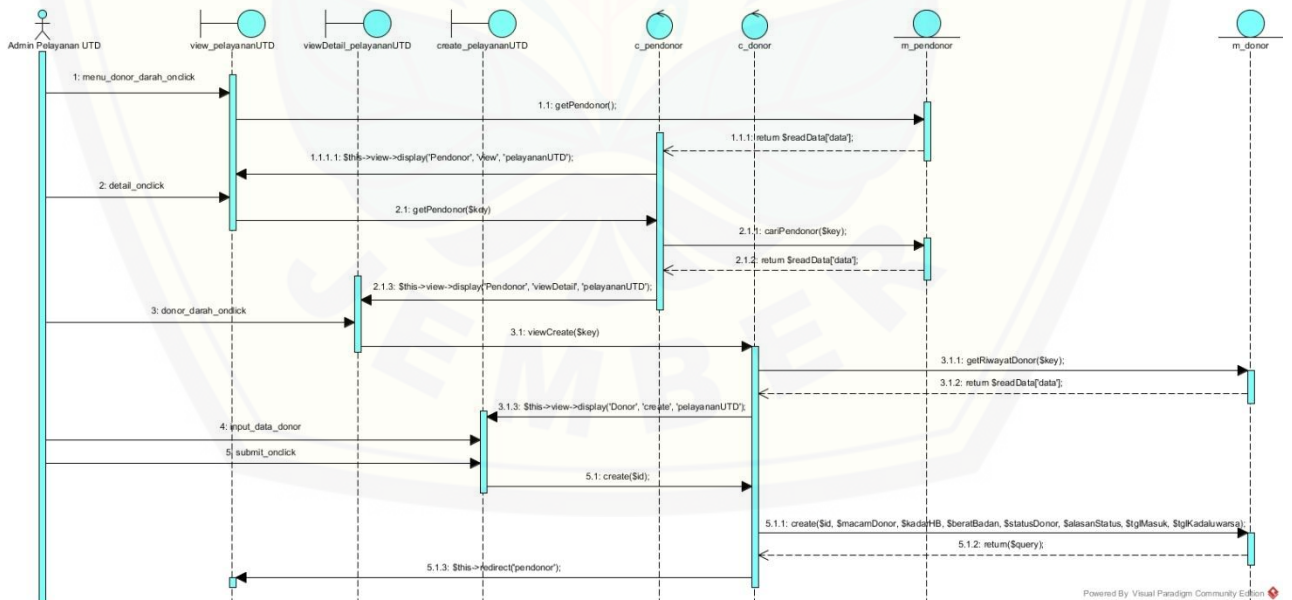
Gambar C. 16 Sequence Diagram Validasi Data Permintaan Darah

C10 Sequence Diagram Mengelola Data Pendonor



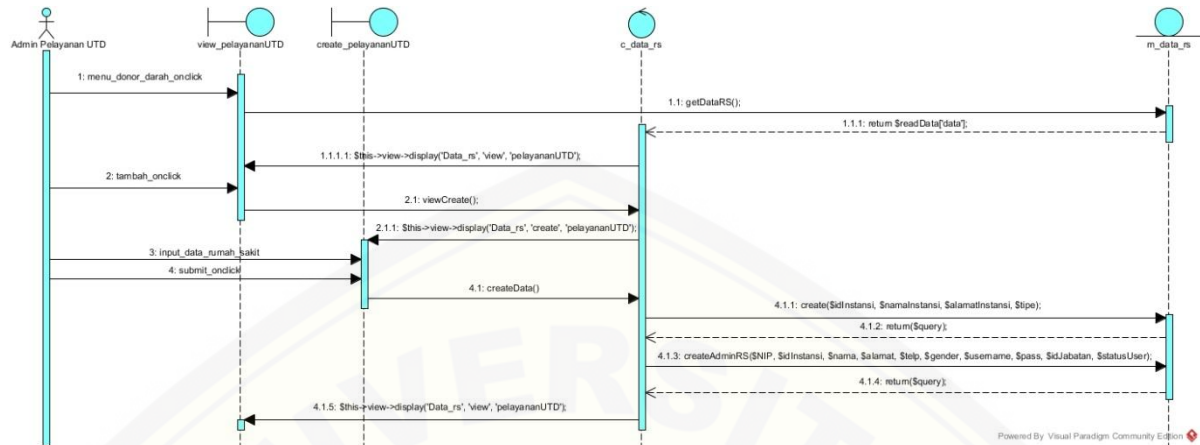
Gambar C. 17 Sequence Diagram Mengelola Data Pendonor

C11 Sequence Diagram Menambah Data Transaksi Donor



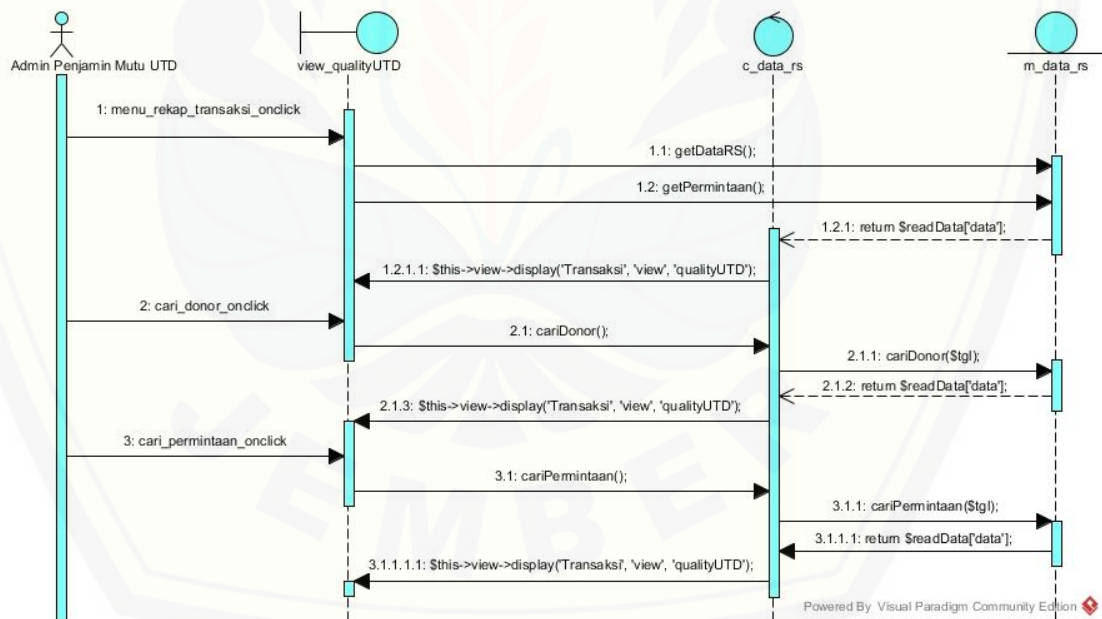
Gambar C. 18 Sequence Diagram Menambah Data Transaksi Donor

C12 Sequence Diagram Menambah Data Rumah Sakit



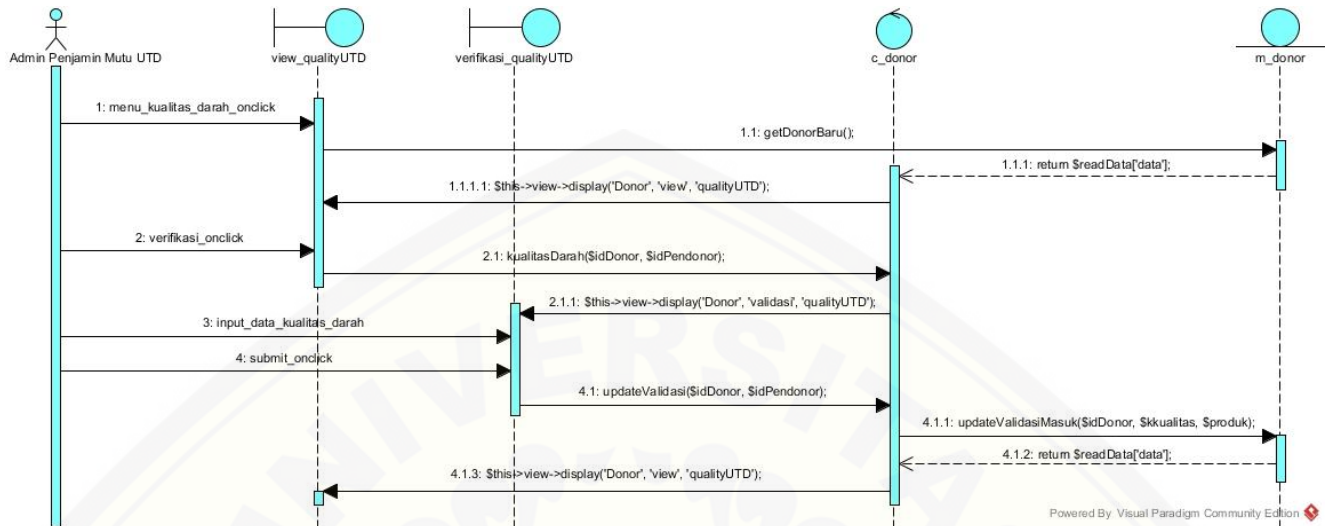
Gambar C. 19 Sequence Diagram Menambah Data Rumah Sakit

C13 Sequence Diagram Melihat Data Transaksi Donor dan Transaksi Permintaan Darah



Gambar C. 20 Sequence Diagram Melihat Data Transaksi Donor dan Transaksi Permintaan Darah

C14 Sequence Diagram Verifikasi Kualitas Darah Donor



Gambar C. 21 Sequence Diagram Verifikasi Kualitas Darah Donor

Lampiran D. Kode Program

D1 Kode program Login

Tabel C. 1 Kode program Login (Controller)

```
function login() {
    if (isset($_POST['submit'])) {
        $username = $_POST['username'];
        $password = $_POST['password'];
        $hasil = $this->model->login($username, $password);
        $data = $hasil['data'];
        if ($hasil['cek'] == true) {
            session::set('username', $username);
            session::set('nip', $data['NIP']);
            session::set('nama', $data['nama']);
            session::set('jabatan', $data['jabatan']);
            session::set('loggedIn', true);
            $this->redirect('stok');
        } else {
            $this->index();
            echo "<script type='text/javascript'>alert('login gagal!')</script>";
            $this->index();
        }
    }
}
```

Tabel C. 2 Kode program Login (Model)

```
function login($username, $password){
    $stmt = $this->db->prepare("SELECT * FROM user u JOIN jabatan j ON u.idJabatan =
        j.idJabatan
        WHERE username = '$username' AND pass = '$password' LIMIT 1");

    $stmt->execute();

    $result = $stmt->fetch(PDO::FETCH_ASSOC);

    if($stmt->rowCount() > 0){
        return array('cek' => true, 'data' => $result);
    } else {
        echo "error";
    }
}
```

D2 Kode program Mengelola Data Permintaan Darah

Tabel C. 3 Kode program Mengelola Data Permintaan Darah (Controller)

```

function create() {
    if (isset($_POST['submit'])) {
        $ruang = $_POST['ruang'];

        $sid = $this->Model->getBagian($ruang);
        $data = $sid['data'];

        if ($sid['cek'] == true) {
            $bagian = $data['bagian'];
        } else {
            echo "error";
        }

        $NIPperawat = session::get('nip');
        $dokter = $_POST['dokter'];
        $namaPasien = $_POST['namaPasien'];
        $gender = $_POST['gender'];
        $usiaPasien = $_POST['usiaPasien'];
        $alamatPasien = $_POST['alamatPasien'];
        $tglPermintaan = date("Y-m-d");
        $alasanTransfusi = $_POST['alasanTransfusi'];
        $goldar = $_POST['goldar'];
        $produk = $_POST['produk'];
        $jumlah = $_POST['jumlah'];
        $sifat = $_POST['sifat'];
        $this->Model->create($ruang, $bagian, $NIPperawat, $dokter, $namaPasien,
            $gender, $usiaPasien, $alamatPasien, $tglPermintaan, $alasanTransfusi,
            $goldar, $produk, $jumlah, $sifat);
        $this->index();
    }
}

function edit($sid) {
    $ruang = $_POST['ruang'];

    $a = $this->Model->getBagian($ruang);
    $data = $a['data'];
    if ($a['cek'] == true) {
        $bagian = $data['bagian'];
    } else {
        echo "error";
    }

    $dokter = $_POST['dokter'];
    $namaPasien = $_POST['namaPasien'];
    $gender = $_POST['gender'];
    $usiaPasien = $_POST['usiaPasien'];
    $alamatPasien = $_POST['alamatPasien'];
    $tglPermintaan = date("Y-m-d");
    $alasanTransfusi = $_POST['alasanTransfusi'];
}

```

```

$goldar = $_POST['goldar'];
$produk = $_POST['produk'];
$jumlah = $_POST['jumlah'];
$sifat = $_POST['sifat'];
$this->Model->edit($id, $ruang, $bagian, $dokter, $namaPasien, $gender, $usiaPasien,
    $alamatPasien, $tglPermintaan, $alasanTransfusi, $goldar, $produk, $jumlah,
    $sifat);
$this->index();
}
function cancel($id){
    $this->Model->cancel($id);
    echo "<script type='text/javascript'>alert('Permintaan Darah Berhasil
        Dibatalkan')</script>";
    $this->index();
}

```

Tabel C. 4 Kode program Mengelola Data Permintaan Darah (Model)

```

function create($ruang, $bagian, $NIPperawat, $dokter, $namaPasien, $gender, $usiaPasien,
    $alamatPasien, $tglPermintaan, $alasanTransfusi, $goldar, $produk, $jumlah, $sifat){
    $stmt = $this->db->prepare("INSERT INTO `permintaan`(`bagian`, `ruang`,
    `NIPperawat`, `namaDokter`, `namaPasien`, `genderPasien`, `usiaPasien`, `alamatPasien`,
    `tglPermintaan`, `alasanTransfusi`, `goldar`, `produkDarah`, `jumlah`, `sifatPermintaan`)
    VALUES (:bagian, :ruang, :NIPperawat, :namaDokter, :namaPasien, :genderPasien,
    :usiaPasien, :alamatPasien, :tglPermintaan, :alasanTransfusi, :goldar, :produk, :jumlah,
    :sifatPermintaan)");
    $stmt->bindParam(':bagian', $bagian, PDO::PARAM_STR);
    $stmt->bindParam(':ruang', $ruang, PDO::PARAM_STR);
    $stmt->bindParam(':NIPperawat', $NIPperawat, PDO::PARAM_STR);
    $stmt->bindParam(':namaDokter', $dokter, PDO::PARAM_STR);
    $stmt->bindParam(':namaPasien', $namaPasien, PDO::PARAM_STR);
    $stmt->bindParam(':genderPasien', $gender, PDO::PARAM_STR);
    $stmt->bindParam(':usiaPasien', $usiaPasien, PDO::PARAM_STR);
    $stmt->bindParam(':alamatPasien', $alamatPasien, PDO::PARAM_STR);
    $stmt->bindParam(':tglPermintaan', $tglPermintaan, PDO::PARAM_STR);
    $stmt->bindParam(':alasanTransfusi', $alasanTransfusi, PDO::PARAM_STR);
    $stmt->bindParam(':goldar', $goldar, PDO::PARAM_STR);
    $stmt->bindParam(':produk', $produk, PDO::PARAM_STR);
    $stmt->bindParam(':jumlah', $jumlah, PDO::PARAM_STR);
    $stmt->bindParam(':sifatPermintaan', $sifat, PDO::PARAM_STR);
    $stmt->execute();
}

function edit($id, $ruang, $bagian, $dokter, $namaPasien, $gender, $usiaPasien,
    $alamatPasien, $tglPermintaan, $alasanTransfusi, $goldar, $produk, $jumlah, $sifat){
    $this->db->update('UPDATE `permintaan` SET `bagian`= :bagian,`ruang`=
    :ruang,`namaDokter`= :namaDokter, `namaPasien`= :namaPasien,`genderPasien`=
    :genderPasien,`usiaPasien`= :usiaPasien, `alamatPasien`= :alamatPasien,`tglPermintaan`=
    :tglPermintaan,`alasanTransfusi`= :alasanTransfusi, `goldar`= :goldar,`produkDarah`=
    :produkDarah,`jumlah`= :jumlah, `sifatPermintaan`= :sifatPermintaan WHERE idPermintaan
    = :idPermintaan',
    array(
        'idPermintaan' => $id,
        'bagian' => $bagian,
        'ruang' => $ruang,

```



```

        'namaDokter' => $dokter,
        'namaPasien' => $namaPasien,
        'genderPasien' => $gender,
        'usiaPasien' => $usiaPasien,
        'alamatPasien' => $alamatPasien,
        'tglPermintaan' => $tglPermintaan,
        'alasanTransfusi' => $alasanTransfusi,
        'goldar' => $goldar,
        'produkDarah' => $produk,
        'jumlah' => $jumlah,
        'sifatPermintaan' => $sifat
    ));
}

function cancel($id){
    $this->db->update('UPDATE `permintaan` SET `statusPermintaan` = "cancel" WHERE
        idPermintaan = :idPermintaan', array('idPermintaan' => $id));
}

```

D3 Kode program Cetak Label Sample Darah

Tabel C. 5 Kode program Cetak Label Sample Darah (Controller)

```

function cetak($id){
    $this->view->display = $this->Model->getDetail($id);
    $this->view->display('Permintaan', 'viewCetak', 'perawat');
}

```

Tabel C. 6 Kode program Cetak Label Sample Darah (Model)

```

function getDetail($id){
    $readData = $this->db->selectWhere('select * from permintaan p join user u on
        p.NIPperawat = u.NIP join Instansi i on u.idInstansi = i.idInstansi where
        idPermintaan = :idPermintaan LIMIT 1', array('idPermintaan' => $id));
    return $readData['data'];
}

```

D4 Kode program Mengelola Data Dokter

Tabel C. 7 Kode program Mengelola Data Dokter (Controller)

```

function create() {
    if (isset($_POST['submit'])) {
        $str = $_POST['str'];

        $user = session::get('username');
        $id = $this->Model->getInstansi($user);
        $data = $id['data'];
        if ($id['cek'] == true) {
            $idInstansi = $data['idInstansi'];

```

```

    } else {
        echo "eror";
    }

    $nama = $_POST['nama'];
    $alamat = $_POST['alamat'];
    $telp = $_POST['telp'];
    $spesialisasi = $_POST['spesialisasi'];
    $status = 'aktif';

    $this->Model->create($str, $idInstansi, $nama, $alamat, $telp, $spesialisasi, $status);
    $this->index();
}
}

function nonaktif($noSTR){
    $this->Model->update($noSTR);
    echo "<script type='text/javascript'>alert('Status Dokter Non-aktif')</script>";
    $this->index();
}

```

Tabel C. 8 Kode program Mengelola Data Dokter (Model)

```

function create($str, $idInstansi, $nama, $alamat, $telp, $spesialisasi, $status){
    $stmt = $this->db->prepare("INSERT INTO `dokter`(`noSTR`, `idInstansi`, `nama`,
        `alamat`, `telp`, `spesialisasi`, `status`)
        VALUES (:noSTR, :idInstansi, :nama, :alamat, :telp, :spesialisasi, :status)");
    $stmt->bindParam(':noSTR', $str, PDO::PARAM_STR);
    $stmt->bindParam(':idInstansi', $idInstansi, PDO::PARAM_STR);
    $stmt->bindParam(':nama', $nama, PDO::PARAM_STR);
    $stmt->bindParam(':alamat', $alamat, PDO::PARAM_STR);
    $stmt->bindParam(':telp', $telp, PDO::PARAM_STR);
    $stmt->bindParam(':spesialisasi', $spesialisasi, PDO::PARAM_STR);
    $stmt->bindParam(':status', $status, PDO::PARAM_STR);
    $stmt->execute();
}

function update($noSTR){
    $this->db->update('UPDATE `dokter` SET `status` = "nonaktif"
        WHERE noSTR = :noSTR',
        array(
            'noSTR' => $noSTR
        ));
}

```

D5 Kode program Mengelola Data Perawat

Tabel C. 9 Kode program Mengelola Data Perawat (Controller)

```

function create() {

    if (isset($_POST['submit'])) {
        $nip = $_POST['nip'];

        $user = session::get('username');
        $id = $this->Model->getInstansi($user);
        $data = $id['data'];
        if ($id['cek'] == true) {
            $idInstansi = $data['idInstansi'];
        } else {
            echo "error";
        }

        $username = $_POST['username'];
        $pass = $_POST['pass'];
        $nama = $_POST['nama'];
        $alamat = $_POST['alamat'];
        $telp = $_POST['telp'];
        $gender = $_POST['gender'];
        $idJabatan = '2';
        $status = 'aktif';

        $this->Model->create($nip, $idInstansi, $username, $pass, $nama, $alamat, $telp,
        $gender, $idJabatan, $status);
        $this->index();
    }

    function nonaktif($nip){
        $this->Model->update($nip);
        echo "<script type='text/javascript'>alert('Status Perawat Non-aktif')</script>";
        $this->index();
    }
}

```

Tabel C. 10 Kode program Mengelola Data Perawat (Model)

```

function create($nip, $idInstansi, $username, $pass, $nama, $alamat, $telp, $gender,
$idJabatan, $status){
    $stmt = $this->db->prepare("INSERT INTO `user`(`NIP`, `idInstansi`, `username`,
`pass`, `nama`, `alamat`, `telp`, `gender`, `idJabatan`, `statusUser`) VALUES (:NIP,
:idInstansi, :username, :pass, :nama, :alamat, :telp, :gender, :idJabatan, :statusUser)");
    $stmt->bindParam(':NIP', $nip, PDO::PARAM_STR);
    $stmt->bindParam(':idInstansi', $idInstansi, PDO::PARAM_STR);
    $stmt->bindParam(':username', $username, PDO::PARAM_STR);
    $stmt->bindParam(':pass', $pass, PDO::PARAM_STR);
    $stmt->bindParam(':nama', $nama, PDO::PARAM_STR);
    $stmt->bindParam(':alamat', $alamat, PDO::PARAM_STR);
    $stmt->bindParam(':telp', $telp, PDO::PARAM_STR);
}

```

```

$stmt->bindParam(':gender', $gender, PDO::PARAM_STR);
$stmt->bindParam(':idJabatan', $idJabatan, PDO::PARAM_STR);
$stmt->bindParam(':statusUser', $status, PDO::PARAM_STR);
$stmt->execute();

}

function update($nip){
    $this->db->update('UPDATE `user` SET `statusUser`= "nonaktif" WHERE NIP = :NIP', array('NIP' => $nip));
}

```

D6 Kode program Mengelola Data Ruang Rumah Sakit

Tabel C. 11 Kode program Mengelola Data Ruang Rumah Sakit (Controller)

```

function create() {

    if (isset($_POST['submit'])) {
        $ruang = $_POST['ruang'];
        $bagian = $_POST['bagian'];

        $user = session::get('username');
        $id = $this->Model->getInstansi($user);
        $data = $id['data'];
        if ($id['cek'] == true) {
            $idInstansi = $data['idInstansi'];
        } else {
            echo "error";
        }

        $status = 'dipakai';

        $this->Model->create($ruang, $idInstansi, $bagian, $status);
        $this->index();
    }
}

function nonaktif($id){
    $this->Model->update($id);
    echo "<script type='text/javascript'>alert('Ruangan Dihapus')</script>";
    $this->index();
}

```

Tabel C. 12 Kode program Mengelola Data Ruang Rumah Sakit (Model)

```

function create($ruang, $idInstansi, $bagian, $status){
    $stmt = $this->db->prepare("INSERT INTO `ruang_rs`(`bagian`, `namaRuang`, `idInstansi`, `statusRuang`) VALUES (:bagian, :namaRuang, :idInstansi, :statusRuang)");
    $stmt->bindParam(':bagian', $bagian, PDO::PARAM_STR);
    $stmt->bindParam(':namaRuang', $ruang, PDO::PARAM_STR);
    $stmt->bindParam(':idInstansi', $idInstansi, PDO::PARAM_STR);
}

```

```

$stmt->bindParam(':statusRuang', $status, PDO::PARAM_STR);
$stmt->execute();

}

function update($id){
    $this->db->update('UPDATE `ruang_rs` SET `statusRuang`= "tidak dipakai" WHERE
idRuang = :idRuang', array('idRuang' => $id));
}

```

D7 Kode program Melihat Data Permintaan Darah

Tabel C. 13 Kode program Melihat Data Permintaan Darah (Controller)

```

function viewDetail($id){
    $this->view->display = $this->Model->getDetail($id);
    $jabatan = session::get('jabatan');
    if ($jabatan == 'Pelayanan UTD') {
        $this->view->display('Permintaan', 'viewDetail', 'pelayananUTD');
    } else if ($jabatan == 'Quality UTD') {
        $this->view->display('Permintaan', 'viewDetail', 'qualityUTD');
    } else if ($jabatan == 'Perawat') {
        $this->view->display('Permintaan', 'viewDetail', 'perawat');
    } else if ($jabatan == 'Admin RS') {
        $this->view->display('Permintaan', 'viewDetail', 'rs');
    }
}
}

```

Tabel C. 14 Kode program Melihat Data Permintaan Darah (Model)

```

function getDetail($id){
    $readData = $this->db->selectWhere('select * from permintaan p join user u on
p.NIPperawat = u.NIP join Instansi i on u.idInstansi = i.idInstansi where idPermintaan =
:idPermintaan LIMIT 1', array ('idPermintaan' => $id));
    return $readData['data'];
}
}

```

D8 Kode program Melihat Stok Darah

Tabel C. 15 Kode program Melihat Stok Darah (Controller)

```

function viewStok($produk, $goldar){
    $this->Model->kadaluwarsa();
    $show = $this->Model->show($produk, $goldar);
    return $show;
}
}

```

Tabel C. 16 Kode program Melihat Stok Darah (Model)

```

function kadaluwarsa(){
    $stmt = $this->db->prepare("UPDATE `donor` set keterangan = 'kadaluwarsa' where
tglKadaluwarsa <= '2017-04-14' and keterangan = 'masuk' ");
    $stmt->execute();
}

function show($produk, $goldar){
    $readData = $this->db->selectWhere("SELECT COUNT(*) AS jumlah FROM donor d
JOIN pendonor p ON d.noID = p.noID
WHERE p.golDarah = :goldar AND d.produk = :produk AND keterangan = 'masuk'
",
    array(
        'goldar' => $goldar,
        'produk' => $produk
    ));
    foreach ($readData['data'] as $key => $value) {
        return $value['jumlah'];
    }
}

```

D9 Kode program Validasi Data Permintaan Darah

Tabel C. 17 Kode program Validasi Data Permintaan Darah (Controller)

```

function terima($id){
    $this->view->display = $id;
    $this->view->display('Permintaan', 'validasi', 'pelayananUTD');
}

function tolak($id){
    $tglValidasi = date("Y-m-d");
    $status = 'tolak';
    $this->Model->validasiTolak($id, $status, $tglValidasi);
    $this->index();
}

function terimaFix($id){
    $tglValidasi = date("Y-m-d");
    $idDonor = $_POST['idDonor'];
    $status = 'terima';
    $this->Model->setDonor($idDonor);
    $this->Model->validasiTerima($id, $status, $tglValidasi, $idDonor);
    $this->index();
}

```

Tabel C. 18 Kode program Validasi Data Permintaan Darah (Model)

```

function validasiTerima($id, $status, $tglValidasi, $idDonor){
    $this->db->update('UPDATE `permintaan` SET `statusPermintaan`=:statusPermintaan,
`tglValidasi`=:tglValidasi, `idDonor`=:idDonor WHERE idPermintaan = :idPermintaan',

```

```

array('idPermintaan' => $id, 'statusPermintaan' => $status, 'tglValidasi' => $tglValidasi,
'idDonor' => $idDonor));
}

function validasiTolak($id, $status, $tglValidasi){
    $this->db->update('UPDATE `permintaan` SET `statusPermintaan`=:statusPermintaan,
`tglValidasi`=:tglValidasi WHERE idPermintaan = :idPermintaan', array('idPermintaan' =>
$id, 'statusPermintaan' => $status, 'tglValidasi' => $tglValidasi));
}

function setDonor($idDonor){
    $this->db->update('UPDATE `donor` SET `keterangan`= "keluar" WHERE idDonor =
:idDonor', array('idDonor' => $idDonor));
}

```

D10 Kode program Mengelola Data Pendoror

Tabel C. 19 Kode program Mengelola Data Pendoror (Controller)

```

function create() {
    if (isset($_POST['submit'])) {
        $noID = $_POST['noID'];
        $nama = $_POST['nama'];
        $alamat = $_POST['alamat'];
        $telp = $_POST['telp'];
        $pekerjaan = $_POST['pekerjaan'];
        $tempatLahir = $_POST['tempatLahir'];
        $tglLahir = $_POST['tglLahir'];
        $lahir = new DateTime($tglLahir);
        $now = new DateTime();
        $selisih = $now->diff($lahir);
        $umur = $selisih->y;
        $goldar = $_POST['goldar'];
        $donorPuasa = $_POST['donorPuasa'];
        $tglAkhirdonor = $_POST['tglAkhirdonor'];
        $banyak = $_POST['banyak'];
        $status = 'normal';
        $this->Model->create($noID, $nama, $alamat, $telp, $pekerjaan, $tempatLahir,
        $tglLahir, $umur, $goldar, $donorPuasa, $tglAkhirdonor, $banyak, $status);
        $this->index();
    }
}

function getPendoror($key){
    $this->view->display = $this->Model->cariPendoror($key);
    $this->view->display('Pendoror', 'viewDetail', 'pelayananUTD');
}

function kembali(){
    $this->index();
}

```

Tabel C. 20 Kode program Mengelola Data Pendonor (Model)

```

function create($noID, $nama, $alamat, $telp, $pekerjaan, $tempatLahir, $tglLahir, $umur,
$goldar, $donorPuasa, $tglAkhirdonor, $banyak, $status){
    $stmt = $this->db->prepare("INSERT INTO `pendonor`(`noID`, `nama`, `alamat`,
`telp`, `pekerjaan`, `tempatLahir`, `tanggalLahir`, `umur`, `golDarah`, `donorPuasa`,
`terakhirDonor`, `kaliDonor`, `status`)
    VALUES (:noID, :nama, :alamat, :telp, :pekerjaan, :tempatLahir, :tanggalLahir,
:umur, :golDarah, :donorPuasa, :terakhirDonor, :kaliDonor, :status)");
    $stmt->bindParam(':noID', $noID, PDO::PARAM_STR);
    $stmt->bindParam(':nama', $nama, PDO::PARAM_STR);
    $stmt->bindParam(':alamat', $alamat, PDO::PARAM_STR);
    $stmt->bindParam(':telp', $telp, PDO::PARAM_STR);
    $stmt->bindParam(':pekerjaan', $pekerjaan, PDO::PARAM_STR);
    $stmt->bindParam(':tempatLahir', $tempatLahir, PDO::PARAM_STR);
    $stmt->bindParam(':tanggalLahir', $tglLahir, PDO::PARAM_STR);
    $stmt->bindParam(':umur', $umur, PDO::PARAM_STR);
    $stmt->bindParam(':golDarah', $goldar, PDO::PARAM_STR);
    $stmt->bindParam(':donorPuasa', $donorPuasa, PDO::PARAM_STR);
    $stmt->bindParam(':terakhirDonor', $tglAkhirdonor, PDO::PARAM_STR);
    $stmt->bindParam(':kaliDonor', $banyak, PDO::PARAM_STR);
    $stmt->bindParam(':status', $status, PDO::PARAM_STR);
    $stmt->execute();
}

function cariPendonor($key){
    $readData = $this->db->selectWhere('select * from pendonor where noID = :key or
nama = :key', array('key' => $key));
    return $readData['data'];
}

```

D11 Kode program Menambah Data Transaksi Donor

Tabel C. 21 Kode program Menambah Data Transaksi Donor (Controller)

```

function create($id) {
    if (isset($_POST['submit'])) {
        $macamDonor = $_POST['macamDonor'];
        $kadarHB = $_POST['kadarHB'];
        $beratBadan = $_POST['bb'];
        $statusDonor = $_POST['statusDonor'];
        $salasanStatus = $_POST['ket'];
        $tglMasuk = date("Y-m-d");
        $proses = strtotime ( '+1 month' , strtotime ( $tglMasuk ) );
        $tglKadaluwarsa = date("Y-m-d",$proses);

        $this->Model->create($id, $macamDonor, $kadarHB, $beratBadan, $statusDonor,
$salasanStatus, $tglMasuk, $tglKadaluwarsa);
        $this->redirect('pendonor');
    }
}

```



```

function getDataDonor($key){
    $this->view->display = $this->Model->cariDataDonor($key);

    $jabatan = session::get('jabatan');
    if ($jabatan == 'Pelayanan UTD') {
        $this->view->display('Donor', 'viewDetail', 'pelayananUTD');
    } else if ($jabatan == 'Quality UTD') {
        $this->view->display('Donor', 'viewDetail', 'qualityUTD');
    }
}
}

```

Tabel C. 22 Kode program Menambah Data Transaksi Donor (Model)

```

function getRiwayatDonor($key){
    $readData = $this->db->selectWhere('select * from donor d join pendonor p on d.noID
= p.noID where p.noID = :key', array ('key' => $key));
    return $readData['data'];
}

function create($id, $macamDonor, $kadarHB, $beratBadan, $statusDonor, $alasanStatus,
$tglMasuk, $tglKadaluwarsa){
    $stmt = $this->db->prepare("INSERT INTO `donor`(`noID`, `macamDonor`,
`kadarHB`, `beratBadan`, `statusDonor`, `alasanStatus`, `tglMasuk`, `tglKadaluwarsa`)
VALUES (:noID, :macamDonor, :kadarHB, :beratBadan, :statusDonor, :alasanStatus,
:tglMasuk, :tglKadaluwarsa)");
    $stmt->bindParam(':noID', $id, PDO::PARAM_STR);
    $stmt->bindParam(':macamDonor', $macamDonor, PDO::PARAM_STR);
    $stmt->bindParam(':kadarHB', $kadarHB, PDO::PARAM_STR);
    $stmt->bindParam(':beratBadan', $beratBadan, PDO::PARAM_STR);
    $stmt->bindParam(':statusDonor', $statusDonor, PDO::PARAM_STR);
    $stmt->bindParam(':alasanStatus', $alasanStatus, PDO::PARAM_STR);
    $stmt->bindParam(':tglMasuk', $tglMasuk, PDO::PARAM_STR);
    $stmt->bindParam(':tglKadaluwarsa', $tglKadaluwarsa, PDO::PARAM_STR);
    $stmt->execute();
}

function cariDataDonor($key){
    $readData = $this->db->selectWhere('select * from donor d join pendonor p on d.noID
= p.noID where idDonor = :key', array ('key' => $key));
    return $readData['data'];
}
}

```

D12 Kode program Menambah Data Rumah Sakit

Tabel C. 23 Kode program Menambah Data Rumah Sakit (Controller)

```

function createData() {

    if (isset($_POST['submit'])) {

```

```

        $idInstansi = $_POST['id'];
        $namaInstansi = $_POST['nama'];
        $alamatInstansi = $_POST['alamat'];
        $tipe = '2';

        $NIP = $_POST['nipAdmin'];
        $nama = $_POST['namaAdmin'];
        $gender = $_POST['genderAdmin'];
        $alamat = $_POST['alamatAdmin'];
        $telp = $_POST['teleponAdmin'];
        $username = $_POST['username'];
        $pass = $_POST['pass'];
        $idJabatan = '1';
        $statusUser = 'aktif';

        $hasil = $this->Model->create($idInstansi, $namaInstansi, $alamatInstansi, $tipe);
        $hasil = $this->Model->createAdminRS($NIP, $idInstansi, $nama, $alamat, $telp,
        $gender, $username, $pass, $idJabatan, $statusUser);
        $this->index();
    }
}

```

Tabel C. 24 Kode program Menambah Data Rumah Sakit (Model)

```

function create($idInstansi, $namaInstansi, $alamatInstansi, $tipe){
    $stmt = $this->db->prepare("INSERT INTO `instansi`(`idInstansi`, `idTipe`,
`namaInstansi`, `alamatInstansi`)
    VALUES (:idInstansi, :idTipe, :namaInstansi, :alamatInstansi)");
    $stmt->bindParam(':idInstansi', $idInstansi, PDO::PARAM_STR);
    $stmt->bindParam(':idTipe', $tipe, PDO::PARAM_STR);
    $stmt->bindParam(':namaInstansi', $namaInstansi, PDO::PARAM_STR);
    $stmt->bindParam(':alamatInstansi', $alamatInstansi, PDO::PARAM_STR);
    $stmt->execute();
}

function createAdminRS($NIP, $idInstansi, $nama, $alamat, $telp, $gender, $username,
$pass, $idJabatan, $statusUser){
    $stmt = $this->db->prepare("INSERT INTO `user`(`NIP`, `idInstansi`, `username`,
`pass`, `nama`, `alamat`, `telp`, `gender`, `idJabatan`, `statusUser`)
    VALUES (:NIP, :idInstansi, :username, :pass, :nama, :alamat, :telp, :gender,
:idJabatan, :statusUser)");
    $stmt->bindParam(':NIP', $NIP, PDO::PARAM_STR);
    $stmt->bindParam(':idInstansi', $idInstansi, PDO::PARAM_STR);
    $stmt->bindParam(':username', $username, PDO::PARAM_STR);
    $stmt->bindParam(':pass', $pass, PDO::PARAM_STR);
    $stmt->bindParam(':nama', $nama, PDO::PARAM_STR);
    $stmt->bindParam(':alamat', $alamat, PDO::PARAM_STR);
    $stmt->bindParam(':telp', $telp, PDO::PARAM_STR);
    $stmt->bindParam(':gender', $gender, PDO::PARAM_STR);
    $stmt->bindParam(':idJabatan', $idJabatan, PDO::PARAM_STR);
    $stmt->bindParam(':statusUser', $statusUser, PDO::PARAM_STR);
    $stmt->execute();
}

```

D13 Kode program Melihat Data Transaksi Donor dan Transaksi Permintaan Darah

Tabel C. 25 Kode program Melihat Data Transaksi Donor dan Transaksi Permintaan Darah (Controller)

```
function index() {
    $this->view->masuk = $this->Model->getDonor();
    $this->view->totalMasuk = $this->Model->viewAllTotalMasuk();
    $this->view->keluar = $this->Model->getPermintaan();
    $this->view->totalKeluar = $this->Model->viewAllTotalKeluar();
    $this->view->display('Transaksi', 'view', 'qualityUTD');
}

function kembali(){
    $this->index();
}

function cariPermintaan(){
    $tgl = $_POST['tgl'];

    $this->view->masuk = $this->Model->getDonor();
    $this->view->totalMasuk = $this->Model->viewAllTotalMasuk();
    $this->view->keluar = $this->Model->cariPermintaan($tgl);
    $this->view->totalKeluar = $this->Model->getTotalPermintaan($tgl);
    $this->view->display('Transaksi', 'view', 'qualityUTD');
}

function cariDonor(){
    $tgl = $_POST['tgl'];

    $this->view->masuk = $this->Model->cariDonor($tgl);
    $this->view->totalMasuk = $this->Model->getTotalDonor($tgl);
    $this->view->keluar = $this->Model->getPermintaan();
    $this->view->totalKeluar = $this->Model->viewAllTotalKeluar();
    $this->view->display('Transaksi', 'view', 'qualityUTD');
}
```

Tabel C. 26 Kode program Melihat Data Transaksi Donor dan Transaksi Permintaan Darah (Model)

```
function getDonor(){ // hari ini
    $readData = $this->db->selectWhere('select * from donor d join pendonor p on d.noID = p.noID where statusDonor = "berhasil" and kualitasDarah = "baik" and `tglMasuk` = "2017-04-14"');
    return $readData['data'];
}

function getPermintaan(){ // hari ini
```

```

    $readData = $this->db->selectWhere('select * from permintaan p join user u on
p.NIPperawat = u.NIP join instansi i on u.idInstansi = i.idInstansi where statusPermintaan =
"terima" and tglPermintaan = "2017-04-14"');
    return $readData['data'];
}

function viewAllTotalMasuk(){ // hari ini
    $readData = $this->db->selectWhere('SELECT COUNT(*) AS jumlah from donor
where statusDonor = "berhasil" and kualitasDarah = "baik" and `tglMasuk` = "2017-04-14"
');
    foreach ($readData['data'] as $key => $value) {
        return $value['jumlah'];
    }
}

function viewAllTotalKeluar(){ // hari ini
    $readData = $this->db->selectWhere('SELECT SUM(jumlah) AS jumlah from
permintaan where statusPermintaan = "terima" and tglPermintaan = "2017-04-14"');
    foreach ($readData['data'] as $key => $value) {
        return $value['jumlah'];
    }
}

function cariPermintaan($tgl){
    $readData = $this->db->selectWhere('select * from permintaan p join user u on
p.NIPperawat = u.NIP join instansi i on u.idInstansi = i.idInstansi where statusPermintaan =
"terima" and tglPermintaan = :tgl', array ('tgl' => $tgl));
    return $readData['data'];
}

function getTotalPermintaan($tgl){ //yang dicari
    $readData = $this->db->selectWhere("SELECT SUM(jumlah) AS jumlah FROM user u
JOIN permintaan p ON u.NIP = p.NIPperawat WHERE statusPermintaan = 'terima' and
tglPermintaan = :tgl",
    array(
        'tgl' => $tgl
    ));
    foreach ($readData['data'] as $key => $value) {
        return $value['jumlah'];
    }
}

function cariDonor($tgl){
    $readData = $this->db->selectWhere('select * from donor d join pendonor p on d.noID
= p.noID where `tglMasuk` = :tgl and statusDonor = "berhasil" and kualitasDarah = "baik",
array ('tgl' => $tgl));
    return $readData['data'];
}

function getTotalDonor($tgl){ //yang dicari
    $readData = $this->db->selectWhere("SELECT COUNT(*) AS jumlah FROM donor d
JOIN pendonor p on d.noID = p.noID WHERE `tglMasuk` = :tgl and statusDonor = 'berhasil'
and kualitasDarah = 'baik' ",
    array(
        'tgl' => $tgl
    ));
}

```

```

    ));
    foreach ($readData['data'] as $key => $value) {
        return $value['jumlah'];
    }
}

```

D14 Kode program Verifikasi Kualitas Darah Donor

Tabel C. 27 Kode program Verifikasi Kualitas Darah Donor (Controller)

```

function kualitasDarah($idDonor, $idPendonor){
    $this->view->idDonor = $idDonor;
    $this->view->idPendonor = $idPendonor;
    $this->view->display('Donor', 'validasi', 'qualityUTD');
}

function updateValidasi($idDonor, $idPendonor){
    $kualitas = $_POST['kualitas'];
    $kkualitas = $_POST['kKualitas'];

    if ($kualitas == "cekal") {
        $this->Model->updateStatusPendonor($idPendonor);
        $this->Model->updateValidasiCekal($idDonor, $kkualitas);
    } else {
        $produk = $_POST['produk'];
        $this->Model->updateValidasiMasuk($idDonor, $kkualitas, $produk);
    }
    $this->index();
}

```

Tabel C. 28 Kode program Verifikasi Kualitas Darah Donor (Model)

```

function getDonorBaru(){
    $readData = $this->db->selectWhere('Select * from donor d join pendonor p on d.noID = p.noID where keterangan != "masuk" and keterangan != "keluar" and keterangan != "kadaluwarsa" and statusDonor = "berhasil");
    return $readData['data'];
}

function updateStatusPendonor($idPendonor){
    $this->db->update('UPDATE `pendonor` SET `status` = "cekal" WHERE noID = :noID', array('noID' => $idPendonor));
}

function updateValidasiCekal($idDonor, $kkualitas){
    $this->db->update('UPDATE `donor` SET `kualitasDarah` = "cekal", `alasanKualitas` = :alasanKualitas, `keterangan` = "keluar" WHERE `idDonor` = :idDonor', array('alasanKualitas' => $kkualitas, 'idDonor' => $idDonor));
}

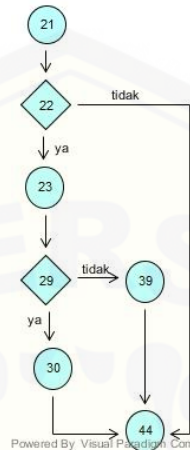
```

```
function updateValidasiMasuk($idDonor, $kkualitas, $produk){
    $this->db->update('UPDATE `donor` SET `kualitasDarah`= "baik", `alasanKualitas`=
:alasanKualitas, `produk`= :produk, `keterangan`= "masuk"
    WHERE `idDonor` = :idDonor ', array('idDonor' => $idDonor, 'produk' => $produk,
'alasanKualitas' => $kkualitas));
}
```



Lampiran E. Pengujian White Box

E1 Pengujian white box function login



Gambar E. 1 Diagram Alir function login

$$CC = E - N + 2 = 8 - 7 + 2 = 3$$

Tabel E. 1 Test Case function login

<i>Test case 1</i>	Jika field tidak terisi
Target yang diharapkan	Menampilkan halaman login
Hasil pengujian	Benar
<i>Path/Jalur</i>	21 – 22 – 44
<i>Test case 2</i>	Jika login berhasil
Target yang diharapkan	Menampilkan halaman dashboard masing-masing <i>user</i>
Hasil pengujian	Benar
<i>Path/Jalur</i>	21 – 22 – 23 – 29 – 30 – 44
<i>Test case 3</i>	Jika <i>username dan password</i> salah
Target yang diharapkan	Menampilkan pop up “login gagal”
Hasil pengujian	Benar
<i>Path/Jalur</i>	21 – 22 – 23 – 29 – 39 – 44

E2 Pengujian white box function showTransaksi



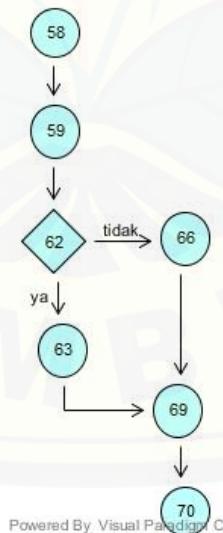
Gambar E. 2 Diagram Alir function showTransaksi

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel E. 2 Test Case function login

Test case 1	Menampilkan data transaksi darah
Target yang diharapkan	Menampilkan data transaksi darah
Hasil pengujian	Benar
Path/Jalur	10 – 11 – 17

E3 Pengujian white box function validasiKualitas



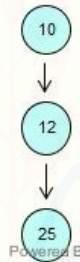
Gambar E. 3 Diagram Alir function validasiKualitas

$$CC = E - N + 2 = 7 - 7 + 2 = 2$$

Tabel E. 3 Test Case function validasiKualitas

<i>Test case 1</i>	Jika kualitas darah baik
Target yang diharapkan	Update status kualitas darah
Hasil pengujian	Benar
<i>Path/Jalur</i>	58 – 59 – 62 – 66 – 69 – 70
<i>Test case 2</i>	Jika kualitas darah cekal
Target yang diharapkan	Update status kualitas darah dan kualitas pendonor
Hasil pengujian	Benar
<i>Path/Jalur</i>	58 – 59 – 62 – 63 – 69 – 70

E4 Pengujian white box function showDokter



Gambar E. 4 Diagram Alir function showDokter

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel E. 4 Test Case function showDokter

<i>Test case 1</i>	Menampilkan data dokter
Target yang diharapkan	Menampilkan data dokter
Hasil pengujian	Benar
<i>Path/Jalur</i>	10 – 12 – 25

Lampiran F. Pengujian Black Box

No	Menu	Fungsi	Aksi	Hasil	Ket
1.	Login	Menu login merupakan menu awal yang digunakan untuk autentikasi user, untuk masuk ke dalam sistem	Mengisi form login berupa <i>username</i> dan <i>password</i> lalu menekan tombol “Login”	<i>Login</i> berhasil dan menampilkan halaman dashboard <i>user</i>	√
			<i>Username</i> atau <i>Password</i> salah	Menampilkan popup “ login gagal ”	√
2.	Permintaan Darah (perawat)	Menu Permintaan Darah berfungsi untuk mengelola data permintaan darah (tambah, lihat, ubah dan cancel)	Memilih menu permintaan darah	Menampilkan tabel data permintaan darah	√
			Menekan tombol “isi formulir”	Menampilkan form input data permintaan darah	√
			Menekan tombol “detail”	Menampilkan detail data permintaan darah	√
			Menekan tombol edit”	Menampilkan form ubah data permintaan darah	√
			Menekan tombol “cancel”	Berhasil mengubah status permintaan darah	√
3.	Dokter (admin rumah sakit)	Menu Dokter berfungsi untuk mengelola data dokter (tambah, lihat, dan nonaktif)	Memilih menu dokter	Menampilkan data dokter rumah sakit	√
			Menekan tombol “tambah dokter”	Menampilkan form input data dokter	√
			Menekan tombol “nonaktif”	Berhasil mengubah data dokter menjadi tidak aktif	√
4.	Perawat (admin rumah sakit)		Memilih menu perawat	Menampilkan data perawat rumah sakit	√

No	Menu	Fungsi	Aksi	Hasil	Ket
		Menu Perawat berfungsi untuk mengelola data perawat (tambah, lihat, dan nonaktif)	Menekan tombol “tambah perawat”	Menampilkan form input data perawat	√
			Menekan tombol “nonaktif”	Berhasil mengubah data perawat menjadi tidak aktif	√
5.	Ruangan (admin rumah sakit)	Menu Ruangan berfungsi untuk mengelola data ruangan rumah sakit (tambah, lihat, dan kosongkan)	Memilih menu ruangan	Menampilkan data ruangan rumah sakit	√
			Menekan tombol “tambah ruang”	Menampilkan form input data ruangan	√
			Menekan tombol “kosongkan”	Berhasil mengubah data ruangan menjadi tidak dipakai	√
4.	Permintaan Darah (admin rumah sakit)	Menu Permintaan Darah berfungsi untuk melihat data permintaan darah	Memilih menu permintaan darah	Menampilkan tabel data permintaan darah	√
			Menekan tombol “detail”	Menampilkan detail data permintaan darah	√
5.	Rumah Sakit (admin pelayanan UTD)	Menu Rumah Sakit berfungsi untuk menambah data rumah sakit	Memilih menu rumah sakit	Menampilkan tabel data rumah sakit	√
			Menekan tombol “tambah”	Menampilkan form input data rumah sakit	√
6.	Donor Darah (admin pelayanan UTD)	Menu Donor Darah berfungsi untuk mengelola data pendonor (tambah, lihat)	Memilih menu transaksi donor darah	Menampilkan tabel data pendonor	√
			Menekan tombol “tambah pendonor”	Menampilkan form input data pendonor	√
7.	Permintaan Darah (admin pelayanan UTD)	Menu permintaan darah berfungsi untuk validasi data permintaan darah	Memilih menu permintaan darah	Menampilkan data permintaan darah yang belum divalidasi	√
			Menekan tombol “detail”	Menampilkan detail data permintaan darah	√

No	Menu	Fungsi	Aksi	Hasil	Ket
			Menekan tombol “tolak” atau “terima” pada halaman detail	Berhasil mengubah status permintaan darah	√
8.	Kualitas Darah (admin penjamin mutu UTD)	Menu kualitas darah berfungsi untuk verifikasi kualitas darah donor	Memilih kualitas darah	Menampilkan data donor darah yang belum di verifikasi	√
			Menekan tombol “verifikasi”	Mengubah status kualitas darah donor	√
9.	Rekap Transaksi (admin penjamin mutu UTD)	Menu rekap transaksi berfungsi untuk menampilkan transaksi donor dan transaksi permintaan darah per hari	Memilih menu rekap transaksi	Menampilkan data transaksi donor dan transaksi permintaan darah	√
10.	Ramalan Permintaan (admin penjamin mutu UTD)	Menu ramalan permintaan berfungsi untuk mendapatkan hasil peramalan dan menambahkan data peramalan.	Memilih menu ramalan permintaan	Menampilkan data ramalan permintaan serta grafik permintaan dan peramalannya dalam jangka waktu satu minggu	√
			Menekan tombol “peramalan”	Menampilkan hasil perhitungan peramalan dan metode yang tepat	√