

**PENGEMBANGAN SISTEM INFORMASI RAWAT INAP  
DENGAN METODE *RATIONAL UNIFIED PROCESS*  
(STUDI KASUS RUMAH SAKIT KHUSUS HAYUNANTO  
*MEDICAL CENTER MALANG*)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Candra Dwi Kusuma  
NIM: 145150401111043



PROGRAM STUDI SISTEM INFORMASI  
JURUSAN SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2019

## PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI RAWAT INAP  
DENGAN METODE *RATIONAL UNIFIED PROCESS*  
(STUDI KASUS RUMAH SAKIT KHUSUS HAYUNANTO *MEDICAL CENTER* MALANG)

### SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Candra Dwi Kusuma  
NIM: 145150401111043

Skripsi ini telah diuji dan dinyatakan lulus pada  
25 Juli 2019

Telah diperiksa dan disetujui oleh:

Pembimbing I

Yusi Tyroni Mursityo, S.Kom., M.A.B.  
NIP: 198002282006041001

Pembimbing II

Welly Purnomo, S.T., M.Kom.  
NIK: 2017088101171001

Mengetahui

Ketua Jurusan Sistem Informasi



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



## PERNYATAAN ORISINALITAS

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

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

Malang, 25 Juli 2019



Candra Dwi Kusuma

NIM: 145150401111043



## PRAKATA

Puji dan syukur kehadiran Tuhan Yang Maha Esa atas berkat rahmat serta kasih-Nya sehingga penulis dapat menyelesaikan skripsi. Naskah skripsi ini diajukan untuk memenuhi salah satu syarat untuk memperoleh gelar sarjana. Judul yang penulis ajukan sebagai skripsi adalah “PENGEMBANGAN SISTEM INFORMASI RAWAT INAP DENGAN METODE *RATIONAL UNIFIED PROCESS* (STUDI KASUS RUMAH SAKIT KHUSUS HAYUNANTO *MEDICAL CENTER* MALANG)”.

Dalam penyusunan dan penulisan naskah skripsi ini penulis mendapatkan banyak bantuan, bimbingan serta dukungan dari berbagai pihak baik secara moril maupun materil. Dalam kesempatan ini penulis ingin mengucapkan banyak terima kasih kepada yang terhormat :

1. Bapak Yusi Tyroni Mursityo, S.Kom., M.AB. selaku Dosen pembimbing pertama yang telah membimbing dan memberikan banyak solusi dalam pengerjaan skripsi.
2. Bapak Welly Purnomo, S.T., M.Kom. selaku Dosen pembimbing kedua yang telah membimbing dan memberikan banyak masukan selama pengerjaan skripsi.
3. Bapak Dr. Eng. Herman Tolle, S.T, M.T selaku Ketua Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Ismiarta Aknuranda, S.T, M.Sc, Ph.D selaku sekretaris Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya Malang.
5. Bapak Yusi Tyroni Mursityo, S.Kom., M.AB. selaku Ketua Program Studi Sistem Informasi Universitas Brawijaya Malang.
6. Bapak Pariyo dan Ibu Partini selaku orang tua penulis dan Yoga Prasetya selaku kakak dari penulis yang telah memberi segenap dukungan dan motivasi kepada penulis.
7. Seluruh Dosen dan seluruh civitas akademik Fakultas Ilmu Komputer yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Universitas Brawijaya.
8. Bapak F.Antono selaku narasumber Rumah Sakit Khusus Hayunanto *Medical Center* Malang yang bersedia membimbing dan mengizinkan serta membantu penulis untuk melakukan penelitian ini.
9. Seluruh teman-teman yang tidak dapat penulis sebutkan satu per satu yang telah memberikan semangat dan dukungan agar skripsi ini cepat terselesaikan.

Semoga Allah SWT melimpahkan rahmat berlipat ganda kepada seluruh pihak yang membantu dan berkontribusi pada proses penelitian ini. Tidak lupa saya sampaikan bahwa penelitian yang telah dilakukan memiliki banyak kekurangan,

diskusi mengenai penelitian lanjutan, saran, dan kritik yang membangun merupakan beberapa hal yang saya harapkan dapat disampaikan kepada saya.

Malang, 5 Juni 2019

Candra Dwi Kusuma

candradwikusuma07@gmail.com



## ABSTRAK

**Candra Dwi Kusuma, Pengembangan Sistem Informasi Rawat Inap dengan Metode *Rational Unified Process* (Studi Kasus Rumah Sakit Khusus Hayunanto Medical Center Malang)**

**Dosen Pembimbing: Yusi Tyrone Mursityo, S.Kom., M.A.B. dan Welly Purnomo, S.T., M.Kom.**

Rumah Sakit Khusus Hayunanto *Medical Center* merupakan salah satu rumah sakit yang memfokuskan diri pada pelayanan kesehatan rehabilitasi mental dan rehabilitasi ketergantungan narkoba di Malang. Dimana Ketepatan dan kecepatan alur informasi dalam layanan kesehatan merupakan salah satu hal terpenting yang sangat mempengaruhi kesehatan dan keselamatan seseorang. Salah satu layanan kesehatan yang membutuhkan alur informasi yang cepat pada rumah sakit ini khususnya di bagian rawat inap. Namun Pengimplementasian proses pencatatan data rawat inap masih dilakukan pada dokumen atau formulir yang selanjutnya hanya disimpan dengan map itu membuat tidak mempercepat alur informasi di Rumah sakit ini membuat terlalu lamanya untuk melakukan pengurutan data sehingga berdampak pada proses pencarian data menjadi lama dan bahkan tidak diketahui karena harus menelusuri satu per satu data yang dicari. Begitu juga pada proses transaksi pemakaian obat atau periksa harus mengentri ulang data pasien, hal ini dikarenakan data pasien belum tersimpan secara rapi dalam sebuah penyimpanan. Dalam penerapan sistem yang manual akan sulit melakukan kontrol karena pemrosesan yang dilakukan oleh manusia maka untuk terjadinya kesalahan akan semakin besar, seperti terjadinya pengulangan pencatatan atau redundansi data data. Dalam menggunakan sistem informasi diharapkan pemrosesan akan berlangsung cepat dan memudahkan pengontrolan sehingga kemungkinan terjadinya *human error* dapat ditekan.

Sistem dikembangkan dengan metode *Rational Unified Process* (RUP) dengan tahapan fase *inception, elaboration, construction, dan transition*. Sistem informasi rawat inap diimplementasi dengan hasil pengujian validasi untuk menguji fungsional sistem dengan hasil 100% *valid* dan *compability testing* untuk menguji kebutuhan non fungsional didapat hasil 2 *critical issues* pada 2 aplikasi peramban. *User Acceptance Testing* yang dilakukan dengan lima koresponden dari masing-masing pengguna menunjukkan sistem informasi layanan rawat inap dapat diterima dengan baik oleh pengguna.

**Kata kunci:** Sistem Informasi, Rawat Inap, RUP, Rumah Sakit, Layanan Kesehatan.

## ABSTRACT

**Candra Dwi Kusuma, *Development of Inpatient Information Systems with the Rational Unified Process Method (Case Study of Hayunanto Medical Center Malang Special Hospital)***

**Supervisor: Yusi Tyroni Mursityo, S.Kom., M.A.B. and Welly Purnomo, S.T., M.Kom.**

*Hayunanto Medical Center Hospital is one of the hospitals that focuses on health services for mental rehabilitation and drug dependence rehabilitation in Malang. Where the accuracy and speed of information flow in health services is one of the most important things that greatly affects a person's health and safety. One health service that requires rapid flow of information in this hospital, especially in the inpatient department. However, the implementation of the process of recording inpatient data is still carried out on documents or forms which are then only stored with maps. This does not speed up the flow of information in this hospital. browse one by one the data sought. Likewise in the process of drug use transactions or checks must re-enter patient data, this is because patient data is not stored neatly in a storage. In the application of a manual system it will be difficult to control because the processing is done by humans so that the occurrence of errors will be greater, such as the occurrence of repetition of recording or reduction of data. In using information systems it is expected that processing will take place quickly and facilitate control so that the possibility of human error can be suppressed.*

*The system was developed using the Rational Unified Process (RUP) method with stages of inception, elaboration, construction, and transition. Inpatient information system is implemented with the results of validation testing to test the functional system with 100% valid results and compability testing to test non-functional requirements obtained by 2 critical issues on 2 browser applications. User Acceptance Testing conducted with five correspondents from each user shows that the inpatient service information system can be well received by the user.*

**Keywords:** *Information System, Inpatient, RUP, Hospital, Health Services.*

## DAFTAR ISI

PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS .....	iii
PRAKATA .....	iv
ABSTRAK .....	vi
ABSTRACT .....	vii
DAFTAR ISI .....	viii
DAFTAR TABEL .....	xii
DAFTAR GAMBAR .....	xvi
DAFTAR LAMPIRAN .....	xvii
BAB 1 PENDAHULUAN .....	1
1.1 Latar belakang .....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan .....	4
1.4 Manfaat .....	4
1.5 Batasan masalah.....	5
1.6 Sistematika pembahasan.....	5
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI.....	7
2.1 Kajian Pustaka.....	7
2.2 Dasar Teori.....	14
2.2.1 Rumah Sakit Khusus Hayunanto <i>Medical Center</i> .....	14
2.2.2 Visi Rumah Sakit Khusus Hayunanto <i>Medical Center</i> .....	14
2.2.3 Misi Rumah Sakit Khusus Hayunanto <i>Medical Center</i> .....	14
2.3 Struktur Organisasi .....	14
2.4 Rawat Inap.....	15
2.5 Sistem Informasi .....	16
2.6 <i>Rational Unified Process (RUP)</i> .....	16
2.6.1 <i>Inception</i> .....	17
2.6.2 <i>Elaboration</i> .....	23
2.6.3 <i>Contruction</i> .....	28
2.6.4 <i>Transition</i> .....	30
BAB 3 METODOLOGI PENELITIAN .....	32





3.1	Studi Literatur .....	33
3.2	Analisa Permasalahan .....	33
3.3	Metode Rational Unfiend Process (RUP) .....	33
3.1.1	<i>Inception</i> .....	34
3.1.2	<i>Elaboration</i> .....	34
3.1.3	<i>Contruction</i> .....	35
3.1.4	<i>Transition</i> .....	36
3.4	Tahap Pengambilan Kesimpulan dan Saran .....	36
BAB 4 ANALISIS KEBUTUHAN .....		37
4.1.1	Identifikasi Tipe Pemangku Kepentingan .....	37
4.1.2	Analisis Masalah .....	38
4.1.3	Pemodelan Proses Bisnis .....	39
4.1.3.1	Proses Bisnis As-Is .....	40
4.1.3.2	Proses Bisnis To-be .....	43
4.2	Kesimpulan Analisis Kebutuhan Fase Inepsi Metode Rational Unifeld Proccess .....	76
BAB 5 PERANCANGAN .....		78
5.1	Pemodelan Interaksi Objek .....	78
5.1.1	Sequence Diagram Pendaftaran Pasien .....	79
5.1.2	Sequence Diagram Rekam medis .....	80
5.1.3	<i>Sequence Diagram</i> Memperbarui Pemeriksaan Pasien .....	81
5.1.4	<i>Sequence Diagram</i> Memperbarui Obat Keluar Pasien .....	82
5.1.5	Sequence Diagram kelola Data Obat .....	83
5.1.6	<i>Sequence Diagram</i> kelola Pasien Keluar .....	84
5.2	Pemodelan Objek .....	85
5.2.1	<i>Diagram Class</i> Perancangan .....	85
5.3	Perancangan Database .....	87
5.4	Perancangan Antarmuka .....	88
5.4.1	Perancangan Antarmuka Halaman Awal Admin .....	88
5.4.2	Perancangan Antarmuka Data User Front Office .....	88
5.4.3	Perancangan Antarmuka Pendaftaran Pasien .....	89
5.4.4	Perancangan Antarmuka Transaksi Pasien .....	90
5.5	Perancangan Algoritma .....	91

5.5.1 Algoritma Menyimpan Data Pendaftaran Pasien Baru .....	91
5.5.2 Algoritma Melihat Data Rekam Medis Pasien.....	92
5.5.3 Algoritma Mengubah Hasil Rekam Medis.....	93
BAB 6 IMPLEMENTASI.....	94
6.1 Spesifikasi Lingkungan Sistem Informasi .....	94
6.2 Implementasi Algoritma .....	95
6.2.1 Implementasi Algoritma Menambah Data Pendaftaran Pasien Baru.....	95
6.2.2 Implementasi Algoritma Mengubah Data Rekam Medis .....	96
6.2.3 Implementasi Algoritma Mengolah Transaksi Biaya Pasien.....	98
6.3 Implementasi Antarmuka Pengguna .....	99
6.3.1 Antarmuka Halaman Admin.....	99
6.3.2 Antarmuka Halaman Daftar Rekam Medis Pasien .....	100
6.3.3 Antarmuka Halaman Pendaftaran Pasien .....	100
6.3.4 Antarmuka Halaman Data Transaksi Pasien.....	101
BAB 7 PENGUJIAN .....	102
7.1 Pengujian <i>Validation Testing</i> .....	102
7.1.1 Perancangan Pengujian <i>Validation Testing</i> .....	102
7.1.2 Hasil Pengujian <i>Validation Testing</i> .....	108
7.1.3 Kesimpulan <i>Validation Testing</i> .....	110
7.2 Pengujian <i>Compatibility Testing</i> .....	111
7.2.1 Perancangan Pengujian <i>Compatibility Testing</i> .....	111
7.2.2 Hasil Pengujian <i>Compatibility Testing</i> .....	115
7.2.3 Kesimpulan Pengujian <i>Compatibility Testing</i> .....	116
7.3 Kesimpulan Pengujian Fase Konstruksi Metode <i>Rational Unified Process</i> .....	116
7.4 Pengujian <i>User Acceptance Testing (UAT)</i> .....	117
7.4.1 Perancangan Pengujian <i>User Acceptance Testing (UAT)</i> .....	117
7.4.2 Hasil Pengujian <i>User Acceptance Testing (UAT)</i> .....	121
7.4.3 Kesimpulan Pengujian <i>User Acceptance Testing (UAT)</i> .....	130
7.5 Kesimpulan Pengujian Fase Transisi Metode <i>Rational Unified Process</i> .....	130
BAB 8 PENUTUP .....	131



8.1 Kesimpulan .....	131
8.2 Saran .....	132
DAFTAR REFERENSI .....	133
LAMPIRAN A HASIL WAWANCARA .....	135
LAMPIRAN B ARTEFAK .....	137
LAMPIRAN C DOKUMENTASI HASIL <i>USER ACCEPTANCE TESTING</i> (UAT) .....	139



## DAFTAR TABEL

Tabel 2. 1 Literatur Review .....	10
Tabel 2. 2 Analisis Permasalahan .....	18
Tabel 2. 3 Simbol <i>Event</i> .....	19
Tabel 2. 4 Simbol <i>Activity</i> .....	19
Tabel 2. 5 Simbol <i>Gateway</i> .....	20
Tabel 2. 6 Simbol <i>Connections</i> .....	21
Tabel 2. 7 Simbol <i>Swimlanes</i> .....	21
Tabel 2. 8 Simbol <i>Artifact</i> .....	22
Tabel 2. 9 Simbol <i>Use Case Diagram</i> .....	23
Tabel 2. 10 Simbol <i>Sequence Diagram</i> .....	25
Tabel 2. 11 Simbol <i>Class Diagram</i> .....	28
Tabel 4. 1 Tipe Pemangku Kepentingan .....	37
Tabel 4. 2 <i>Problem Statement</i> .....	38
Tabel 4. 3 Perubahan Aktivitas Proses Bisnis Pendaftaran Pasien .....	47
Tabel 4. 4 Perubahan Aktivitas Proses Bisnis Pemeriksaan Fisik .....	48
Tabel 4. 5 Perubahan Aktivitas Proses Bisnis Pelayanan Obat Pasien .....	48
Tabel 4. 6 Perubahan Aktivitas Proses Bisnis Pelayanan Pasien Keluar .....	48
Tabel 4. 7 Analisis Waktu .....	49
Tabel 4. 8 Hasil Identifikasi Pengguna .....	49
Tabel 4. 9 Hasil Identifikasi Kebutuhan Pengguna .....	51
Tabel 4. 10 Perubahan dan Penambahan Identifikasi Kebutuhan Pengguna .....	54
Tabel 4. 11 Identifikasi Fitur .....	55
Tabel 4. 12 Perbaikan Identifikasi Fitur .....	55
Tabel 4. 13 Perbaikan Hubungan Kebutuhan Pengguna Dengan Fitur .....	56
Tabel 4. 14 Kebutuhan Fungsional .....	56
Tabel 4. 15 Perbaikan Kebutuhan Fungsional .....	58
Tabel 4. 16 Analisis Kebutuhan Non-Fungsional .....	58
Tabel 4. 17 Identifikasi Aktor .....	60
Tabel 4. 18 <i>Use case Scenario login</i> .....	61
Tabel 4. 19 <i>Usecase Scenario Mengelola Data Persediaan Obat</i> .....	62
Tabel 4. 20 <i>Use case Scenario Mengelola Pemeriksaan Pasien</i> .....	65

Tabel 4. 21 <i>Use case Scenario</i> Mengelola Perkembangan Medis Pasien.....	67
Tabel 4. 22 <i>Use case Scenario</i> Mengelola Rincian Biaya Pemakaian Obat .....	69
Tabel 4. 23 <i>Use case Scenario</i> Mengelola Rincian Biaya Keseluruhan Pasien .....	71
Tabel 4. 24 <i>Use case Scenario</i> Mengelola Pendaftaran dan Data Pasien Rawat Inap .....	74
Tabel 4. 25 <i>Use case Scenario</i> Log out .....	76
Pemangku Kepentingan.....	37
Tabel 4. 2 <i>Problem Statement</i> .....	38
Tabel 4. 3 Perubahan Aktivitas Proses Bisnis Pendaftaran Pasien.....	47
Tabel 4. 4 Perubahan Aktivitas Proses Bisnis Pemeriksaan Fisik .....	48
Tabel 4. 5 Perubahan Aktivitas Proses Bisnis Pelayanan Obat Pasien .....	48
Tabel 4. 6 Perubahan Aktivitas Proses Bisnis Pelayanan Pasien Keluar .....	48
Tabel 4. 7 Analisis Waktu .....	49
Tabel 4. 8 Hasil Identifikasi Pengguna.....	49
Tabel 4. 9 Hasil Identifikasi Kebutuhan Pengguna .....	51
Tabel 4. 10 Perubahan dan Penambahan Identifikasi Kebutuhan Pengguna .....	54
Tabel 4. 11 Identifikasi Fitur .....	55
Tabel 4. 12 Perbaikan Identifikasi Fitur.....	55
Tabel 4. 13 Perbaikan Hubungan Kebutuhan Pengguna Dengan Fitur .....	56
Tabel 4. 14 Kebutuhan Fungsional.....	56
Tabel 4. 15 Perbaikan Kebutuhan Fungsional.....	58
Tabel 4. 16 Analisis Kebutuhan Non-Fungsional.....	58
Tabel 4. 17 Identifikasi Aktor .....	60
Tabel 4. 18 <i>Use case Scenario</i> login .....	61
Tabel 4. 19 <i>Usecase Scenario</i> Mengelola Data Persediaan Obat.....	62
Tabel 4. 20 <i>Use case Scenario</i> Mengelola Pemeriksaan Pasien .....	65
Tabel 4. 21 <i>Use case Scenario</i> Mengelola Perkembangan Medis Pasien.....	67
Tabel 4. 22 <i>Use case Scenario</i> Mengelola Rincian Biaya Pemakaian Obat .....	69
Tabel 4. 23 <i>Use case Scenario</i> Mengelola Rincian Biaya Keseluruhan Pasien .....	71
Tabel 4. 24 <i>Use case Scenario</i> Mengelola Pendaftaran dan Data Pasien Rawat Inap .....	74
Tabel 4. 25 <i>Use case Scenario</i> Log out .....	76
Tabel 5.1 Algoritma Menyimpan Data Pendaftaran Pasien.....	92

Tabel 5.2	Algoritma Melihat Data Hasil Rekam medis .....	92
Tabel 5.3	<i>Pseudocode</i> Mengubah Hasil Rekam medis .....	93
Tabel 6.1	Spesifikasi Perangkat Keras .....	94
Tabel 6.2	Spesifikasi Perangkat Lunak .....	94
Tabel 6.3	Spesifikasi Minimal Lingkungan <i>Deployment</i> .....	94
Tabel 6.4	Implementasi Algoritma Menambah Data Pendaftaran Pasien Baru ..	95
Tabel 6.5	Implementasi Algoritma Mengubah Data Rekam medis.....	96
Tabel 6.6	Implementasi Algoritma Mengolah Transaksi Biaya Pasien.....	98
Tabel 7.1	Rancangan Tampilan Hasil Pengujian <i>Validation Testing</i> .....	103
Tabel 7.2	Rancangan Skenario Testing Menambah Data Pendaftaran Baru .....	103
Tabel 7.3	Rancangan <i>Validation Testing</i> Menambah Data Pendaftaran Pasien Baru .....	103
Tabel 7.4	Rancangan <i>Validation Testing</i> Inputan Data Pendaftaran Baru Tidak Sesuai.....	104
Tabel 7.5	Rancangan Skenario Testing Menambah Data Pemeriksaan Pasien..	104
Tabel 7.6	Rancangan <i>Validation Testing</i> Menambah Data Pemeriksaan Pasien	104
Tabel 7.7	Rancangan <i>Validation Testing</i> Inputan Data Pemeriksaan Pasien Tidak Sesuai.....	105
Tabel 7.8	Rancangan Skenario Testing Mengubah Data Hasil Rekam Medis ....	105
Tabel 7.9	Rancangan <i>Validation Testing</i> Mengubah Data Hasil Rekam Medis..	106
Tabel 7.10	Rancangan <i>Validation Testing</i> Mengubah Data Hasil Rekam Medis Belum Lengkap.....	106
Tabel 7.11	Rancangan Skenario Testing Mengisi Data Transaksi Pembayaran Pasien.....	107
Tabel 7.12	Rancangan <i>Validation Testing</i> Mengisi Data Transaksi Pembayaran Pasien.....	107
Tabel 7.13	Rancangan <i>Validation Testing</i> Inputan Data Transaksi Pembayaran Pasien Tidak Sesuai.....	108
Tabel 7.14	Hasil Pengujian <i>Validation Testing</i> .....	108
Tabel 7.15	Kategori Masalah Kompatibilitas.....	111
Tabel 7.16	Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>Internet Explorer</i> .....	112
Tabel 7.17	Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>Microsoft Edge</i> .....	112



Tabel 7.18 Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>Firefox</i>	112
Tabel 7.19 Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>Safari</i>	113
Tabel 7.20 Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>Opera</i>	113
Tabel 7.21 Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>Chrome</i>	114
Tabel 7.22 Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>iOS</i>	114
Tabel 7.23 Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>Android</i>	114
Tabel 7.24 Tampilan Hasil Pengujian <i>Compatibility</i>	115
Tabel 7.26 Rancangan Kasus Uji UAT untuk Petugas <i>Front office</i>	118
Tabel 7.27 Rancangan Kasus Uji UAT untuk Dokter	119
Tabel 7.35 Hasil Pengujian <i>User Acceptance Testing</i> untuk Petugas Apoteker	126
Tabel 7.37 Hasil Pengujian <i>User Acceptance Testing</i> untuk Perawat	128



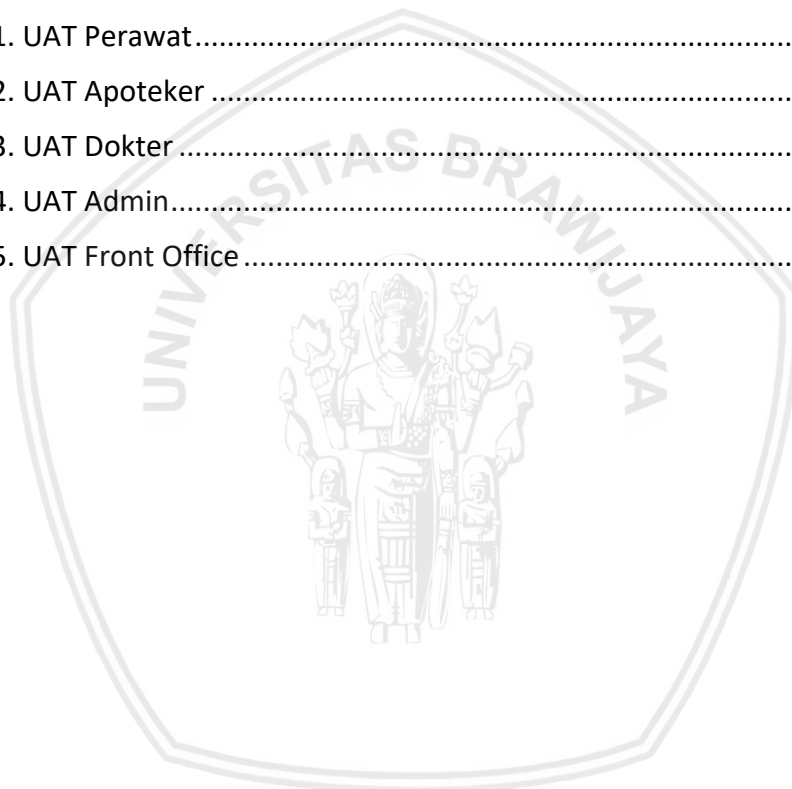
## DAFTAR GAMBAR

Gambar 3.1 Diagram Alur Metodologi.....	32
Gambar 4.1 Kodifikasi Kebutuhan Pengguna.....	50
Gambar 5.1 <i>Sequence Diagram</i> Pendaftaran Pasien .....	79
Gambar 4. 1 Proses Bisnis As-is Pendaftaran Rawat Inap.....	41
Gambar 4. 2 Proses Bisnis As-is Pemeriksaan Pasien Rutin Dokter .....	42
Gambar 4. 3 Proses Bisnis As-is Pelayanan Obat Pasien.....	42
Gambar 4. 4 Proses Bisnis As-is Pelayanan Pasien Keluar .....	43
Gambar 4. 5 Proses Bisnis To-be Pendaftaran Rawat Inap.....	44
Gambar 4. 6 Proses Bisnis To-be Pemeriksaan Pasien Rutin Dokter .....	45
Gambar 4. 7 Proses Bisnis To-be Pelayanan Obat Pasien .....	46
Gambar 4. 8 Proses Bisnis To-be Pelayanan Pasien Keluar.....	47
Gmbaar 5.2 <i>Sequence Diagram</i> Memperbarui Pemeriksaan Pasien .....	80
Gambar 6.1 Antarmuka Halaman <i>Home</i> .....	99
Gambar 6.2 Antarmuka Halaman Daftar Pasien Harian .....	100
Gambar 6.3 Antarmuka Halaman Pendaftaran Pasien .....	100
Gambar 6.4 Antarmuka Halaman Hasil Pemeriksaan Pasien.....	101
Gambar 7.1 Pengkodean Uji <i>Validation Testing</i> .....	102
Gambar 7.2 Pengkodean Uji <i>Compatibility Testing</i> .....	111
Gambar 7.3 Hasil <i>Compatibility Testing</i> .....	116
Gambar 7.4 Kode Uji <i>User Acceptance Testing</i> (UAT).....	117



## DAFTAR LAMPIRAN

Lampiran A HASIL WAWANCARA.....	135
A.1 Sesi Wawancara .....	136
LAMPIRAN B ARTEFAK .....	137
B1. Dokumen Pendaftaran .....	137
B2. Dokumen Pemeriksaan Perkembangan Medis.....	138
LAMPIRAN C DOKUMENTASI HASIL <i>USER ACCEPTANCE TESTING</i> (UAT) .....	139
C1. UAT Perawat.....	139
C2. UAT Apoteker .....	141
C3. UAT Dokter .....	143
C4. UAT Admin.....	145
C5. UAT Front Office .....	147



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Perkembangan di dalam teknologi informasi sangat berkembang begitu pesat dan memiliki peran yang sangat penting. Prosedur dalam pemrosesan data yang didasari oleh teknologi informasi dan diintegrasikan dengan prosedur manual dan juga prosedur yang lain bertujuan menghasilkan informasi yang efektif dan efisien untuk mendukung proses pengambilan keputusan manajemen disebut juga sebagai sistem Informasi. Saat ini sistem informasi mempunyai nilai strategis dan memiliki peranan yang begitu penting di dalam daya saing dan kompetensi utama dalam sebuah organisasi yang menyongsong era informasi ini.

Rumah Sakit memiliki peranan dan tujuan sebagai pelayanan kesehatan yang memiliki beberapa kegiatan yaitu seperti pelayanan pelayan gawat darurat, rawat jalan, pelayanan rawat inap, dan pelayanan rujukan yang sudah mencakup pelayanan rekam medis (AA, 2004). Satu dari beberapa unit yang menyediakan informasi yang penting untuk meningkatkan pelayanan terhadap pasien adalah Unit rawat inap. Pelayanan yang diberikan oleh unit rawat inap terdapat kegiatan pelayanan medis dan perawat seperti hasil anamnesa, diagnosa, pemeriksaan fisik, pemeriksaan penunjang, riwayat penyakit, terapi serta beberapa tindakan lain yang di dokumentasikan dalam formulir pasien, untuk itu perlu adanya upaya meningkatkan mutu dari pelayanan yang baik juga pelayanan medis, keperawatan maupun non medis.

Rumah Sakit khusus Hayunanto *Medical Center* merupakan rumah sakit khusus pusat terapi yang mefokuskan diri pada pelayanan kesehatan rehabilitasi mental dan rehabilitasi ketergantungan narkoba. Rumah sakit khusus ini memberikan penuh layanan sarana dan prasarana serta dituntut untuk memberikan pelayanan yang maksimal terhadap para pasiennya. Terdapat bagian rawat jalan, instalasi gawat darurat (IGD) dan rawat inap. Terlebih pada bagian rawat inap mempunyai peranan penting bagi suatu instansi ini sebagai penyedia jasa perawatan pasien dalam jangka waktu yang lama di dalam Rumah Sakit khusus Hayunanto *Medical Center* Malang.

Ketepatan dan kecepatan alur informasi dalam layanan kesehatan merupakan salah satu hal terpenting yang sangat mempengaruhi kesehatan dan keselamatan seseorang. Salah satu faktor utama yang membutuhkan ketepatan dan kecepatan informasi adalah tenaga medis dan tenaga administrasi. Salah satu layanan kesehatan yang membutuhkan alur informasi yang lebih baik adalah Rumah Sakit khusus Hayunanto *Medical Center* khususnya di bagian rawat inap.

Pengimplementasian proses pencatatan data rawat inap masih dilakukan pada dokumen atau formulir yang selanjutnya hanya disimpan dengan map itu membuat tidak mempercepat alur informasi di Rumah sakit ini, karena data tersebut hanya bisa diakses secara local oleh staff administrasi atau rekam medis. Setiap tenaga medis harus melakukan pengecekan manual melalui staff rumah

sakit untuk mengetahui daftar pasien dan jadwal pasien yang perlu divisite. Juga pada saat pasien menanyakan berapa besar biaya sementara yang harus dibayar, pihak rumah sakit tidak segera dapat menjawab karena harus mengambil dokumen lain seperti dokumen biaya periksan dan dokumen pemakaian obat untuk kemudian baru dihitung dengan kalkulator. Kondisi sistem lama dimana dokter mengizinkan pasien untuk pulang, kemudian keluarga pasien mengurus kewajiban pembayaran, setelah sampai dibagian pembayaran ternyata bagian pembayaran belum menyiapkan dokumen ataupun slip pembayaran dan butuh waktu lama untuk menunggu pembuatan rincian pembayaran.

Belum adanya otomatisasi pencatatan data pasien yang keluar menyebabkan status pasien belum berubah menjadi keluar, sehingga bagian pelayanan belum mengetahui kalau pasien sudah keluar, hal ini menyebabkan kesalahan informasi yang diberikan kepada pembesuk. Permasalahan ini tidak terlepas dari faktor data *redundancy* data khususnya pada pencatatan pengkodeannya, pencatatan kode yang kembar mengakibatkan kesalahan informasi yang dihasilkan contohnya nomor pasien yang sama, pasien pertama menderita A dan satu pasien lagi menderita penyakit B. Jika hal ini dibiarkan maka bisa menjadikan hasil pemeriksaan rawat inap dengan informasi yang salah. Pada informasi ataupun laporan yang dihasilkan juga akan terjadi pelaporan yang salah, dimana kode pasien yang sama akan memiliki penyakit yang bukan semestinya diderita.

Terlalu lama untuk melakukan pengurutan data sehingga berdampak pada proses pencarian data menjadi lama dan bahkan tidak diketahui karena harus menelusuri satu per satu data yang dicari. Begitu juga pada proses transaksi pemakaian obat atau periksa harus mengentri ulang data pasien, hal ini dikarenakan data pasien belum tersimpan secara rapi dalam sebuah penyimpanan. Terdapat juga faktor yang melatarbelakanginya adalah data masih belum terintegrasi, dimana untuk pencatatan identitas pasien harus ditulis ulang, yaitu didalam penulisan nomor pasien, nama pasien dan alamat pasien. Data laporan dimungkinkan juga berbeda dengan dapat yang diinputkan (inkonsistensi data), sebagai contoh penulisan nomor pasien ketika terjadi pendataan adalah P001, sedangkan dipenyusunan laporan karena belum otomatisasi hanya di tulis P01, ini sudah menimbulkan kesalahan yang sangat berarti.

Jadi dengan sistem yang sekarang dalam berlangsungnya pemrosesan data dan informasi yang memakan banyak waktu lama, dalam penerapan sistem yang manual akan sulit melakukan kontrol karena pemrosesan yang dilakukan oleh manusia maka untuk terjadinya kesalahan akan semakin besar, seperti terjadinya pengulangan pencatatan atau redundansi data. Dalam menggunakan sistem informasi diharapkan pemrosesan akan berlangsung cepat dan memudahkan pengontrolan sehingga kemungkinan terjadinya *human error* dapat ditekan.

Dari studi literatur yang penulis lakukan, terdapat penelitian yang pernah dilakukan oleh (Limbong, 2010) dengan judul Pengembangan Sistem Informasi Rawat Inap Pelayanan Penyakit Dalam Guna Mendukung Keputusan Manajemen Pelayanan Studi Kasus Rsud Dr H Soemarno Sosroadmojo Bulungan di Kalimantan

Timur, dalam penelitian ini Limbong mengembangkan sistem informasi dalam bentuk laporan rawat inap yang di butuhkan manajemen pelayanan penyakit yang datanya diambil dari catatan rekam medis kemudian diolah menjadi informasi seperti diagnosa medis, informasi penggunaan obat, dan lainnya. Berdasarkan hasil penelitian peneliti Ada perbedaan kualitas informasi sebelum dan sesudah pengembangan sistem informasi rawat inap penyakit dalam yang signifikan guna mendukung keputusan manajemen pelayanan rumah sakit tersebut.

Metode rekayasa perangkat lunak yang mengembangkan dengan mengumpulkan *best practices* yang terdapat dalam pengembangan perangkat lunak disebut juga *Rational Unified Process* (RUP). Ciri-ciri utama di dalam metode ini ialah menggunakan *use-case driven* dan pendekatan iteratif. Iteratif merupakan fase pengulangan di dalam RUP yang mempunyai beberapa fase *inception*, *elaboration*, *construction* dan *transition*. Iterasi terdapat di setiap tahap bertujuan untuk mengakomodasikan perubahan kebutuhan perangkat lunak. Pengembangan sistem informasi rawat inap ini akan dilakukan dengan beberapa tahap dan terdapat perubahan atau juga penambahan fungsi sesuai kebutuhan pihak rumah sakit khusus HMC Malang hingga dihasilkan sistem informasi rawat inap yang memiliki fungsi lengkap untuk mengatasi permasalahan. Untuk mengakomodasi kebutuhan pemangku kepentingan yang selalu berubah yang mungkin terjadi dalam proses pengembangan sistem menjadi alasan dipilihnya metode RUP ini (Kinasih, 2018) selain memudahkan menganalisa permasalahan dengan cara pendefinisian dengan baik, metode ini juga mengakomodasi perubahan kebutuhan perangkat lunak, dan dalam setiap fase RUP terdapat evaluasi bertujuan kesalahan sistem dan resiko dapat diketahui sejak dini juga management waktu yang tepat.

Berdasarkan pernyataan diatas, peneliti akan melakukan penelitian tentang pengembangan sistem informasi rawat inap, yang nantinya akan diterapkan di rumah sakit khusus HMC Malang. Judul penelitian yang dibuat oleh peneliti yakni "Pengembangan Sistem Informasi Rawat Inap dengan Metode *Rational Unified Process* Studi Kasus Rumah Sakit Khusus Hayunanto *Medical Center* Malang". Melalui penelitian ini, diharapkan dapat memudahkan petugas menginputkan data pasien baru dan memberikan pelayanan yang optimal kepada pasien.

## 1.2 Rumusan masalah

Berdasarkan latar belakang yang telah di jelaskan, rumusan masalah pada penelitian ini sebagai berikut:

1. Bagaimanakah proses bisnis layanan rawat inap yang diterapkan Pada Rumah Sakit Khusus HMC Malang saat ini?
2. Bagaimana merancang proses bisnis layanan rawat inap yang lebih baik dari proses bisnis saat ini sesuai dengan perkembangan saat ini?

3. Bagaimana mengimplementasikan perancangan Sistem Informasi Rawat Inap Pada Rumah Sakit Khusus HMC Malang dari hasil analisis proses bisnis baru menggunakan metode *Rational Unified Process*?
4. Bagaimanakah hasil pengujian implementasi Sistem Informasi Rawat Inap Pada Rumah Sakit Khusus HMC Malang dari proses bisnis yang baru?

### 1.3 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini, ialah:

1. Menganalisa proses bisnis layanan rawat inap yang diterapkan pada Rumah Sakit Khusus HMC Malang saat ini.
2. Membuat rancangan proses bisnis layanan rawat inap yang lebih baik dari proses bisnis saat ini sesuai dengan perkembangan saat ini.
3. Mengimplementasikan perancangan Sistem Informasi Rawat Inap p (Kinasih, 2018)ada Rumah Sakit Khusus HMC Malang dari hasil analisis proses bisnis baru menggunakan metode *Rational Unified Process*.
4. Menguji hasil implementasi Sistem Informasi Rawat Inap Pada Rumah Sakit Khusus HMC Malang dari proses bisnis yang baru.

### 1.4 Manfaat

Manfaat yang diharapkan didapat dari penelitian adalah sebagai berikut:

1. Bagi Penulis
  - a. Menambah ilmu dalam menganalisis kebutuhan dari pemodelan proses bisnis sistem informasi yang akan dikembangkan.
  - b. Menambah ilmu dalam perancangan untuk pengembangan sistem infromasi.
  - c. Sarana pengembangan pengetahuan terkait implementasi sistem berdasarkan rancangan sistem.
  - d. Menambah ilmu mengenai mekanisme pengujian dari sistem informasi yang dikembangkan.
2. Bagi Rumah Sakit Khusus HMC Malang
  - a. Mempermudah kinerja petugas rumah sakit dalam mengolah data dan informasi.
  - b. Memberikan pelayanan administrasi kesehatan secara terkomputerisasi sehingga menghasilkan laporan transaksi yang cepat dan akurat.
3. Bagi Pembaca
  - a. Memberikan ilmu mengenai analisis kebutuhan dan pemodelan proses bisnis.
  - b. Memberikan wawasan mengenai perancangan suatu sistem.

- c. Memberikan wawasan mengenai implementasi sistem.
- d. Memberikan wawasan terkait pengujian sistem.

## 1.5 Batasan masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Perancangan sistem yang dibuat merupakan sistem informasi rawat inap pada rumah sakit khusus HMC Malang.
2. Data yang diperoleh merupakan hasil wawancara langsung dengan pihak Rawat Inap Pada Rumah Sakit Khusus HMC Malang.
3. Perancangan sistem yang dilakukan adalah perancangan sistem berbasis *website*.
4. Penelitian ini dilakukan mulai analisis kebutuhan, perancangan, implementasi, hingga pengujian sistem.

## 1.6 Sistematika pembahasan

### **BAB I      Pendahuluan**

Memuat latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

### **BAB II     Landasan Kepustakaan**

Menjelaskan tentang teori dasar dan teori pendukung dalam pengembangan sistem informasi rawat inap pekerjaan berbasis web dan menjelaskan secara singkat perihal profil dari kantor terkait.

### **BAB III    Metode Penelitian**

Berisi tentang penjelasan metode dan tahap-tahap yang akan dilakukan dalam penelitian.

### **BAB IV    Analisis Kebutuhan**

Menjelaskan analisis kebutuhan sistem informasi rawat inap Rumah Sakit Khusus HMC Malang yang akan menjadi dasar dalam perancangan sistem.

### **BAB V     Perancangan**

Membahas tentang perancangan sistem untuk sistem informasi rawat inap pada Rumah Sakit Khusus HMC Malang yang akan dibangun.

### **BAB VI    Implementasi**

Membahas tentang implementasi sistem informasi rawat inap pada Rumah Sakit Khusus HMC Malang sesuai dengan perancangan sistem yang sudah dibuat.

**BAB VII Pengujian**

Membahas terkait proses dan hasil dari pengujian sistem informasi rawat inap berbasis *web* pada Rumah Sakit Khusus HMC Malang.

**BAB VIII Penutup**

Memuat kesimpulan dan saran dari hasil penelitian yang dilakukan.



## BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab dua ini membahas kajian pustaka dan dasar teori yang diperlukan untuk penelitian. Kajian pustaka menjelaskan penelitian yang telah ada dan yang diusulkan. Dasar teori menjelaskan teori yang diperlukan untuk menyusun penelitian yang diusulkan

### 2.1 Kajian Pustaka

Dari studi literatur yang telah penulis lakukan, ada penelitian serupa yang pernah dikembangkan oleh peneliti sebelumnya yaitu (Krol & Reich, 1999) dalam penelitian yang berjudul *“Object-Oriented Analysis and Design of a Health Care Management Information System”* menjelaskan bahwa *Object-Oriented Analysis and Design (OOAD)*. Model berorientasi objek penuh menggabungkan tiga pandangan dari suatu sistem dalam bentuk objek, dinamis, dan model fungsional. Model objek menggambarkan struktur objek dalam suatu sistem (identitas mereka, hubungan, atribut, dan operasi) dalam bentuk struktur hirarkis yang berisi kelas objek. Kelas menentukan nilai atribut yang dibawa oleh setiap instance objek dan operasi yang dilakukan setiap objek atau mengalami. Kelas dapat dibagi menjadi subclass, yang masing-masing mewarisi atribut dan operasi superclassnya. Pada saat yang sama, sesuai dengan konsep polymorphism, operasi yang sama dapat didefinisikan secara berbeda di kelas yang berbeda. Model dinamis mewakili urutan operasi tepat waktu. Secara grafis, itu direpresentasikan sebagai kumpulan diagram keadaan untuk kelas objek dalam sistem. Setiap state diagram menunjukkan urutan keadaan dan peristiwa yang diizinkan dalam sistem untuk satu kelas objek. Diagram fungsional merepresentasikan transformasi data dalam suatu sistem dan direpresentasikan dengan diagram aliran data. Model fungsional menjelaskan fungsi yang dipanggil oleh operasi dalam model objek dan tindakan dalam model dinamis. Model fungsional juga menunjukkan kendala pada nilai-nilai objek. Hasil penelitian tersebut menjadi dasar pemilihan teknik OOAD sebagai panduan untuk melaksanakan kegiatan analisis persyaratan dan pemodelan sistem informasi kesehatan yang akan dikembangkan dalam penelitian ini.

Referensi kedua dari jurnal yang berjudul *“The Application of the Unified Modeling Language in Object-Oriented Analysis of Healthcare Information”* Vinod Aggarwal (2002). Dijelaskan dalam *Object-Oriented Analysis* adalah teknik yang sesuai dalam pengembangan sistem informasi kesehatan dimana OOA yang di dalam penelitian tersebut mempunyai representasi persyaratan sistem informasi dari berbagai sudut pandang, lalu representasi persyaratan tersebut dapat ditampilkan dengan UML. Banyak manfaat UML dari hasil penelitian tersebut dalam memvisualisasi, membangun, mendokumentasikan, dan mengkomunikasikan hasil dari aktivitas OOA. Semua itu akan menjadi dasar dalam pemanfaatan UML sebagai pemodelan pada proses pengembangan sistem informasi kesehatan untuk memodelkan aktivitas, interaksi objek, dan memodelkan objek. Lalu hasil dari



pemodelan bisa menjadi masukan dalam merancang *database* juga bisa menjadi panduan implementasi sistem informasi kesehatan.

Untuk referensi ketiga diambil dari jurnal yang berjudul “*System modeling of vehicle management based on RUP and UML*” (Lan & Xiujian, 2012). Penelitian ini menjelaskan kombinasi RUP dan UML dalam membangun suatu sistem informasi manajemen kendaraan dengan menggunakan fase pada RUP. Khususnya di 3 proses alurkerja pada RUP yaitu *business modelling, requirements analysis dan design*, diwaktu yang sama menggunakan UML bertujuan memodelkan sistem yaitu *usecase diagram, class diagram, class diagram dan sequence diagram*. paper ini menggunakan UML untuk membangun pemodelan sistem. Dalam proses pemodelan sistem, pembentukan model dapat berupa derivasi dan konversi yang sesuai. Panduan pengembangan perangkat lunak gabungan RUP dengan UML, telah meningkatkan kualitas dan efisiensi pengembangan perangkat lunak dan pemeliharaan.

Praktek menunjukkan bahwa RUP memberikan kriteria yang diperlukan, template dan panduan alat dalam aspek sebagai use case driven, pengembangan iterasi, pemodelan visual, arsitektur-sentris, dan sebagainya. Ini untuk setiap pengembang di semua kegiatan pengembangan penting. Bahasa pemodelan yang sangat baik, dalam penggunaan UML dapat membantu seorang pengembang untuk memiliki wawasan yang jelas dalam keseluruhan sistem disebut dengan UML. Sehingga kita bisa membuat persiapan penuh untuk pengkodean nanti melalui pembuatan berbagai model sistem menggunakan UML. Menggabungkan RUP dengan UML untuk memandu pengembangan perangkat lunak dapat mengurangi risiko produk, menstandarisasi proses manajemen dan pengembangan, dan meningkatkan kualitas, efisiensi, dan pemeliharaan pengembangan perangkat lunak. Jadi kesimpulan yang didapat bahwa menggunakan RUP dengan UML dalam mengembangkan perangkat lunak dapat mengurangi risiko produk dalam mengembangkan proses untuk meningkatkan kualitas, efisiensi, juga standarisasi manajemen dan pemeliharaan pengembangan perangkat lunak.

Pada penelitian yang telah dilakukan oleh Mina Zaminkar dan Mohammad R. Reshadinezhad yaitu membandingkan 2 model *software engineering process* yang berjudul “*A Comparison Between Two Software Engineering Processes, RUP and Waterfall Models*” menjelaskan bahwa membandingkan Model RUP dan Waterfall yang memiliki kelebihan dan kekurangan masing-masing. Kelebihan dari RUP dibandingkan dengan Waterfall Models yaitu model rup adalah metodologi yang lengkap dengan penekanan pada dokumentasi yang akurat, secara proaktif mampu menyelesaikan risiko proyek yang terkait dengan persyaratan klien yang berkembang yang memerlukan manajemen permintaan perubahan yang hati-hati, dibutuhkan waktu yang sedikit untuk integrasi karena proses integrasi berlangsung sepanjang siklus pengembangan perangkat lunak. Dibutuhkan waktu yang sebentar untuk pengembangan karena penggunaan komponen berulang-ulang, Ada tutorial dan pelatihan online yang tersedia bagi pengguna untuk menggunakan proses ini Kesimpulan yang dihasilkan yaitu dua model RUP dan waterfall untuk rekayasa perangkat lunak dibandingkan. Setiap model diperkenalkan secara terpisah dan kelebihan dan kekurangan masing-masing model diilustrasikan secara lengkap. Untuk kedepannya, model siklus pengembangan perangkat lunak lainnya seperti spiral dan inkremental harus

dipertimbangkan dan juga disimulasikan, memungkinkan manajer proyek untuk memilih metodologi pengembangan perangkat lunak terbaik.

Referensi yang terakhir diambil dari jurnal berjudul “*A Review of RUP (Rational Unified Process)*” (Anwar,2014) yang berisi yaitu menjelaskan definisi dari RUP sejarahnya dan juga langkah-langkah pengerjaan, juga keuntungan dan kerugian RUP. Kesimpulan dari jurnal tersebut, *software* dari pengembangan tersebut sering atau banyak digunakan dalam dunia industri disebabkan dalam penerapannya telah di uji beberapa kali dalam beberapa proyek dan hasilnya terbukti berhasil. Sebuah platform dimana proses pengembangan perangkat lunaknya *delivers practises* dan arsitektur yang dapat dikonfigurasi disebut juga RUP. Pengembang dapat memilih dan menerapkan proses komponen di setiap tahapannya di dalam proyek. Dengan menggunakan iteratif dengan pengujian yang konstan,lalu umpan balik berkelanjutan dari pemangku kepentingan, dan fokus pengembangan perangkat lunak, RUP akan dapat meningkatkan kualitas dari perangkat lunak itu sendiri.



Tabel 2. 1 Literatur Review

No	Nama Penulis + Nama Jurnal +Tahun	Nama Paper	Tujuan Penelitian	Metode Penelitian	Hasil dan Kesimpulan atau Saran
1	Ashraf Anwar (2014). International Journal of Software Engineering (IJSE), Volume (5) : Issue (2) : 2014	"A Review of RUP ( <i>Rational Unified Process</i> )"	<i>software</i> dari pengembangan tersebut sering atau banyak digunakan dalam dunia industri disebabkan dalam penerapannya telah di uji beberapa kali dalam beberapa proyek dan hasilnya terbukti berhasil. Sebuah platform dimana proses pengembangan perangkat lunaknya <i>delivers practises</i> dan arsitektur yang dapat dikonfigurasi disebut juga RUP.	-	Pengembang dapat memilih dan menerapkan proses komponen di setiap tahapannya di dalam proyek. Dengan menggunakan iteratif dengan pengujian yang konstan,lalu umpan balik berkelanjutan dari pemangku kepentingan, dan fokus pengembangan perangkat lunak, RUP akan dapat meningkatkan kualitas dari perangkat lunak itu sendiri.
2	Zamika, Mina dan Reshadinezhad ,Mohammad R. .2013. International Journal of Engineering Research &	"A <i>Comparison Between Two Software Engineering Processes, RUP And Waterfall Models</i> "	Perhatian utama dalam penelitian ini adalah untuk mewakili model pengembangan perangkat lunak yaitu RUP, waterfall dan membuat perbandingan di antara keduanya untuk menunjukkan kelebihan dan	-	Kesimpulan yang dihasilkan yaitu dua model RUP dan waterfall untuk rekayasa perangkat lunak dibandingkan. Setiap model diperkenalkan secara terpisah dan kelebihan dan kekurangan masing-masing model diilustrasikan secara lengkap. Untuk kedepannya, model siklus pengembangan perangkat lunak lainnya seperti spiral dan inkremental harus dipertimbangkan dan juga disimulasikan, memungkinkan manajer proyek untuk

No	Nama Penulis + Nama Jurnal +Tahun	Nama Paper	Tujuan Penelitian	Metode Penelitian	Hasil dan Kesimpulan atau Saran
	<i>Technology</i> (IJERT) . <i>University of Isfahan, Iran.</i>		kekurangan masing-masing model.		memilih metodologi pengembangan perangkat lunak terbaik.
3	Marina Krol dan David L. Reich (1999). <i>Journal of Medical Systems, Vol. 23, No. 2, 1999</i>	<i>Object-Oriented Analysis and Design of a Health Care Management Information System”</i> menjelaskan bahwa <i>Object-Oriented Analysis and Design (OOAD)</i>	Keuntungan dalam pengembangan ini merupakan pengembangan dalam sistem komputer yang kompleks. OOAD terbukti diterima oleh komunitas informatika medis dalam mengembangkan sistem kesehatan, terutama dalam informasi rumah sakit. berfokus dalam pemodelan objek, pemodelan urutan operasi dalam masing-masing objek, dan pemodelan fungsi sistem dengan pendekatan OOAD.		Hasil penelitian tersebut menjadi dasar pemilihan teknik OOAD sebagai panduan untuk melaksanakan kegiatan analisis persyaratan dan pemodelan sistem informasi kesehatan yang akan dikembangkan dalam penelitian ini.
4	Vinod Aggarwal (2002). <i>Journal</i>	<i>“The Application of the Unified</i>	Menguraikan proses analisis <i>object oriented (OOA)</i> menggunakan UML dan		keuntungan dari Unified Modeling Language (UML), sebuah standar pemodelan objek nonproprietary, dalam menetapkan, memvisualisasikan, membangun,

No	Nama Penulis + Nama Jurnal +Tahun	Nama Paper	Tujuan Penelitian	Metode Penelitian	Hasil dan Kesimpulan atau Saran
	of Medical Systems, Vol. 26, No. 5, October 2002	<i>Modeling Language in Object-Oriented Analysis of Healthcare Information</i>	menggambarkan hal ini dengan contoh-contoh perawatan kesehatan untuk menunjukkan kepraktisan penerapan UML oleh personel perawatan kesehatan untuk masalah sistem informasi dunia nyata. Serta menjelaskan efek menguntungkan dari Unified Modeling Language (UML) dalam menetapkan, memvisualisasikan, membangun, mendokumentasikan, dan mengkomunikasikan model sistem informasi kesehatan dari perspektif pengguna.		mendokumentasikan, dan mengkomunikasikan model sistem informasi perawatan kesehatan dari perspektif pengguna. Penulis menguraikan proses analisis objectoriented (OOA) menggunakan UML dan menggambarkan hal ini dengan contoh-contoh perawatan kesehatan untuk menunjukkan kepraktisan penerapan UML oleh personel perawatan kesehatan untuk masalah sistem informasi dunia nyata.
5	(Jin & Liang, 2012) .IEEE(2012)	System modeling of vehicle management based on	Pada penelitian ini membahas kombinasi RUP dan UML dalam pembangunan sistem informasi manajemen kendaraan. Dengan mengikuti		Penulis memberikan kesimpulan bahwa Memadukan RUP dengan UML dalam pengembangan perangkat lunak dapat mengurangi risiko produk, standarisasi manajemen dan pengembangan proses, dan

No	Nama Penulis + Nama Jurnal + Tahun	Nama Paper	Tujuan Penelitian	Metode Penelitian	Hasil dan Kesimpulan atau Saran
		RUP and UML”	siklus fase pada RUP. terutama 3 proses alurkerja pada RUP yaitu <i>business modelling, requirements analysis dan design</i> , diwaktu bersamaan menggunakan UML untuk momodelkan sistem diantaranya <i>usecase diagram, class diagram, class diagram dan sequence diagram</i> .		meningkatkan kualitas, efisiensi, dan pemeliharaan pengembangan perangkat lunak.

Tabel 2.1 memberikan *review* dari literatur yang telah dikumpulkan oleh peneliti dengan memberikan judul jurnal, tahun, tujuan penelitian, metode yang digunakan, dan hasil kesimpulan atau saran.



## 2.2 Dasar Teori

### 2.2.1 Rumah Sakit Khusus Hayunanto *Medical Center*

Hayunanto *Medical Center* merupakan rumah sakit khusus untuk terapi mental dan narkoba. Rumah sakit Hayunanto *Medical Center* beralamat di Jl. Raya Sengkaling, Mulyoagung, Dau, Malang, Jawa Timur. Fokus pelayanan rumah sakit ini adalah pelayanan yang paripurna mencakup tindakan promotif, preventif, kuratif dan rehabilitatif untuk kesehatan mental. Sasaran rumah sakit khusus HMC adalah pasien dengan gangguan kesehatan mental, keluarga dan masyarakat luas untuk membantu menciptakan suasana yang kondusif bagi peningkatan kualitas kesehatan mental. Didirikan pada tanggal 25 Februari 2010 dibawah Naungan PT. Hayunanto Trisna Mukti.

Institusi rumah sakit khusus HMC ini berdiri di tengah stigma negative masyarakat yang masih menganggap tabu terhadap kesehatan mental, namun fakta di lapangan, sebenarnya kebutuhan masyarakat tersebut masih sangat tinggi terhadap layanan kesehatan mental. HMC hadir sebagai pengubah *mindset* masyarakat bahwa penderita gangguan mental bukanlah untuk dijauhi, diasingkan, bahkan dipasung, namun untuk justru untuk dirangkul, dan didorong untuk mendapatkan layanan kesehatan yang terbaik. Dalam menjawab kebutuhan masyarakat sekitar yang mengharapkan layanan kesehatan yang baik ada disekitar lingkungan mereka kini HMC berusaha mewujudkan dengan menyediakan layanan psikiatri, (jiwa/mental) untuk bisa menerima dan melayani peserta BPJS.

### 2.2.2 Visi Rumah Sakit Khusus Hayunanto *Medical Center*

Menjadi salah satu Rumah Sakit rujukan pertama untuk terapi narkoba dan kesehatan mental

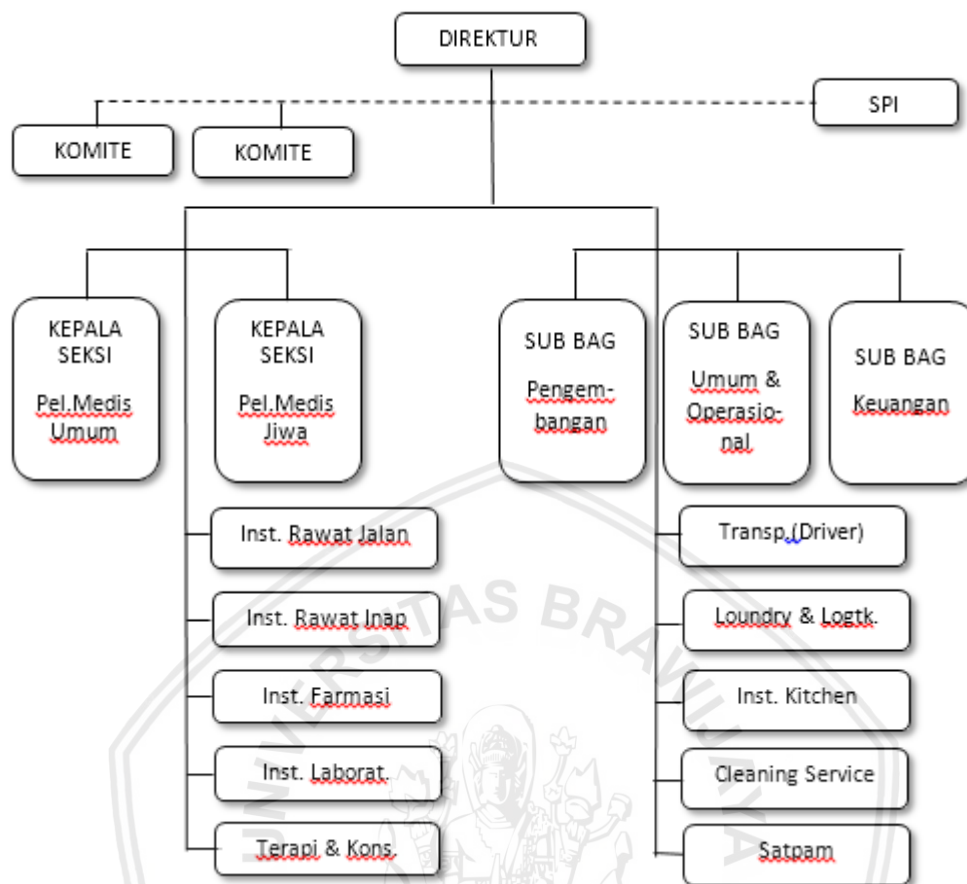
### 2.2.3 Misi Rumah Sakit Khusus Hayunanto *Medical Center*

Untuk mencapai Rumah Sakit Khusus pilihan masyarakat, ditempuh melalui Misi sebagai berikut:

1. Peningkatan kayanan kesehatan jiwa dan terapi narkoba yang berkualitas
2. Komprehensif dan terjangkau melalui pemenuhan kompetensi para professional
3. Sarana prasarana yang mendukung
4. Manajemnen cerdas dan inovatif
5. Kerjasama lintas sektoral dalam mencapai kepuasan pelanggan secara berkelanjutan

## 2.3 Struktur Organisasi

Struktur Organisasi pada Rumah Sakit Khusus sebagai berikut:



## 2.4 Rawat Inap

Menurut KepMendKes RI no 71/MENKES/2013. Rawat Inap Tingkat Pertama merupakan pelayanan kesehatan perorangan yang bersifat non spesialisik dan dilakukan di fasilitas kesehatan tingkat pertama untuk keperluan observasi, perawatan, diagnosa, pengobatan, dan atau pelayanan medis lainnya, dimana peserta dan anggota keluarganya dirawat paling singkat selama 1 hari (Menkes, 2008).

Di rawat inap Rumah Sakit Khusus Hayunanto *Medical Center* Malang memiliki tindakan khusus yang berbeda dengan rumah sakit lainnya karena rumah sakit ini berfokus pada terapi mental dan narkoba. Terdapat keistimewaan yang dimiliki pada saat di rawat inap yaitu adanya kegiatan terapi motorik, lalu terdapat layanan rehabilitasi narkoba, juga ada layanan housekeeping dan antar jemput pasien. Di rawat inap ini terdapat tenaga medis professional yaitu seperti dokter yang berjumlah 5, perawat 10 orang dan yang tidak hanya merawat tetapi selalu memberikan motivasi terhadap pasien untuk ingin sembuh dan yang lebih penting lagi terdapat kenyamanan privasi dari pihak pasien sendiri. Adapun Fasilitas rawat inap di Rumah Sakit Khusus Hayunanto *Medical Center* Malang berkapasitas 44





tempat tidur yang terdiri dari berbagai kelas perawatan mulai dari Kelas 1 (VIP), Kelas 2 sampai dengan kelas 3.

## 2.5 Sistem Informasi

Sistem informasi bisa didefinisikan sebagai gabungan dari orang, perangkat lunak, perangkat keras, sumber data, jaringan komunikasi, prosedur dan kebijakan yang digunakan untuk menyimpan, mengambil, mengubah serta menyebarluaskan suatu informasi kedalam organisasi (O'brien, 2008).

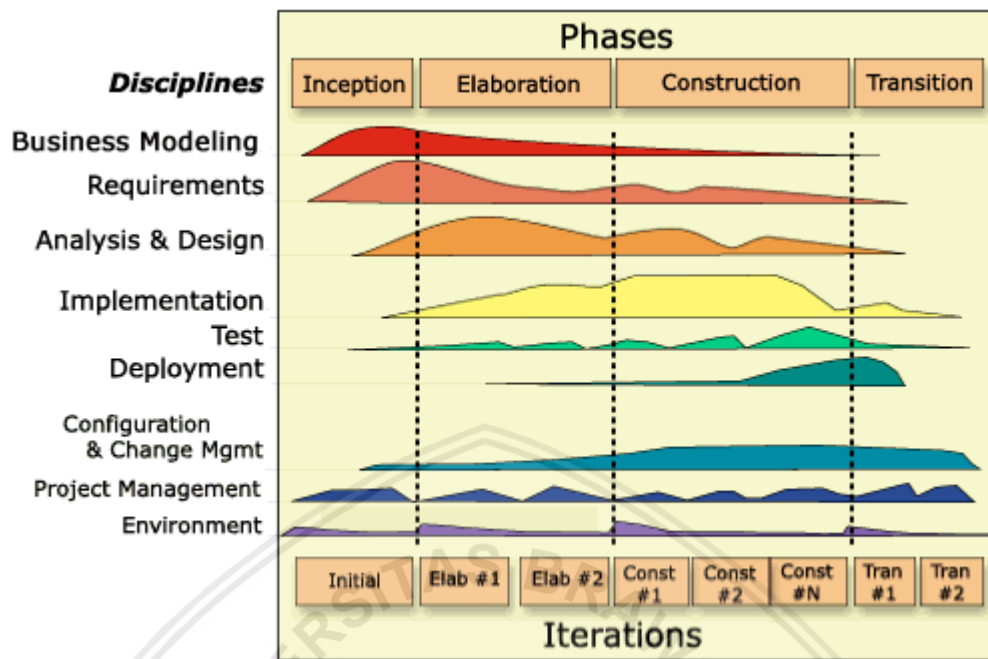
Sistem informasi didefinisikan juga sebagai kombinasi dari perangkat lunak, perangkat keras dan juga jaringan telekomunikasi yang dibangun lalu dipakai oleh orang untuk pengumpulan, pembuatan dan pendistribusian data yang berguna dalam suatu pengaturan organisasi. Perangkat lunak mengacu pada rangkaian program yang dapat memberitahu komputer dalam melakukan tugas tertentu. Perangkat lunak mengacu pada alat-alat pada komputer fisik, misalnya keyboard dan monitor. Sedangkan jaringan telekomunikasi adalah serangkaian komputer yang terhubung secara bersama-sama dengan alat-alat komunikasi (Valacich, 2010).

Sehingga dari pengertian tersebut bisa disimpulkan bahwa sistem informasi merupakan suatu sistem yang bisa mengolah sebuah *input* data dan menghasilkan output berupa informasi yang bisa digunakan dalam membantu organisasi dalam mencapai tujuannya.

## 2.6 Rational Unified Process (RUP)

Proses Rekayasa Perangkat Lunak (*Software Engineering Process*) disebut *Rational Unified Process* (RUP). RUP menyediakan pendekatan disiplin untuk menetapkan tugas dan tanggung jawab dalam pengembangan organisasi. Tujuannya adalah untuk memastikan produksi perangkat lunak berkualitas tinggi yang memenuhi kebutuhan pengguna dalam masalah prediksi dan anggaran. RUP juga merupakan kerangka proses yang dapat diadaptasi dan diperluas untuk memenuhi kebutuhan organisasi (Kroll, 2003).

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Gambar dibawah ini dapat dilihat bahwa RUP memiliki 2 dimensi yaitu pada gambar 2.3:



**Gambar 2.1** Fase RUP

Sumber (IBM, 2007)

- a. Dimensi pertama digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari phase selanjutnya. Setiap phase dapat berdiri dari satu beberapa iterasi. Dimensi ini terdiri atas *Inception*, *Elaboration*, *Construction*, dan *Transition*.
- b. Dimensi kedua digambarkan secara vertical. Dimensi mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing*, *what*, *how* dan *when*. Dimensi ini terdiri atas *Business Modelling*, *Requirement*, *Analysis and Design*, *Implementation*, *Test*, *Deployment*, *Configuration dan change Management*, *Project Management*, *Environtment*.

RUP mendefinisikan struktur yang baik untuk alur hidup proyek perangkat lunak. Dalam Metode RUP terdapat beberapa Fase yaitu:

### 2.6.1 Inception

Fase ini pengembang mendefinisikan batasan kegiatan dengan memodelkan proses bisnis yang dibutuhkan, dan melakukan analisis kebutuhan user, dan melakukan perancangan awal perangkat lunak (perancangan

arsitektural dan use case). Adapun kegiatan di fase ini yaitu merumuskan ruang lingkup proyek, Perencanaan dan mempersiapkan kasus bisnis, Mempersiapkan lingkungan untuk proyek.

### 2.6.1.1 Analisis Proses Bisnis

Pada analisis proses bisnis dilakukan pemetaan terhadap peroses bisnis yang berlaku (*as-is*) pada organisasi. Dalam melakukan analisis proses bisnis *as-is* pertama-tama mengumpulkan informasi berkaitan dengan *stakeholder* yang terlibat dalam suatu proses kemudian mengumpulkan informasi mengenai kondisi proses bisnis yang sedang berjalan (Brandenburg, 2017). Selanjutnya informasi proses bisnis yang didapatkan dimodelkan kedalam bentuk BPMN. Setelah itu informasi yang diperoleh berkaitan dengan masalah yang ada dalam proses bisnis yang sedang berjalan dilakukan analisis agar ada solusi untuk dilakukan perbaikan. Berikut Tabel 2.7 cara untuk melakukan analisis permasalahan.

**Tabel 2. 2 Analisis Permasalahan**

Masalah	[Deskripsi masalah]
Mempengaruhi	[Pemangku kepentingan yang terpengaruh oleh masalah]
Dampak	[Dampak dari masalah]
Solusi	[Solusi beserta manfaatnya]

Sumber : (Ramadhani, 2015)

Setelah melakukan analisis permasalahan kemudian diperlukan pemodelan usulan proses bisnis (*to-be*) sebagai gambaran atau penjelasan mengenai perbaikan atau usulan terhadap permasalahan yang terdapat pada organisasi, apakah sudah memenuhi kebutuhan dari *stakeholder* yang terlibat dalam proses bisnis tersebut (Brandenburg, 2017).

BPMN (*Business Process Model Notation*) merupakan metodologi yang digunakan untuk pemodelan proses bisnis yang menggambarkan desain sistem yang kompleks dari berbagai sudut pandang. BPMN memuat notasi yang memudahkan untuk digunakan dan dimengerti oleh pihak-pihak yang terlibat dari berbagai tingkatan manajemen dalam proses bisnis yang membantu dalam pengambilan keputusan. BPMN memiliki kemampuan untuk memodelkan aliran pesan secara sekuensial dari awal sampai akhir. Di dalam BPMN terdapat empat kategori dari elemen-elemennya yang menunjang pembuatan notasi pemodelan bisnis yaitu *Flow Object*, *Connections*, *Swimlanes*, dan *Artifacts*.

### Flow Object




Flow Object terdiri dari tiga elemen yaitu *Event*, *Activity*, dan *Gateway*.

#### 1. Event

Event merupakan sesuatu yang terjadi dan memberikan pengaruh dalam proses bisnis yang dapat berasal dari dalam dan luar suatu proses. Event terdiri dari start event, intermediate event, dan end event. Dalam suatu

proses bisnis akan selalu memiliki start dan end event. Berikut penjelasan lebih lanjut mengenai simbol event yang ditunjukkan pada tabel 2.3.

**Tabel 2. 3 Simbol Event**

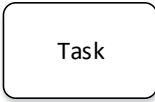
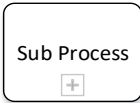
No.	Nama	Simbol	Keterangan
1	<i>Start</i>		Memulai suatu proses bisnis.
2	<i>Intermediate</i>		Terjadi diantara awal dan akhir proses bisnis yang mempengaruhi alur proses bisnis.
3	<i>End</i>		Akhir dari suatu proses bisnis.

Pada Tabel 2.3 menunjukkan symbol event dimana Event ini mempengaruhi alur proses dan biasanya memiliki sebuah tindakan atau hasil. Simbolnya berupa berupa lingkaran. Event ini sesuatu yang terjadi selama rangkaian dari sebuah business proses.

## 2. Activity

*Activity* merupakan elemen yang mendeskripsikan aktivitas yang dilakukan dalam suatu proses bisnis. *Activity* terdiri dari *Task* dan *Sub Process*. Berikut penjelasan lebih lanjut mengenai simbol *activity* yang ditunjukkan pada tabel 2.2.

**Tabel 2. 4 Simbol Activity**

No.	Nama	Simbol	Keterangan
1	<i>Task</i>		Aktivitas yang dilakukan pada proses bisnis.
2	<i>Sub Process</i>		Aktivitas majemuk di dalam proses bisnis yang dapat dijelaskan dengan lebih detail.







Pada Tabel 2.4 menunjukkan symbol activity digunakan untuk mewakili berbagai makna dalam kehidupan sehari-hari. Aktivitas dianggap mencakup berbagai kegiatan yang dapat diselesaikan dalam waktu 5 menit, satu minggu atau lebih. Biasanya simbolnya berupa persegi dengan ujung sisi yang melengkung.

### 3. Gateway

Gateway merupakan elemen yang mengendalikan alur dari sebuah proses di dalam proses bisnis. Gateway terdiri dari *exclusive*, *inclusive*, dan *complex*. Berikut penjelasan lebih lanjut mengenai simbol gateway yang ditunjukkan pada tabel 2.5.

**Tabel 2. 5 Simbol Gateway**

No.	Nama	Simbol	Keterangan
1	<i>Exclusive</i>		Sebagai proses divergen untuk membuat jalur alternative proses yang memungkinkan satu proses yang dipilih dan sebagai proses konvergen untuk menggabungkan jalur alternative.
2	<i>Parallel</i>		Proses yang dijalankan secara bersamaan.
3	<i>Inclusive</i>		Proses yang dipecah menjadi beberapa jalur proses.
4	<i>Complex</i>		Alur proses yang kompleks dalam sebuah proses bisnis.

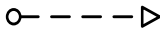


Pada Tabel 2.5 menunjukkan symbol gateway. Dalam BPMN , Gateway dapat mendefinisikan semua tindakan Arus Urutan Proses Bisnis. Sebuah Gateway kadang-kadang memainkan salah satu dari dua peran, dan kadang-kadang bermain baik pada waktu yang sama. Bentuknya berupa belah ketupat dengan berisi symbol bermacam didalamnya.



### Connections

*Connections* merupakan elemen yang menghubungkan antar *flow object* di dalam proses bisnis. *Connections* terdiri dari *sequence flow*, *association*, dan *message flow*. Berikut adalah penjelasan mengenai simbol *connections* yang ditunjukkan pada tabel 2.6.

Tabel 2. 6 Simbol *Connections*


No.	Nama	Simbol	Keterangan
1	<i>Sequence Flow</i>		Menunjukkan urutan aktivitas dalam sebuah proses bisnis.
2	<i>Assosiation</i>		Menunjukkan hubungan antara teks, data, <i>flow object</i> , atau artefak dalam proses bisnis.
3	<i>Message Flow</i>		Alur pesan antara aktivitas yang dapat saling berkiriman dan menerima pesan satu sama lain.


Pada Tabel 2.6 menunjukkan symbol connections. Dimana symbol ini merupakan penghubung antara *flow object* dengan *flow object* lain seperti menghubungkan event ke *gateway* atau pun sebaliknya. Bentuknya berupa garis lurus adapun terdapat garis putus dan berupa titik .

### Swimlanes

*Swimlanes* adalah tempat/wadah aktivitas di dalam proses bisnis yang dapat membagi set aktivitas dengan aktivitas lain. *Swimlanes* terdiri dari *pool* dan *lanes*. Berikut penjelasan lebih lanjut mengenai simbol *swimlanes* yang ditunjukkan pada tabel 2.7.

Tabel 2. 7 Simbol *Swimlanes*

No.	Nama	Simbol	Keterangan
1	<i>Pool</i>		Wadah yang berisi <i>flow objects</i> , <i>connections</i> , maupun artefak.



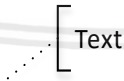

2	Lane		Bagian mendetail dari <i>pool</i> .
---	------	---	-------------------------------------

Pada Tabel 2.7 menunjukkan symbol *swimlanes* yaitu pengelompokan dari beberapa model elemen. *Swimlines* digunakan untuk memisahkan dan mengatur kegiatan oleh peserta sehingga kita secara intuitif dapat memahami siapa yang bertanggung jawab untuk setiap event. *Swimlines* ada 2 jenis yaitu *pools* dan *Lanes*.

### Artifacts

*Artifacts* menggambarkan sesuatu yang berada di luar sebuah proses. *Artifacts* dapat mendeskripsikan dan mengelola data atau catatan dari sebuah proses. *Artifacts* terdiri dari *data object*, *data store*, *annotation*, dan *group*. Berikut penjelasan lebih lanjut mengenai simbol *artifacts* yang ditunjukkan pada tabel 2.8.

**Tabel 2. 8 Simbol Artifact**

No.	Nama	Simbol	Keterangan
1	<i>Data Object</i>		Data yang dihasilkan dan dibutuhkan oleh sebuah aktivitas di dalam proses bisnis.
2	<i>Data Store</i>		Tempat membaca dan menulis data.
3	<i>Annotation</i>		Informasi tambahan untuk yang menjelaskan diagram kepada pembaca diagram.
4	<i>Group</i>		Mengelompokkan aktivitas dalam proses bisnis.

Pada Tabel 2.8 menunjukkan symbol *artifacts* dimana elemen yang digunakan untuk memberikan informasi tambahan dari sebuah proses. Bentuk dan penggunaan *artifacts* itu bermacam-macam dan bisa lebih luas tergantung dari standar pengertian BPMN yang digunakan.

### 2.6.1.2 Analisis Kebutuhan Fungsional dan Non Fungsional

Kebutuhan fungsional merupakan pernyataan layanan yang harus disediakan dalam sistem, bagaimana reaksi sistem dengan masukan tertentu dan bagaimana perilaku sistem dengan situasi tertentu. Pada beberapa kasus lain di persyaratan fungsional juga bisa secara eksplisit menyatakan hal yang tidak dilakukan oleh sistem. Sedangkan kebutuhan non fungsional merupakan batasan dalam fungsi atau layanan yang disediakan oleh sistem seperti waktu, proses dan batasan yang diberlakukan standar (Sommerville, 2011).

### 2.6.2 Elaboration


Pada fase ini dilakukan perancangan perangkat lunak, perancangan pada tahap ini lebih memperinci dari kebutuhan sistem yang telah di lakukan pada fase inception. Perancangan pada tahap ini menentukan arsitektur awal dari perangkat lunak berdasarkan usecase dan semua diagram perancangan yang mengacu pada usecase sistem. Kegiatan yang dilakukan di fase ini diantaranya menganalisa berbagai resiko dan persyaratan, menetapkan batasan perancangan.

#### **Use case Diagram**

Fungsionalitas yang diharapkan di dalam sebuah sistem, dengan diutamakan “apa” yang diperbuat sistem, dan bukan “bagaimana” disebut dengan *Use Case Diagram*. *Use case* merupakan sebuah representasi dari interaksi antara actor dan sistem misalnya melakukan pekerjaan tertentu seperti login ke sistem atau menambah list barang yang dibutuhkan dan lain-lainnya. Di *use case* terdapat *actor* yang merupakan entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan tertentu. *Use case* diagram sangatlah bermanfaat dalam menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan klien dan merancang *test case* untuk semua *feature* pada sistem.






Dalam membuat *Use Case Diagram* perlu memperhatikan simbol-simbol yang ada pada Table 2.9, disediakan semua simbol yang bisa digunakan dalam pembuatan *Use Case Diagram*.

**Tabel 2. 9 Simbol Use Case Diagram**

No	Nama	Gambar	Fungsi
1	<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>



Tabel 2. 9 Simbol Use Case Diagram

No	Nama	Gambar	Fungsi
2	Actor		Orang, proses, atau sistem lain yang bisa berinteraksi dengan sistem yang akan dibangun.
3	Association		Komunikasi yang berpartisipasi dalam use case, terdapat antara hubungan aktor dan use case
4	Extend		Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan
5	Generalization		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi salah satu usecase merupakan fungsi yang lebih umum dari lainnya
6	Include		Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat jalannya use case tersebut

Sumber: (A.S & Shalahudin, 2015)

Pada table 2.9 menunjukkan symbol-simbol dalam use case untuk penamaan pada diagram use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua yang utama di dalam sebuah use case diagram yaitu aktor dan use case. Aktor di gambarkan orang, proses, atau sistem yang berinteraksi dengan aplikasi yang akan dibuat di luar sistem informasi. Simbol actor adalah gambar orang, tapi aktor belum tentu merupakan orang. Use case merupakan fungsionalitas yang disediakan sistem sebagai berbagai unit yang saling bertukar informasi antar unit atau aktor.

### Use Case Scenario

Penjelasan secara tekstual dari sekumpulan skenario interaksi disebut juga Use case scenario. Setiap skenario mendeskripsikan urutan aksi/ langkah yang

dilakukan aktor ketika berinteraksi dengan sistem, baik yang berhasil maupun gagal. Penjelasan use case scenario dilakukan secara tekstual pada beberapa format sesuai dengan kebutuhan dan penjelasannya bisa dibuat dengan tabel 1 kolom atau 2 kolom, yaitu singkat (*brief*), informal (*casual*) atau lengkap (*fully*). Penjelasan untuk format yang singkat cukup menuliskan 1 paragraf dengan mengacu pada skenario yang berhasil. Penjelasan untuk format informal bisa dibuat beberapa paragraf yang meliputi skenario berhasil dan gagal. Sedangkan penjelasan untuk format yang lengkap sering digunakan dalam banyak praktik, terdapat bagian penting yang terkandung dalam format lengkap diantaranya aktor primer, prakondisi, alur utama, alur alternatif, kondisi akhir.




*Aktor* primer merupakan inisiasi layanan sistem yang dilakukan oleh aktor supaya tujuan tercapai, jumlah aktor primer bisa lebih dari satu. Prakondisi merupakan kondisi spesifik yang harus dipenuhi sebelum sebuah use case dieksekusi oleh aktor primer. Alur utama merupakan jalur yang menjelaskan keberhasilan sebuah skenario hingga tujuannya tercapai. Alur alternatif merupakan suatu jalur percabangan (pilihan) interaksi antar aktor dan sistem atau skenario gagal yang membuat tujuan aktor tidak terpenuhi. Kondisi akhir merupakan kondisi akhir saat sebuah use case berhasil dijalankan atau representasi dari tujuan yang ingin dicapai.

### **Sequence Diagram**



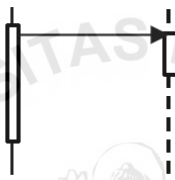
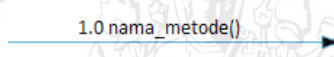
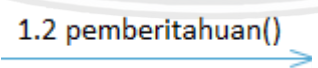
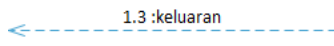
Kolaborasi yang dinamis antara obyek satu dengan yang lain. Kolaborasi ini ditunjukkan dengan adanya interaksi antara obyek di dalam dan di sekitar sistem yang berupa pesan atau intruksi yang berurutan disebut juga *Sequence diagram*. *Sequence diagram* secara general digunakan dalam penggambaran suatu scenario atau urutan langkah-langkah yang dilakukan baik oleh *actor* maupun sistem yang merupakan respon dari sebuah kejadian untuk mendapatkan hasil output (Prayitno, 2006).

Terdapat simbol yang bisa digunakan dalam pembuatan struktur class diagram pada Tabel 2.10 sebagai berikut :

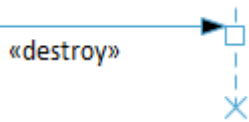
**Tabel 2. 10 Simbol Sequence Diagram**

No	Nama	Gambar	Fungsi
1	<i>Actor</i>		Mendekripsikan orang yang berinteraksi dengan sistem
2	<i>Boundary</i>		Mendekripsikan <i>interface</i> dan interaksi satu atau lebih antara aktor dan sistem
3	<i>Controller</i>		Mendekripsikan perilaku mengatur, mengkoordinasikan perilaku sistem dan dinamika dari

Tabel 2. 10 Simbol *Sequence Diagram*

No	Nama	Gambar	Fungsi
			suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem
4	<i>Entity</i>		Mendekripsikan data atau informasi yang disimpan dalam sistem
5	<i>Lifeline</i>		Mengindikasikan keberadaan sebuah obyek dalam basis waktu
6	<i>Message</i>		Mendekripsikan hubungan atau pesan antara objek satu dengan objek yang lain, dan juga meyakinkan urutan kejadian yang terjadi
7	Pesan tipe <i>call</i> ( <i>synchronous</i> )		Menyatakan suatu objek memanggil fungsi yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki fungsi, karena ini memanggil sebuah fungsi maka fungsi yang dipanggil harus ada pada <i>class diagram</i> sesuai dengan <i>class</i> objek yang berinteraksi.
8	Pesan tipe <i>send</i> ( <i>asynchronous</i> )		Menyatakan bahwa suatu objek mengirimkan data/masukan ke objek lainnya
9	Pesan tipe <i>return</i>		Menyatakan bahwa suatu objek yang telah menjalankan suatu fungsi menghasilkan kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

**Tabel 2. 10 Simbol Sequence Diagram**

No	Nama	Gambar	Fungsi
10	Pesan tipe <i>destroy</i>		Objek dapat diakhiri selama interaksi yang berlangsung. <i>Lifeline</i> objek berakhir apabila menerima pesan <i>destroy</i> . Bila ada pesan tipe <i>create</i> maka sebaiknya ada pesan tipe <i>destroy</i> .

Sumber: (A.S & Shalahudin, 2015)

Pada table 2.10 menunjukkan simbol-simbol yang terdapat di sequence diagram Banyaknya *sequence diagram* yang harus dibuat adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri, maka dari itu semakin banyak *use case* yang didefinisikan maka *sequence diagram* yang harus dibuat juga semakin banyak. Dalam pembuatan stuktur *sequence diagram* harus memperhatikan dan mengerti simbol yang ada dikarenakan apabila ada kesalahan dalam penggambaran simbol maka akan terjadi kesalahpahaman dalam memahaminya dan sulit untuk tahap implementasi selanjutnya.

### **Class Diagram**

Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Dalam setiap kelas memiliki 3 area pokok:

1. Nama (dan *stereotype*)
2. Atribut merupakan sebuah variable yang dimiliki oleh suatu kelas
3. Metode atau operasi adalah fungsi-fungsi yang dimiliki oleh suatu kelas






Atribut dan metode dalam class diagram dapat memiliki salah satu sifat seperti berikt ini (Prayitno, 2006):

1. *Private*, yaitu hanya dapat diakses oleh class itu sendiri
2. *Protected*, yaitu hanya dapat diakses oleh class itu sendiri dan turunan dari class tersebut
3. *Public*, dapat diakses oleh *class* selain dari *class* yang bersangkutan

Dalam pembuatan stuktur *class diagram* harus memperhatikan dan mengerti simbol yang ada dikarenakan apabila ada kesalahan dalam penggambaran simbol maka akan terjadi kesalahpahaman dalam memahaminya dan sulit untuk tahap implementasi selanjutnya. Terdapat simbol yang bisa digunakan dalam pembuatan struktur class diagram pada Tabel 2.11 sebagai berikut :



Tabel 2. 11 Simbol *Class Diagram*

No	Nama	Gambar	Fungsi
	<i>Class</i>		<i>Class</i> di struktur sistem. <i>Class</i> mendefinisikan atribut, operasi dan <i>instance</i> .
2.	<i>Association</i>		Relasi yang menunjukkan hubungan antara dua <i>class</i> .
3.	<i>Generalization</i>		Relasi yang menunjukkan pewarisan dari <i>class</i> utama ( <i>parent</i> ) ke <i>class</i> anak ( <i>child</i> )
4.	<i>Dependencies</i>		Relasi yang menunjukkan suatu <i>class</i> bergantung pada <i>class</i> yang lain
5.	<i>Agregasi / Aggregation</i>		Relasi yang menunjukkan sebuah elemen yang terdiri dari beberapa komponen kecil

Sumber: (A.S & Shalahudin, 2015)

Pada table 2.11 menunjukkan simbol-simbol yang terdapat pada class diagram. *Class diagram* dibuat supaya *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antar dokumentasi rancangan dan perangkat lunak saling sinkron. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas seperti kelas yang memiliki fungsi tampilan (*view*), kelas untuk mengambil dari pendefinisian *use case* (*controller*), kelas yang diambil dari pendefinisian data (*model*)

### 2.6.3 Contruction

Contruction merupakan tahap dimana pengembang mengimplementasikan perancangan yang sudah dibuat sebelumnya dan melakukan pengujian hasil implementasi. Tahapan ini aka melakukan sebuah proses perulangan terhadap kegiatan yang berjalan. Pada setiap perulangan akan melibatkan proses analisa desain, implementasi dan testing. Memperbaiki hasil implementasi dari fase sebelumnya jika ditemukan kesalahan dan melengkapi implementasi sistem.

#### 2.6.3.1 Implementasi

Pada langkah ini dilakukan proses pembuatan kode program dan pengujian untuk menemukan kesalahan (Pressman, 2010). Pembuatan kode program

merupakan penerjemahan desain kedalam sebuah bahasa yang bisa dibaca oleh komputer. Setelah pembuatan kode program selesai, maka selanjutnya dilakukan proses pengujian (*test*) perangkat lunak. Hal ini bertujuan agar membantu dalam proses perbaikan perangkat lunak apabila menemukan kesalahan yang ada dan memastikan perangkat lunak sudah berfungsi dengan layak.

### **PHP**

PHP (PHP Hypertext Processing) adalah bahasa pemrograman untuk pengembangan web yang menggunakan script *server-side* dan bahasa pemrograman umum. PHP sendiri dikembangkan oleh Rasmus Lerdorf pada tahun 1995 yang selanjutnya dikelola oleh The PHP Group sampai sekarang. PHP diproses pada komputer server sehingga disebut sebagai bahasa pemrograman *server-side*. Sedangkan bahasa pemrograman yang diproses pada web browser (*client*) yaitu JavaScript, disebut dengan bahasa pemrograman *client-side*.

PHP yang pada awalnya memiliki kepanjangan Personal Home Page dan merupakan bahasa pemrograman yang berfungsi untuk membuat website pribadi, kini menjelma menjadi bahasa pemrograman web yang powerful dan populer dan merubah kepanjangan menjadi Hypertext Preprocessor, sebuah kepanjangan rekursif yang diambil dari permainan kata dari PHP, PHP menjadi semakin populer dengan dipakainya Bahasa pemrograman PHP dalam website seperti wordpress, Wikipedia, dan masih banyak lainnya.

#### **2.6.3.2 Pengujian**

Pengujian perangkat lunak memerlukan perancangan *test case* agar dapat menemukan kesalahan dalam waktu singkat dan usaha minimum. Berbagai macam metode perancangan kasus uji telah berevolusi. Metode ini menyediakan developer pendekatan sistematis untuk pengujian. Terlebih lagi metode- metode ini menyediakan mekanisme yang dapat membantu memastikan kelengkapan dari pengujian dan menyediakan kemungkinan tertinggi untuk menemukan kesalahan-kesalahan dalam perangkat lunak (Pressman, 2010).

#### **Pengujian Black Box**

*Black Box Testing* lebih fokus terhadap spesifikasi fungsional yang ada dalam perangkat lunak. Penguji bisa mendefinisikan kumpulan dari beberapa kondisi masukan dan menguji spesifikasi fungsional program. *Black box testing* bukan termasuk dari solusi alternatif selain dari *white box testing* namun lebih mengarah pada pelengkap untuk melakukan pengujian yang tidak dicakup dalam *white box testing* (Mustaqbal, Firdaus, & Rahmadi, 2015). Proses pada *black box testing* merupakan pengujian yang mengacu pada spesifikasi sistem, dimana perilakunya hanya dapat ditentukan dengan memahami dan mempelajari hubungan antara masukan dan hasil keluaran.

*Black box testing* lebih difokuskan pada fungsionalitas dan bukan pengimplementasi dari perangkat lunak. Dalam *black box testing* lebih condong

dalam menemukan beberapa hal seperti berikut (Mustaqbal, Firdaus, & Rahmadi, 2015):

1. Fungsi yang tidak ada atau tidak benar
2. Terdapat kesalahan dalam antarmuka
3. Terdapat dalam struktur data dan juga akses basis data
4. Terdapat kesalahan dalam performansi
5. Terdapat kesalahan dalam inialisasi dan terminasi

Pada *black box testing* untuk menguji kebutuhan fungsional peneliti menggunakan *validation testing*.

#### 1. Validation Testing

*Validation testing* digunakan agar mengetahui apakah perangkat lunak yang dibangun telah benar sesuai dengan apa yang dibutuhkan. Item-item yang sudah dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi tidak memerlukan konsentrasi terhadap alur algoritma program dan lebih ditekankan dalam menemukan konformitas antara daftar kebutuhan dan kinerja sistem (Indriati, 2010). Terdapat beberapa poin yang ada dalam pengujian validasi yaitu :

- a. Nama kasus uji
- b. Hasil yang diharapkan
- c. Hasil yang didapatkan
- d. Status Validitas

### Compability Testing

Pengujian kompatibilitas perangkat lunak (*compability Testing*) adalah penjaminan mutu yang bertujuan untuk memastikan sistem berbasis multi-component yang dibangun atau dijalankan dengan baik pada lingkungan yang berbeda. Pengujian kompatibilitas juga melibatkan pemilihan konfigurasi lingkungan tertentu (Yoon, 2008).

*Cross Compability Website* adalah aspek pengujian yang sangat signifikan. Beberapa aplikasi mungkin sangat bergantung pada browser. Browser yang berbeda memiliki konfigurasi dan pengaturan yang berbeda. Pengujian dilakukan pada berbagai browser seperti Internet Explorer, Firefox, Navigator Netscape, AOL, Safari, Browser Opera dengan berbagai versi. (Sanjay Dahiya, 2012).

#### 2.6.4 Transition

Pada Tahap akhir ini dilakukan pengujian, instalasi dan sosialisasi perangkat lunak. Pengujian sistem ini ditujukan menunjukkan bahwa sistem benar-benar telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan sistem. terdapat kegiatan membuat dokumentasi tambahan seperti training, user guides.

## User Acceptance Testing

Sebelum sebuah perangkat lunak yang telah dibangun diberikan kepada klien maka perlu dilakukan pengujian agar memastikan seluruh persyaratan pengguna yang ada pada *Specification Requirement Software* telah terpenuhi dan sesuai. Pengujian yang dilakukan pada tahap ini adalah *User Acceptance Testing* (UAT). UAT dilakukan setelah semua sistem siap digunakan dan sebelum perangkat lunak digunakan oleh klien atau pengguna.

*User Acceptance Testing* (UAT) merupakan sarana formal dimana perusahaan mengkonfirmasi bahwa sistem yang telah dibangun sudah benar-benar memenuhi persyaratan pengguna. Tahap ini merupakan pengujian akhir yang dilakukan setelah melakukan pengujian fungsional. *User Acceptance Testing* (UAT) dilakukan oleh pengguna dan pengembang perangkat lunak, sebelum perangkat lunak ini dioperasikan pengujian ini dilakukan sebagai tahap akhir pengujian perangkat lunak. Tujuan utama dari *User Acceptance Testing* (UAT) adalah mengecek perangkat lunak terhadap kebutuhan bisnis, hal ini dilakukan oleh pengguna akhir yang sudah familiar dengan kebutuhan bisnis (Goel & Gupta, 2014).

Inti dari user acceptance testing terdapat pada seperangkat *acceptance criteria* yang harus dipenuhi oleh perangkat lunak agar bisa diterima (Naik, 2008). Terdapat *acceptance criteria* yang digunakan oleh peneliti berdasarkan buku "SOFTWARE TESTING AND QUALITY ASSURANCE Theory and Practice" yang dibuat oleh Kshirasagar Naik diantaranya yaitu *Usability, Functional Correctness and Completeness, Confidentiality and Availability*.

*Usability* terdapat pertanyaan mengenai seberapa mudah dalam menggunakan sistem dan seberapa mudah dalam mempelajarinya. Tujuan dari *usability* untuk memastikan bahwa sistemnya fleksibel, mudah untuk mengonfigurasi, antarmuka *friendly*, bantuan *online* tersedia, pekerjaan tersedia (Naik, 2008).

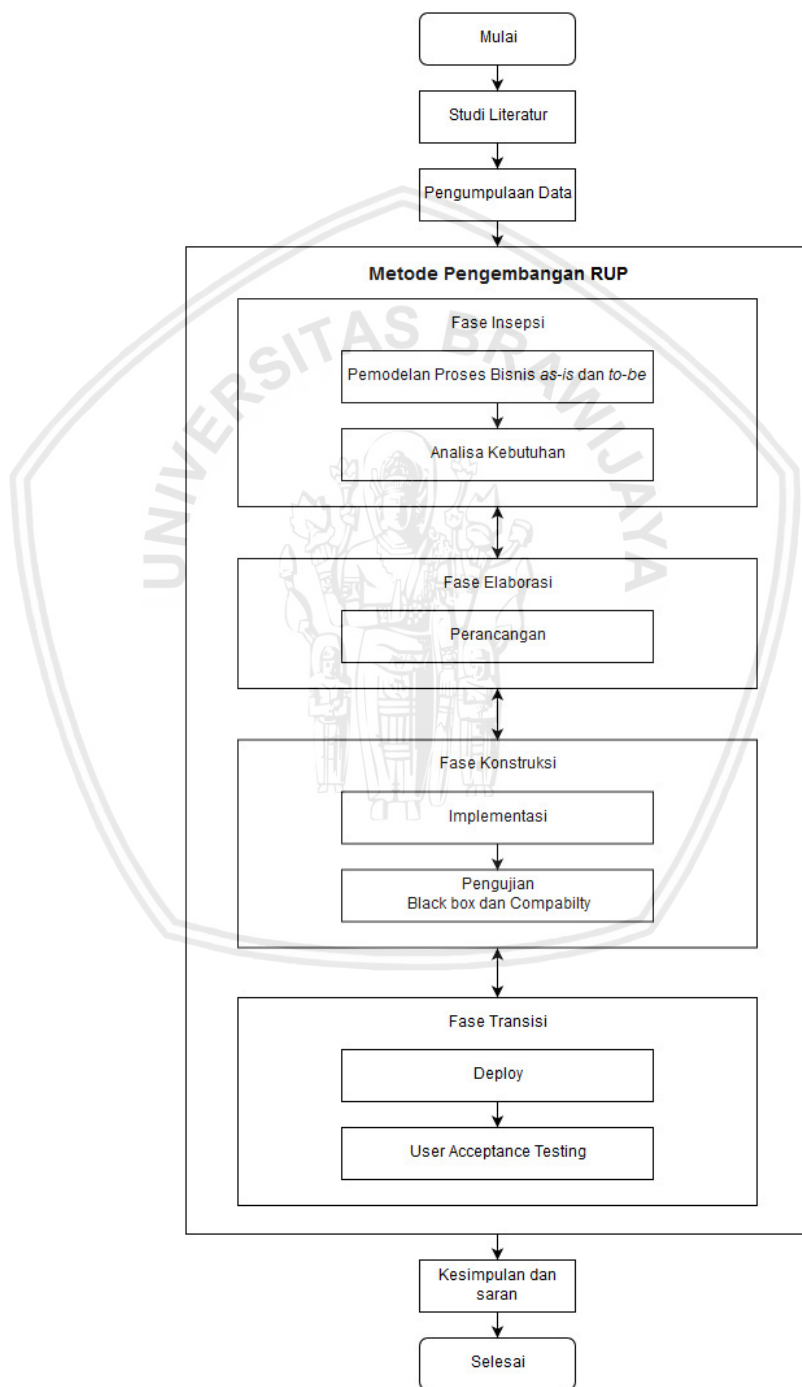
*Functional Correctness and Completeness* terdapat pertanyaan mengenai apakah sistem melakukan apa yang diinginkan klien. Semua fitur yang dijelaskan dalam spesifikasi persyaratan harus ada dalam perangkat lunak yang di kirimkan (*delivered*) ke klien. Penting agar menunjukkan bahwa sistem bekerja dengan benar, setidaknya terdapat dua sampai tiga kondisi untuk fitur sebagai bagian dari penerimaan (Naik, 2008).

*Confidentiality and Availability* terdapat pertanyaan mengenai kriteria penerimaan *confidentiality* yang mengacu pada persyaratan bahwa data harus dilindungi dari pengungkapan yang tidak sah dan kriteria penerimaan *availability* dengan persyaratan bahwa data harus dilindungi dari *denial of service* (DoS) kepada pengguna yang berwenang (Naik, 2008).



### BAB 3 METODOLOGI PENELITIAN

Pada bab metode penelitian ini akan membahas langkah-langkah yang akan dilakukan dalam merancang dan membangun Sistem Informasi Rawat inap pada Rumah Sakit Khusus Hayunanto *Medical Center* menggunakan metode Rationl Unified Process. Di bawah ini merupakan diagram alir dari metode Penelitian yang dilakukan.



**Gambar 3.1 Diagram Alur Metodologi**



Gambar 3.1 menunjukkan alur atau proses pengembangan Sistem Informasi Rawat inap pada Rumah Sakit Khusus Hayunanto *Medical Center*, dimulai dari studi literatur, Analisis Masalah, Pengembangan Metode RUP, Kesimpulan dan Saran.

### 3.1 Studi Literatur

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

1. Proses bisnis pada Rumah Sakit Khusus HMC Malang
2. Pemodelan proses bisnis *as-is* dan *to-be*
3. Pengembangan perangkat lunak dengan metode RUP
4. Unified Modelling Language
5. Pemrograman aplikasi web menggunakan bahasa PHP
6. Database MySQL
7. Pengujian dan Analisis Perangkat Lunak

### 3.2 Analisa Permasalahan

Pada tahap ini dilakukan analisa permasalahan pada Rumah Sakit Khusus Hayunanto *Medical Center* Malang yaitu permasalahan yang terjadi pada pelayanan rawat inap. Berikut beberapa kegiatan yang dilakukan pada tahap ini adalah:

a. Observasi

Kegiatan ini merupakan metode mengumpulkan data dengan mengamati secara langsung di Rumah Sakit Khusus HMC khususnya pada pelayanan rawat inap, mulai dari data yang berupa proses bisnis, struktur organisasi, serta pelayanan rawat inap pada Rumah Sakit Khusus HMC.

b. Wawancara

Kegiatan ini merupakan metode mengumpulkan data dengan cara bertanya langsung terhadap responden yang dianggap layak untuk memberikan informasi mengenai sistem pelayanan rawat inap. Kegiatan ini bertujuan melakukan analisis kebutuhan dan batasan kegiatan.

### 3.3 Metode Rational Unfiend Process (RUP)

Pada Fase ini yaitu melakukan pengembangan sistem informasi dengan metode yang telah dipilih dan sesuai dengan penelitian ini yaitu *Rational Unified*

*Process* (RUP). RUP memiliki 4 fase pengulangan diantaranya fase insepisi, fase elaborasi, fase konstruksi, fase transisi. Dalam penelitian ini setiap fase metode RUP penggunaannya disesuaikan dengan pengembangan sistem informasi rawat inap yang dilakukan.

### **3.1.1 Inception**

Di dalam fase ini yaitu Pada Fase ini yaitu melakukan pemahaman tentang Analisis dan pemodelan proses bisnis dengan menggunakan konsep BPMN dengan melakukan studi literatur mengenai BPMN serta pemodelan proses bisnis as-is dari hasil data yang dikumpulkan dari observasi atau wawancara pada Layanan rawat inap Rumah Sakit Khusus HMC Malang untuk Mengetahui proses bisnis yang sedang berjalan saat ini. Setelah data berhasil dikumpulkan, kemudian dilakukan analisis prosedur pelaksanaan proyek dan digambarkan dalam bentuk diagram dengan menggunakan konsep BPMN. Setelah itu membangun model to-be yaitu menggambarkan perubahan dan peningkatan proses bisnis as-is. Model to-be ini akan menjadi proses bisnis usulan perbaikan pada proses bisnis Layanan rawat inap Rumah Sakit Khusus HMC Malang dengan diterapkannya sistem informasi yang dikembangkan di penelitian ini. Dengan penggunaan BPMN diharapkan memudahkan dalam menggambarkan alur-alur proses bisnis Layanan rawat inap Rumah Sakit Khusus HMC Malang.

### **3.1.2 Elaboration**

Fase elaborasi merupakan fase kedua yang menganalisa kebutuhan sistem serta perancangan. Dalam fase ini akan dilakukan kegiatan yang hampir mirip pada fase sebelumnya yaitu memperbaiki atau mengevaluasi apabila ditemukan kesalahan atau perubahan.

#### **a. Analisa Kebutuhan**

Pada tahap ini dilakukan perancangan aplikasi, disini perancangan lebih terperinci dari kebutuhan sistem yang telah dilakukan pada fase sebelumnya. Perancangan ini merupakan penentuan arsitektur awal dari perancangan mengacu pada kebutuhan sistem yang akan dibangun baik fungsional atau pun non-fungsional. Kebutuhan fungsional merupakan kebutuhan utama yang diperlukan oleh sistem sedangkan kebutuhan non-fungsional merupakan proses menganalisa kebutuhan diluar kebutuhan utama sistem seperti keamanan dan kemudahan dalam mengakses sistem itu sendiri. Pada fase ini juga mengidentifikasi aktor dan use case. identifikasi aktor dapat mengetahui karakter dari masing-masing pengguna.

#### **b. Perancangan**

Di tahap ini dilakukan perancangan atau desain arsitektur dari sistem yang akan dikembangkan. Pada langkah ini mengacu pada hasil dari analisa kebutuhan yang sudah digali sebelumnya. Hasil dari langkah ini akan digunakan untuk referensi dalam merancang arsitektur pada proses implementasi. Selanjutnya dilakukan perancangan atau desain arsitektur dari sistem yang akan dikembangkan. Pembuatan diagram berdasarkan hasil dari analisis kebutuhan fungsional dan

kebutuhan non-fungsional yang ada pada tahap analisis kebutuhan sistem sebelumnya. Selanjutnya dengan merancang *User Interface (UI)* atau tampilan yang nantinya akan diimplementasikan dalam sistem. Tampilan yang dirancang berupa tata letak terhadap setiap fungsi sebagai acuan dalam membuat halaman *web* yang akan di implementasikan. Nantinya di setiap proses perancangan dilakukan evaluasi dan perbaikan terhadap desain arsitektur sistem, database, dan antar muka sistem kepada *stakeholder* untuk perubahan kebutuhan serta pemahaman *stakeholder* terhadap sistem informasi yang dikembangkan.

### 3.1.3 Construction

Pada tahap ini dilakukan implementasi sistem dari perancangan yang sudah dibuat sebelumnya. Tahap ini juga melakukan pengujian terhadap sistem. Dalam fase ini akan dilakukan kegiatan perbaikan daftar kebutuhan, arsitektur sistem fase sebelumnya apabila ditemukan kesalahan.

#### a. Implementasi

Pada langkah ini dilakukan implementasian sistem atau tahapan penulisan code program menggunakan bahasa pemrograman *Hypertext Markup Language (HTML)*, *Cascading Style Sheets (CSS)*, *Javascript*, *Asynchronous JavaScript and XML (AJAX)*, *Jquery*, *Hypertext Processor (PHP)*, dilengkapi dengan fasilitas yang telah disediakan oleh *framework CodeIgniter* serta menggunakan database MySQL. Evaluasi dan perbaikan juga terus dilakukan dalam tahap ini untuk tetap dapat memajemen perubahan yang mungkin terjadi selama proses implementasi agar tetap dapat memenuhi kebutuhan dan persyaratan terkini dari *stakeholder*.

#### b. Pengujian

Fase ini dilakukan pengujian perangkat lunak yang telah selesai dibangun. Pengujian ini merupakan langkah penting karena menentukan kualitas baik dan buruknya dari suatu perangkat lunak. pengujian ini tujuannya untuk mengetahui kelemahan serta kesalahan yang mungkin terjadi saat pembangunan yang telah dibangun agar dapat melakukan perbaikan dan dapat berjalan sesuai dengan harapan pengguna. Pengujian yang dilakukan oleh peneliti menggunakan black box testing dan compability testing.

Pengujian black box dilakukan untuk memastikan apakah aplikasi yang dibuat berfungsi sesuai dengan keinginan atau tidak. Sedangkan compability testing dilakukan dengan menguji sistem pada platform yang berbeda apakah sistem dapat berjalan dengan baik atau tidak. Pengujian dilakukan pada berbagai browser seperti Internet Explorer, Firefox, Chrome dan lain-lain. Hal ini dilakukan dalam upaya peningkatan fleksibilitas dalam penggunaan sistem oleh pengguna.

### 3.1.4 Transition

fase transisi ini adalah melakukan pengujian akhir sistem dan memperkenalkan sistem yang telah jadi kepada *stakeholder*. Dalam fase ini dilakukan *Deploy* dan *User Acceptance Testing*.

#### a. *Deploy*

Sistem Informasi Rawat Inap yang telah dikembangkan dan diuji pada tahap sebelumnya akan di *release* dan diberikan kepada *stakeholder* untuk dilakukan instalasi di Rumah Sakit Khusus Hayunanto *Medical Center* Malang. Setelah dilakukan instalasi juga diberikan bantuan serta pengarahan penggunaan sistem kepada pengguna agar penerapan sistem dapat diterapkan dan diterima dengan optimal.

#### b. *User Acceptance Testing (UAT)*

Demi memastikan apakah sistem ini dapat diterima sesuai dengan kebutuhan dan fungsi yang diinginkan pengguna maka dibutuhkan pengujian *User Acceptance Testing*. Pengujian ini dimulai dengan menentukan atribut *acceptance requirement* (persyaratan penerimaan) hingga memberikan kuesioner kepada *user* mengenai sistem yang telah dibuat.

## 3.4 Tahap Pengambilan Kesimpulan dan Saran

Disini dalam pengambilan kesimpulan akan dilakukan setelah tahap perancangan, implementasi, pengujian sistem dan di *deploy* telah dilakukan dan didasarkan pada kesesuaian teori dan praktik. Nantinya kesimpulan akan diambil analisis dan hasil pengujian terhadap sistem yang dibuat. Tahap akhir dari penulisan ini adalah saran yang nantinya untuk memperbaiki kesalahan-kesalahan yang mungkin terjadi dan menyempurnakan penulisan serta memberikan saran atau pertimbangan dengan pengembangan sistem selanjutnya.

## BAB 4 ANALISIS KEBUTUHAN

Pada bab ini akan membahas tentang proses analisis kebutuhan dari sistem informasi rawat inap pada rumah sakit khusus Hayunanto *Medical Center* di Malang. Aktivitas merupakan fase inepsi dan fase elaborasi dalam metode RUP. Dalam fase pertama inepsi akan menghasilkan analisa proses bisnis, analisis permasalahan, analisa kebutuhan fungsional dan non fungsional. Lalu masuk dalam tahap elaborasi perancangan awal untuk perancangan usecase diagram serta activity diagram. Analisis yang diperoleh berdasarkan analis pada hasil wawancara dan memetakan ke dalam tabel analisis permasalahan. Informasi berupa proses bisnis rumah sakit ini akan digambarkan dengan diagram BPMN. umum lalu dilanjutkan dengan spesifikasi kebutuhan yang digunakan agar mendapatkan kebutuhan fungsional dan non-fungsional yang akan dimanfaatkan dalam pembuatan Use Case Diagram, Use Case Scenario dan Activity Diagram sehingga bisa digunakan dalam tahap perancangan sistem yang ada pada bab selanjutnya.

### 4.1.1 Identifikasi Tipe Pemangku Kepentingan

Pada identifikasi tipe pemangku kepentingan dilakukan pengelompokan pemangku kepentingan sesuai dengan karakteristik pemangku kepentingan dan hubungan pemangku kepentingan dengan sistem yang akan dikembangkan. Hasil dari identifikasi pemangku kepentingan ini digunakan untuk informasi kegiatan analisis masalah sehingga dapat memperoleh informasi masalah dari berbagai pemangku kepentingan di Rumah Sakit Khusus Hayunanto *Medical Center* Malang. Penejelasan dari tipe pemangku kepentingan yang dilakukan dengan melakukan analisis terhadap hasil wawancara dengan Wakil Manajer Rumah Sakit Khusus Hayunanto *Medical Center* Malang terdapat pada tabel 4.1.

Tabel 4. 1 Tipe Pemangku Kepentingan

No	Tipe Pemangku Kepentingan	Keterangan	Pemangku Kepentingan
1	Pengguna	Individu yang melakukan interaksi secara langsung dengan sistem informasi serta berperan sebagai actor pada <i>use case</i> .	<ul style="list-style-type: none"> <li>• Petugas <i>Front Office</i></li> <li>• Perawat</li> <li>• Dokter</li> <li>• Petugas Farmasi</li> </ul>
2	Pengembang	Individu atau organisasi yang melakukan pengembangan sistem informasi.	Peneliti

No	Tipe Pemangku Kepentingan	Keterangan	Pemangku Kepentingan
3	Pihak yang berwenang	Individu atau organisasi yang dapat memberikan dukungan dengan memberikan informasi yang dibutuhkan terkait regulasi yang berjalan, sehingga solusi dapat diterapkan sesuai aturan yang berlaku.	Rumah Sakit Khusus Hayunanto <i>Medical Center</i> Malang
4	Pelanggan	Individu atau organisasi yang akan mendapatkan manfaat dari penerapan sistem informasi.	Pasien Rawat Inap Rumah Sakit Khusus Hayunanto <i>Medical Center</i> Malang

#### 4.1.2 Analisis Masalah

Analisis masalah dilakukan untuk menganalisa permasalahan di Rumah Sakit Khusus Hayunanto *Medical Center* Malang, permasalahan yang dianalisa difokuskan pada salah satu layanan unggulan yaitu Rawat Inap. Analisa permasalahan dilakukan dengan melakukan Observasi secara langsung di Rumah Sakit Khusus Hayunanto *Medical Center* Malang serta Wawancara dengan pihak yang berkepentingan di Layanan Rawat Inap Rumah Sakit Khusus Hayunanto *Medical Center* Malang. *Problem statement* yang didapatkan melalui analisis hasil observasi dan wawancara pengurus Humas serta Wakil Manajer terdapat pada tabel 4.2.

**Tabel 4. 2 Problem Statement**

No	Analisis	Deskripsi
1	<i>The problem of</i>	<ul style="list-style-type: none"> <li>• Terdapat data yang sama namun memiliki nilai-nilai berbeda dalam file-file yang berbeda (tidak ada integrasi data) misal: dua rincian biaya yang seharusnya ditulis dalam satu dokumen,yaitu pada dokumen rincian biaya perawatan dan rincian biaya pengeluaran obat</li> <li>• Perekapan data membutuhkan waktu yang lama disebabkan perekapan data dilakukan diakhir setelah seluruh pasien</li> </ul>

No	Analisis	Deskripsi
		<p>melakukan pemeriksaan menggunakan Ms. Excel.</p> <ul style="list-style-type: none"> <li>• Data tidak fleksibel karena banyaknya berkas/dokumen yang tersimpan. Data yang tersimpan dalam file sulit diakses karena dalam bentuk berkas bukan pada sever.</li> <li>• Ada 2 file yang berisi data yang sama, karena di edit oleh dua orang yang berbeda.</li> </ul>
2	<i>Affects</i>	Pengguna, Pihak yang berwenang, Pelanggan.
3	<i>The impact of which is</i>	<ul style="list-style-type: none"> <li>• Proses bisnis yang berjalan membutuhkan waktu yang lebih.</li> <li>• Dengan tidak terpusatnya data menyebabkan kurang efisiensi waktu yang digunakan , terutama ketika dibutuhkan untuk pelaporan</li> <li>• Terjadinya kemungkinan adanya duplikasi data, sehingga data menjadi tidak relevan</li> <li>• Informasi yang diolah secara berulang-ulang sehingga dari segi waktu dianggap kurang efisien.</li> </ul>
4	<i>A successful solution would</i>	Sistem informasi yang dapat membantu dalam perekapan data pasien secara langsung serta memberikan penyebaran informasi yang cepat dalam Pelayanan Rawat Inap di Rumah Sakit HMC Malang.

#### 4.1.3 Pemodelan Proses Bisnis

Pemodelan proses bisnis dilakukan untuk memodelkan serangkaian aktivitas yang dilakukan oleh fungsi-fungsi bisnis pada rumah sakit khusus Hayunanto *Medical Center* di Malang untuk memenuhi kebutuhan pengguna



bisnis (internal maupun eksternal). Penggalian informasi mengenai proses bisnis yang akan dimodelkan didapatkan dengan cara melakukan wawancara dengan pegawai rumah sakit mengenai alur proses bisnis pada rumah sakit dan observasi pada lokasi studi kasus penelitian. Dalam penelitian ini, pemodelan proses bisnis akan menghasilkan beberapa diagram proses bisnis yang terbagi menjadi 2 kelompok, yaitu proses bisnis yang saat ini berjalan (*as-is*) dan proses bisnis usulan (*to-be*).

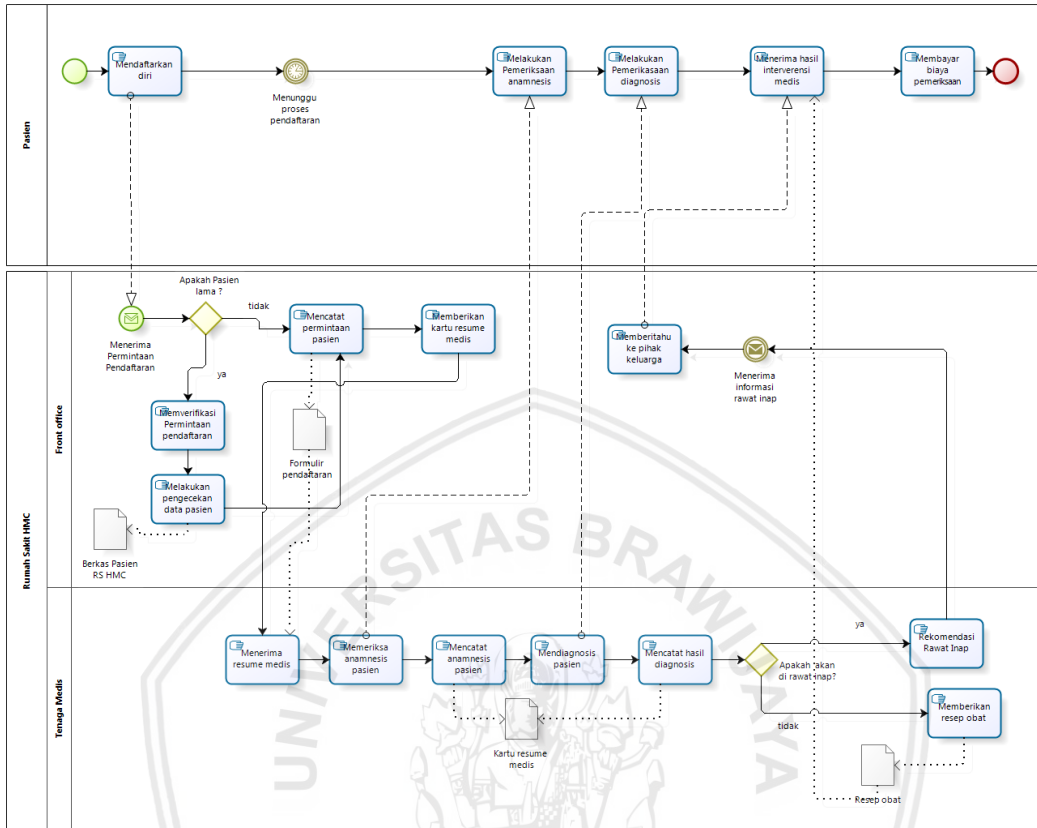
#### 4.1.3.1 Proses Bisnis As-Is

Dalam penelitian ini proses bisnis *as-is* dapat diidentifikasi berdasarkan keadaan proses bisnis yang berjalan pada lokasi studi kasus sebelum solusi yang diusulkan dalam penelitian ini diusulkan pada lokasi studi kasus. Analisis dan pemodelan terhadap proses bisnis *as-is* dilakukan agar peneliti dapat mengidentifikasi kemungkinan perbaikan atau peningkatan terhadap proses bisnis melalui solusi yang akan ditawarkan dalam penelitian ini. Analisis proses bisnis *as-is* dilakukan dengan melakukan wawancara dengan pegawai Rumah Sakit HMC Malang mengenai alur aktivitas bisnis pada poliklinik dan observasi pada lokasi studi kasus penelitian. Kemudian hasil analisis dimodelkan dengan notasi BPMN untuk membantu pemangku kepentingan memahami hasil analisis proses bisnis *as-is*.

##### a. Proses Bisnis As-Is Pendaftaran Rawat Inap

Proses bisnis di mulai dari pasien mendaftarkan diri pada front office. Lalu front office mengecek apakah pasien lama atau pasien baru. Jika pasien lama maka front office akan memverifikasi permintaan pendaftaran dengan mencari di berkas pasien berupa buku pendaftaran. Selanjutnya front office mencatat permintaan pasien yang dicatat di kertas formulir pendaftaran. Setelah itu menyiapkan kartu resume medis dan menyerahkan kepada tenaga medis untuk memeriksa pasien. Tenaga medis mencatat hasil anamnesis dan diagnosis pasien yang dicatat di kartu resume medis. Jika pasien perlu ditangani lebih intens maka akan direkomendasikan rawat inap dan pihak front office memberitahukan kepada pihak keluarga dari hasil intervensi medis. Dan pasien atau keluarga membayar biaya pemeriksaan. Jika direkomendasikan rawat jalan maka hanya diberi resep

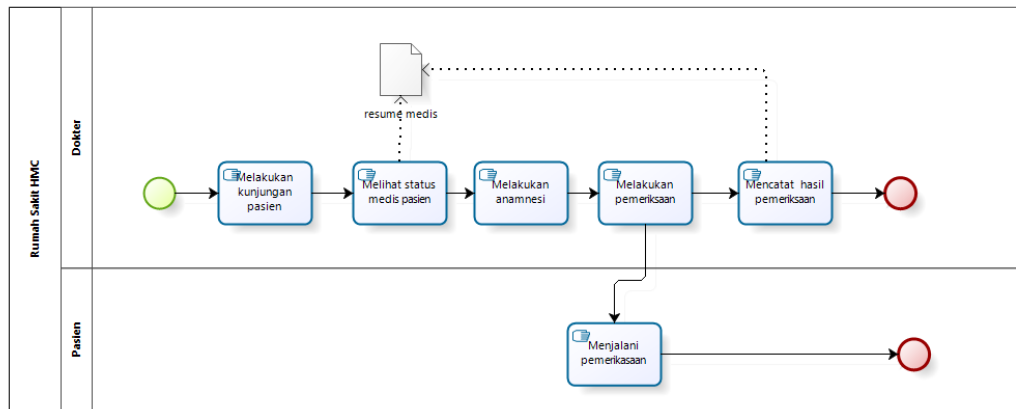
obat. Pemodelan proses bisnis as-is Pendaftaran rawat inap melalui terdapat gambar 4.1



**Gambar 4. 1 Proses Bisnis As-is Pendaftaran Rawat Inap**

**b. Proses Bisnis As-Is Pemeriksaan Pasien Rutin Dokter**

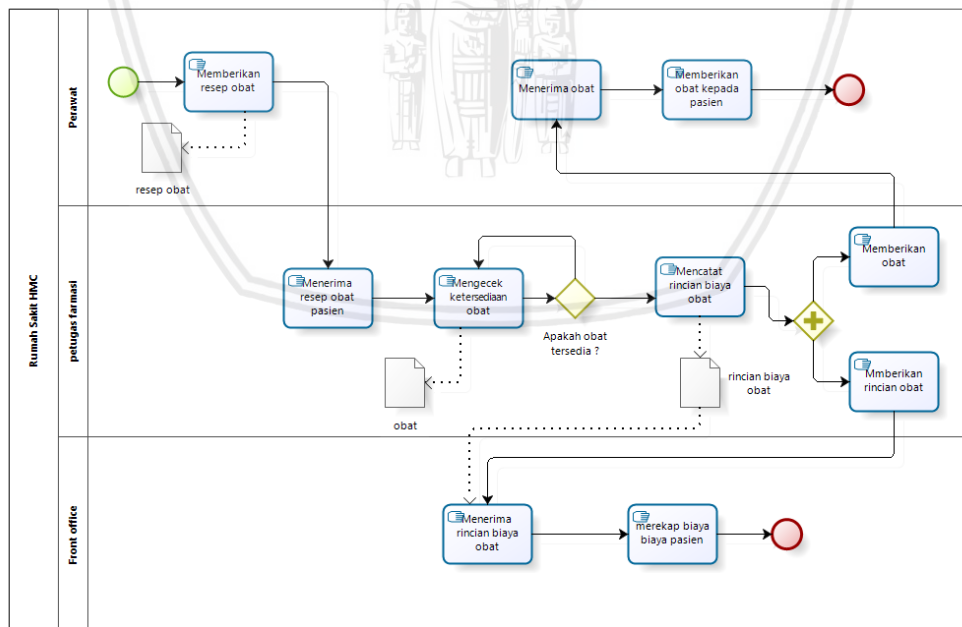
Proses bisnis pemeriksaan pasien rutin dokter diawali dengan dokter melakukan kunjungan kepada pasien. Lalu dokter melihat status medis sebelumnya berupa kertas resume medis. Setelah itu dokter melakukan anamnesis terhadap pasien sekaligus melakukan pemeriksaan terhadap pasien yang nantinya akan di catat hasil pemeriksaannya di kertas sebelumnya yaitu berupa kertas resume medis. Pemodelan proses bisnis as-is Pemeriksaan pasien rutin dokter terdapat gambar 4.2



Gambar 4. 2 Proses Bisnis As-is Pemeriksaan Pasien Rutin Dokter

c. Proses Bisnis As-Is Pelayanan Obat Pasien

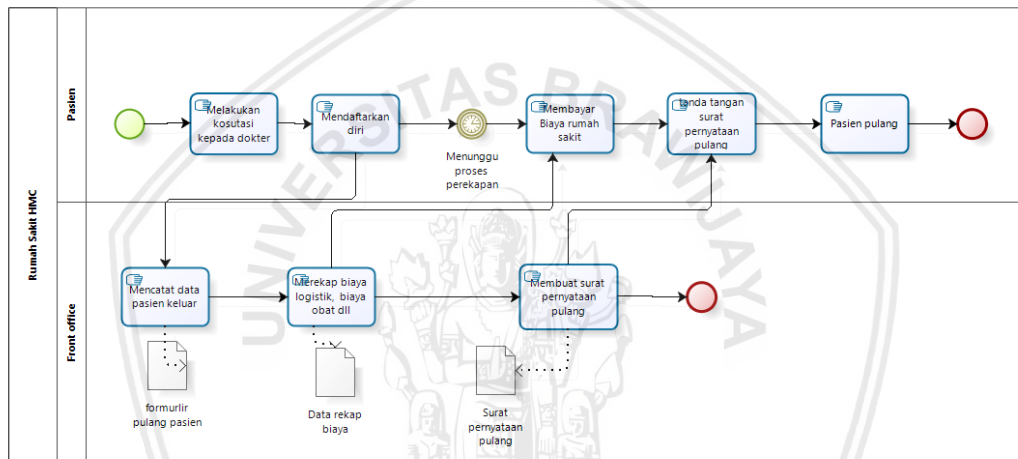
Jika pasien membutuhkan obat perawat akan menuliskan memberikan resep obat kepada pegawai farmasi berupa kertas resep obat. Pegawai farmasi menerima lalu langsung mengecek ketersediaan obat secara manual karena tidak terdapat sistem informasi maka pegawai mencari obat secara manual mencari satu persatu obat. Setelah menemukan obat yang dibutuhkan lalu pegawai farmasi memberikan obatnya langsung ke perawat lalu memberikannya kepada pasien. Lalu rincian biaya obat pasien tersebut diberikan kepada front office. Front office lalu merekap biaya pasien tersebut. Pemodelan proses bisnis as-is pelayanan obat pasien terdapat gambar 4.3



Gambar 4. 3 Proses Bisnis As-is Pelayanan Obat Pasien

**d. Proses Bisnis As-Is Pelayanan Pasien Keluar**

Proses bisnis pelayanan pasien keluar diawali keluarga pasien melakukan konsultasi kepada dokter tentang kondisi pasien. Lalu jika diperbolehkan pasien pulang maka keluarga pasien mendaftarkan diri ke front office. Front office mencatat data pasien keluar lalu merekap biaya keseluruhan pasien secara manual melalui excel dengan biaya rekap terpisah mulai dari data rekap biaya logistik, biaya obat, biaya kamar inap dll. Front office juga membuat surat pernyataan pulang pasien. Setelah pasien menunggu perekapan biaya rumah sakit maka pihak keluarga harus membayar biaya pasien selama di rumah sakit sekaligus tanda tangan surat pernyataan bahwa pasien telah keluar dari rumah sakit HMC Malang ini. Lalu pasien telah dinyatakan keluar dari rumah sakit HMC Malang. Pemodelan proses bisnis as-is pelayanan pasien keluar terdapat gambar 4.4



**Gambar 4. 4 Proses Bisnis As-is Pelayanan Pasien Keluar**

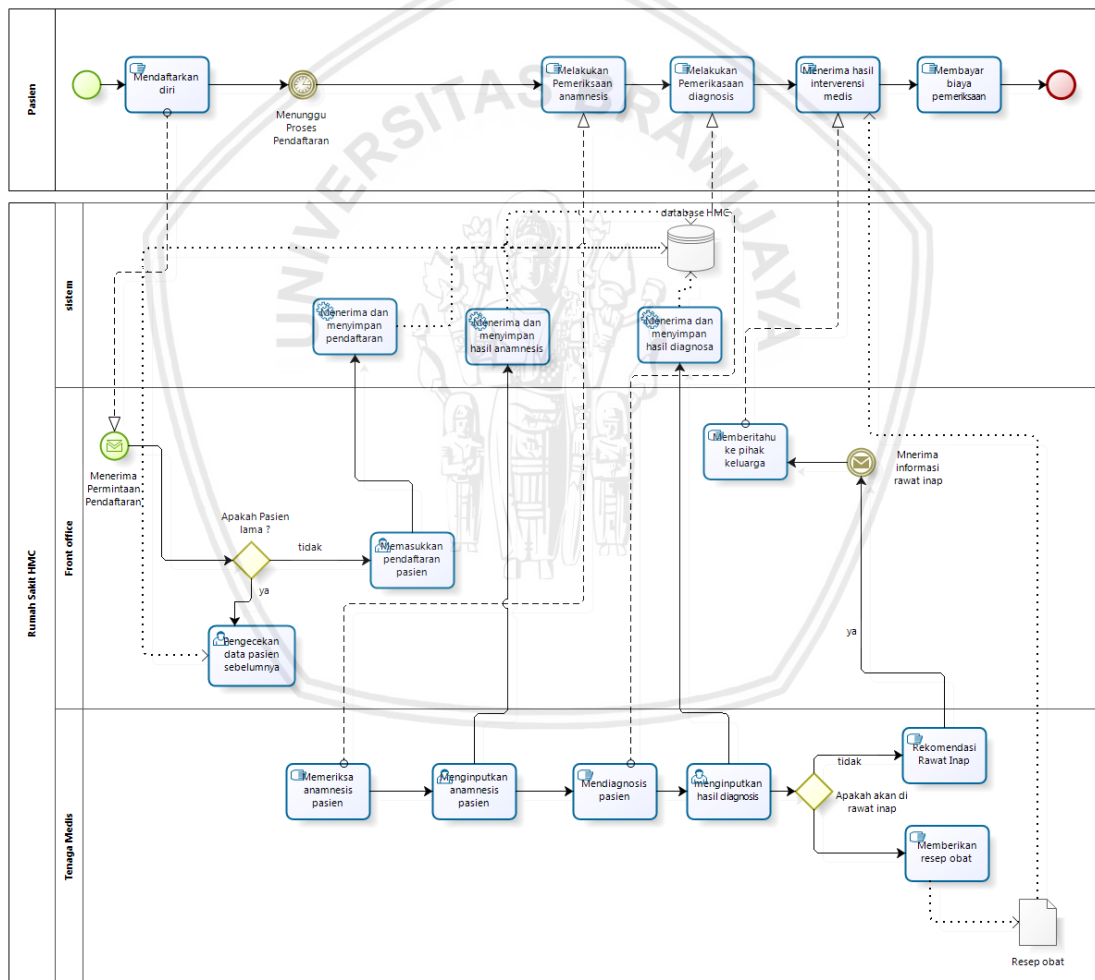
**4.1.3.2 Proses Bisnis To-be**

Proses bisnis to-be yang dimodelkan pada penelitian ini merupakan proses bisnis usulan yang ditawarkan dalam penelitian ini kepada pemangku kepentingan. Proses bisnis to-be diidentifikasi berdasarkan hasil analisis terhadap proses bisnis as-is. Dalam penelitian ini, analisis proses bisnis to-be dilakukan dengan melakukan eliminasi pada aktivitas yang bersifat redundan, kemudian menentukan aktivitas-aktivitas yang membutuhkan dukungan sistem informasi. Selanjutnya, proses bisnis to-be dimodelkan untuk menggambarkan perubahan atau tambahan yang diusulkan melalui penelitian ini. Hasil pemodelan proses bisnis to-be digunakan sebagai input kegiatan analisis persyaratan.



**a. Proses Bisnis To-be Pendaftaran Rawat Inap Melalui Poli**

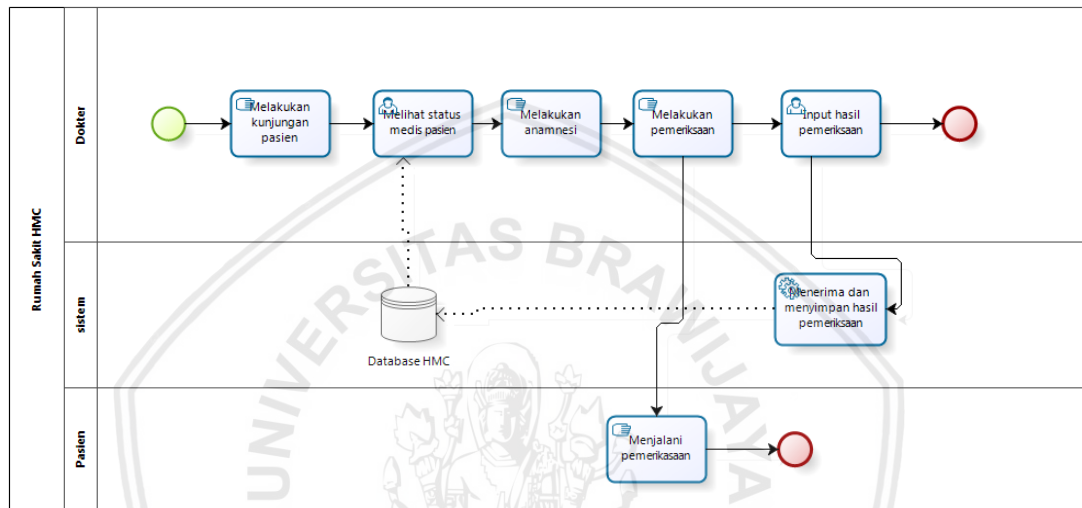
Proses bisnis di mulai dari pasien mendaftarkan diri pada front office. Lalu front office mengecek apakah pasien lama atau pasien baru. Jika pasien lama maka front office akan memverifikasi permintaan pendaftaran dengan mencari di sistem informasi rawat inap. Selanjutnya front office menginputkan permintaan pasien yang di disimpan di database. Tenaga medis menginputkan hasil anamnesis dan diagnosis pasien ke dalam sistem informasi di simpan di database. Jika pasien perlu ditangani lebih intens maka akan direkomendasikan rawat inap dan pihak front office memberitahukan kepada pihak keluarga dari hasil intervensi medis. Dan pasien atau keluarga membayar biaya pemeriksaan. Jika direkomendasikan rawat jalan makan hanya diberi resep obat. Pemodelan proses bisnis as-is Pendaftaran rawat inap melalui terdapat gambar 4.5



**Gambar 4. 5 Proses Bisnis To-be Pendaftaran Rawat Inap**

**b. Proses Bisnis To-be Pemeriksaan Pasien Rutin Dokter**

Proses bisnis pemeriksaan pasien rutin dokter diawali dengan dokter melakukan kunjungan kepada pasien. Lalu dokter melihat status medis sebelumnya di dalam sistem informasi dalam database. Setelah itu dokter melakukan anamnesis terhadap pasien sekaligus melakukan pemeriksaan terhadap pasien yang nantinya akan di inputkan hasil pemeriksaannya di simpan di database. Pemodelan proses bisnis to-be Pemeriksaan pasien rutin dokter terdapat gambar 4.6

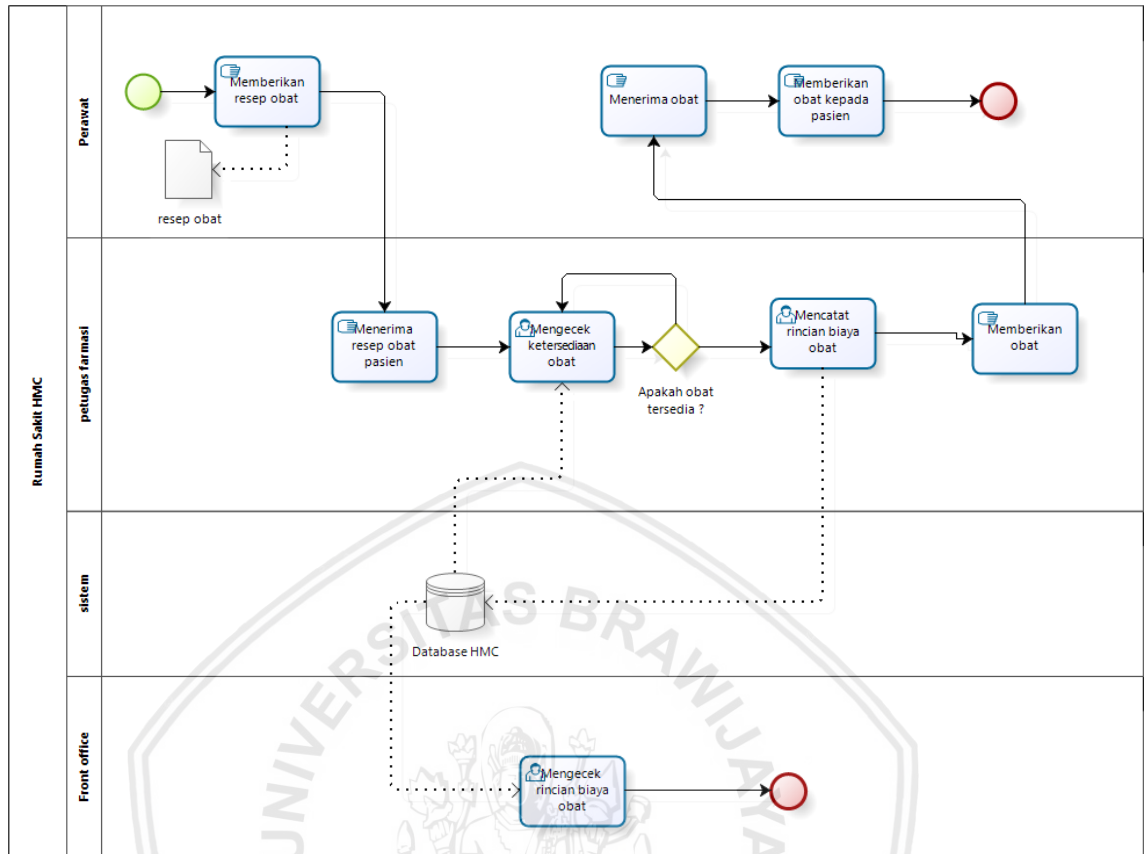


**Gambar 4. 6 Proses Bisnis To-be Pemeriksaan Pasien Rutin Dokter**

**c. Proses Bisnis To-be Pelayanan Obat Pasien**

Jika pasien membutuhkan obat perawat akan menuliskan memberikan resep obat kepada pegawai farmasi berupa kertas resep obat. Pegawai farmasi menerima lalu langsung mengecek ketersediaan obat di sistem informasi. Setelah menemukan obat yang dibutuhkan lalu pegawai farmasi memberikan obatnya langsung ke perawat lalu memberikannya kepada pasien. Lalu rincian biaya obat pasien tersebut diberikan kepada front office otomastis melalui sistem informasi. Front office hanya merekap biaya pasien tersebut di dalam sistem informasi dari dalam database. Pemodelan proses bisnis to-be pelayanan obat pasien terdapat gambar 4.7

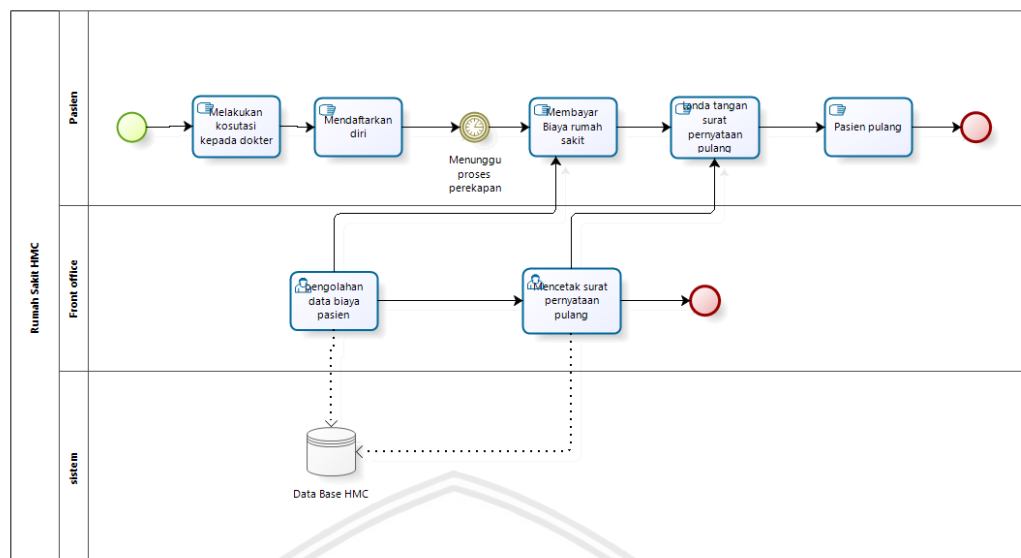




Gambar 4. 7 Proses Bisnis To-be Pelayanan Obat Pasien

**d. Proses Bisnis To-be Pelayanan Pasien Keluar**

Proses bisnis pelayanan pasien keluar diawali keluarga pasien melakukan konsultasi kepada dokter tentang kondisi pasien. Lalu jika diperbolehkan pasien pulang maka keluarga pasien mendaftarkan diri ke front office. Front office mencatat data pasien keluar lalu merekap biaya keseluruhan pasien secara melalui sistem informasi. Front office juga membuat surat pernyataan pulang pasien melalui sistem yang hanya tinggal mencetak. Setelah pasien menunggu perekapan biaya rumah sakit maka pihak keluarga harus membayar biaya pasien selama di rumah sakit sekaligus tanda tangan surat pernyataan bahwa pasien telah keluar dari rumah sakit HMC Malang ini. Lalu pasien telah dinyatakan keluar dari rumah sakit HMC Malang. Pemodelan proses bisnis to-be pelayanan pasien keluar terdapat gambar 4.8



Gambar 4. 8 Proses Bisnis To-be Pelayanan Pasien Keluar

#### 4.1.4 Analisis *Improvement* Proses Bisnis

Setelah pemodelan proses bisnis *to-be* dibuat sebagai rekomendasi proses bisnis baru yang lebih baik terdapat beberapa perubahan dan peningkatan proses. Pada proses bisnis yang baru ada beberapa task yang diubah serta perbaikan dalam jumlah waktu yang dibutuhkan untuk melakukan sebuah proses. Berikut merupakan perubahan aktivitas serta analisis waktu dari rawat inap setelah diterapkan sistem informasi.

##### a. Perubahan Aktivitas Proses Bisnis

Setelah dilakukan pemodelan proses bisnis *as-is* dan *to-be* selanjutnya dapat kita bandingkan perubahan yang akan dilakukan dengan menganalisis perubahan aktivitas proses bisnis. Perubahan aktivitas proses bisnis yang dilakukan yaitu pengurangan *task*, penambahan *task*, dan pengubahan *task*. Hasil dari perubahan aktivitas proses bisnis dapat dilihat pada tabel berikut yang menjelaskan unit bisnis mana yang dilakukan perubahan dan keterangan perubahan yang dilakukan.

Tabel 4. 3 Perubahan Aktivitas Proses Bisnis Pendaftaran Pasien

No	Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses bisnis <i>to-be</i>	Keterangan
1	Front Office	Melakukan pengecekan pasien melalui berkas pasien	Melakukan pengecekan pasien sebelumnya dengan sistem	Diubah
2	Front Office	Memberikan kartu resume medis	-	Dieliminasi



No	Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses bisnis <i>to-be</i>	Keterangan
3	Tenaga medis	Menerima kartu resume medisa	-	Dieliminasi
4	Tenaga medis	Menulis hasil diagnosis	Menginputkan hasil diagnosis	diubah

**Tabel 4. 4 Perubahan Aktivitas Proses Bisnis Pemeriksaan Fisik**

No	Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses bisnis <i>to-be</i>	Keterangan
1	Dokter	Mecatata hasil pemeriksaan	Menginputkan hasil pemeriksaan ke sistem	Diubah

**Tabel 4. 5 Perubahan Aktivitas Proses Bisnis Pelayanan Obat Pasien**

No	Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses bisnis <i>to-be</i>	Keterangan
1	Petugas Farmasi	Memberikan rincian obat	-	Dieliminasi
2	Front Office	Menerima rincian obat	-	Dieliminasi
3	Petugas Farmasi	Mengecek ketersediaan obat	Melihat ketersediaan obat disistem	Diubah
4	Front office	Merekap rincian biaya	Mengolah rincian biaya	Diubah

**Tabel 4. 6 Perubahan Aktivitas Proses Bisnis Pelayanan Pasien Keluar**

No	Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses bisnis <i>to-be</i>	Keterangan
1	Pasien	Mendaftarkan diri	-	Dieliminasi
2	Front office	Menerima formulir pendaftaran pasien keluar	-	Dieliminasi
3	Front office	Merekap biaya keseluruhan pasien	Mengolah biaya keseluruhan pasien	Diubah

No	Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses bisnis <i>to-be</i>	Keterangan
4	Front office	Membuat Surat pernyataan pasien keluar	Mencetak surat pernyataan pasien keluar	Diubah

#### 4.1.4.1 Analisis Improvement Waktu Proses Bisnis

Proses bisnis *as-is* yang dilakukan peningkatan dengan memodelkan proses bisnis *to-be* dilakukan analisis waktu. Analisis waktu dilakukan untuk menunjukkan peningkatan proses bisnis setelah dilakukan perubahan dalam hal waktu yang dibutuhkan untuk melakukan proses bisnis. Dalam Tabel 4.7 menunjukkan analisis waktu dari proses bisnis yang dilakukan peningkatan. Analisis waktu diambil dari proses unit proses bisnis yang mengalami peningkatan waktu.

Tabel 4. 7 Analisis Waktu

Proses Bisnis	Waktu proses bisnis <i>as-is</i>	Waktu proses bisnis <i>to-be</i>	Keterangan
Pendaftaran pasien baru	10 menit +/-	5 menit +/-	Mempersingkat 5 menit +/-
Pemeriksaan Pasien rutin	14 menit +/-	10 menit +/-	Mempersingkat 4 menit +/-
Pelayanan obat pasien	6 menit +/-	3 menit +/-	Mempersingkat 3 menit +/-
Pelayanan pasien keluar	20 menit +/-	13 menit +/-	Mempersingkat 7 menit +/- detik

#### 4.1.5 Identifikasi Pengguna

Pada tahap ini juga dilakukan identifikasi pengguna (aktor) yang terlibat langsung dalam penggunaan sistem. Dengan dilakukan identifikasi pengguna (aktor) dapat mengetahui karakter dari masing-masing pengguna (aktor).

Tabel 4. 8 Hasil Identifikasi Pengguna

No	Pengguna	Deskripsi
1	Petugas <i>Front Office</i>	Bertugas untuk melakukan pencatatan pendaftaran pasien dan Mengelola data pasien serta sekaligus menjadi administrasi dari Rumah Sakit Khusus HMC Malang .

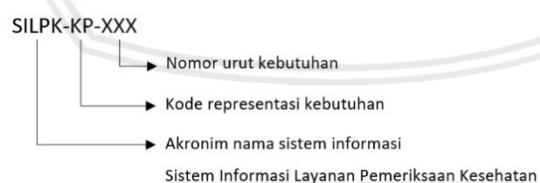
2	Perawat	Bertugas untuk melakukan pemeriksaan terhadap pasien mulai dari mengelola dan resume medik pasien, tindak medis, terapi mencatat pemakaian obat , alat kesehatan, resep obat.
3	Dokter	Bertugas melakukan pemeriksaan fisik lanjutan, melakukan tanya jawab, serta pemeriksaan kejiwaan dari pasien.
4	Petugas Farmasi	Bertugas yang berkaitan dengan kesediaan farmasi, melayani resep dokter, memberikan informasi pemakaian obat, dan melakukan pengolahan apotek.

Penjelasan hasil identifikasi pengguna terdapat pada Tabel 4.8. Informasi mengenai identifikasi pengguna digunakan untuk melakukan identifikasi aktor yang akan dimodelkan ke dalam diagram use case.

#### 4.1.6 Identifikasi Kebutuhan Pengguna

Informasi mengenai kebutuhan pengguna diperoleh dengan melakukan analisis terhadap *problem statement* dan hasil wawancara dengan Wakil Manajer Rumah Sakit Khusus HMC Malang. Kebutuhan pengguna berisi sekumpulan pernyataan yang berhubungan dengan masalah yang telah diidentifikasi pada tahap analisis masalah.

Hasil identifikasi kebutuhan pengguna digunakan sebagai dasar informasi tahap analisis fitur, yang merupakan tahap untuk mengidentifikasi solusi yang ditawarkan untuk memenuhi kebutuhan pengguna. Masing-masing kebutuhan pengguna diberi kode sebagai identitas, keterangan kodifikasi kebutuhan pengguna seperti pada Gambar.



**Gambar 4.9 Kodifikasi Kebutuhan Pengguna**

Tabel 4. 9 Hasil Identifikasi Kebutuhan Pengguna

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi saat ini	Solusi yang ditawarkan
SIRI-KP-1	Sistem harus dapat memberikan layanan pengelola akun pengguna yang berhak melakukan akses terhadap sistem.	Petugas <i>Front Office, Admin</i>	-	Sistem informasi yang menyediakan layanan pengelola akun pengguna yang berhak melakukan akses terhadap sistem.
SIRI-KP-2	Informasi di dalam sistem harus dapat diakses sesuai dengan identitas.	Petugas <i>Front Office, Perawat, Dokter, Petugas Farmasi</i>	-	Sistem informasi yang menyediakan layanan untuk membatasi akses terhadap pengguna dalam menggunakan sistem.
SIRI-KP-3	Sistem harus menyediakan layanan untuk pengelolaan data pasien rawat inap	Petugas front office	Pencatatan dilakukan dengan mengisi formulir data yang disimpan dalam bentuk dokumen cetak.	Sistem informasi yang menyediakan layanan pencatatan data rawat inap. Data disimpan secara historis, di dalam sebuah basis data.

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi saat ini	Solusi yang ditawarkan
SIRI-KP-4	Sistem dapat memberikan layanan untuk pengelolaan data tindakan pada pasien.	Perawat, Dokter	Informasi pencatatan tindakan pemeriksaan pasien ditulis kedalam dokumen tambahan dan diberikan kepada front office	Sistem informasi yang menyediakan layanan pencatatan informasi tindakan pemeriksaan pasien yang dapat ditampilkan oleh petugas kesehatan rumah sakit.
SIRI-KP-5	Sistem dapat diakses sesuai dengan identitas petugas pemeriksaan kesehatan.	Petugas <i>Front Office</i> , Perawat, Dokter, Farmasi	Tidak ada.	Sistem informasi dapat menyediakan layanan untuk membatasi akses terhadap informasi tertentu sesuai dengan hak akses yang dimiliki oleh setiap petugas pemeriksaan kesehatan.
SIRI-KP-6	Sistem dapat membantu proses pendataan rincian biaya perawatan rawat inap	Petugas front office	Perekapan data biaya perawatan rawat inap dilakukan dengan memasukan data biaya setiap pasien satu persatu kedalam Ms. Excel	Sistem informasi yang menyediakan layanan perekapan dan laporan biaya perawatan rawat inap pasien disimpan di dalam sebuah basis data yang dapat diakses melalui sistem informasi.

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi saat ini	Solusi yang ditawarkan
SIRI-KP-7	Sistem harus menyediakan layanan pemakaian obat	Perawat, Dokter.	Resep obat dari dokter yang diberikan kepada pasien dicatat dalam dokumen cetak oleh dokter ataupun perawat.	Sistem informasi yang menyediakan layanan pencatatan resep obat yang merekam informasi mengenai identitas pasien yang menerima resep, obat yang diberikan kepada pasien dan jumlah obat yang diberikan kepada pasien. Setelah itu juga alat kesehatan dan resep tersimpan di dalam sistem
SIRI-KP-8	Sistem menyediakan layanan pendaftaran rawat inap	Petugas front office	Pendaftaran dilakukan dengan mengisi formulir data yang disimpan dalam bentuk dokumen cetak	Sistem informasi yang menyediakan layanan pencatatan data rawat inap. Data disimpan secara historis, di dalam sebuah basis data.

Pada Tabel 4.9 berisi informasi kode identitas untuk masing-masing kebutuhan pengguna, pernyataan kebutuhan pengguna, pemangku kepentingan yang termasuk dalam kategori pengguna dan pengembang, yang memiliki keterkaitan dengan kebutuhan yang telah diidentifikasi, situasi saat ini yang berkaitan dengan kegiatan yang dilakukan untuk memenuhi kebutuhan pengguna,

dan solusi yang ditawarkan untuk memenuhi masing-masing kebutuhan yang telah teridentifikasi.

#### 4.1.6.1 Perubahan dan Penambahan Kebutuhan Pemangku Kepentingan dan Pengguna

Terdapat perubahan pada kebutuhan pemangku kepentingan dan pengguna yaitu penambahan kebutuhan pemangku kepentingan dan pengguna baru pada saat melakukan pengidentifikasian ulang kepada *stakeholder* pada fase elaborasi metode *Rational Unified Process*. setelah dilakukan iterasi fase inception, terdapat penambahan kebutuhan pemangku kepentingan dan pengguna baru pada saat melakukan pengidentifikasian ulang kepada *stakeholder*, yaitu kebutuhan untuk melihat informasi progres pengerjaan data adalah semua pengguna yang diberikan kode SIRI-KP-9. Kebutuhan pemangku kepentingan dan pengguna yang telah dan ditambahkan sesuai dengan Tabel 4.10.

**Tabel 4. 10 Perubahan dan Penambahan Identifikasi Kebutuhan Pengguna**

Kode Kebutuhan	Kebutuhan	Pengguna	Situasi Saat Ini	Solusi yang Ditawarkan
SIRI-KP-9	Kemudahan melihat informasi progress pekerjaan data	Petugas Front Office, Perawat, Dokter, Farmasi	Informasi progress pengerjaan didapatkan dari pemberian paraf pada formulir ketika pemeriksaan selesai dan hasil pemeriksaan telah diisi dalam formulir.	Sistem informasi dapat menyediakan informasi progress pengerjaan data pemeriksaan kesehatan pasien serta menampilkan informasi pengolahan data pemeriksaan.

#### 4.1.7 Identifikasi Fitur

Identifikasi fitur merupakan deskripsi singkat dari layanan-layanan yang disediakan oleh sistem untuk dapat memenuhi kebutuhan pengguna. Nantinya identifikasi fitur ini digunakan sebagai informasi identifikasi kebutuhan fungsional dan non fungsional dari sistem yang akan dikembangkan.

**Tabel 4. 11 Identifikasi Fitur**

No	Kode Fitur	Deskripsi
1	FEAT-01	Sistem menyediakan layanan untuk mengenali pengguna dan membatasi akses pengguna sesuai hak akses pengguna.
2	FEAT-02	Sistem menyediakan layanan untuk menampilkan informasi data detail dari pasien di rawat inap.
3	FEAT-03	Sistem menyediakan layanan untuk menampilkan informasi pemeriksaan kesehatan yaitu catatan dari dokter dan catatan perkembangan dari pasien.
4	FEAT-04	Sistem dapat digunakan untuk mencatat dan menampilkan resep obat yang berisi informasi identitas pasien penerima resep obat, nama obat yang diberikan kepada pasien, dan jumlah obat yang diberikan kepada pasien.
5	FEAT-05	Sistem informasi yang menyediakan layanan pasien keluar yaitu proses perekapan biaya pasien
6	FEAT-06	Sistem menyediakan proses pendaftaran pasien berupa form inputan yang dengan data identitas pasien
7	FEAT-07	Sistem dapat digunakan untuk mengelola akun pengguna.

Dalam Tabel 4.11 menunjukkan informasi fitur beserta penjelasan singkat dari fitur tersebut dari Sistem Informasi Rawat Inap Rumah Sakit Khusus HMC Malang.

#### 4.1.7.1 Perbaikan Identifikasi Fitur

Perbaikan dilakukan pada Identifikasi Fitur disebabkan perbaikan yang dilakukan pada Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna. Tabel 4.12 menunjukkan perbaikan fitur dimana dilakukan penambahan fitur untuk kode fitur FEAT-08. Fitur selanjutnya diperbaiki sesuai penambahan pada Tabel 4.12 sehingga hubungan kebutuhan pengguna dengan fitur diperbaiki seperti pada Tabel 4.14.

**Tabel 4. 12 Perbaikan Identifikasi Fitur**

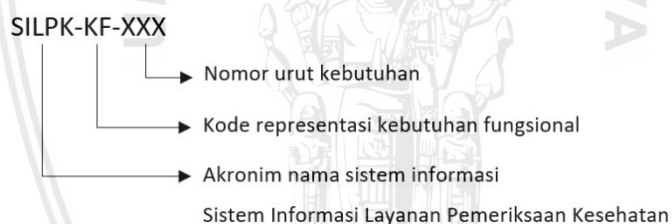
Kode Fitur	Deskripsi
FEAT-08	Sistem dapat menampilkan informasi pelaporan progres data pemeriksaan kepada pengguna.



**Tabel 4. 13 Perbaikan Hubungan Kebutuhan Pengguna Dengan Fitur**

Kebutuhan Pengguna	Fitur
SIRI-KP-001	FEAT-01
SIRI -KP-002	
SIRI -KP-003	FEAT -02
SIRI -KP-004	FEAT -03
	FEAT -04
SIRI -KP-005	FEAT -01
SIRI -KP-006	FEAT -05
SIRI -KP-007	FEAT -06
SIRI -KP-008	FEAT -07
SIRI -KP-009	FEAT-08

#### 4.1.8 Kebutuhan Fungsional



**Gambar 4.10 Kodifikasi Kebutuhan Fungsional**

Pada tahap ini menjelaskan kebutuhan fungsional dari sistem yang akan dikembangkan. Kebutuhan fungsional merupakan kondisi atau kemampuan yang harus dipenuhi oleh sistem agar fitur sistem dapat berjalan dan memenuhi kebutuhan pengguna.

**Tabel 4. 14 Kebutuhan Fungsional**

No	Kode Fitur	Kode Kebutuhan Fungsional	Deskripsi
1	FEAT-01	SIRI-KF-01	Sistem dapat menampilkan halaman login pengguna.
		SIRI-KF-02	Sistem dapat mengidentifikasi pengguna berdasarkan data pengguna didalam sistem.

No	Kode Fitur	Kode Kebutuhan Fungsional	Deskripsi
		SIRI-KF-03	Sistem dapat menyediakan fasilitas bagi pengguna untuk keluar dari sistem.
2	FEAT-02	SIRI-KF-04	Sistem dapat menampilkan identitas detail pasien.
		SIRI-KF-05	Sistem dapat menampilkan cari, view, edit print pasien.
		SIRI-KF-06	Sistem dapat menampilkan riwayat penyakit pasien.
3	FEAT-03	SIRI-KF-07	Sistem dapat menampilkan form inputan catatan dan intruksi dokter
		SIRI-KF-08	Sistem dapat menampilkan cari, view, edit, print catatan intruksi dokter
		SIRI-KF-09	Sistem dapat menampilkan form inputan catatan perkembangan pasien.
		SIRI-KF-10	Sistem dapat cari, view,edit, print catatan perkembangan pasien.
4	FEAT-04	SIRI-KF-11	Sistem dapat menampilkan input rincian pemakaian obat, alat kesehatan, dan stok obat
		SIRI-KF-12	Sistem dapat cari, view,edit, print rincian pemakaian obat, alat kesehatan, dan stock obat
5	FEAT-05	SIRI-KF-13	Sistem dapat menampilkan form input rincian biaya perawatan
		SIRI-KF-14	Sistem dapat menampilkan cari, view,edit, print rincian biaya perawatan
6	FEAT-06	SIRI-KF-15	Sistem dapat menampilkan form pendaftaran pasien rawat inap
7	FEAT-07	SIRI-KF-16	Sistem dapat menambah data pengguna baru
		SIRI-KF-17	Sistem dapat menghapus data pengguna

Tabel 4.14 menunjukkan hubungan antara fitur dengan kebutuhan fungsional sistem yang akan dikembangkan.

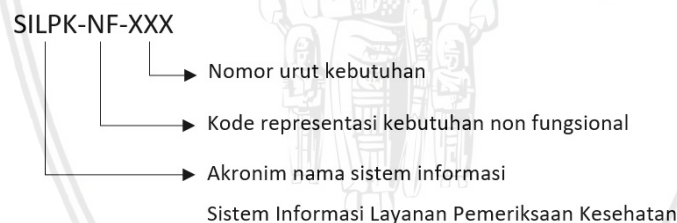
#### 4.1.8.1 Perbaikan Persyaratan Fungsional

Perbaikan dilakukan pada persyaratan fungsional disebabkan perbaikan yang dilakukan pada Identifikasi Fitur dan saat melakukan identifikasi ulang kepada pemangku kepentingan fase insepisi. Tabel 4.15 menunjukkan perbaikan kebutuhan fungsional dimana dilakukan penambahan kebutuhan untuk kode kebutuhan fungsional SIRI-KF-18 dan SIRI-KF-19. Kebutuhan fungsional selanjutnya diperbaiki sesuai penambahan pada Tabel 4.15.

**Tabel 4. 15 Perbaikan Kebutuhan Fungsional**

Kode Fitur	Kode Kebutuhan Fungsional	Deskripsi
FEAT-08	SIRI-KF-18	Sistem dapat menampilkan data jumlah pemeriksaan hari ini dan bulan ini
	SIRI-KF-19	Sistem dapat menampilkan data progress disetiap aktor.

#### 4.1.9 Kebutuhan Non Fungsional



**Gambar 4.11 Kodifikasi Kebutuhan Non Fungsional**

Dalam kebutuhan non fungsional merupakan kebutuhan yang harus dipenuhi oleh sistem agar dapat berjalan sepertiseharusnya. Kebutuhan non fungsional ini didapatkan melalui identifikasi aplikasi *browser* serta sistem operasi yang digunakan pada Rumah Sakit HMC Malang.

**Tabel 4. 16 Analisis Kebutuhan Non-Fungsional**

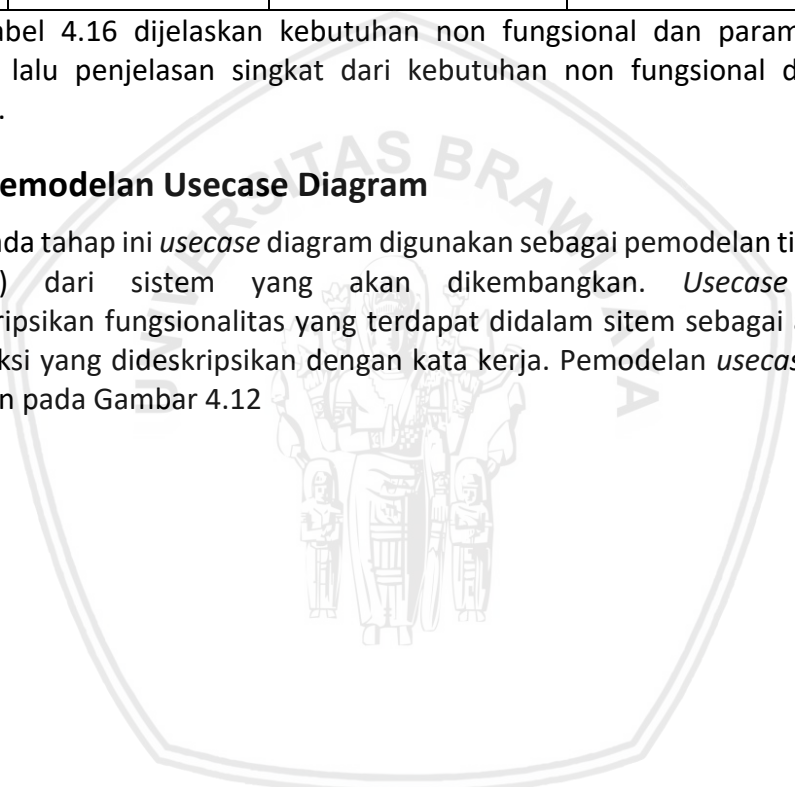
Kode Fitur	Kode Kebutuhan Non Fungsional	Parameter	Deskripsi
FEAT-01	SIRI-NF-1	<i>Compatibility</i>	Perangkat lunak bisa dibuka dan diakses dengan berbagai macam <i>web browser</i> yang berbeda
FEAT-02			

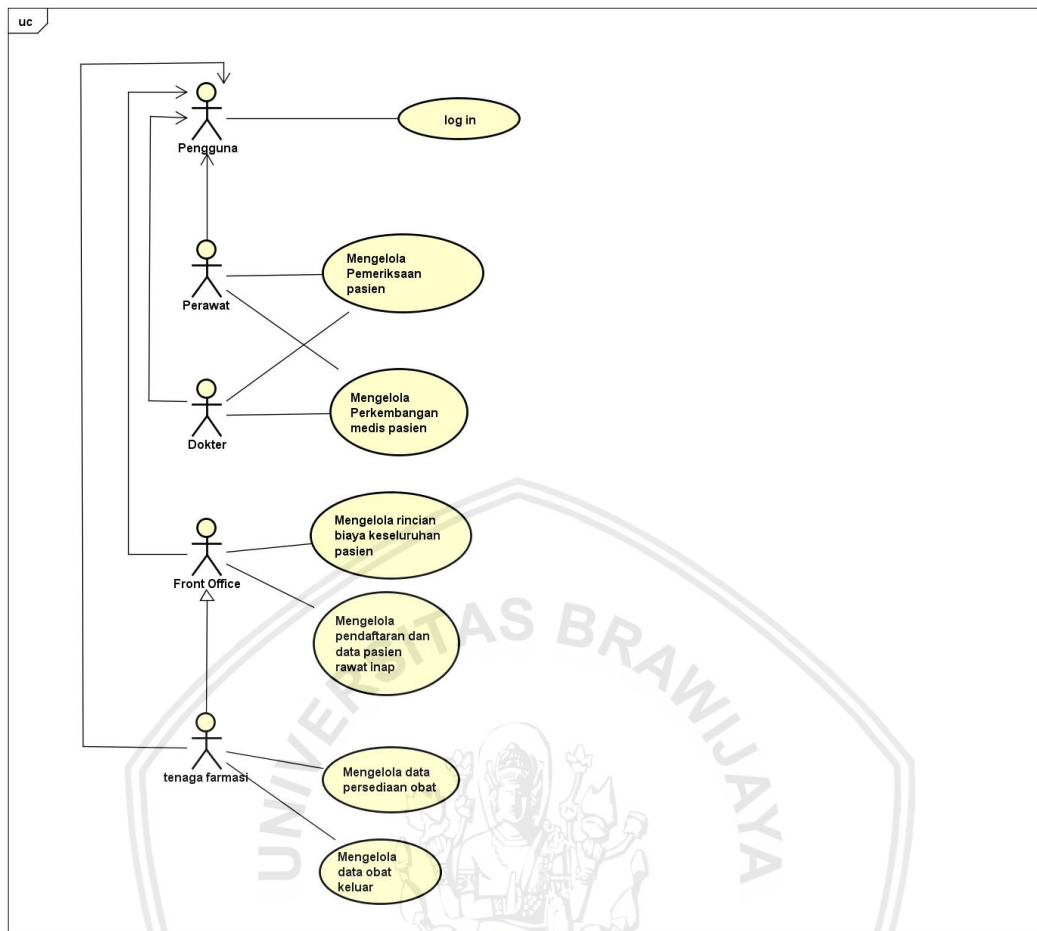
FEAT-03			
FEAT-04			
FEAT-05			
FEAT-06			
FEAT-07			

Pada Tabel 4.16 dijelaskan kebutuhan non fungsional dan parameter yang digunakan lalu penjelasan singkat dari kebutuhan non fungsional dari sistem informasi.

#### 4.1.10 Pemodelan Usecase Diagram

Pada tahap ini *usecase* diagram digunakan sebagai pemodelan tingkah laku (*behavior*) dari sistem yang akan dikembangkan. *Usecase* Diagram mendeskripsikan fungsionalitas yang terdapat didalam sitem sebagai aktor yang berinteraksi yang dideskripsikan dengan kata kerja. Pemodelan *usecase* diagram ditunjukkan pada Gambar 4.12





powered by Astah

Gambar 4.12 Pemodelan usecase diagram

#### 4.1.11 Identifikasi Aktor

Identifikasi aktor merupakan penjelasan singkat mengenai tanggung jawab aktor yang telah teridentifikasi. Aktor sendiri disini merupakan seseorang atau apa saja yang berhubungan dengan sistem yang sedang dibangun yang berada di luar ruang lingkup sistem. Pada Tabel 4.17 menunjukkan identifikasi aktor *front office*, perawat, dokter, dan pengguna.

Tabel 4. 17 Identifikasi Aktor

No	Nama Aktor	Deskripsi
1	Pengguna	Aktor pengguna adalah seluruh staff Rumah Sakit Khusus HMC Malang yang menggunakan sistem secara langsung. Pengguna dapat menggunakan sistem sesuai hak akses yang diberikan oleh sistem.
2	Front Office	Aktor <i>Front Office</i> adalah petugas <i>Front Office</i> yang bertugas sebagai penerima pasien. Dalam sistem <i>Front Office</i> menggunakan untuk mengelola data



No	Nama Aktor	Deskripsi
		permintaan pemeriksaan kesehatan pasien dan data pendaftaran pasien.
3	Perawat	Aktor Perawat adalah perawat yang bertugas untuk melakukan pemeriksaan terhadap pasien mulai dari mengelola dan resume medik pasien, tindak medis, terapi mencatat pemakaian obat, alat kesehatan, resep obat.
4	Dokter	Aktor dokter adalah dokter yang melakukan pemeriksaan fisik lanjutan pada pasien. Dokter menggunakan sistem untuk melihat permintaan pemeriksaan kesehatan dan megolah data hasil pemeriksaan kesehatan fisik.
5	Tenaga Farmasi	Aktor Tenaga farmasi adalah orang yang bertanggung jawab kepada pasien yang berkaitan dengan kesediaan farmasi, melayani resep dokter, memberikan informasi pemakaian obat, dan melakukan pengolahan apotek.

Tabel 4.17 menunjukkan identifikasi aktor dari dalam sistem dimana terdapat penjelasan singkat mengenai tanggung jawab aktor yang telah teridentifikasi. Pada tahap ini juga disajikan penjelasan singkat mengenai tujuan aktor dalam sistem.

#### 4.1.12 Usecase Scenario

Pada tahap ini dilakukan penjelasan mengenai scenario yang berjalan pada masing-masing usecase. Dalam *usecase scenario* memuat informasi mengenai aktor yang menjalankan use case, kondisi sebelum dan sesudah *usecase* dijalankan, serta tahap-tahap yang dilakukan ketika *usecase* dilakukan. Tabel berikut berisi *usecase scenario* dari masing-masing *usecase*.

**Tabel 4. 18 Use case Scenario login**

Flow of Events untuk use case melakukan login	
<b>Brief Description</b>	Use case ini menggambarkan bagaimana aktor masuk kedalam sistem
<b>Actor</b>	Pengguna, <i>Front Office</i> , Dokter, Perawat
<b>Pre-conditions</b>	1. Aktor telah terdaftar dan memiliki akun dalam sistem
<b>Basic flow of Events</b>	1. Aktor mengakses halaman login

	<p>Aktor mengakses halaman utama login sistem informasi rawat inap Rumah Sakit HMC.</p> <p><b>1. Sistem menampilkan kolom login</b></p> <p>Sistem menampilkan kolom <i>username</i> dan <i>password</i>.</p> <p><b>2. Aktor menginputkan username dan password</b></p> <p>Aktor memasukkan <i>username</i> dan <i>password</i> yang sesuai untuk dapat masuk ke dalam sistem informasi rawat inap Rumah Sakit HMC.</p> <p><b>3. Sistem validasi <i>username</i> dan <i>password</i></b></p> <p>Sistem melakukan validasi <i>username</i> dan <i>password</i> yang dimasukkan bila sudah sesuai maka <i>user</i> akan masuk ke dalam sistem sesuai dengan hak aksesnya.</p>
<b>Alternative Flows</b>	<p><b>A1. Login gagal</b></p> <ol style="list-style-type: none"> <li>1. Username atau <i>password</i> salah</li> <li>2. Login gagal dan menampilkan pemberitahuan</li> <li>3. Kembali kehalaman login</li> </ol>
<b>Key Scenarios</b>	Data tidak sesuai format : Basic flow, A1, Basic flow
<b>Post-conditions</b>	Jika Aktor berhasil maka <i>user</i> masuk ke dalam sistem sesuai hak akses, jika tidak maka status sistem tidak berubah.

Penjelasan tabel 4.18 tentang use case scenario login dimulai dari sistem menampilkan halaman *login* ketika *user* ingin mengakses sistem. Aktor *user* lalu memasukkan *username* dan *password* untuk autentifikasi. Bila data yang dimasukkan aktor *user* sesuai maka akan masuk ke dalam sistem sesuai hak akses. Sementara bila data yang dimasukkan tidak sesuai maka sistem akan kembali menampilkan halaman *form login*.

**Tabel 4. 19 Usecase Scenario Mengelola Data Persediaan Obat**

Flow of Events untuk use case mengelola data persediaan obat	
<b>Brief Description</b>	Use case ini menggambarkan bagaimana aktor mengelola data persediaan obat
<b>Actor</b>	<i>Tenaga Farmasi</i>
<b>Pre-conditions</b>	1. Aktor telah masuk kedalam sistem

	<p>4. Data pendaftaran pasien telah tersimpan dalam sistem</p>
<b>Basic flow of Events</b>	<p><b>1. Aktor memilih fungsi rincian persediaan obat</b> Perawat memilih fungsi rincian persediaan obat untuk dapat mengelola data persediaan obat pasien di rawat inap</p> <p><b>2. Sistem menampilkan list rincian persediaan obat</b> Sistem menampilkan list rincian persediaan obat pasien di rawat inap yang telah tersimpan di database.</p> <p><b>3. Aktor memilih fungsi input tambah obat</b> Perawat memilih fungsi input tambah persediaan untuk dapat menambahkan data rincian persediaan obat.</p> <p><b>4. Sistem menampilkan form persediaan obat</b> Sistem menampilkan form persediaan obat seperti tanggal, nama obat, jumlah.</p> <p><b>5. Aktor mengisi form persediaan obat</b> Perawat mengisi data sesuai dengan formulir persediaan obat pasien di rawat inap</p> <p><b>6. Aktor melakukan submit data persediaan obat</b> Pegawai farmasi yang telah memasukan data persediaan obat memilih fungsi <i>submit</i> untuk memasukan data persediaan obat ke dalam sistem dan tersimpan dalam <i>database</i>.</p> <p><b>7. Sistem Menampilkan kembali list rincian persediaan obat</b> Data rincian persediaan obat pasien yang berhasil dimasukan akan menampilkan kembali seluruh daftar rincian persediaan obat .</p>
<b>Alternative Flows</b>	<p><b>A1. Invalid data</b> Ketika Perawat melakukan <i>submit</i> data persediaan obat, sistem menampilkan pesan <i>error</i> pada kolom dengan masukan yang tidak sesuai dengan format. Data tersebut tidak disimpan ke dalam sistem.</p> <p><b>A2. Menyunting data pemakaian obat</b> Saat sistem menampilkan data persediaan obat, bila pegawai farmasi ingin melakukan perubahan</p>





	<p>data persediaan obat, perawat memilih fungsi sunting. Dengan fungsi sunting maka sistem akan menampilkan formulir persediaan obat yang telah terisi dengan isian yang disimpan sebelumnya. Use case kembali pada saat pengisian data persediaan obat.</p> <p><b>A3. Menghapus data pemakaian obat</b></p> <p>Ketika perawat ingin melakukan penghapusan data pemakaian obat, perawat memilih fungsi hapus. Dengan fungsi hapus maka sistem akan menghapus data pemakaian obat pasien yang ingin di hapus.</p>
<b>Key Scenarios</b>	<ol style="list-style-type: none"> <li>1. Menambah data rincian persediaan obat : Basic flow</li> <li>2. Data tidak sesuai format : Basic flow, A1, Basic flow</li> <li>3. Menyunting data rincian persediaan obat: Basic flow. A2, Basic flow</li> <li>4. Menghapus data rincian persediaan obat : Basic flow, A3</li> </ol>
<b>Post-conditions</b>	<ul style="list-style-type: none"> <li>• Sistem berhasil menyimpan rincian persediaan obat</li> <li>• Aktor berhasil melihat data rincian persediaan obat</li> </ul>

Penjelasan tabel 4.19 tentang use case scenario data persediaan obat yaitu jika pengguna memilih fungsi tambah maka akan menjalankan scenario menambah persediaan obat, jika pengguna memilih fungsi ubah maka akan menjalankan scenario mengubah persediaan obat, jika pengguna memilih fungsi hapus maka akan menjalankan scenario menghapus data persediaan obat, lalu jika pengguna memilih fungsi melihat data keseluruhan maka akan menjalankan scenario melihat rincian data persediaan obat.

Tabel 4. 20 Use case Scenario Mengelola Pemeriksaan Pasien

Flow of Events untuk use case mengelola pemeriksaan fisik pasien	
<b>Brief Description</b>	Use case ini menggambarkan bagaimana aktor mengelola pemeriksaan fisik pasien
<b>Actor</b>	Dokter, Perawat
<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. Aktor telah masuk kedalam sistem</li> <li>2. Data pendaftaran pasien telah tersimpan dalam sistem</li> </ol>
<b>Basic flow of Events</b>	<ol style="list-style-type: none"> <li> <b>1. Aktor memilih fungsi pemeriksaan fisik pasien</b>                      Aktor memilih fungsi pemeriksaan fisik untuk dapat mengelola data pemeriksaan fisik pasien pada saat pemeriksaan awal.                 </li> <li> <b>2. Sistem menampilkan list pemeriksaan fisik pasien</b>                      Sistem menampilkan list pemeriksaan fisik pasien pada saat pemeriksaan awal yang telah tersimpan di database.                 </li> <li> <b>3. Aktor memilih fungsi input tambah pemeriksaan fisik</b>                      Aktor memilih fungsi input tambah pemeriksaan fisik untuk dapat menambahkan data pemeriksaan fisik pasien.                 </li> <li> <b>4. Sistem menampilkan form pemeriksaan fisik</b>                      Sistem menampilkan form pemeriksaan fisik Nama, usia, jenis kelamin, diakhnosa primer, diakhnosa sekunder tindakan medis dll.                 </li> <li> <b>5. Aktor mengisi form pemeriksaan fisik</b>                      Aktor mengisi data sesuai dengan formulir pemeriksaan fisik pasien .                 </li> <li> <b>6. Aktor melakukan submit data pemeriksaan fisik</b>                      Aktor yang telah memasukan data pemeriksaan fisik memilih fungsi <i>submit</i> untuk memasukan data pemeriksaan fisik ke dalam sistem dan tersimpan dalam <i>database</i>.                 </li> </ol>



	<p><b>7. Sistem Menampilkan kembali list pemeriksaan fisik</b></p> <p>Data pemeriksaan fisik pasien yang berhasil dimasukan akan menampilkan kembali seluruh daftar pemeriksaan fisik seluruh pasien.</p>
<b>Alternative Flows</b>	<p><b>A1. Invalid data</b></p> <p>Ketika Terapis melakukan <i>submit</i> data pemeriksaan fisik, sistem menampilkan pesan <i>error</i> pada kolom dengan masukan yang tidak sesuai dengan format. Data tersebut tidak disimpan ke dalam sistem.</p> <p><b>A2. Menyunting data pemeriksaan fisik</b></p> <p>Saat sistem menampilkan data pemeriksaan fisik, bila aktor ingin melakukan perubahan data pemeriksaan fisik memilih fungsi sunting. Dengan fungsi sunting maka sistem akan menampilkan formulir pemeriksaan fisik yang telah terisi dengan isian yang disimpan sebelumnya. Use case kembali pada saat pengisian data pemeriksaan fisik.</p> <p><b>A3. Menghapus data pemeriksaan fisik</b></p> <p>Ketika aktor ingin melakukan penghapusan data pemeriksaan fisik, aktor memilih fungsi hapus. Dengan fungsi hapus maka sistem akan menghapus data pemeriksaan fisik pasien yang ingin di hapus.</p>
<b>Key Scenarios</b>	<ol style="list-style-type: none"> <li>1. Menambah data pemeriksaan fisik: Basic flow</li> <li>2. Data tidak sesuai format : Basic flow, A1, Basic flow</li> <li>3. Menyunting data pemeriksaan fisik: Basic flow. A2, Basic flow</li> <li>4. Menghapus data pemeriksaan fisik: Basic flow, A3</li> </ol>
<b>Post-conditions</b>	<p>Sistem berhasil menyimpan data pemeriksaan fisik.</p> <p>Aktor berhasil melihat data pemeriksaan fisik.</p>

Penjelasan tabel 4.20 tentang use case scenario data pemeriksaan fisik pasien yaitu jika pengguna memilih fungsi tambah maka akan menjalankan scenario menambah pemeriksaan fisik, jika pengguna memilih fungsi ubah maka akan menjalankan scenario mengubah pemeriksaan fisik, jika pengguna memilih fungsi hapus maka akan menjalankan scenario menghapus data pemeriksaan fisik,

lalu jika pengguna memilih fungsi melihat data keseluruhan pemeriksaan pasien maka akan menjalankan scenario melihat rincian data pemeriksaan fisik pasien.

**Tabel 4. 21 Use case Scenario Mengelola Perkembangan Medis Pasien**

Flow of Events untuk use case mengelola data perkembangan medis pasien	
<b>Brief Description</b>	Use case ini menggambarkan bagaimana aktor mengelola data perkembangan medis pasien
<b>Actor</b>	Perawat, Dokter
<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. Aktor telah masuk kedalam sistem</li> <li>2. Data pendaftaran pasien telah ada dalam sistem</li> </ol>
<b>Basic flow of Events</b>	<ol style="list-style-type: none"> <li> <b>1. Aktor memilih fungsi perkembangan medis pasien</b>                      Aktor memilih fungsi perkembangan medis pasien untuk dapat mengelola data perkembangan pasien selama di rawat inap.                 </li> <li> <b>2. Sistem menampilkan list perkembangan medis pasien</b>                      Sistem menampilkan list perkembangan medis pasien selama di rawat inap yang telah tersimpan di database.                 </li> <li> <b>3. Aktor memilih fungsi input tambah perkembangan medis pasien</b>                      Aktor memilih fungsi input tambah perkembangan medis pasien untuk dapat menambahkan data perkembangan medis pasien.                 </li> <li> <b>4. Sistem menampilkan form perkembangan medis pasien</b>                      Sistem menampilkan form perkembangan medis pasien seperti Nama, tanggal, Keluhan, keadaan, masalah pasien, diagnose, tindakan keperawatan.                 </li> <li> <b>5. Aktor mengisi form perkembangan medis pasien</b>                      Aktor mengisi data sesuai dengan formulir perkembangan medis pasien .                 </li> <li> <b>6. Aktor melakukan submit data perkembangan medis pasien</b>                      Aktor yang telah memasukan data perkembangan medis pasien memilih fungsi                 </li> </ol>

	<p><i>submit</i> untuk memasukkan data perkembangan medis pasien ke dalam sistem dan tersimpan dalam <i>database</i>.</p> <p><b>7. Sistem Menampilkan kembali list perkembangan medis pasien</b></p> <p>Data perkembangan medis pasien yang berhasil dimasukan akan menampilkan kembali seluruh daftar perkembangan medis seluruh pasien.</p>
<b>Alternative Flows</b>	<p><b>A1. Invalid data</b></p> <p>Ketika Aktor melakukan <i>submit</i> data perkembangan medis pasien, sistem menampilkan pesan <i>error</i> pada kolom dengan masukan yang tidak sesuai dengan format. Data tersebut tidak disimpan ke dalam sistem.</p> <p><b>A2. Menyunting data perkembangan medis pasien</b></p> <p>Saat sistem menampilkan data perkembangan medis pasien, bila aktor ingin melakukan perubahan data perkembangan medis pasien memilih fungsi sunting. Dengan fungsi sunting maka sistem akan menampilkan formulir perkembangan medis pasien yang telah terisi dengan isian yang disimpan sebelumnya. Use case kembali pada saat pengisian data perkembangan medis pasien.</p> <p><b>A3. Menghapus data perkembangan medis pasien</b></p> <p>Ketika aktor ingin melakukan penghapusan data perkembangan medis pasien, aktor memilih fungsi hapus. Dengan fungsi hapus maka sistem akan menghapus data perkembangan medis pasien yang ingin di hapus.</p>
<b>Key Scenarios</b>	<ol style="list-style-type: none"> <li>1. Menambah data perkembangan medis pasien: Basic flow</li> <li>2. Data tidak sesuai format : Basic flow, A1, Basic flow</li> <li>3. Menyunting data perkembangan medis pasien: Basic flow. A2, Basic flow</li> </ol>

	4. Menghapus data perkembangan medis pasien: Basic flow, A3
<b>Post-conditions</b>	Sistem berhasil menyimpan data perkembangan medis pasien.  Aktor berhasil melihat data perkembangan medis pasien.

Penjelasan tabel 4.21 tentang use case scenario data perkembangan medis pasien yaitu jika pengguna memilih fungsi tambah maka akan menjalankan scenario menambah perkembangan medis pasien, jika pengguna memilih fungsi ubah maka akan menjalankan scenario mengubah perkembangan medis pasien, jika pengguna memilih fungsi hapus maka akan menjalankan scenario menghapus data perkembangan medis pasien, lalu jika pengguna memilih fungsi melihat data keseluruhan perkembangan medis pasien maka akan menjalankan scenario melihat rincian data perkembangan medis pasien.

**Tabel 4. 22 Use case Scenario Mengelola Rincian Biaya Pemakaian Obat**

Flow of Events untuk use case mengelola data rincian biaya pemakaian obat	
<b>Brief Description</b>	Use case ini menggambarkan bagaimana aktor mengelola data rincian biaya pemakaian obat
<b>Actor</b>	Petugas farmasi
<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. Aktor telah masuk kedalam sistem</li> <li>2. Data pendaftaran pasien telah ada dalam sistem</li> <li>3. Data rincian pemakaian obat dari perawat telah ada dalam sistem</li> </ol>
<b>Basic flow of Events</b>	<ol style="list-style-type: none"> <li>1. <b>Aktor memilih fungsi rincian biaya pemakaian obat</b>  Aktor memilih fungsi rincian biaya pemakaian obat untuk dapat mengelola data rincian biaya pemakaian obat pasien selama di rawat inap.</li> <li>2. <b>Sistem menampilkan list rincian biaya pemakaian obat</b>  Sistem menampilkan list rincian biaya pemakaian obat pasien selama di rawat inap yang telah tersimpan di database.</li> <li>3. <b>Sistem menyunting data rincian biaya pemakaian obat</b>  Sistem menampilkan data rincian biaya pemakaian obat, bila aktor harus melakukan</li> </ol>



	<p>perubahan data rincian biaya pemakaian obat memilih fungsi sunting. Dengan fungsi sunting maka sistem akan menampilkan formulir pemeriksaan fisik yang telah terisi dengan isian yang disimpan sebelumnya dari rincian biaya obat dari perawat.</p> <p><b>4. Sistem menampilkan form rincian biaya pemakaian obat</b></p> <p>Sistem menampilkan form rincian biaya pemakaian obat seperti tanggal, nama pasien, nama obat, jumlah, dan harga</p> <p><b>5. Aktor mengisi form biaya pemakaian obat</b></p> <p>Aktor mengisi data sesuai dengan formulir rincian biaya pemakaian obat pasien terutama pada kolom harga.</p> <p><b>6. Aktor melakukan submit data biaya pemakaian obat</b></p> <p>Aktor yang telah memasukan data biaya pemakaian obat memilih fungsi <i>submit</i> untuk memasukkan data biaya pemakaian obat ke dalam sistem dan tersimpan dalam <i>database</i>.</p> <p><b>7. Sistem Menampilkan kembali list rincian biaya pemakaian obat</b></p> <p>Data biaya rincian biaya pemakaian obat pasien yang berhasil dimasukan akan menampilkan kembali seluruh daftar biaya pemakaian obat seluruh pasien.</p>
<p><b>Alternative Flows</b></p>	<p><b>A1. Invalid data</b></p> <p>Ketika Terapis melakukan <i>submit</i> data rincian biaya pemakaian obat, sistem menampilkan pesan <i>error</i> pada kolom dengan masukan yang tidak sesuai dengan format. Data tersebut tidak disimpan ke dalam sistem.</p> <p><b>A3. Menghapus data rincian biaya pemakaian obat</b></p> <p>Ketika aktor ingin melakukan penghapusan data data rincian biaya pemakaian obat, aktor memilih fungsi hapus. Dengan fungsi hapus maka sistem</p>

	akan menghapus data rincian biaya pemakaian obat pasien yang ingin di hapus.
<b>Key Scenarios</b>	<ol style="list-style-type: none"> <li>1. Menyunting data pemeriksaan fisik pasien: Basic flow.</li> <li>2. Data tidak sesuai format : Basic flow, A1, Basic flow</li> <li>3. Menghapus data pemeriksaan fisik pasien: Basic flow, A3</li> </ol>
<b>Post-conditions</b>	<p>Sistem berhasil menyimpan data rincian biaya pemakaian obat.</p> <p>Aktor berhasil melihat data rincian biaya pemakaian obat.</p>

Penjelasan tabel 4.22 tentang use case scenario data biaya pemakaian obat yaitu jika pengguna memilih fungsi tambah maka akan menjalankan scenario menambah biaya pemakaian obat, jika pengguna memilih fungsi ubah maka akan menjalankan scenario mengubah biaya pemakaian obat, jika pengguna memilih fungsi hapus maka akan menjalankan scenario menghapus data biaya pemakaian obat, lalu jika pengguna memilih fungsi melihat data keseluruhan maka akan menjalankan scenario melihat rincian data biaya pemakaian obat pasien.

**Tabel 4. 23 Use case Scenario Mengelola Rincian Biaya Keseluruhan Pasien**

Flow of Events untuk use case melihat rincian biaya keseluruhan pasien	
<b>Brief Description</b>	Use case ini menggambarkan bagaimana aktor mngelola rincian biaya keseluruhan pasien
<b>Actor</b>	Front office
<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. Aktor telah masuk kedalam sistem</li> <li>2. Data permintaan pemeriksaan kesehatan telah tersimpan dalam sistem</li> <li>3. Data rincian biaya pemakaian obat dari petugas farmasi telah ada dalam sistem</li> </ol>
<b>Basic flow of Events</b>	<p><b>1. Aktor memilih fungsi rincian biaya keseluruhan pasien</b></p> <p>Perawat memilih fungsi rincian biaya keseluruhan pasien untuk dapat mengelola data biaya keseluruhan pasien selama di rawat inap</p>



	<p><b>2. Sistem menampilkan list rincian biaya keseluruhan pasien</b></p> <p>Sistem menampilkan list rincian biaya keseluruhan pasien selama di rawat inap termasuk yaitu biaya pemakaian obat dan biaya kamar inap dll yang telah tersimpan di database.</p> <p><b>3. Aktor memilih fungsi input tambah biaya pasien</b></p> <p>Perawat memilih fungsi input tambah pemakaian untuk dapat menambahkan data rincian biaya keseluruhan pasien.</p> <p><b>4. Sistem menampilkan form biaya pasien</b></p> <p>Sistem menampilkan form biaya keseluruhan pasien seperti tanggal, nama pasien, nama obat, jumlah, biaya obat, biaya kamar, biaya logistic, biaya tambahan lainnya.</p> <p><b>5. Aktor mengisi form biaya pasien</b></p> <p>Front Office mengisi data sesuai dengan formulir biaya keseluruhan pasien selama di rawat inap</p> <p><b>6. Aktor melakukan submit data biaya pasien</b></p> <p>Front Office yang telah memasukan data biaya keseluruhan pasien memilih fungsi <i>submit</i> untuk memasukkan data biaya keseluruhan ke dalam sistem dan tersimpan dalam <i>database</i>.</p> <p><b>7. Sistem Menampilkan kembali list rincian biaya keseluruhan pasien</b></p> <p>Data rincian biaya keseluruhan pasien yang berhasil dimasukan akan menampilkan kembali seluruh daftar rincian biaya keseluruhan pasien.</p>
<p><b>Alternative Flows</b></p>	<p><b>A1. Invalid data</b></p> <p>Ketika Terapis melakukan <i>submit</i> data rincian biaya keseluruhan pasien, sistem menampilkan pesan <i>error</i> pada kolom dengan masukan yang tidak sesuai dengan format. Data tersebut tidak disimpan ke dalam sistem.</p> <p><b>A2. Menyunting data rincian biaya keseluruhan pasien</b></p> <p>Saat sistem menampilkan data rincian biaya keseluruhan pasien, bila aktor ingin melakukan perubahan data rincian biaya keseluruhan pasien</p>

	<p>memilih fungsi sunting. Dengan fungsi sunting maka sistem akan menampilkan formulir rincian biaya keseluruhan pasien yang telah terisi dengan isian yang disimpan sebelumnya. Use case kembali pada saat pengisian data rincian biaya keseluruhan pasien.</p> <p><b>A3. Menghapus data rincian biaya keseluruhan pasien</b></p> <p>Ketika aktor ingin melakukan penghapusan data rincian biaya keseluruhan pasien, aktor memilih fungsi hapus. Dengan fungsi hapus maka sistem akan menghapus data rincian biaya keseluruhan pasien yang ingin di hapus.</p>
<b>Key Scenarios</b>	<ol style="list-style-type: none"> <li>1. Menambah data rincian biaya keseluruhan pasien: Basic flow</li> <li>2. Data tidak sesuai format : Basic flow, A1, Basic flow</li> <li>3. Menyunting data rincian biaya keseluruhan pasien: Basic flow. A2, Basic flow</li> <li>4. Menghapus data rincian biaya keseluruhan pasien: Basic flow, A3</li> </ol>
<b>Post-conditions</b>	<p>Sistem berhasil menyimpan data rincian biaya keseluruhan pasien.</p> <p>Aktor berhasil melihat data rincian biaya keseluruhan pasien.</p>

Penjelasan tabel 4.23 tentang use case scenario data biaya keseluruhan pasien yaitu jika pengguna memilih fungsi tambah maka akan menjalankan scenario menambah biaya keseluruhan pasien, jika pengguna memilih fungsi ubah maka akan menjalankan scenario mengubah biaya keseluruhan pasien, jika pengguna memilih fungsi hapus maka akan menjalankan scenario menghapus data biaya keseluruhan pasien, lalu jika pengguna memilih fungsi melihat data keseluruhan maka akan menjalankan scenario melihat rincian data biaya keseluruhan pasien.

Tabel 4. 24 *Use case Scenario* Mengelola Pendaftaran dan Data Pasien Rawat Inap

Flow of Events untuk use case mengelola pendaftaran dan data pasien rawat inap	
<b>Brief Description</b>	Use case ini menggambarkan bagaimana aktor melihat mengelola pendaftaran dan data pasien rawat inap
<b>Actor</b>	Front Office
<b>Pre-conditions</b>	1. Aktor telah masuk kedalam sistem
<b>Basic flow of Events</b>	<p><b>1. Aktor memilih fungsi pendaftaran dan data pasien</b></p> <p>Front Office memilih fungsi pendaftaran dan data pasien untuk dapat mengelola data pasien selama di rawat inap</p> <p><b>2. Sistem menampilkan list pendaftaran dan data pasien</b></p> <p>Sistem menampilkan list pendaftaran dan data pasien selama di rawat inap yang telah tersimpan di database.</p> <p><b>3. Aktor memilih fungsi input tambah pendaftaran pasien</b></p> <p>Perawat memilih fungsi input tambah pendaftaran pasien untuk dapat menambahkan data pasien baru.</p> <p><b>4. Sistem menampilkan form pendaftaran pasien</b></p> <p>Sistem menampilkan form pendaftaran pasien seperti tanggal, nama pasien, usia, jenis kelamin, pekerjaan, staus, alamat dll.</p> <p><b>5. Aktor mengisi form layanan pendaftaran data pasien</b></p> <p>Aktor mengisi data sesuai dengan formulir pendaftaran pasien .</p> <p><b>6. Aktor melakukan submit data pendaftaran pasien</b></p> <p>Aktor yang telah memasukan data pendaftaran pasien memilih fungsi <i>submit</i> untuk memasukkan data pendaftar baru ke dalam sistem dan tersimpan dalam <i>database</i>.</p>

	<p><b>7. Sistem Menampilkan kembali list pendaftaran dan data pasien rawat inap</b></p> <p>Data pendaftar pasien yang berhasil dimasukan akan menampilkan kembali seluruh daftar pasien di rumah sakit HMC.</p>
<b>Alternative Flows</b>	<p><b>A1. Invalid data</b></p> <p>Ketika Terapis melakukan <i>submit</i> data pendaftaran pasien keseluruhan pasien, sistem menampilkan pesan <i>error</i> pada kolom dengan masukan yang tidak sesuai dengan format. Data tersebut tidak disimpan ke dalam sistem.</p> <p><b>A2. Menyunting data pendaftaran pasien</b></p> <p>Saat sistem menampilkan data pendaftaran pasien, bila aktor ingin melakukan perubahan data pendaftaran pasien memilih fungsi sunting. Dengan fungsi sunting maka sistem akan menampilkan formulir pendaftaran pasien yang telah terisi dengan isian yang disimpan sebelumnya. Use case kembali pada saat pengisian data pendaftaran pasien.</p> <p><b>A3. Menghapus data rincian pendaftaran pasien</b></p> <p>Ketika aktor ingin melakukan penghapusan data pendaftaran pasien, aktor memilih fungsi hapus. Dengan fungsi hapus maka sistem akan menghapus data pendaftaran pasien yang ingin di hapus.</p>
<b>Key Scenarios</b>	<ol style="list-style-type: none"> <li>1. Menambah data rincian pendaftaran pasien: Basic flow</li> <li>2. Data tidak sesuai format : Basic flow, A1, Basic flow</li> <li>3. Menyunting data pendaftaran pasien: Basic flow. A2, Basic flow</li> <li>4. Menghapus data pendaftaran pasien: Basic flow, A3</li> </ol>
<b>Post-conditions</b>	<p>Sistem berhasil menyimpan data pendaftaran pasien.</p> <p>Aktor berhasil melihat data pendaftaran pasien.</p>

Penjelasan tabel 4.24 tentang use case scenario data pendaftaran pasien yaitu jika pengguna memilih fungsi tambah maka akan menjalankan scenario

menambah pendaftaran pasien, jika pengguna memilih fungsi ubah maka akan menjalankan scenario mengubah data pasien, jika pengguna memilih fungsi hapus maka akan menjalankan scenario menghapus data pasien, lalu jika pengguna memilih fungsi melihat data keseluruhan maka akan menjalankan scenario melihat rincian data seluruh pasien.

**Tabel 4. 25 Use case Scenario Log out**

Flow of Events untuk use case Log out	
<b>Brief Description</b>	Use Case ini menjelaskan pengguna keluar dari sistem informasi Rawat Inap Rumah Sakit HMC Malang
<b>Actor</b>	Pengguna
<b>Pre-conditions</b>	1. Aktor telah masuk kedalam sistem
<b>Basic flow of Events</b>	<p><b>1. User memilih fungsi logout</b></p> <p>User mengakses fitur <i>logout</i> dengan melakukan aksi klik.</p> <p><b>2. Sistem menghapus user session</b></p> <p>Sistem menghapus <i>user session</i> lalu menampilkan halaman utama login.</p>
<b>Alternative Flows</b>	-
<b>Key Scenarios</b>	-
<b>Post-conditions</b>	User keluar dari sistem informasi Rawat Inap Rumah Sakit HMC Malang.

Penjelasan tabel 4.25 tentang use case scenario logout dimulai oleh aktor user/administrator memilih fungsi logout, sistem lalu akan menghapus *user session*. Sistem menampilkan halaman awal login.

#### **4.1.13 Kesimpulan Analisis Kebutuhan Fase Insepsi Metode Rational Unifield Proccess**

Analisis kebutuhan pada fase insepsi telah dilakukan, dimulai dengan melakukan analisis proses bisnis saat ini *as-is* yang menghasilkan usulan proses bisnis *to-be* sebagai hasil proses bisnis rekomendasi yang ditawarkan pemangku kepentingan dan sebagai masukan dalam kegiatan analisis persyaratan. Analisis persyaratan kemudian dilakukan yang memuat identifikasi tipe pemangku kepentingan, masalah yang dihadapi oleh pemangku kepentingan, dan identifikasi kebutuhan pemangku kepentingan dan pengguna. Sehingga hasil analisis persyaratan didapat 8 fitur, 22 persyaratan fungsional dan 1 persyaratan non fungsional.

Untuk memastikan kebutuhan dari pemangku kepentingan dan pengguna peneliti melakukan pengulangan analisis kebutuhan kepada pemangku kepentingan. Hasil perbaikan identifikasi kebutuhan pengguna pada kode kebutuhan SIRI-KP-09 dimana pengguna menambahkan untuk sistem agar dapat menampilkan informasi progress pemeriksaan. Dengan adanya perubahan kebutuhan pengguna dilakukan juga perbaikan fitur dengan melakukan penambahan fitur dengan kode FEAT-08 Sistem dapat menampilkan informasi pelaporan pengolahan data pemeriksaan kepada pengguna. Perbaikan yang dilakukan pada persyaratan fungsional pada kode SIRI-KF-18 Sistem dapat menampilkan data jumlah pemeriksaan hari ini dan bulan ini , SILPK-KF-19 Sistem dapat menampilkan data progress disetiap aktor. Hasil akhir analisis persyaratan didapat 8 fitur, 19 persyaratan fungsional dan 1 persyaratan nonfungsional. Kebutuhan analisis persyaratan kemudian telah diterima atau disetujui oleh pemangku kepentingan. *Lifecycle objective milestone* fase insepisi pada RUP maka telah terpenuhi.

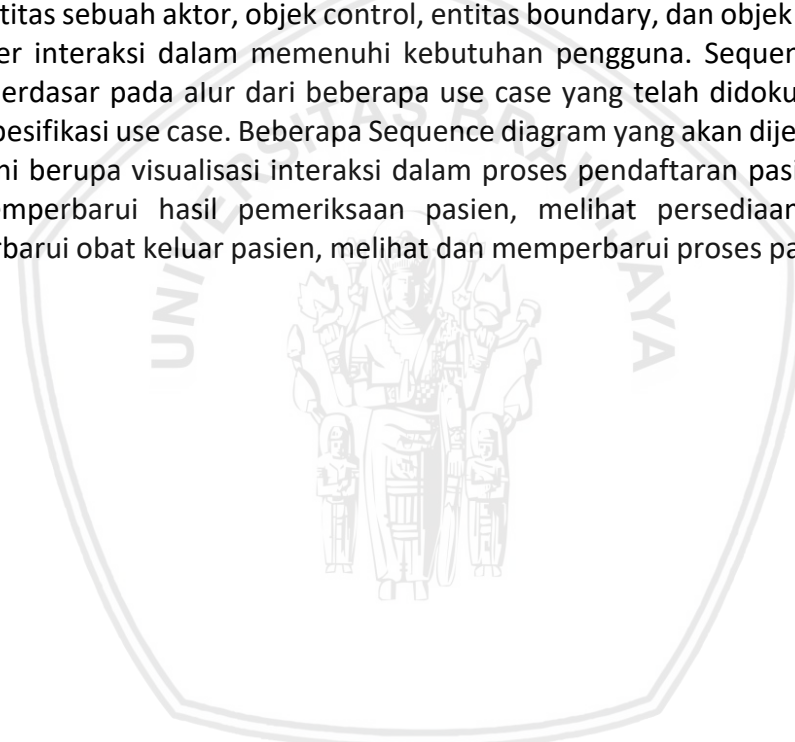


## BAB 5 PERANCANGAN

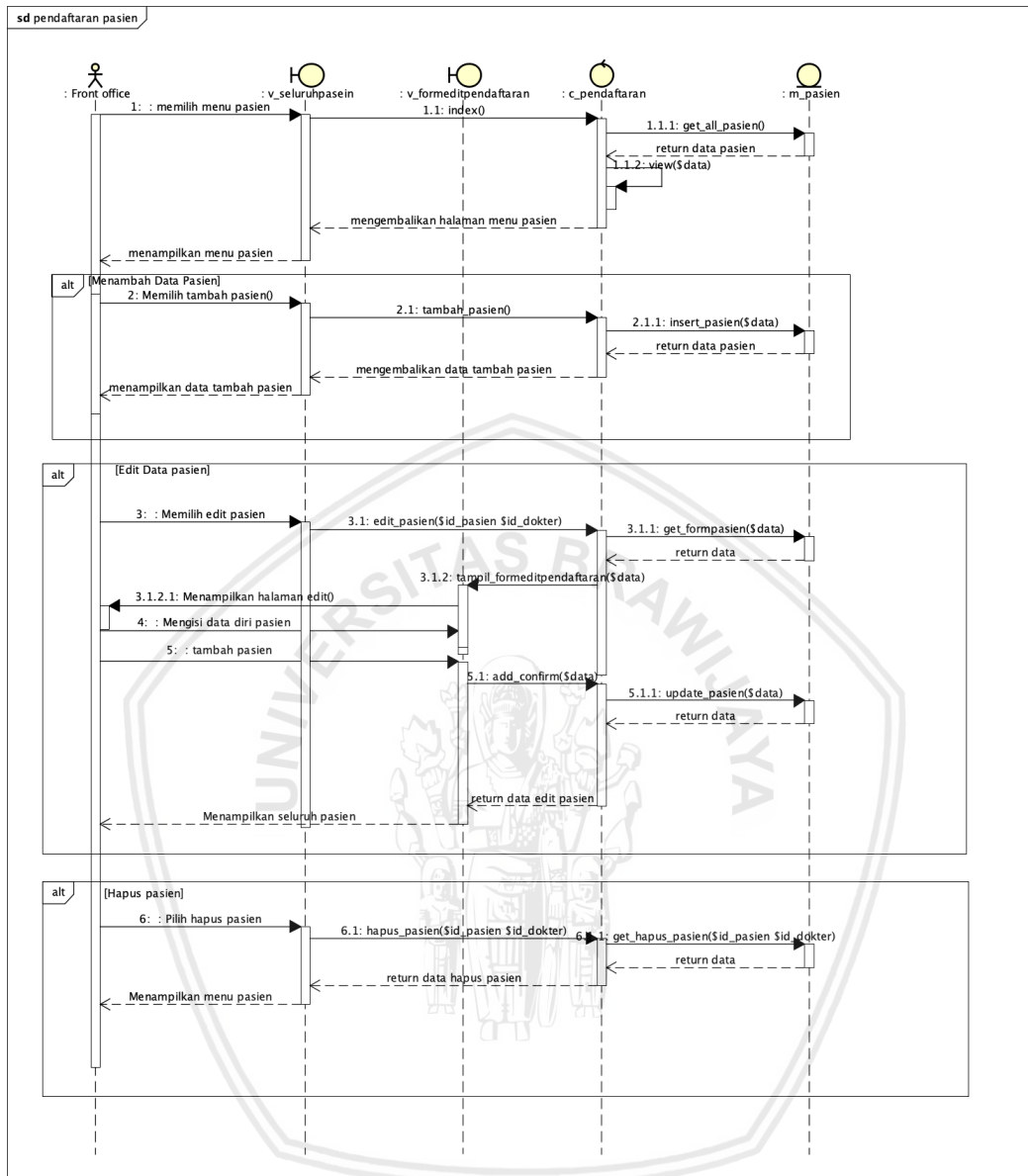
Dalam Pada bab ini menjelaskan kegiatan dalam metode Rational Unified Process (RUP) fase elaborasi lanjutan dilakukan, yaitu melakukan perancangan-perancangan sistem yang dibangun sesuai kebutuhan yang telah didefinisikan sebelumnya. Perancangan sistem meliputi gambaran sequence diagram, class diagram, perancangan basis data, beberapa perancangan antarmuka, beberapa perancangan algoritma dan perancangan pengujian.

### 5.1 Pemodelan Interaksi Objek

Sequence diagram merupakan dokumentasi dari pemodelan interaksi objek. Pemodelan interaksi objek dilakukan guna memvisualisasikan pertukaran pesan antar entitas sebuah aktor, objek control, entitas boundary, dan objek model yang saling ber interaksi dalam memenuhi kebutuhan pengguna. Sequence diagram dibuat berdasar pada alur dari beberapa use case yang telah didokumentasikan dalam spesifikasi use case. Beberapa Sequence diagram yang akan dijelaskan pada bagian ini berupa visualisasi interaksi dalam proses pendaftaran pasien, melihat dan memperbarui hasil pemeriksaan pasien, melihat persediaan obat dan memperbarui obat keluar pasien, melihat dan memperbarui proses pasien keluar.



### 5.1.1 Sequence Diagram Pendaftaran Pasien

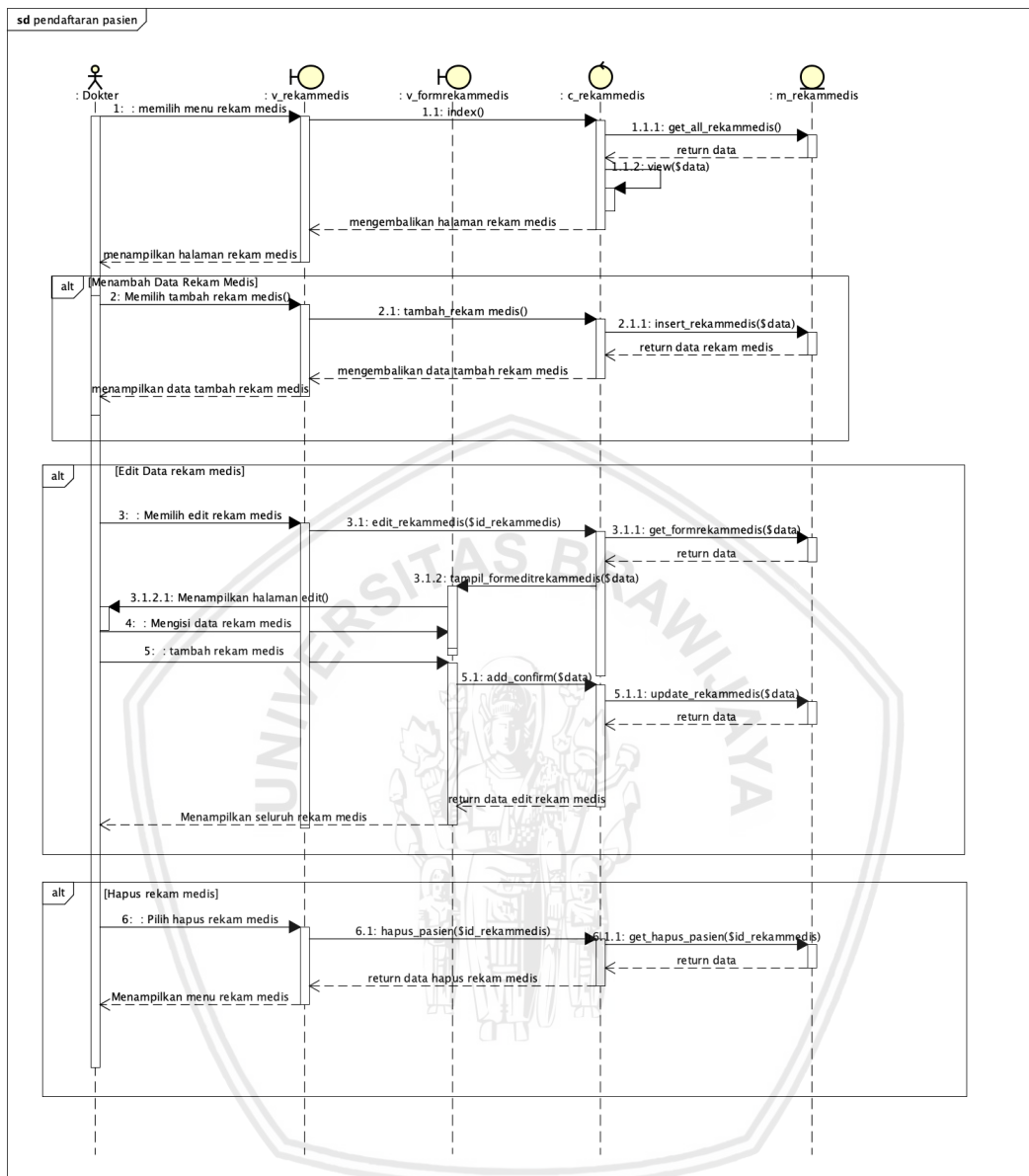


Gambar 5.1 Sequence Diagram Pendaftaran Pasien

Sequence diagram Gambar 5.1 merupakan visualisasi interaksi antar objek pada saat aktor yaitu *Front Office* yang menggunakan sistem untuk menambah data pasien baru rumah sakit HMC Malang. Objek-objek yang terlibat di dalam interaksi di antaranya adalah *Front office* sebagai aktor, *table\_seluruhpasein* dan *v\_formpendaftaran* sebagai *interface*, kelas *c\_pendaftaran* sebagai objek *control*, dan kelas *m\_pasien* sebagai objek *model*.



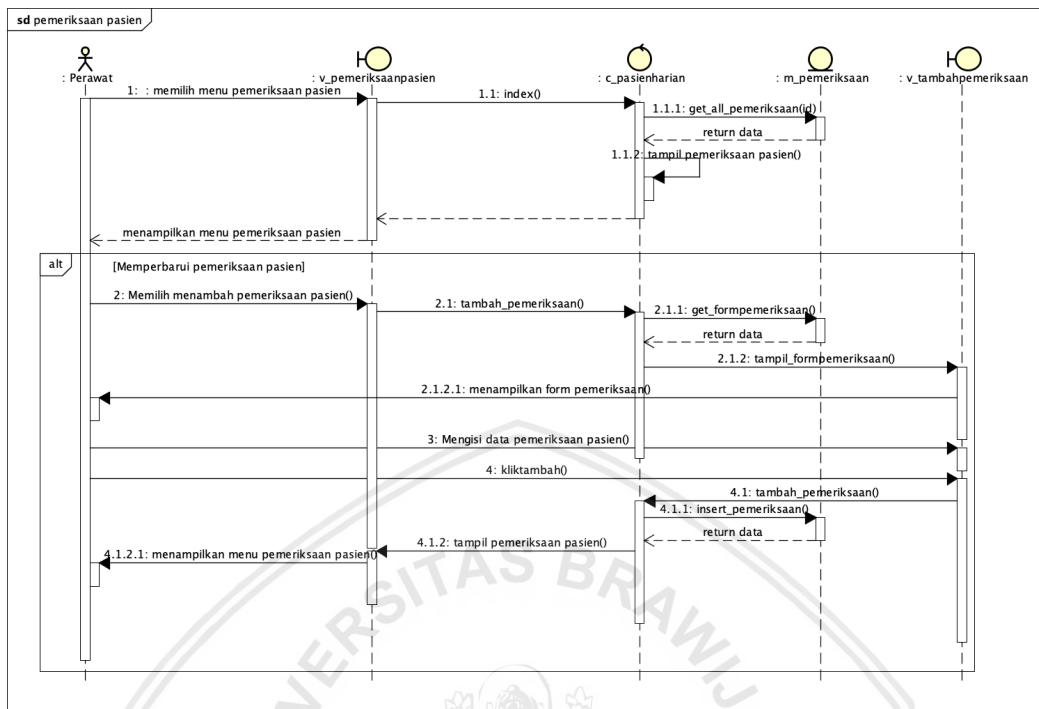
## 5.1.2 Sequence Diagram Rekam medis



**Gambar 5.2 Sequence Diagram Memperbarui Pemeriksaan Pasien**

Sequence diagram Gambar 5.2 merupakan visualisasi interaksi antar objek pada saat aktor yaitu Perawat yang menggunakan sistem untuk melihat dan menambah data rekam medis pasien selama rawat inap di rumah sakit HMC Malang. Objek-objek yang terlibat di dalam interaksi di antaranya adalah Perawat sebagai aktor, tabel\_seuruhrekammids dan v\_formrekammedis sebagai *interface*, kelas c\_rekammedis objek *control*, dan kelas m\_rekammedis sebagai objek *model*.

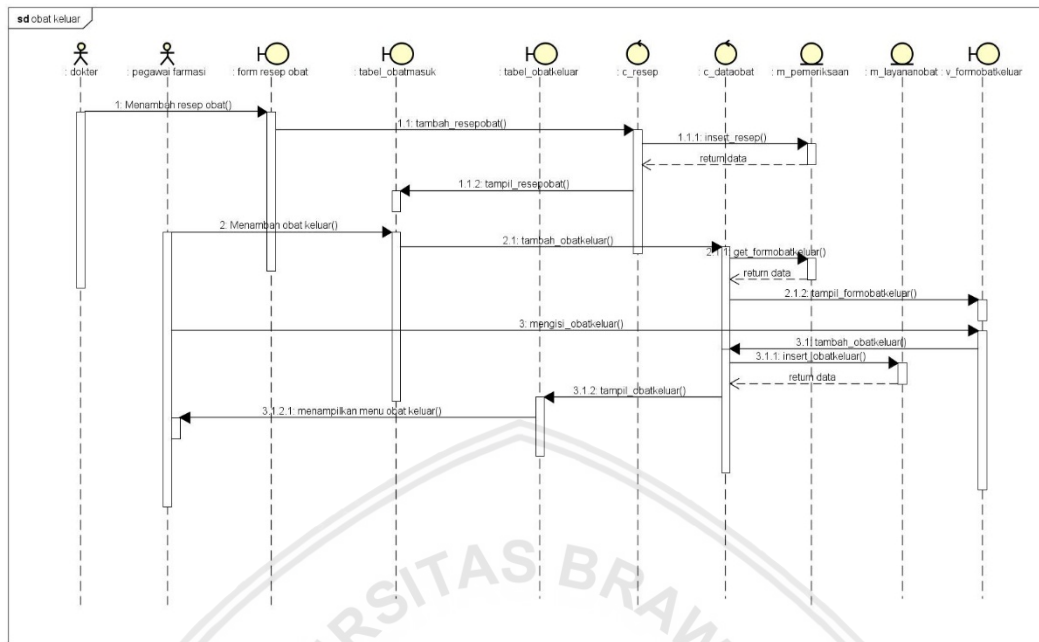
### 5.1.3 Sequence Diagram Memperbarui Pemeriksaan Pasien



**Gambar 5.3 Sequence Diagram Memperbarui Pemeriksaan Pasien**

Sequence diagram Gambar 5.3 merupakan visualisasi interaksi antar objek pada saat aktor yaitu Perawat yang menggunakan sistem untuk melihat dan memperbarui data pemeriksaan pasien selama rawat inap di rumah sakit HMC Malang. Objek-objek yang terlibat di dalam interaksi di antaranya adalah Perawat sebagai aktor, tabel\_pemeriksaanpasien dan v\_tambahpemeriksaan sebagai *interface*, kelas c\_pasienharian objek *control*, dan kelas m\_pemeriksaan sebagai objek *model*.

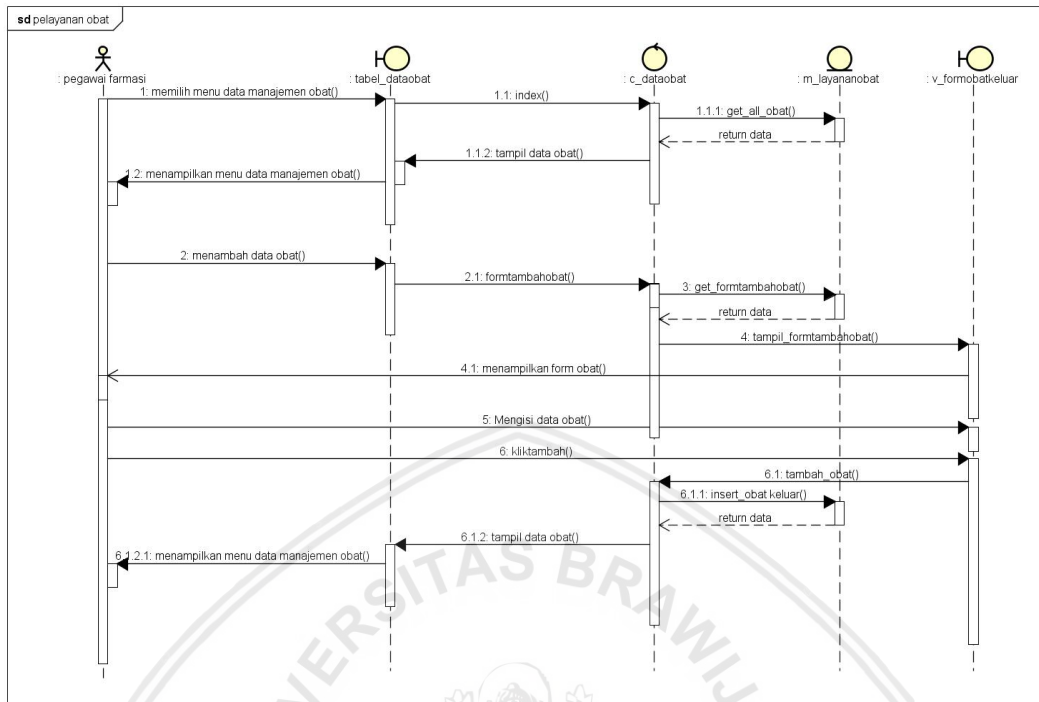
### 5.1.4 Sequence Diagram Memperbarui Obat Keluar Pasien



**Gambar 5.4 Sequence Diagram Memperbarui Obat Keluar Pasien**

Sequence diagram Gambar 5.4 merupakan visualisasi interaksi antar objek pada saat aktor yaitu pegawai farmasi yang menggunakan sistem untuk melihat dan memperbarui data obat keluar pasien. Objek-objek yang terlibat di dalam interaksi di antaranya adalah Pegawai farmasi sebagai aktor dan dokter, tabel\_dataobat, tabel\_obatkeluar dan v\_formobatkeluar sebagai *interface*, kelas c\_dataobat sebagai objek *control*, dan kelas m\_layananobat sebagai objek *model*.

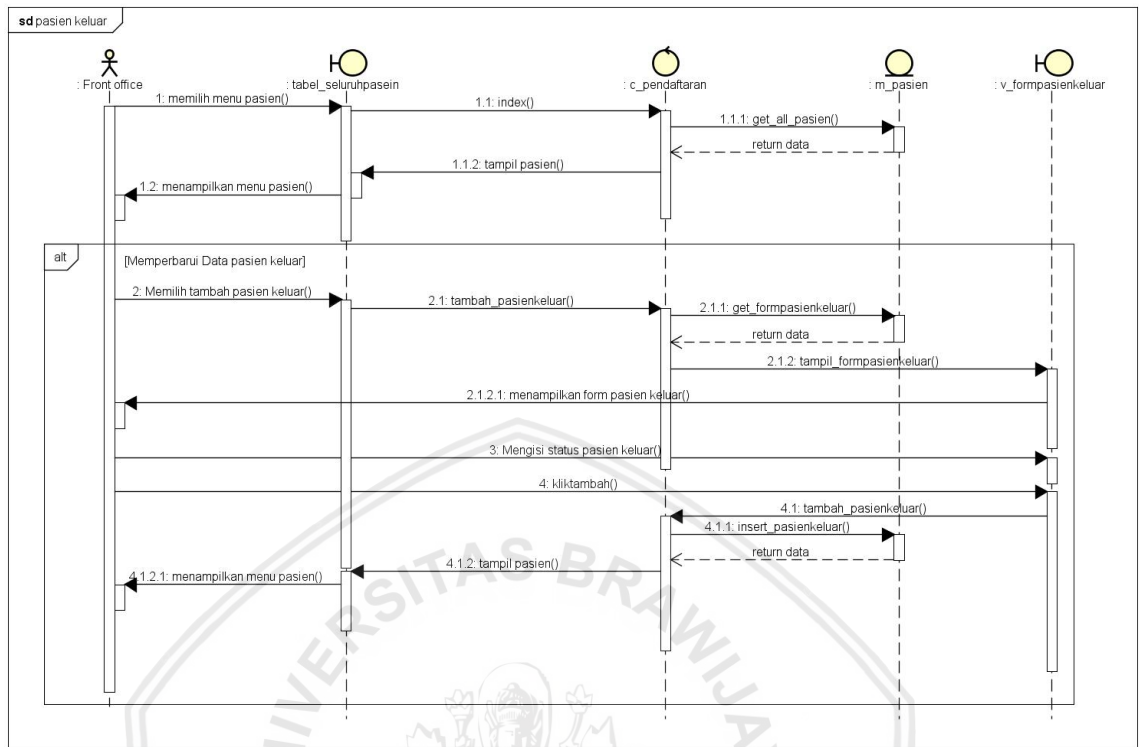
### 5.1.5 Sequence Diagram kelola Data Obat



**Gambar 5.5 Sequence Diagram Memperbarui Pasien Keluar**

Sequence diagram Gambar 5.5 merupakan visualisasi interaksi antar objek pada saat aktor yaitu pegawai farmasi yang menggunakan sistem untuk melihat dan memperbarui data stock obat. Objek-objek yang terlibat di dalam interaksi di antaranya adalah Pegawai farmasi sebagai aktor, tabel\_dataobat, tabel\_dataobat dan v\_formobatkeluar sebagai *interface*, kelas c\_dataobat sebagai objek *control*, dan kelas m\_layananobat sebagai objek *model*.

### 5.1.6 Sequence Diagram kelola Pasien Keluar



**Gambar 5.6 Sequence Diagram Memperbarui Pasien Keluar**

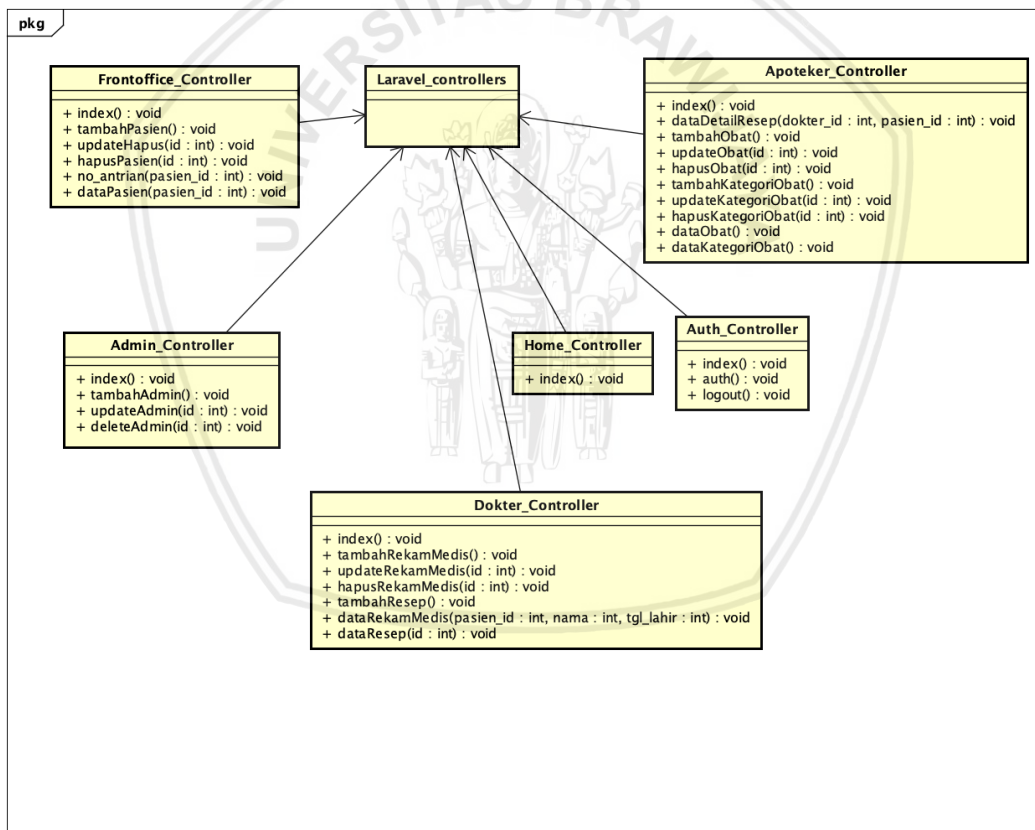
Sequence diagram Gambar 5.6 merupakan visualisasi interaksi antar objek pada saat aktor yaitu Front office yang menggunakan sistem untuk melihat dan memperbarui data pasien keluar dari rumah sakit HMC Malang. Objek-objek yang terlibat di dalam interaksi di antaranya adalah Front office sebagai aktor, tabel\_seluruhpasein dan v\_formpasienkeluar sebagai *interface*, kelas c\_pendaftaran sebagai objek *control*, dan kelas m\_pasien sebagai objek *model*.

## 5.2 Pemodelan Objek

Pemodelan objek ini dilakukan untuk mendokumentasikan sebuah kumpulan objek dari sistem yang dibangun. Pemodelan objek yang akan divisualisasikan dalam diagram kelas. Diagram kelas analisis dan diagram kelas perancangan akan dijelaskan pada bagian ini. Diagram kelas analisis akan digunakan sebagai panduan untuk melakukan pembuatan perancangan basis data. Diagram kelas perancangan adalah visualisasi kelas dan hubungan di antara tiap-tiap kelas sebagai panduan dalam melakukan pengembangan sebuah sistem.

### 5.2.1 Diagram Class Perancangan

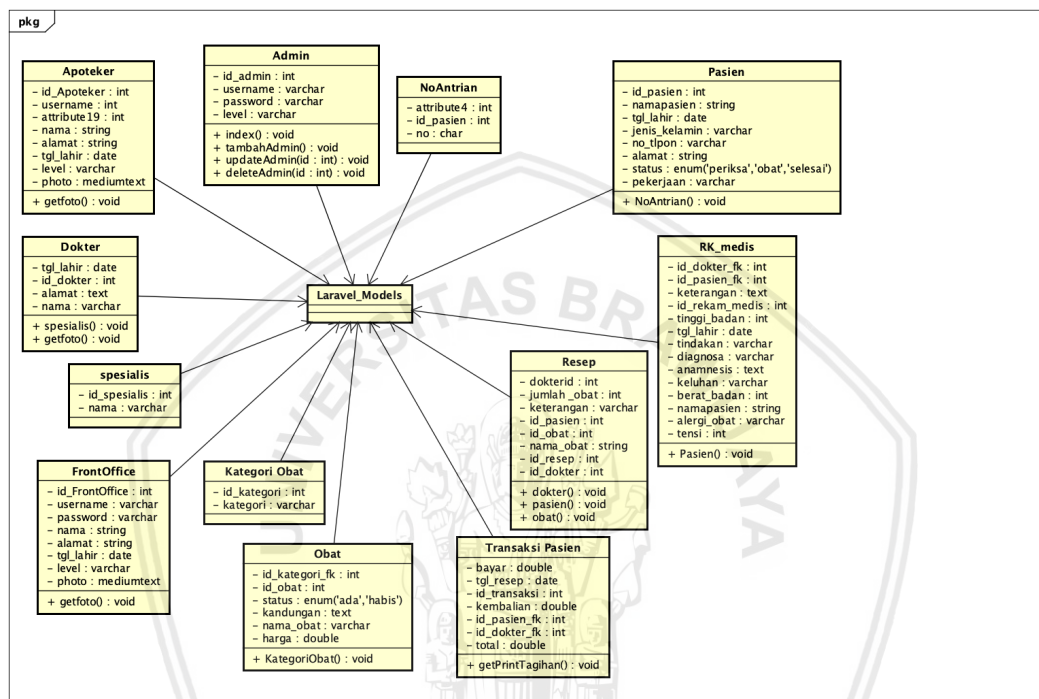
Diagram kelas perancangan merupakan diagram kelas hasil pengembangan diagram kelas analisis yang telah dibuat, pada dalam Gambar 5.8 dan Gambar 5.9. Diagram kelas perancangan di buat menyesuaikan diagram kelas analisis, identifikasi objek-objek pada sequence diagram, dan dengan menyesuaikan framework MVC pemrograman PHP *Laravel*.



**Gambar 5.7 Class Diagram Perancangan Controllers**

Gambar 5.7 menjelaskan secara umum tentang fungsi logika yang merepresentasikan kemampuan sistem untuk memenuhi kebutuhan pengguna. Fungsi logika yang merepresentasikan kemampuan sistem untuk melakukan pengolahan data pasien dan pendaftaran pasien ditangani oleh fungsi pada subkelas `frontoffice_controller`. Fungsi logika yang merepresentasikan kemampuan sistem untuk melakukan pengolahan data pengguna yang menggunakan sistem

ditangani oleh fungsi pada subkelas admin\_controller. Subkelas dokter\_controller mempresentasikan kemampuan sistem untuk melakukan pengolahan data pemeriksaan pasien serta hasil akhir pemeriksaan berupa rekam medis. Subkelas home\_controller mempresentasikan kemampuan sistem untuk memberikan informasi halaman awal dari sistem ini. Subkelas auth\_controller merupakan fungsi logika yang mempresentasikan kemampuan sistem untuk mengatur hak akses dari data pengguna. Subkelas Apoteker mempresentasikan kemampuan sistem untuk melakukan pengolahan data resep, obat, dan kategori obat.



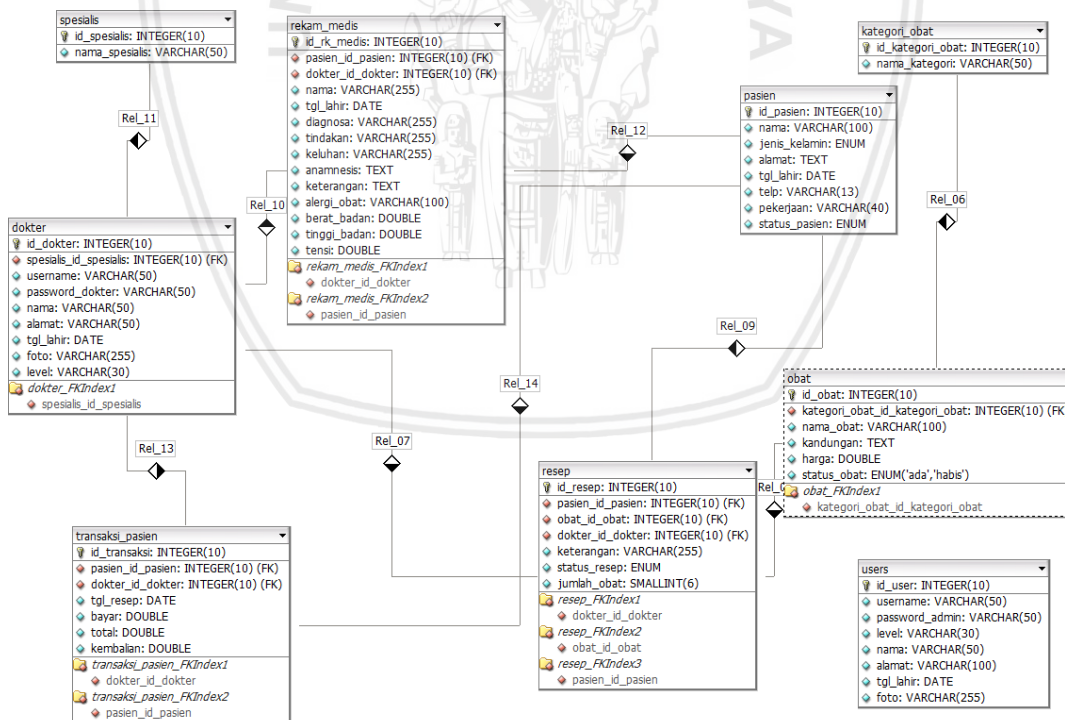
Gambar 5.8 Class Diagram Perancangan Models

Diagram kelas perancangan model Gambar 5.8 secara umum menyediakan fungsi untuk melakukan akses data yang tersimpan pada basis data. Fungsi akses pengolahan data user admin ditangani oleh subkelas model Admin. Subkelas apoteker menangani pengolahan data user Apoteker. Fungsi akses pengolahan data user dokter ditangani oleh subkelas Dokter. Fungsi akses pemilihan dokter spesialis ditangani oleh subkelas Spesialis. Fungsi akses pengolahan data user frontoffice ditangani oleh subkelas FrontOffice. Subkelas KategoriObat menangani pemilihan kategori obat sesuai dengan Subkelas Obat. Subkelas Obat menangani pengolahan obat yang tersedia dari apoteker. Fungsi akses pengolahan biaya transaksi pasien ditangani oleh subkelas TransaksiPasien. Fungsi akses pengolahan resep pasien dari pemeriksaan akan ditangani oleh subkelas Resep. Subkelas RK\_medis menangani pengolahan pengolahan hasil dari pemeriksaan yang berupa rekam medis. Subkelas antrian menangani antrian pasien pada saat pendaftaran. Subkelas Pasien menangani pengolahan data diri pasien pada saat pendaftaran.

### 5.3 Perancangan Database

Rancangan basis data yang ditujukan pada Gambar 5.9 dibuat berdasarkan pemodelan kelas diagram analisis yang terdapat pada Gambar 5.8. Masing-masing kelas yang ada pada diagram kelas analisis direpresentasikan ke tabel yang terdapat pada PDM, sedangkan atribut yang ada di dalam setiap kelas analisis direpresentasikan oleh nama-nama kolom pada setiap tabel pada PDM. Hubungan antar tabel pada PDM dibuat berdasarkan hubungan antar kelas pada diagram kelas analisis.

Tabel spesialis menyimpan data spesialis yang merupakan kategori dari dokter. Data dokter tersimpan dalam tabel dokter. Tabel spesialis memiliki relasi *OnetoMany* dengan dokter memungkinkan untuk data spesialis yang memiliki beberapa dari dokter. Data rekam medis tersimpan dalam tabel rekam\_medis. Data pasien yang terdaftar tersimpan di tabel pasien. Tabel dokter dan pasien memiliki relasi *OnetoMany* dengan rekam medis sehingga data pasien dan data dokter memungkinkan memiliki beberapa dari rekam medis. Data kategori obat disimpan di dalam tabel kategori\_obat. Data pengolahan obat disimpan di dalam di tabel obat. Lalu data resep akan disimpan di tabel resep. Tabel kategori obat memiliki relasi *OnetoMany* dengan obat. Untuk data transaksi pasien akan disimpan di dalam tabel transaksi\_pasien. Data pengguna sistem informasi tersimpan dalam tabel user.



Gambar 5.9 PDM Sistem Informasi Rawat inap HMC Malang



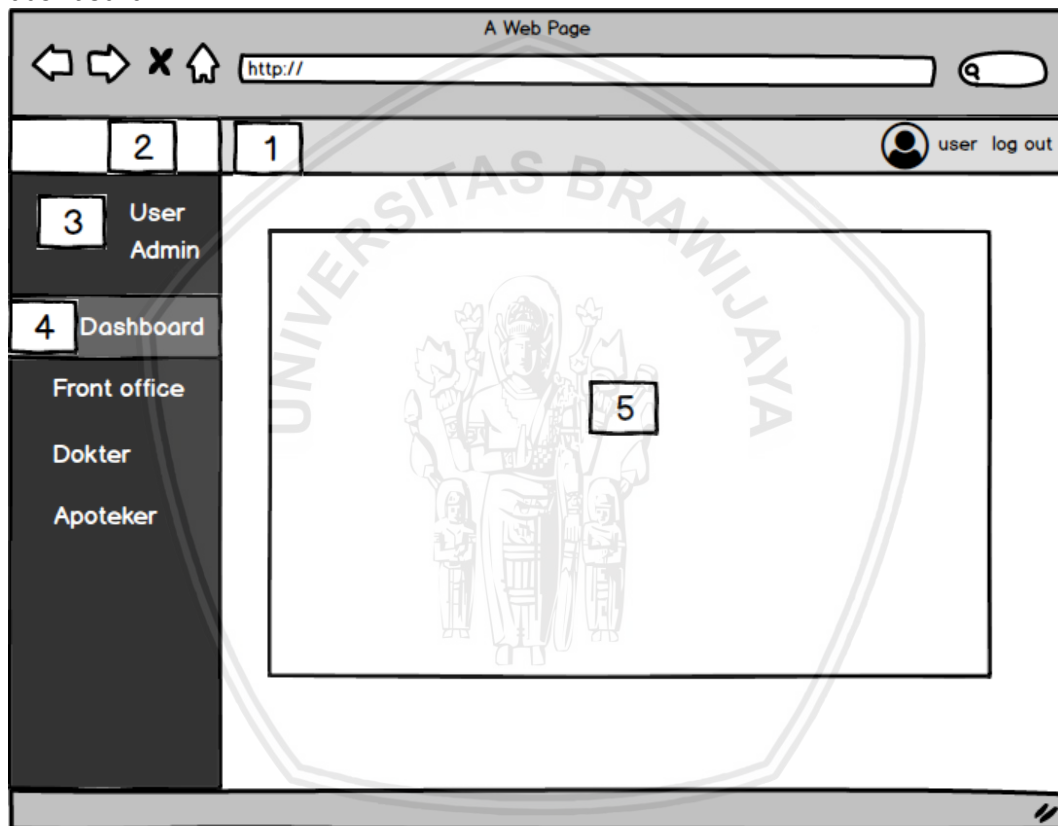


## 5.4 Perancangan Antarmuka

Subbab ini merupakan perancangan antarmuka pengguna sebagai dasar tampilan antarmuka sistem yang akan dibangun. Beberapa gambaran antarmuka pengguna sistem di antaranya adalah antarmuka halaman awal, daftar pasien harian, database pasien, form hasil pemeriksaan, dan form pendaftaran.

### 5.4.1 Perancangan Antarmuka Halaman Awal Admin

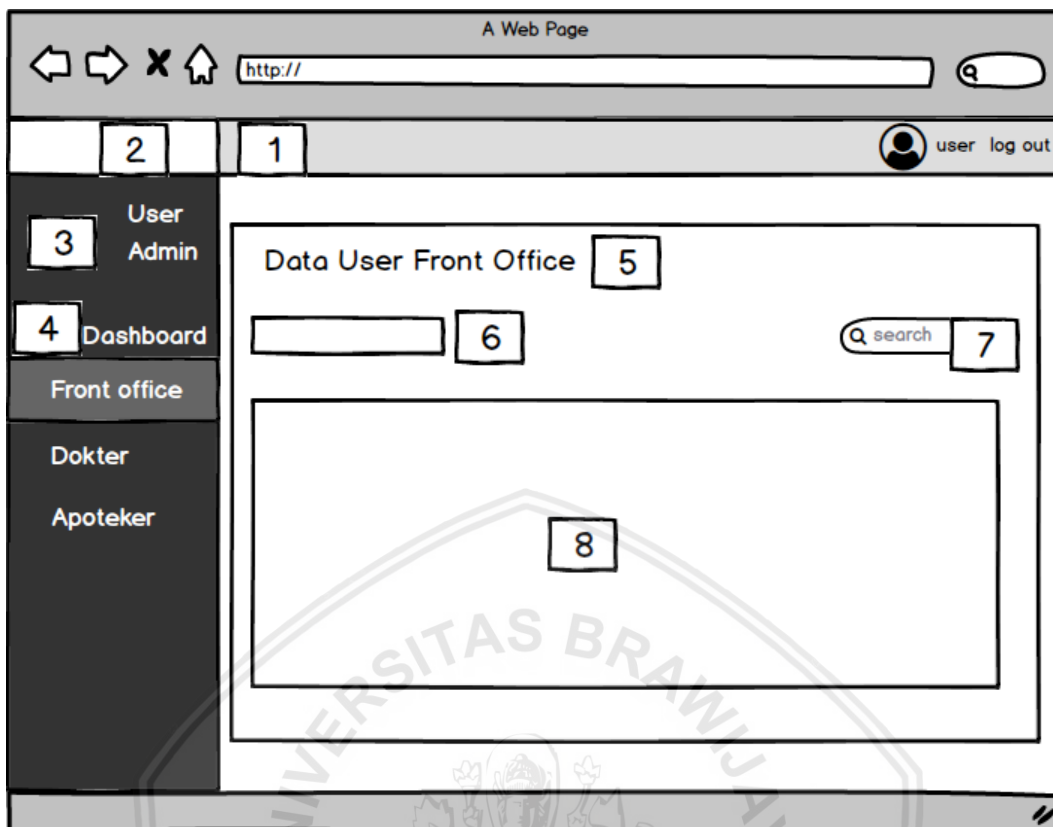
Pada Gambar 5.10 merupakan tampilan antarmuka ketika pengguna melihat halaman awal Admin. Komponen pada antarmuka ini diantaranya (1) top-nav, (2) logo, (3) profile, (4) side-menu. Halaman konten memiliki komponen (5) dashboard.



Gambar 5.10 Perancangan Antarmuka Halaman Awal Admin

### 5.4.2 Perancangan Antarmuka Data User Front Office

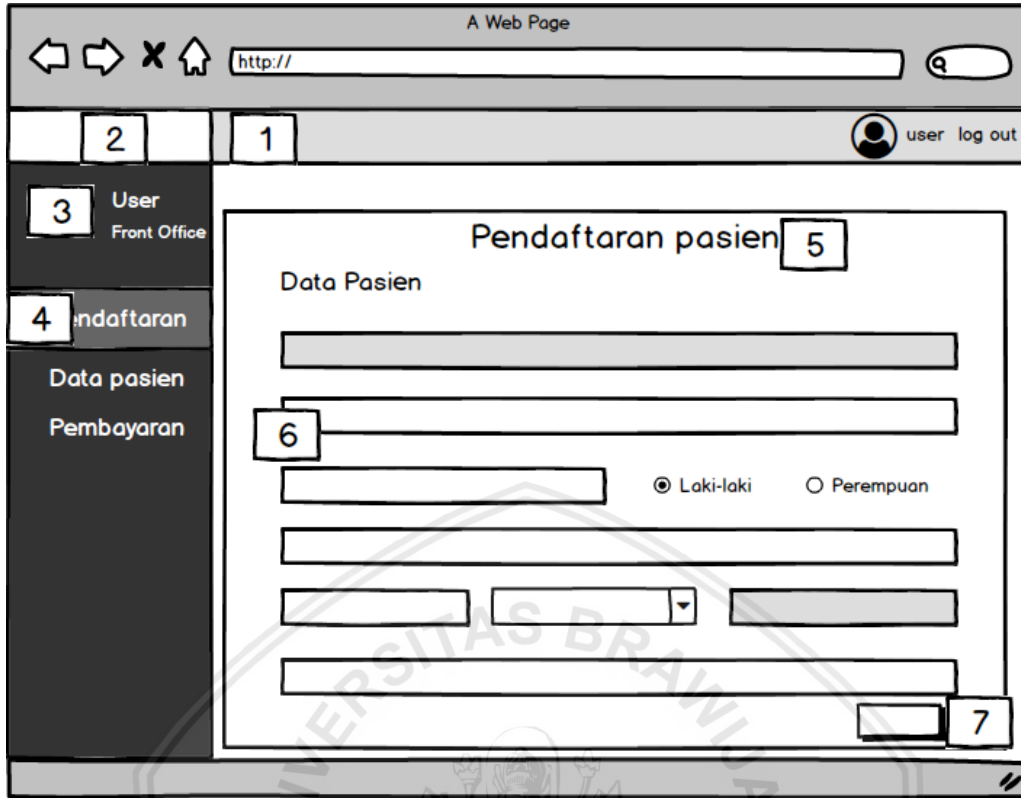
Pada Gambar 5.11 merupakan tampilan antarmuka ketika pengguna melihat Data User Front Office. Komponen pada antarmuka ini diantaranya (1) top-nav, (2) logo, (3) profile, (4) side-menu. Halaman konten memiliki komponen (5) konten (6) tambah data, (7) pencarian (8) tabel.



Gambar 5.11 Perancangan Antarmuka Data User Front Office

#### 5.4.3 Perancangan Antarmuka Pendaftaran Pasien

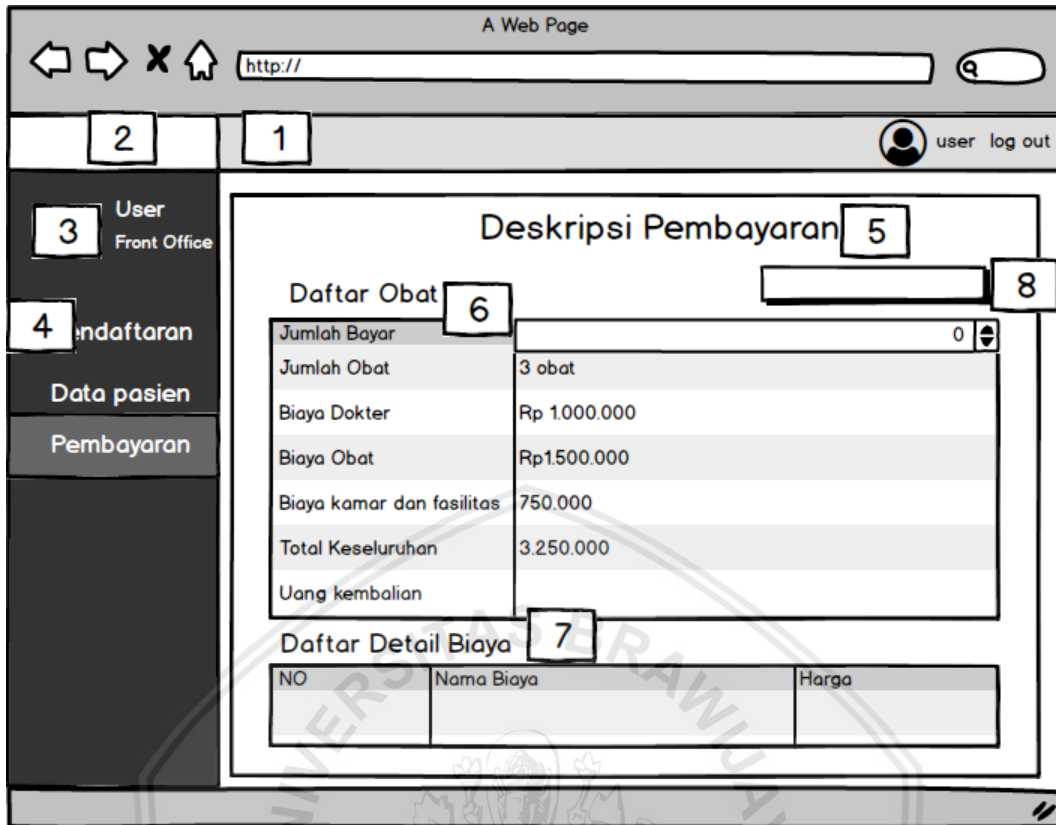
Pada Gambar 5.12 merupakan tampilan antarmuka ketika pengguna mendaftarkan pasien. Komponen pada antarmuka ini diantaranya (1) top-nav, (2) logo, (3) profile, (4) side-menu . Halaman konten memiliki komponen (5) konten (6) form elemet data diri,(7) tombol konfirmasi data.



Gambar 5.12 Perancangan Antarmuka Pendaftaran pasien

#### 5.4.4 Perancangan Antarmuka Transaksi Pasien

Pada Gambar 5.13 merupakan tampilan antarmuka ketika pengguna Mengolah data Transaksi Pasien. Komponen pada antarmuka ini diantaranya (1) top-nav, (2) logo, (3) profile, (4)side-menu . Halaman konten memiliki komponen (5) konten (6) form elemet biaya keseluruhan ,(7) Daftar Detail biaya pasien,(8) Konfirmasi pembayaran.



Gambar 5.13 Perancangan Antarmuka Transaksi Pasien

## 5.5 Perancangan Algoritma

Perancangan algoritma menjelaskan tahap yang dilakukan dalam menyusun sekumpulan alur operasi logika sebagai konsep membangun fungsi fungsi di sistem informasi. Hasil perancangan algoritma dalam penelitian ini dijelaskan ke dalam beberapa tabel algoritma. Algoritma menjadi panduan dalam mengembangkan sistem informasi pada tahap implementasi.

### 5.5.1 Algoritma Menyimpan Data Pendaftaran Pasien Baru

Subbab ini menjelaskan algoritma dari sebuah fungsi dari sistem saat pengguna menggunakan sistem untuk menyimpan data pendaftaran pasien baru. Algoritma pada Tabel 5.1 merupakan rancangan algoritma fungsi untuk menyimpan data pendaftaran pasien baru. Algoritma dimulai dengan membuat variable array untuk menangkap data inputan dari formulir yang telah diisi pengguna yaitu variabel array yang menampung data diri dan data pemeriksaan pasien. Kemudian data array tersebut disimpan ke database dengan memanggil fungsi penyimpanan data. Setelah itu dilakukan pengecekan jika proses penyimpanan data tersebut berhasil maka menampilkan pesan berhasil serta menampilkan data pada halaman daftar pasien harian namun jika penyimpanan data gagal dilakukan akan menampilkan pesan gagal.

**Tabel 5.1 Algoritma Menyimpan Data Pendaftaran Pasien**

Baris	Algoritma
1	Mulai
2	
3	Buat array untuk menangkap inputan data diri
4	Buat array untuk menangkap inputan data pemeriksaan
5	
6	
7	Memanggil fungsi untuk menyimpan data diri
8	Memanggil fungsi untuk menyimpan data pemeriksaan
9	
10	Jika proses penyimpanan berhasil
11	Menampilkan pesan berhasil
12	Mengalihkan ke halamanan pasien
13	Menampilkan data baru
14	
15	Jika proses penyimpanan gagal
16	Menampilkan halaman dengan pesan gagal
17	
18	Selesai

### 5.5.2 Algoritma Melihat Data Rekam Medis Pasien

Subbab ini menjelaskan algoritma dari sebuah fungsi dari sistem saat pengguna menggunakan sistem untuk melihat data hasil rekam medis pasien. Algoritma pada Tabel 5.2 merupakan rancangan algoritme fungsi untuk menampilkan data rekam medis. Algoritma dimulai dengan memanggil fungsi menampilkan data hasil reka medis. Fungsi akan mengambil data dari model *Rk\_medis*, pasien dan dokter. Data akan dimasukkan kedalam variable untuk dibawa ke halaman tampilan.

**Tabel 5.2 Algoritma Melihat Data Hasil Rekam medis**

Baris	Algoritma
1	Mulai
2	
3	Memanggil fungsi menampilkan data hasil rekam medis
4	
5	Meminta parameter id rekam medisid pasien dan id dokter
6	
7	Fungsi mengambil data model <i>Rk_medis</i> dengan berdasarkan id rekam medis
8	
9	Fungsi mengambil data model pasien dengan berdasarkan id pasien
10	
11	

12	Fungsi mengambil data model dokter berdasarkan
13	id dokter
14	
15	Memasukan data model kedalam variabel
16	
17	Menuju tampilan hasil pemeriksaan dengan compact
18	variabel

### 5.5.3 Algoritma Mengubah Hasil Rekam Medis

Subbab ini menjelaskan algoritma dari sebuah fungsi dari sistem saat pengguna menggunakan sistem untuk mengubah hasil rekam medis. Algoritma pada Tabel 5.2 merupakan rancangan algoritme fungsi untuk mengubah hasil rekam medis. Algoritma dimulai dengan memanggil fungsi mengubah hasil rekam medis. Fungsi mengambil data dari model Rk\_medis, pasien dan dokter lalu memasukan data model kedalam variable. Pada pengecekan pertama jika salah satu data pada variable *null* maka akan menampilkan halaman dengan pesan hasil pemeriksaan belum lengkap. Pada pengecekan kedua jika data status\_pengisian pada semua variable sudah terisi maka akan memanggil fungsi menyimpan data hasil akhir pemeriksaan dan menampilkan halaman dengan pesan berhasil.

**Tabel 5.3 Pseudocode Mengubah Hasil Rekam medis**

Baris	Algoritma
1	Mulai
2	
3	Memanggil fungsi mengubah hasil rekam medis
4	Meminta parameter request, id rekam medis id
5	pasien dan id dokter
6	
7	Fungsi mengambil data model rk_medis id rekam
	medis
8	Fungsi mengambil data model pasien berdasarkan
	id pasien
9	Fungsi mengambil data model dokter berdasarkan
	id dokter
10	
11	Memasukan data model kedalam variabel
12	
13	Melakukan pengecekan ifelse pada variable
14	Jika salah satu data variable null
	Jika data status_pengisian disetiap variable
15	sudah terisi
	Memanggil fungsi menyimpan data hasil akhir
16	pemeriksaan
17	Menampilkan halaman dengan pesan berhasil
18	



## BAB 6 IMPLEMENTASI

Pada bab ini menjelaskan kegiatan dalam metode *Rational Unified Process* (RUP) fase konstruksi tahap implementasi sistem dilakukan, yaitu implementasi dalam membangun sistem informasi berdasarkan hasil perancangan pada bab sebelumnya. Bahasan pada sub bab terdiri dari spesifikasi lingkungan implementasi, implementasi beberapa algoritma, dan implementasi beberapa antarmuka pengguna sistem.

### 6.1 Spesifikasi Lingkungan Sistem Informasi

Implementasi Algoritma ini menjelaskan spesifikasi dari perangkat keras dan perangkat lunak yang digunakan dalam membangun sistem informasi. Perangkat keras yang digunakan dalam membangun sistem informasi adalah sebuah perangkat laptop dengan spesifikasi yang dijelaskan pada Tabel 6.1. Sedangkan spesifikasi perangkat lunak yang digunakan dalam pengembangan dijelaskan pada Tabel 6.2 dan spesifikasi lingkungan *deployment* ditunjukkan pada Tabel 6.3.

**Tabel 6.1 Spesifikasi Perangkat Keras**

Laptop Acer Aspire 4741G	
<b>Processor</b>	Core i3 Processor (Up to 2.13 GHz, L3 cache)
<b>Kapasitas RAM</b>	2.00 GB
<b>Kapasitas Hardisk</b>	500 GB
<b>Kartu Grafis</b>	Nvidia 310 Geforce
<b>Maksimal Resolusi Layar</b>	1366 x 768

**Tabel 6.2 Spesifikasi Perangkat Lunak**

Laptop Lenovo G40 A8	
<b>Sistem Operasi</b>	Windows 7 64-bit
<b>Bahasa Pemrograman</b>	PHP 7
<b>Editor Kode Program</b>	Sublime Text 3.1.1
<b>Perangkat Lunak Pendukung</b>	phpMyAdmin, Visual Paradigm, Google Chrome, Asta, Photoshop, draw.io
<b>Library Pendukung</b>	Jquery, database (Laravel), bootstrap, session (Laravel), datetimepicker, chart

**Tabel 6.3 Spesifikasi Minimal Lingkungan *Deployment***

<b>Kapasitas RAM (min.)</b>	256 MB
<b>Kapasitas Memori (min.)</b>	200 MB

Versi PHP	PHP 5.3 atau lebih
-----------	--------------------

Tabel 6.1 menjelaskan spesifikasi perangkat keras yang digunakan untuk pengembangan sistem informasi dimana perangkat keras yang digunakan adalah Laptop Acer Aspire 4741G dengan Core i3 Processor, RAM 2.00 GB, dan grafis Nvidia 310 Geforce. Pada Tabel 6.2 selanjutnya menjelaskan spesifikasi perangkat lunak yang digunakan untuk pengembangan sistem informasi dimana sistem operasi yang digunakan Windows 7 64-bit dan menggunakan bahasa pemrograman PHP 5.6.30. Spesifikasi minimal lingkungan pengembangan dijelaskan pada Tabel 6.3 dimana lingkungan implementasi harus memenuhi standar minimal RAM 256 MB, Memori 200 MB, dan versi PHP 5.3 atau lebih.

## 6.2 Implementasi Algoritma

Subbab implementasi algoritma merupakan hasil implementasi program berdasarkan algoritma yang telah dibuat pada tahap perancangan. Beberapa hasil implementasi algoritme diantaranya implementasi menyimpan semua data pendaftaran pasien baru, mengubah data pemeriksaan, dan mengubah hasil akhir pemeriksaan pasien.

### 6.2.1 Implementasi Algoritma Menambah Data Pendaftaran Pasien Baru

Kode program Menambah data pendaftaran pasien berfungsi untuk mengarahkan tampilan ke halaman pendaftaran pasien dan menyimpan data inputan. Implementasi algoritma ini dibuat berdasarkan hasil rancangan algoritma menambah data pendaftaran pasien baru pada bab sebelumnya. Implementasi algoritma menambah data pendaftaran pasien baru dapat dilihat pada Tabel 6.4.

**Tabel 6.4 Implementasi Algoritma Menambah Data Pendaftaran Pasien Baru**

Baris	Kode Program
1	public function postPendaftaranPasien(Request \$request) {
2	try {
3	DB::beginTransaction();
4	
5	\$data = Pasien::create(\$request->all());
6	
7	\$pasien_id = \$this->getLastNoAntrian();
8	
9	\$pasien = Pasien::whereDate('created_at',
10	'=', date('Y-m-d'))->where('status', '=', 'antri')->get();
11	\$total = Pasien::where('status',
12	'selesai')->get()->toArray();
13	\$bulan = Pasien::whereMonth('created_at',
14	'=', date('m'))->whereYear('created_at', '=', date('Y'))-
15	>get()->toArray();
16	
17	// create no antrian
18	\$id = NoAntrian::select('id')->get()-
19	>last();
20	\$id=\$id['id'];
21	if (\$id == null) {



<p>22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51</p>	<pre>                 \$id = 1;             } else {                 \$id = (int) \$id;                 \$id += 1;             }             \$id = str_pad(\$id, 3, "0", STR_PAD_LEFT);             \$antrian = NoAntrian::create(["no" =&gt; \$id, 'pasien_id' =&gt; \$data['id']]);              DB::commit();             return response()-&gt;json([                 "success" =&gt; [                     "data" =&gt; \$data,                     "id" =&gt; \$pasien_id,                     "id_antrian" =&gt; \$antrian-&gt;id,                     "no_antrian" =&gt; \$antrian-&gt;no,                     "pasien_hari_ini" =&gt; count(\$pasien),                     "total_pasien" =&gt; count(\$total),                     "total_per_bulan" =&gt; count(\$bulan)                 ],                 "errors" =&gt; null             ], 200);         } catch (\Exception \$e) {             DB::rollback();             \$errors = \$e-&gt;errorInfo[2];             return response()-&gt;json([                 "success" =&gt; null,                 "errors" =&gt; \$errors             ], 400);         }     }         </pre>
--	---

### 6.2.2 Implementasi Algoritma Mengubah Data Rekam Medis

Kode program mengubah data Rekam medis memiliki fungsi untuk mengarahkan tampilan ke halaman data pemeriksaan edit dan melakukan penyimpanan data perubahan yang dilakukan. Implementasi algoritma ini dibuat berdasarkan hasil rancangan algoritma mengubah data rekam medis pada bab sebelumnya. Implementasi algoritma mengubah data rekam medis dapat dilihat pada Tabel 6.5.

**Tabel 6.5 Implementasi Algoritma Mengubah Data Rekam medis**

Baris	Kode Program
1	public function postRekamMedisPasien(Request \$request) {
2	if (\$request->ajax()) {
3	\$rekamMedis = RK_Medis::create([
4	'id' => \$request->id,
5	'nama' => \$request->nama,
6	'bb' => \$request->bb,
7	'tb' => \$request->tb,
8	'tensi' => \$request->tensi,
9	'bw' => \$request->bw,



```

10         'pasien_id' => $request->pasien_id,
11         'dokter_id' => $request->dokter_id,
12         'diagnosa' => $request->diagnosa,
13         'keluhan' => $request->keluhan,
14         'anamnesis' => $request->anamnesis,
15         'tindakan' => $request->tindakan,
16         'keterangan' => $request->deskripsi,
17         'alergi_obat' => $request->alergi_obat,
18     });
19
20     $pasien = Pasien::find($request->pasien_id)-
21 >update(['status' => 'obat']);
22
23     for ($i=0; $i <count($request['obat']) ; $i++)
24     {
25         $resep = Resep::create([
26             'dokter_id' =>
27 $request['dokter_id'],
28             'pasien_id' =>
29 $request['pasien_id'],
30             'obat_id' =>
31 $request['obat'][$i]['value'],
32             'keterangan' =>
33 $request['keterangan'][$i]['value'],
34             'jumlah' =>
35 $request['jumlah'][$i]['value'],
36             'biaya_dokter' =>
37 $request['biaya_dokter'],
38             'status' => 'belum'
39         ]);
40     }
41
42     return response()->json($rekamMedis);
43 }
44 }
45
46 public function getObat() {
47     $data = Obat::get()->toJson();
48     return response($data);
49 }
50
51 public function getRekamMedis() {
52     $rekamMedis = RK_Medis::with('pasien')-
53 >where('dokter_id', Session::get('id'))->get()-
54 >toArray();
55     $HariIni = RK_Medis::where('dokter_id',
56 Session::get('id'))->whereDate('created_at', date('Y-m-
57 d'))->get()->toArray();
58     return view('dokter.rekam-medis', ['rekamMedis' =>
59 $rekamMedis, 'HariIni' => $HariIni]);
60 }
61
62 public function postUpdateRekamMedis(Request
63 $request) {
64     if ($request->ajax()) {
65         $data = RK_Medis::find($request->id)-
66 >update($request->all());
67         return response()->json($data);
68     }
69 }

```

	}

### 6.2.3 Implementasi Algoritma Mengolah Transaksi Biaya Pasien

Kode program mengolah transaksi biaya pasien memiliki fungsi untuk melakukan pembayaran biaya obat pasien biaya rumah sakit dan dokter selama di rumah sakit. Implementasi algoritma ini dibuat berdasarkan hasil rancangan algoritma mengubah hasil akhir pemeriksaan pasien pada bab sebelumnya. Tabel 6.3 merupakan implementasi algoritma fungsi untuk mengubah hasil akhir pemeriksaan pasien.

**Tabel 6.6 Implementasi Algoritma Mengolah Transaksi Biaya Pasien**

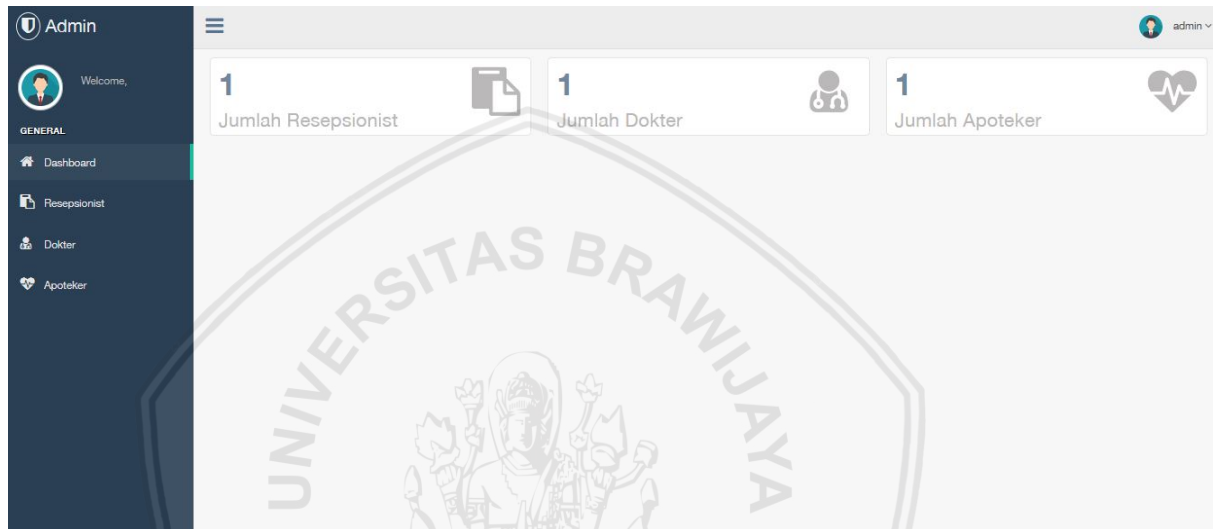
Baris	Kode Program
1	public function postResep(Request \$request) {
2	if (\$request->ajax()) {
3	\$pasien = Pasien::find(\$request->pasien_id)-
4	>update(['status' => 'selesai']);
5	for (\$i=0; \$i <count(\$request['id']); \$i++) {
6	\$data = Resep::where('id',
7	\$request['id'][\$i]['value']->first();
8	\$data->status = 'selesai';
9	\$data->save();
10	}
11	// create transaksi pasien
12	\$transaksi_pasien = TransaksiPasien::create([
13	'dokter_id' => \$data['dokter_id'],
14	'pasien_id' => \$request['pasien_id'],
15	'bayar' => \$request['bayar'],
16	'kembalian' => \$request['kembalian'],
17	'total' => \$request['total'],
18	'tgl_resep' => \$request['tgl_resep']
19	]);
20	return response()->json(\$transaksi_pasien);
21	}
22	}
23	
24	public function getPrintTagihan(\$tagihan_id, \$dokter_id,
25	\$pasien_id) {
26	\$size = array(0,0,204,650);
27	\$resep = Resep::with(['obat', 'dokter', 'pasien'])-
28	>where(['dokter_id' => \$dokter_id, 'pasien_id' => \$pasien_id])-
29	>get()->toArray();
30	\$tagihan = TransaksiPasien::where('id', \$tagihan_id)-
31	>first();
32	\$pdf = PDF::loadView('apoteker.print', ['resep' => \$resep,
33	'tagihan' => \$tagihan])->setOptions(['dpi' => 72,'defaultFont' =>
34	'sans-serif']);
35	return \$pdf->stream('Tagihan Klinik HMC.pdf');
36	}
37	



## 6.3 Implementasi Antarmuka Pengguna

Hasil tampilan antarmuka pengguna sistem informasi dijelaskan dalam subbab implementasi antarmuka pengguna. Subbab ini hanya menyertakan beberapa hasil implementasi antarmuka pengguna. Implementasi antarmuka pengguna yang ditampilkan adalah antarmuka halaman admin, halaman pendaftaran pasien, dan halaman pemeriksaan pasien, halaman rekam medis dan transaksi pembayaran.

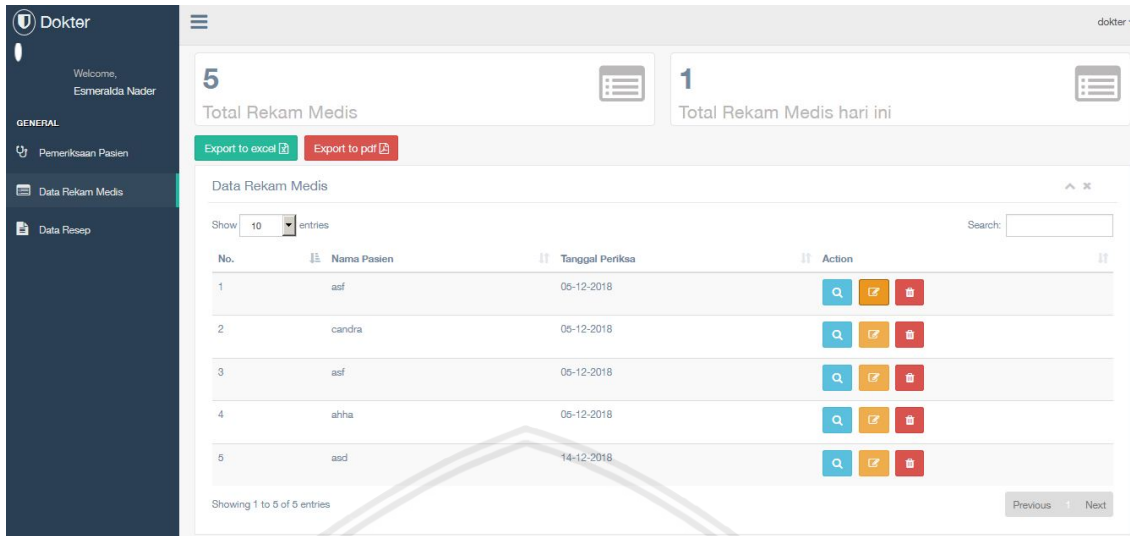
### 6.3.1 Antarmuka Halaman Admin



**Gambar 6.1 Antarmuka Halaman Home**

Gambar 6.1 merupakan hasil implementasi dari perancangan halaman *Admin* yang telah dibuat pada bab sebelumnya. Halaman ini digunakan pengguna untuk menambahkan data pengguna dari sistem ini yaitu menambah FrontOffice Dokter dan Apoteker. Komponen yang terdapat pada halaman ini adalah penambahan data Front Office, dokter dan apoteker data jumlah dari masing pengguna.

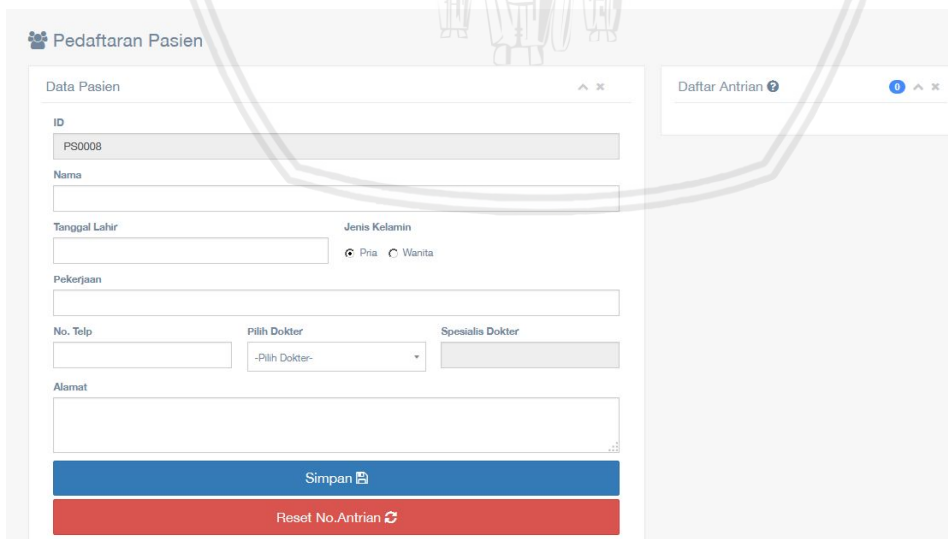
### 6.3.2 Antarmuka Halaman Daftar Rekam Medis Pasien



**Gambar 6.2 Antarmuka Halaman Daftar Pasien Harian**

Gambar 6.2 merupakan hasil implementasi dari perancangan halaman daftar pasien harian yang telah dibuat pada bab sebelumnya. Halaman ini digunakan pengguna untuk melihat Rekam medis yang sedang dilakukan pada hari ini dan memantau progres pengerjaan pemeriksaan. Komponen yang terdapat pada halaman ini adalah tabel yang menampilkan seluruh data rekam medis dari hasil pemeriksaan yang sedang dilakukan hari ini, kolom progres pengerjaan data pemeriksaan, kolom aksi terdapat detail dan hapus data, dan tanggal pemeriksaan.

### 6.3.3 Antarmuka Halaman Pendaftaran Pasien



**Gambar 6.3 Antarmuka Halaman Pendaftaran Pasien**

Gambar 6.3 merupakan hasil implementasi dari perancangan halaman pendaftaran pasien yang telah dibuat pada bab sebelumnya. Halaman ini digunakan pengguna untuk menambahkan data pendaftaran pasien pemeriksaan

baru. Halaman pendaftaran pasien terdapat tiga komponen formulir yang diisi oleh pengguna yaitu data diri pasien, dan daftar antrian.

### 6.3.4 Antarmuka Halaman Data Transaksi Pasien

No.	Nama Obat	Harga	Jumlah	Stok Obat
1	paramex	1000	2	Tersedia ✓
2	parsetamol	2000	3	Tersedia ✓

**Gambar 6.4 Antarmuka Halaman Hasil Pemeriksaan Pasien**

Gambar 6.4 merupakan hasil implementasi dari perancangan halaman hasil pemeriksaan pasien yang telah dibuat pada bab sebelumnya. Halaman ini digunakan pengguna untuk melihat data Transaksi pasien. Halaman hasil pemeriksaan pasien memiliki komponen data deskripsi pembayaran, data detail obat yang digunakan, dan tombol konfirmasi pembayaran.

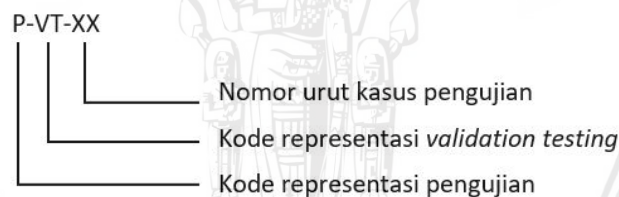
## BAB 7 PENGUJIAN

Pada bab ini menjelaskan kegiatan dalam metode Rational Unified Process (RUP) fase konstruksi tahap pengujian sistem dan fase transisi pengujian kepada pengguna terhadap sistem yang dikembangkan. Pengujian fase konstruksi dilakukan dengan melakukan pengujian *validation testing* untuk menguji persyaratan fungsional pada sistem, pengujian *compability testing* sebagai bagian dari persyaratan non-fungsional sistem dan pengujian fase transisi menggunakan pengujian *user acceptance testing* untuk mengetahui sistem yang dibuat diterima oleh pengguna. Tahap dalam melakukan pengujian pertama dilakukan perancangan pengujian, selanjutnya melakukan pengujian kepada sistem dan pengguna yang menggunakan sistem untuk mengetahui hasil dari pengujian.

### 7.1 Pengujian *Validation Testing*

#### 7.1.1 Perancangan Pengujian *Validation Testing*

Pengujian validasi merupakan salah satu teknik yang terdapat pada metode pengujian black box. Menurut Pressman (2010) pengujian validasi berbasis skenario pada *usecase* dapat mengetahui kesalahan interaksi pada sistem. Dalam melakukan pengujian, terdapat pemberian kode sebagai penanda kasus uji yang dirancang. Ketentuan pemberian kode pengujian ditunjukkan dalam Gambar 7.1.



**Gambar 7.1** Pengkodean Uji *Validation Testing*

Pengujian validasi dilakukan untuk mengetahui bahwa sistem dapat berjalan sesuai harapan pengguna. Terdapat beberapa rancangan pengujian fungsi yang akan digunakan sebagai rangkain pengujian diantaranya menambah data pendaftaran pasien baru, menambah pemeriksaan pasien, dan mengubah data hasil rekam medis, mengisi data transaksi pembayaran pasien. Beberapa rancangan pengujian validasi tiap kasus uji ditunjukkan pada tabel rancangan *Validation Testing*. Tabel 7.1 merupakan tabel rancangan tampilan dari hasil pengujian *Validation Testing* dimana terdapat kode pengujian, tujuan pengujian, hasil yang didapat, dan status.

**Tabel 7.1 Rancangan Tampilan Hasil Pengujian *Validation Testing***

No	Kode pengujian	Tujuan Pengujian	Hasil yang didapat	Status
1				
2				

**7.1.1.1 Perancangan *Validation Testing* Menambah Data Pendaftaran Pasien Baru**

Perancangan *Validation Testing* ini menjelaskan rancangan pengujian pada beberapa fungsi yang dapat digunakan pengguna untuk menambah data pendaftaran baru melalui sistem. Tabel 7.2 menunjukkan beberapa skenario yang dimiliki alur *use case* menambah data pendaftaran baru. Skenario yang ditunjukkan pada Tabel 7.1 akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.2 Rancangan Skenario Testing Menambah Data Pendaftaran Baru**

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	-
Skenario 2	<i>Basic flow</i>	A.2 Inputan data tidak sesuai

Tabel 7.3 menampilkan rancangan pengujian *Validation Testing* Menambah Data Pendaftaran Baru dengan kode P-VT-01. Kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menyimpan data pendaftaran baru pasien.

**Tabel 7.3 Rancangan *Validation Testing* Menambah Data Pendaftaran Pasien Baru**

<b>Kode Pengujian</b>	P-VT-01
<b>Nama Kasus Uji</b>	Kasus uji menambah data pendaftaran pasien baru
<b>Kode Skenario</b>	Skenario 1
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menyimpan data pendaftaran baru.
<b>Test Case</b>	Pengujian dengan memasukan data sesuai format yang ditentukan pada formulir pendaftaran.
<b>Prosedur Uji</b>	.1.1 Penguji mengakses halaman pendaftaran pasien . 1.1 Penguji mengisi formulir pendaftaran sesuai format. 1.2 Penguji menekan tombol simpan.
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan data berhasil ditambahkan dan menampilkan halaman daftar pasien harian bersama data yang baru ditambahkan.

Tabel 7.4 menampilkan rancangan pengujian *Validation Testing* Inputan Data Pendaftaran Baru Tidak Sesuai dengan kode P-VT-02. Kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menampilkan kesalahan inputan data yang tidak sesuai format.



**Tabel 7.4 Rancangan *Validation Testing* Inputan Data Pendaftaran Baru Tidak Sesuai**

<b>Kode Pengujian</b>	P-VT-02
<b>Nama Kasus Uji</b>	Kasus uji Inputan Data Pendaftaran Baru Tidak Sesuai
<b>Kode Skenario</b>	Skenario 2
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan kesalahan inputan data yang tidak sesuai format.
<b>Test Case</b>	Pengujian dengan mengkosongkan nama lengkap pada formulir.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji mengakses halaman pendaftaran pasien.</li> <li>2. Penguji mengisi formulir pendaftaran sesuai format kecuali dengan mengkosongkan nama lengkap pada formulir</li> <li>3. Penguji menekan tombol simpan.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan nama lengkap harus diisi

#### 7.1.1.2 Perancangan *Validation Testing* Menambah Data Pemeriksaan Pasien

Perancangan *Validation Testing* ini menjelaskan rancangan pengujian pada beberapa fungsi yang dapat digunakan pengguna untuk menambah data Pemeriksaan Pasien melalui sistem. Tabel 7.5 menunjukkan beberapa skenario yang dimiliki alur *use case* menambah data pemeriksaan Pasien. Skenario yang ditunjukkan pada Tabel 7.4 akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.5 Rancangan Skenario Testing Menambah Data Pemeriksaan Pasien**

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	-
Skenario 2	<i>Basic flow</i>	A.2 Inputan data tidak sesuai

Tabel 7.6 menampilkan rancangan pengujian *Validation Testing* Menambah Data Pemeriksaan Pasien dengan kode P-VT-03. Kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menyimpan data pemeriksaan baru pasien.

**Tabel 7.6 Rancangan *Validation Testing* Menambah Data Pemeriksaan Pasien**

<b>Kode Pengujian</b>	P-VT-03
<b>Nama Kasus Uji</b>	Kasus uji menambah data Pemeriksaan Pasien
<b>Kode Skenario</b>	Skenario 1
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menyimpan data Pemeriksaan baru.
<b>Test Case</b>	Pengujian dengan memasukan data sesuai format yang ditentukan pada formulir pemeriksaan baru.

<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji mengakses halaman pemeriksaan pasien.</li> <li>2. Penguji mengisi formulir pemeriksaan baru sesuai format.</li> <li>3. Penguji menekan tombol simpan.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan data berhasil ditambahkan dan menampilkan halaman data pendaftaran bersama data yang baru ditambahkan.

Tabel 7.7 menampilkan rancangan pengujian *Validation Testing* Inputan Data Pemeriksaan Baru Tidak Sesuai dengan kode P-VT-04. Kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menampilkan kesalahan inputan data yang tidak sesuai format.

**Tabel 7.7 Rancangan *Validation Testing* Inputan Data Pemeriksaan Pasien Tidak Sesuai**

<b>Kode Pengujian</b>	P-VT-04
<b>Nama Kasus Uji</b>	Kasus uji Inputan Data Pemeriksaan Pasien Tidak Sesuai
<b>Kode Skenario</b>	Skenario 2
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan kesalahan inputan data yang tidak sesuai format.
<b>Test Case</b>	Pengujian dengan mengkosongkan tindakan pada formulir dan memberi inputan tensi dengan karakter random.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji mengakses halaman pemeriksaan pasien.</li> <li>2. Penguji mengisi formulir pemeriksaan baru sesuai format kecuali dengan mengkosongkan tindakan pemeriksaan pada formulir dan memberi inputan tensi dengan karakter random.</li> <li>3. Penguji menekan tombol simpan.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan tanggal pemeriksaan harus diisi dan inputan salah harus menggunakan angka.

### 7.1.1.3 Perancangan *Validation Testing* Mengubah Data Hasil Rekam Medis

Perancangan *Validation Testing* ini menjelaskan rancangan pengujian pada beberapa fungsi yang dapat digunakan pengguna untuk Mengubah Data Hasil Rekam Medis melalui sistem. Tabel 7.8 menunjukkan beberapa skenario yang dimiliki alur *use case* Mengubah Data Hasil Rekam Medis. Skenario yang ditunjukkan pada Tabel 7.7 akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.8 Rancangan Skenario Testing Mengubah Data Hasil Rekam Medis**

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	-
Skenario 2	<i>Basic flow</i>	A.2 Data pemeriksaan belum lengkap terisi

Tabel 7.9 menampilkan rancangan pengujian *Validation Testing* Mengubah Data Hasil Akhir Pemeriksaan dengan kode P-VT-05. Kasus uji ini dilakukan untuk memastikan bahwa sistem dapat mengubah data hasil Rekam Medis.

**Tabel 7.9 Rancangan *Validation Testing* Mengubah Data Hasil Rekam Medis**

<b>Kode Pengujian</b>	P-VT-05
<b>Nama Kasus Uji</b>	Kasus uji Mengubah Data Hasil Rekam Medis
<b>Kode Skenario</b>	Skenario 1
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat mengubah data hasil Rekam medis
<b>Test Case</b>	Pengujian dengan menekan tombol Mengubah pada Rekam Medis pasien yang seluruh data hasil pemeriksaanya telah terisi.
<b>Prosedur Uji</b>	L.1 Penguji mengakses halaman Data Rekam Medis. L.2 Penguji menekan tombol Mengubah dan mengisi data rekam medis. L.3 Penguji menyimpan Data Rekam Medis
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan data berhasil diperbarui dan menampilkan data hasil Rekam medis pasien keseluruhan.

Tabel 7.10 menampilkan rancangan pengujian *Validation Testing* Mengubah Data Rekam Medis Belum Lengkap dengan kode P-VT-06. Kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menampilkan pemberitahuan bahwa pemeriksaan belum lengkap ketika menekan tombol simpan.

**Tabel 7.10 Rancangan *Validation Testing* Mengubah Data Hasil Rekam Medis Belum Lengkap**

<b>Kode Pengujian</b>	P-VT-06
<b>Nama Kasus Uji</b>	Kasus uji Mengubah Data Hasil Rekam medis belum lengkap
<b>Kode Skenario</b>	Skenario 2
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan pesan bahwa hasil rekam medis belum lengkap terisi.
<b>Test Case</b>	Pengujian dengan menekan tombol Mengubah pada Rekam medis yang data hasil pemeriksaanya terisi sebagian.
<b>Prosedur Uji</b>	1. Penguji mengakses halaman Rekam Medis. 2. Penguji menekan tombol Mengubah. Lalu tidak mengisi Data rekam medis

<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan data Rekam medis hasil pemeriksaan belum lengkap.
------------------------------	--

#### 7.1.1.4 Perancangan *Validation Testing* Mengisi Data Transaksi Pembayaran Pasien

Perancangan *Validation Testing* ini menjelaskan rancangan pengujian pada beberapa fungsi yang dapat digunakan pengguna untuk Mengisi Data Transaksi Pembayaran Pasien melalui sistem. Tabel 7.11 menunjukkan beberapa skenario yang dimiliki alur *use case* Mengisi Data Transaksi Pembayaran Pasien. Skenario yang ditunjukkan pada Tabel 7.10 akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.11 Rancangan Skenario Testing Mengisi Data Transaksi Pembayaran Pasien**

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	-
Skenario 2	<i>Basic flow</i>	A.2 Inputan data tidak sesuai

Tabel 7.12 menampilkan rancangan pengujian *Validation Testing* Mengisi Data Transaksi Pembayaran Pasien dengan kode P-VT-07. Kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menyimpan data inputan hasil Transaksi Pembayaran Pasien.

**Tabel 7.12 Rancangan *Validation Testing* Mengisi Data Transaksi Pembayaran Pasien**

<b>Kode Pengujian</b>	P-VT-07
<b>Nama Kasus Uji</b>	Kasus uji Mengisi Data Transaksi Pembayaran Pasien
<b>Kode Skenario</b>	Skenario 1
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menyimpan data hasil Transaksi Pembayaran Pasien.
<b>Test Case</b>	Pengujian dengan memasukan data sesuai format yang ditentukan pada halaman data Transaksi Pembayaran Pasien.
<b>Prosedur Uji</b>	1.1 Penguji mengakses halaman Transaksi Pembayaran Pasien. 1.2 Penguji mengisi hasil pemeriksaan sesuai format. 1.3 Penguji menekan tombol simpan.
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan data berhasil terisi dan menampilkan halaman daftar pasien.

Tabel 7.13 menampilkan rancangan pengujian *Validation Testing* Mengisi Transaksi Pembayaran Pasien Belum Lengkap dengan kode P-VT-06. Kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menampilkan pemberitahuan

bahwa Transaksi Pembayaran Pasien belum lengkap ketika menekan tombol konfirmasi pembayaran.

**Tabel 7.13 Rancangan *Validation Testing* Inputan Data Transaksi Pembayaran Pasien Tidak Sesuai**

<b>Kode Pengujian</b>	P-VT-08
<b>Nama Kasus Uji</b>	Kasus uji Inputan Data Transaksi Pembayaran Pasien Sesuai
<b>Kode Skenario</b>	Skenario 2
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan kesalahan inputan data yang tidak sesuai format.
<b>Test Case</b>	Pengujian dengan mengkosongkan nominal pembayaran pada formulir dan memberi inputan <i>text</i> pada nominal pembayaran.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji mengakses halaman Transaksi Pembayaran Pasien</li> <li>2. Penguji mengisi formulir Transaksi Pembayaran Pasien sesuai format kecuali dengan mengkosongkan nominal pembayaran pada formulir dan memberi inputan <i>text</i> pada nominal pembayaran.</li> <li>3. Penguji menekan tombol konfirmasi pembayaran.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan nominal pembayaran harus diisi dan inputan harus berupa angka.

### 7.1.2 Hasil Pengujian *Validation Testing*

Tabel 7.14 merupakan hasil dari pengujian yang telah dilakukan yang terdiri dari kode pengujian, tujuan pengujian, hasil yang didapat, dan status. *Validation testing* dilakukan berdasarkan sub bab perancangan pengujian pada setiap kasus uji yang sudah dibuat. Terdapat 8 kasus uji yang sudah didefinisikan yang berasal dari empat *use case scenario* yang berbeda diantaranya tambah data menambah data pendaftaran pasien baru, menambah pemeriksaan pasien, dan mengubah data hasil rekam medis, mengisi data transaksi pembayaran pasien.

**Tabel 7.14 Hasil Pengujian *Validation Testing***

No	Kode Pengujian	Tujuan Pengujian	Hasil yang didapat	Status
1	P-VT-01	Pengujian untuk memastikan bahwa sistem dapat menyimpan data pendaftaran pasien baru.	Data pendaftaran baru berhasil disimpan dan mengembalikan pada tampilan daftar pasien harian dengan pesan berhasil serta	Valid

No	Kode Pengujian	Tujuan Pengujian	Hasil yang didapat	Status
			menampilkan data yang baru ditambahkan.	
2	P-VT-02	Pengujian untuk memastikan bahwa sistem dapat menampilkan kesalahan inputan data yang tidak sesuai format pada penambahan data pendaftaran pasien baru.	Sistem berhasil menampilkan kesalahan inputan berdasarkan <i>test case</i> yang diberikan.	Valid
3	P-VT-03	Pengujian untuk memastikan bahwa sistem dapat menyimpan data Pemeriksaan pasien.	Data pemeriksaan pasien berhasil disimpan dan mengembalikan pada tampilan data pendaftaran dengan pesan berhasil serta menampilkan data yang baru ditambahkan.	Valid
4	P-VT-04	Pengujian untuk memastikan bahwa sistem dapat menampilkan kesalahan inputan data yang tidak sesuai format pada penambahan data pemeriksaan baru.	Sistem berhasil menampilkan kesalahan inputan berdasarkan <i>test case</i> yang diberikan.	Valid
5	P-VT-05	Pengujian untuk memastikan bahwa sistem dapat mengubah data hasil rekam medis.	Sistem berhasil menampilkan pesan data berhasil diperbarui dan menampilkan data hasil rekam medis.	Valid
6	P-VT-06	Pengujian untuk memastikan bahwa sistem dapat menampilkan pesan bahwa hasil pemeriksaan belum lengkap terisi pada mengubah data hasil akhir pemeriksaan.	Sistem berhasil menampilkan pesan pemeriksaan belum lengkap dengan <i>test case</i> yang diberikan.	Valid

No	Kode Pengujian	Tujuan Pengujian	Hasil yang didapat	Status
7	P-VT-07	Pengujian untuk memastikan bahwa sistem dapat menyimpan data transaksi pembayaran pasien.	Data hasil transaksi pembayaran pasien berhasil disimpan dan mengembalikan pada tampilan daftar pasien dengan pesan berhasil.	Valid
8	P-VT-08	Pengujian untuk memastikan bahwa sistem dapat menampilkan kesalahan inputan data yang tidak sesuai format pada mengisi data hasil pemeriksaan.	Sistem berhasil menampilkan kesalahan inputan berdasarkan <i>test case</i> yang diberikan.	Valid

Hasil Pengujian dengan menggunakan metode *validation testing* dilakukan pada beberapa fitur dalam sistem informasi. Persentase validitas pengujian dihitung menggunakan rumus berikut :

$$\text{Persentase} = (\text{Jumlah Uji Kasus Valid} / \text{Jumlah Uji Kasus yang Diujikan}) \times 100\%$$

$$\text{Persentase} = (8 / 8) \times 100\% \quad \text{Persentase} = 100\% \text{ (Valid)}$$

### 7.1.3 Kesimpulan *Validation Testing*

Presman (2010) menjelaskan bahwa setiap uji kasus yang telah selesai dilakukan akan menunjukkan 2 kemungkinan. Pertama hasil pengujian kasus uji menunjukkan karakteristik fungsionalitas atau kinerja sesuai dengan spesifikasi yang telah ditentukan maka sistem yang dibangun dapat diterima pengguna. Kedua jika terjadi penyimpangan fungsionalitas dari spesifikasi ditemukan maka daftar kekurangan/cacat pada sistem harus dibuat.





Hasil dari *Validation Testing* yang telah dilakukan pada beberapa kasus uji yang sudah didefinisikan yang berasal dari empat *use case scenario* yang berbeda diantaranya tambah data pemeriksaan pasien baru, menambah data pemeriksaan pasien, mengubah data hasil rekam medis, dan mengisi data transaksi pembayaran pasien menunjukkan hasil status valid dengan presentase 100%. Maka dapat ditarik kesimpulan bahwa sistem yang telah dibangun dapat diterima karena telah memenuhi kriteria fungsi dan karakteristik yang sudah ditentukan.

## 7.2 Pengujian *Compatibility Testing*

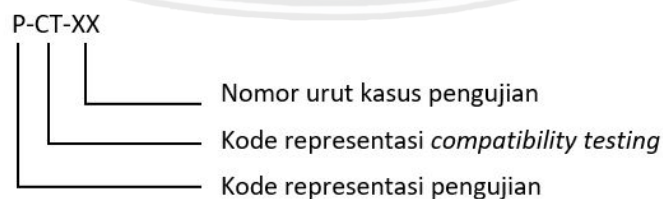
### 7.2.1 Perancangan Pengujian *Compatibility Testing*

Pengujian *Compatibility Testing* dilakukan untuk mengetahui kesesuaian sistem yang akan dikembangkan dengan beberapa aplikasi peramban. Dalam pengujian ini digunakan aplikasi perangkat lunak Sortsite versi 5.32.838.0 yang akan memeriksa komponen tag HTML, CSS, script, kesalahan dalam pemanggilan halaman, teknologi yang tidak didukung oleh *browser* tertentu. Terdapat beberapa kategori masalah untuk mengetahui tingkat kompatibilitas sistem sesuai dengan yang dijelaskan dalam tabel 7.15.

**Tabel 7.15 Kategori Masalah Kompatilitas**

	Tidak terjadi permasalahan pada konten atau fungsionalitas
	Konten atau fungsionalitas hilang
	Masalah mayor pada <i>layout</i> atau performa
	Masalah minor pada <i>layout</i> atau performa

*Compatibility testing* akan menguji komponen-komponen penyusun dalam mengembangkan Sistem Informasi Rawat Inap untuk mengetahui komponen tersebut kompatibel dengan beberapa jenis peramban yang menjadi kasus uji. Komponen-komponen penyusun dalam mengembangkan sistem akan diuji pada beberapa peramban untuk mengetahui kompatibilitas sistem pada tiap peramban. Terdapat beberapa rancangan kasus uji dalam melakukan serangkaian rancangan pengujian tersebut. Pada Gambar 7.2 merupakan kodifikasi *Compatibility testing* yang akan digunakan untuk rancangan kasus uji.



**Gambar 7.2 Pengkodean Uji *Compatibility Testing***

*Compatibility testing* dalam pelaksanaan terdapat beberapa rancangan kasus uji sebagai panduan pengujian untuk mengetahui tujuan, prosedur dan hasil yang diharapkan pada tiap pengujian. Terdapat 8 rancangan kasus uji yang diambil dari tiap jenis dan versi peramban untuk melakukan pengujian ini yang ditujukan pada



tabel-tabel dibawah ini. Peramban yang dilakukan pengujian adalah *internet explorer, edge, firefox, safari, opera, chrome, iOS, dan Android*.

**Tabel 7.16 Rancangan Pengujian *Compatibility* Sistem Kasus Uji Peramban *Internet Explorer***

<b>Kode Pengujian</b>	P-CT-01
<b>Nama Kasus Uji</b>	Kasus uji kompabiitas pada peramban <i>internet explorer</i>
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk mengetahui apakah sistem dapat berjalan baik pada peramban <i>internet explorer</i>
<b>Test Case</b>	Menguji kompabilitas sistem terhadap peramban <i>Internet Explorer</i>
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>4. Penguji menjalankan aplikasi SortSite</li> <li>5. Penguji memasukkan url sistem pada menu <i>bar address</i></li> <li>6. Penguji melakukan checking dengan menekan tombol check</li> <li>7. Penguji melihat hasil <i>compability testing</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat berjalan dengan baik pada peramban <i>Internet Explorer</i>

**Tabel 7.17 Rancangan Pengujian *Compatibility* Sistem Kasus Uji Peramban *Microsoft Edge***

<b>Kode Pengujian</b>	P-CT-02
<b>Nama Kasus Uji</b>	Kasus uji kompabiitas pada peramban <i>Microsoft Edge</i>
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk mengetahui apakah sistem dapat berjalan baik pada peramban <i>Microsoft Edge</i>
<b>Test Case</b>	Menguji kompabilitas sistem terhadap peramban <i>Microsoft Edge</i>
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji menjalankan aplikasi SortSite</li> <li>2. Penguji memasukkan url sistem pada menu <i>bar address</i></li> <li>3. Penguji melakukan checking dengan menekan tombol check</li> <li>4. Penguji melihat hasil <i>compability testing</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat berjalan dengan baik pada peramban <i>Microsoft Edge</i>

**Tabel 7.18 Rancangan Pengujian *Compatibility* Sistem Kasus Uji Peramban *Firefox***

<b>Kode Pengujian</b>	P-CT-03
<b>Nama Kasus Uji</b>	Kasus uji kompabiitas pada peramban <i>Firefox</i>

<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk mengetahui apakah sistem dapat berjalan baik pada peramban <i>Firefox</i>
<b>Test Case</b>	Menguji kemampuan sistem terhadap peramban <i>Firefox</i>
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji menjalankan aplikasi SortSite</li> <li>2. Penguji memasukkan url sistem pada menu <i>bar address</i></li> <li>3. Penguji melakukan checking dengan menekan tombol check</li> <li>4. Penguji melihat hasil <i>compability testing</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat berjalan dengan baik pada peramban <i>Firefox</i>

**Tabel 7.19 Rancangan Pengujian *Compatibility* Sistem Kasus Uji Peramban *Safari***

<b>Kode Pengujian</b>	P-CT-04
<b>Nama Kasus Uji</b>	Kasus uji kemampuan pada peramban <i>Safari</i>
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk mengetahui apakah sistem dapat berjalan baik pada peramban <i>Safari</i>
<b>Test Case</b>	Menguji kemampuan sistem terhadap peramban <i>Safari</i>
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji menjalankan aplikasi SortSite</li> <li>2. Penguji memasukkan url sistem pada menu <i>bar address</i></li> <li>3. Penguji melakukan checking dengan menekan tombol check</li> <li>4. Penguji melihat hasil <i>compability testing</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat berjalan dengan baik pada peramban <i>Safari</i>

**Tabel 7.20 Rancangan Pengujian *Compatibility* Sistem Kasus Uji Peramban *Opera***

<b>Kode Pengujian</b>	P-CT-05
<b>Nama Kasus Uji</b>	Kasus uji kemampuan pada peramban <i>Opera</i>
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk mengetahui apakah sistem dapat berjalan baik pada peramban <i>Opera</i>
<b>Test Case</b>	Menguji kemampuan sistem terhadap peramban <i>Opera</i>
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji menjalankan aplikasi SortSite</li> <li>2. Penguji memasukkan url sistem pada menu <i>bar address</i></li> <li>3. Penguji melakukan checking dengan menekan tombol check</li> </ol>

	4. Penguji melihat hasil <i>compability testing</i>
<b>Hasil yang diharapkan</b>	Sistem dapat berjalan dengan baik pada peramban <i>Opera</i>

**Tabel 7.21 Rancangan Pengujian *Compatibility* Sistem Kasus Uji Peramban *Chrome***

<b>Kode Pengujian</b>	P-CT-06
<b>Nama Kasus Uji</b>	Kasus uji kompabiitas pada peramban <i>Chrome</i>
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk mengetahui apakah sistem dapat berjalan baik pada peramban <i>Chrome</i>
<b>Test Case</b>	Menguji kompabilitas sistem terhadap peramban <i>Chrome</i>
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji menjalankan aplikasi SortSite</li> <li>2. Penguji memasukkan url sistem pada menu <i>bar address</i></li> <li>3. Penguji melakukan checking dengan menekan tombol check</li> <li>4. Penguji melihat hasil <i>compability testing</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat berjalan dengan baik pada peramban <i>Chrome</i>

**Tabel 7.22 Rancangan Pengujian *Compatibility* Sistem Kasus Uji Peramban *iOS***

<b>Kode Pengujian</b>	P-CT-07
<b>Nama Kasus Uji</b>	Kasus uji kompabiitas pada peramban <i>iOS</i>
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk mengetahui apakah sistem dapat berjalan baik pada peramban <i>iOS</i>
<b>Test Case</b>	Menguji kompabilitas sistem terhadap peramban <i>iOS</i>
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji menjalankan aplikasi SortSite</li> <li>2. Penguji memasukkan url sistem pada menu <i>bar address</i></li> <li>3. Penguji melakukan checking dengan menekan tombol check</li> <li>4. Penguji melihat hasil <i>compability testing</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat berjalan dengan baik pada peramban <i>iOS</i>

**Tabel 7.23 Rancangan Pengujian *Compatibility* Sistem Kasus Uji Peramban *Android***

<b>Kode Pengujian</b>	P-CT-08
<b>Nama Kasus Uji</b>	Kasus uji kompabiitas pada peramban <i>Android</i>
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk mengetahui apakah sistem dapat berjalan baik pada peramban <i>Android</i>

<b>Test Case</b>	Menguji kompatibility sistem terhadap peramban <i>Android</i>
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji menjalankan aplikasi SortSite</li> <li>2. Penguji memasukkan url sistem pada menu <i>bar address</i></li> <li>3. Penguji melakukan checking dengan menekan tombol check</li> <li>4. Penguji melihat hasil <i>compability testing</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat berjalan dengan baik pada peramban <i>Android</i>

Hasil dari beberapa kasus uji pengujian kompatibility akan ditampilkan dalam bentuk ringkasan. Tabel 7.24 merupakan hasil pengujian yang akan ditampilkan pada aplikasi *sortsite* Dimana pada kolom vertikal terdapat kategori penilaian dan kolom horizontal terdapat kolom jenis dan versi browser. Titik temu antara kategori dan jenis browser merupakan hasil dari pengujian kompatibility.

**Tabel 7.24 Tampilan Hasil Pengujian *Compatibility***

<i>Browser Version</i>	IE 11	Edge 17	Firefox 61	Safari =<10 11	Opera 54	Chrome 68	iOS =<9 10 11	Android =<3 4
Critical Issues								
Major Issues								
Minor Issues								

### 7.2.2 Hasil Pengujian *Compatibility Testing*

Hasil dari pengujian *Compatibility Testing* berdasarkan perancangan pengujian yang sebelumnya dilakukan dijelaskan pada subbab ini. Pengujian dilakukan dengan bantuan aplikasi peramban *SortSite*. Hasil pengujian dari pengujian *Compatibility Testing* dapat dilihat pada Gambar 7.3.

Browser Version	IE 11	Edge 17	Firefox 61	Safari ≤ 10 11	Opera 54	Chrome 68	iOS ≤ 9 10 11	Android ≤ 3 4*
Critical Issues	✓	✓	●	✓	✓	✓	●	✓
Major Issues	✓	✓		✓	✓	✓	✓	✓
Minor Issues	✓	✓		✓	✓	✓	✓	✓

**Gambar 7.3 Hasil *Compatibility Testing***

Gambar 7.3 menunjukkan hasil yaitu terdapat 2 critical issues pada peramban firefox dan chrome disebabkan tidak diaktifkannya fungsi SSL (*secure socket layer*). Peramban selain firefox dan chrome dalam Gambar 7.3 tidak menunjukkan masalah *critical issues*, *major issues*, dan *minor issues*.

### 7.2.3 Kesimpulan Pengujian *Compatibility Testing*

Pengujian kompatibilitas didapat hasil 2 critical issues pada peramban firefox dan chrome karena disebabkan tidak diaktifkannya fungsi SSL (*secure socket layer*) dan pada peramban lain tidak terjadi masalah. Berdasarkan kebutuhan non-fungsional yang didefinisikan maka dapat disimpulkan sistem dapat diterima dengan baik atau telah memenuhi persyaratan non-fungsional pengguna karena error yang terjadi pada peramban tersebut tidak termasuk pada kebutuhan persyaratan non-fungsional yang sudah didefinisikan sebelumnya.

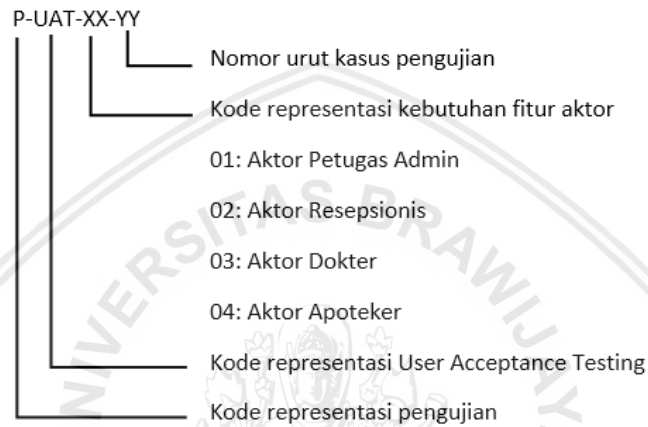
## 7.3 Kesimpulan Pengujian Fase Konstruksi Metode *Rational Unifield Process*

Pengujian pada fase konstruksi metode rational unifield process telah dilakukan. pengujian dilakukan berkaitan dengan sistem yang sudah dibangun. Pengujian meliputi pengujian fungsional dengan *validation testing* dan non-fungsional dengan *compabilty testing*. Berdasarkan hasil pengujian yang telah dilakukan sebelumnya menunjukkan sistem telah sesuai dengan persyaratan pengguna, sehingga standar pencapaian fase konstruksi atau initial operational capability pada RUP telah terpenuhi.

## 7.4 Pengujian *User Acceptance Testing* (UAT)

### 7.4.1 Perancangan Pengujian *User Acceptance Testing* (UAT)

Pada subbab pengujian terdapat rancangan pengujian user acceptance testing (UAT) untuk landasan dalam melakukan pengujian sistem yang telah dibuat. *User Acceptance Testing* diperlukan untuk sistem yang telah dikembangkan yang didapatkan dengan baik menurut kriteria yang sudah pengguna tentukan. Untuk melakukan pengujian UAT, ada pemberian kode dalam penanda kasus uji yang akan dirancang. Berikut terdapat Ketentuan untuk pemberian kode pada pengujian dalam Gambar 7.4.



**Gambar 7.4 Kode Uji *User Acceptance Testing* (UAT)**

Dalam melakukan pengujian, program didemonstrasikan oleh aktor pengguna sistem. lalu responden mengisi kuisioner dari penguji bertujuan mengetahui penilaian dari responden terhadap sistem yang dikembangkan. Responden pada pengujian diambil dari petugas admin, *Front office*, dokter, dan petugas apoteker. Terdapat 4 kriteria pertanyaan yang digunakan dalam menilai sistem yaitu *performance, usability, functional correctness and completeness, and Confidentially and Availability* untuk memastikan bahwa keluaran pada sistem telah sesuai. Berikut adalah uji kasus untuk validasi sistem berdasarkan fungsionalitas yang dimiliki tiap aktor dibedakan menjadi empat tabel yaitu fungsionalitas untuk petugas admin, *Front office*, dokter, dan petugas apoteker. Setiap tabel disajikan informasi kriteria, kode kasus uji, dan daftar pertanyaan.

**Tabel 7.25 Rancangan Kasus Uji UAT untuk Petugas Admin**

No	Kriteria	Kode Kasus Uji	Daftar Pertanyaan
1	<i>Performance</i>	P-UAT-01-01	Apakah Sistem Informasi Layanan Rawat Inap HMC dapat diakses dengan cepat ?
		P-UAT-01-02	Apakah Sistem Informasi Rawat Inap HMC dapat mempercepat pengguna untuk mendapatkan informasi seluruh data

No	Kriteria	Kode Kasus Uji	Daftar Pertanyaan
			pengguna dan data pemeriksaan pasien rawat inap ?
2	<i>Usability</i>	P-UAT-01-03	Apakah Sistem Informasi Rawat Inap HMC mudah digunakan ?
		P-UAT-01-04	Apakah Sistem Informasi Rawat Inap HMC mempermudah pengguna mendapatkan informasi seluruh data pengguna dan data pemeriksaan pasien rawat inap?
		P-UAT-01-05	Apakah Sistem Informasi Rawat Inap HMC mempermudah pengguna mengelola seluruh data pengguna dan data pemeriksaan pasien rawat inap?
3	<i>Functional Correctness and Completeness</i>	P-UAT-01-06	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam mendapatkan informasi seluruh data pengguna dan data pemeriksaan pasien rawat inap?
		P-UAT-01-07	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam mengelola seluruh data pengguna dan data pemeriksaan pasien rawat inap?
4	<i>Confidentially and Availability</i>	P-UAT-01-08	Apakah login pengguna pada Sistem Informasi Rawat Inap HMC membantu keamanan informasi data pengguna dan data pemeriksaan pasien rawat inap ?

**Tabel 7.25 Rancangan Kasus Uji UAT untuk Petugas *Front office***

No	Kriteria	Kode Kasus Uji	Daftar Pertanyaan
1	<i>Performance</i>	P-UAT-02-01	Apakah Sistem Informasi Rawat Inap HMC dapat diakses dengan cepat ?
		P-UAT-02-02	Apakah Sistem Informasi Rawat Inap HMC dapat mempercepat pendaftaran pasien baru?
		P-UAT-02-03	Apakah Sistem Informasi Rawat Inap HMC dapat mempercepat pengguna untuk mendapatkan informasi data pendaftaran pasien ?
2	<i>Usability</i>	P-UAT-02-04	Apakah Sistem Informasi Rawat Inap HMC mudah digunakan ?

		P-UAT-02-05	Apakah Sistem Informasi Rawat Inap HMC mempermudah pendaftaran pasien baru?
		P-UAT-02-06	Apakah Sistem Informasi Rawat Inap HMC mempermudah pengguna untuk mendapatkan informasi data pendaftaran pasien ?
3	<i>Functional Correctness and Completeness</i>	P-UAT-02-07	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam melihat seluruh data pendaftaran pasien ?
		P-UAT-02-08	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu mencari data pendaftaran pasien tertentu ?
		P-UAT-02-09	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam pendaftaran pasien baru untuk di rawat inap?
4	<i>Confidentially and Availability</i>	P-UAT-02-10	Apakah login pengguna pada Sistem Informasi Rawat Inap HMC membantu keamanan informasi data pendaftaran pasien ?

**Tabel 7.26 Rancangan Kasus Uji UAT untuk Dokter**

No	Kriteria	Kode Kasus Uji	Daftar Pertanyaan
1	<i>Performance</i>	P-UAT-03-01	Apakah Sistem Informasi Rawat Inap HMC dapat diakses dengan cepat ?
		P-UAT-03-02	Apakah Sistem Informasi Rawat Inap HMC dapat mempercepat pengguna untuk mendapatkan informasi pemeriksaan kesehatan fisik dan kejiwaan pasien ?
2	<i>Usability</i>	P-UAT-03-03	Apakah Sistem Informasi Rawat Inap HMC mudah digunakan ?
		P-UAT-03-04	Apakah Sistem Informasi Rawat Inap HMC mempermudah pengelolaan data hasil pemeriksaan fisik dan kejiwaan pasien ?
3	<i>Functional Correctness and Completeness</i>	P-UAT-03-05	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam penyajian informasi permintaan pemeriksaan kesehatan fisik dan kejiwaan pasien ?
		P-UAT-03-06	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam mengelola data hasil pemeriksaan fisik dan kejiwaan pasien ?





4	<i>Confidentially and Availability</i>	P-UAT-03-07	Apakah login pengguna pada Sistem Informasi Rawat Inap HMC membantu keamanan informasi data pemeriksaan fisik dan kejiwaan pasien untuk rawat inap?
---	--	-------------	---

**Tabel 7.28 Rancangan Kasus Uji UAT untuk Apoteker**

No	Kriteria	Kode Kasus Uji	Daftar Pertanyaan
1	<i>Performance</i>	P-UAT-04-01	Apakah Sistem Informasi Rawat Inap HMC dapat diakses dengan cepat ?
		P-UAT-04-02	Apakah Sistem Informasi Rawat Inap HMC dapat mempercepat pengguna untuk mendapatkan informasi data obat dan data resep obat dari hasil pemeriksaan pasien ?
2	<i>Usability</i>	P-UAT-04-03	Apakah Sistem Informasi Rawat Inap HMC mudah digunakan ?
		P-UAT-04-04	Apakah Sistem Informasi Rawat Inap HMC mempermudah pengelolaan data obat dan data resep obat dari hasil pemeriksaan pasien ?
3	<i>Functional Correctness and Completeness</i>	P-UAT-04-05	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam penyajian informasi permintaan resep obat dari hasil pemeriksaan pasien?
		P-UAT-04-06	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam mengelola data obat dan data resep obat dari hasil pemeriksaan pasien ?
4	<i>Confidentially and Availability</i>	P-UAT-04-07	Apakah login pengguna pada Sistem Informasi Rawat Inap HMC membantu keamanan informasi data obat dan resep obat dari hasil pemeriksaan pasien ?

## 7.4.2 Hasil Pengujian *User Acceptance Testing* (UAT)

*User acceptance testing* dilakukan berdasarkan perancangan pengujian yang dibuat sebelumnya kepada setiap kasus uji yang dibuat. Terdapat empat rancangan *user acceptance testing* yang sudah didefinisikan untuk lima aktor yang berbeda. Pengujian ini melibatkan 5 responden yaitu 1 orang petugas admin, 1 petugas *Front office*, 1 orang dokter, 1 petugas apoteker, 1 perawat. Berikut adalah hasil dari *user acceptance testing* sistem informasi yang dikembangkan.

### 7.4.2.1 Hasil Pengujian *User Acceptance Testing* untuk Petugas Admin

Subbab ini merupakan hasil kuisioner *User Acceptance Testing* untuk Petugas Admin yang berisi pertanyaan, kode uji, dan jawaban. Responden dalam *User Acceptance Testing* ini adalah satu petugas Administrasi. Hasil kuisioner *User Acceptance Testing* untuk Petugas Administrasi terdapat pada Tabel 7.29.

**Tabel 7.29 Hasil Pengujian *User Acceptance Testing* untuk Petugas Admin**

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
1	Apakah Sistem Informasi Layanan Rawat Inap HMC dapat diakses dengan cepat ?	P-UAT-01-01	0	1	0	0	0
2	Apakah Sistem Informasi Rawat Inap HMC dapat mempercepat pengguna untuk mendapatkan informasi seluruh data pengguna dan data pemeriksaan pasien rawat inap ?	P-UAT-01-02	1	0	0	0	0
3	Apakah Sistem Informasi Rawat Inap HMC mudah digunakan ?	P-UAT-01-03	1	0	0	0	0
4	Apakah Sistem Informasi Rawat Inap HMC mempermudah pengguna mendapatkan informasi seluruh data pengguna dan data pemeriksaan pasien rawat inap?	P-UAT-01-04	1	1	0	0	0
5	Apakah Sistem Informasi Rawat Inap HMC mempermudah pengguna mengelola seluruh data pengguna dan data	P-UAT-01-05	0	1	0	0	0

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
	pemeriksaan pasien rawat inap?						
6	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam mendapatkan informasi seluruh data pengguna dan data pemeriksaan pasien rawat inap?	P-UAT-01-06	1	0	0	0	0
7	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam mengelola seluruh data pengguna dan data pemeriksaan pasien rawat inap?	P-UAT-01-07	0	1	0	0	0
8	Apakah login pengguna pada Sistem Informasi Rawat Inap HMC membantu keamanan informasi data pengguna dan data pemeriksaan pasien rawat inap ?	P-UAT-01-08	0	1	0	0	0

Hasil dari *User Acceptance Testing* yang dilakukan selanjutnya dianalisis menggunakan skala likert. Teori dan rumus perhitungan skala likert diimplementasikan sesuai dengan perancangan pada pengujian *User Acceptance Testing* ini. Berikut merupakan uraian poin-poin perhitungan hasil *User Acceptance Testing* menggunakan skala likert:

- a. Menghitung total nilai berdasarkan jawaban dari satu responden petugas Admin.

**Tabel 7.30 Hasil Pengujian *User Acceptance Testing* untuk Petugas Admin**

Jawaban	Bobot nilai	Jumlah jawaban	Hasil
Sangat Setuju (A)	5	2	10
Setuju (B)	4	5	20
Netral (C)	3	0	0
Tidak Setuju (D)	2	0	0
Sangat Tidak Setuju (E)	1	0	0
Total			30

b. Nilai Y.

$Y = \text{Jumlah Skala} \times \text{Jumlah Responden} \times \text{Jumlah Pertanyaan}$

$Y = 5 \times 1 \times 8 = 40$

c. Nilai Presentase.

$\text{Rumus Index} = (\text{Total Nilai} \div Y) \times 100\%$

$\text{Rumus Index} = (30 \div 40) \times 100\% = 75\% \text{ (Setuju)}$

**7.4.2.2 Hasil Pengujian User Acceptance Testing untuk Petugas Front office**

Subbab ini merupakan hasil kuisioner *User Acceptance Testing* untuk *Petugas Front office* yang berisi pertanyaan, kode uji, dan jawaban. Responden dalam *User Acceptance Testing* ini adalah satu petugas *Front office*. Hasil kuisioner *User Acceptance Testing* untuk *Petugas Front office* terdapat pada Tabel 7.31.

**Tabel 7.31 Hasil Pengujian User Acceptance Testing untuk Petugas Front office**

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
1	Apakah Sistem Informasi Rawat Inap HMC dapat diakses dengan cepat ?	P-UAT-02-01		1	0	0	0
2	Apakah Sistem Informasi Rawat Inap HMC dapat mempercepat pendaftaran pasien baru?	P-UAT-02-02	0	1	0	0	0
3	Apakah Sistem Informasi Rawat Inap HMC dapat mempercepat pengguna untuk mendapatkan informasi data pendaftaran pasien ?	P-UAT-02-03	0	1	0	0	0
4	Apakah Sistem Informasi Rawat Inap HMC mudah digunakan ?	P-UAT-02-04	1	0	0	0	0
5	Apakah Sistem Informasi Rawat Inap HMC mempermudah pendaftaran pasien baru?	P-UAT-02-05	1	0	0	0	0
6	Apakah Sistem Informasi Rawat Inap HMC mempermudah pengguna untuk mendapatkan informasi data pendaftaran pasien ?	P-UAT-02-06	0	1	0	0	0
7	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam	P-UAT-02-07	0	0	1	0	0

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
	melihat seluruh data pendaftaran pasien ?						
8	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu mencari data pendaftaran pasien tertentu ?	P-UAT-02-08	1	0	0	0	0
9	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam pendaftaran pasien baru untuk di rawat inap?	P-UAT-02-09	0	1	0	0	0
10	Apakah login pengguna pada Sistem Informasi Rawat Inap HMC membantu keamanan informasi data pendaftaran pasien ?	P-UAT-02-10	0	1	0	0	0

Hasil dari *User Acceptance Testing* yang dilakukan selanjutnya dianalisis menggunakan skala likert. Teori dan rumus perhitungan skala likert diimplementasikan sesuai dengan perancangan pada pengujian *User Acceptance Testing* ini. Berikut merupakan uraian poin-poin perhitungan hasil *User Acceptance Testing* menggunakan skala likert:

- d. Menghitung total nilai berdasarkan jawaban dari satu responden petugas *Respsionis*.

**Tabel 7.32 Hasil Pengujian *User Acceptance Testing* untuk Petugas *Front office***

Jawaban	Bobot nilai	Jumlah jawaban	Hasil
Sangat Setuju (A)	5	3	15
Setuju (B)	4	5	20
Netral (C)	3	1	3
Tidak Setuju (D)	2	0	0
Sangat Tidak Setuju (E)	1	0	0
Total			38

- e. Nilai Y.

$$Y = \text{Jumlah Skala} \times \text{Jumlah Responden} \times \text{Jumlah Pertanyaan}$$

$$Y = 5 \times 1 \times 10 = 50$$

- f. Nilai Presentase.

$$\text{Rumus Index} = (\text{Total Nilai} \div Y) \times 100\%$$

$$\text{Rumus Index} = (38 \div 50) \times 100\% = 76\% (\text{Setuju})$$

### 7.4.2.3 Hasil Pengujian *User Acceptance Testing* untuk Dokter

Subbab ini merupakan hasil kuisioner *User Acceptance Testing* untuk Dokter yang berisi pertanyaan, kode uji, dan jawaban. Responden dalam *User Acceptance Testing* ini adalah satu Dokter. Hasil kuisioner *User Acceptance Testing* untuk Dokter terdapat pada Tabel 7.33.

**Tabel 7.33 Hasil Pengujian *User Acceptance Testing* untuk Dokter**

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
1	Apakah Sistem Informasi Rawat Inap HMC dapat diakses dengan cepat ?	P-UAT-02-01	0	1	0	0	0
2	Apakah Sistem Informasi Rawat Inap HMC dapat mempercepat pengguna untuk mendapatkan informasi pemeriksaan kesehatan fisik dan kejiwaan pasien ?	P-UAT-02-02	1	0	0	0	0
3	Apakah Sistem Informasi Rawat Inap HMC mudah digunakan ?	P-UAT-02-03	1	0	1	0	0
4	Apakah Sistem Informasi Rawat Inap HMC mempermudah pengelolaan data hasil pemeriksaan fisik dan kejiwaan pasien ?	P-UAT-02-04	0	1	0	0	0
5	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam penyajian informasi permintaan pemeriksaan kesehatan fisik dan kejiwaan pasien ?	P-UAT-02-05	0	1	0	0	0
6	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam mengelola data hasil pemeriksaan fisik dan kejiwaan pasien ?	P-UAT-02-06	0	1	0	0	0
7	Apakah login pengguna pada Sistem Informasi Rawat Inap HMC membantu keamanan	P-UAT-02-07	0	0	1	0	0

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
	informasi data pemeriksaan fisik dan kejiwaan pasien untuk rawat inap?						

Hasil dari *User Acceptance Testing* yang dilakukan selanjutnya dianalisis menggunakan skala likert. Teori dan rumus perhitungan skala likert diimplementasikan sesuai dengan perancangan pada pengujian *User Acceptance Testing* ini. Berikut merupakan uraian poin-poin perhitungan hasil *User Acceptance Testing* menggunakan skala likert:

- g. Menghitung total nilai berdasarkan jawaban dari satu responden Dokter.

**Tabel 7.34 Hasil Pengujian *User Acceptance Testing* untuk Dokter**

Jawaban	Bobot nilai	Jumlah jawaban	Hasil
Sangat Setuju (A)	5	2	10
Setuju (B)	4	4	16
Netral (C)	3	1	3
Tidak Setuju (D)	2	0	0
Sangat Tidak Setuju (E)	1	0	0
Total			29

- h. Nilai Y.

$$Y = \text{Jumlah Skala} \times \text{Jumlah Responden} \times \text{Jumlah Pertanyaan}$$

$$Y = 5 \times 1 \times 7 = 35$$

- i. Nilai Presentase.

$$\text{Rumus Index} = (\text{Total Nilai} \div Y) \times 100\%$$

$$\text{Rumus Index} = (29 \div 35) \times 100\% = 83\% \text{ (Sangat Setuju)}$$

#### 7.4.2.4 Hasil Pengujian *User Acceptance Testing* untuk Petugas Apoteker

Subbab ini merupakan hasil kuisioner *User Acceptance Testing* untuk Petugas Apoteker yang berisi pertanyaan, kode uji, dan jawaban. Responden dalam *User Acceptance Testing* ini adalah dua petugas Apoteker. Hasil kuisioner *User Acceptance Testing* untuk Petugas Apoteker terdapat pada Tabel 7.35.

**Tabel 7.27 Hasil Pengujian *User Acceptance Testing* untuk Petugas Apoteker**

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
1	Apakah Sistem Informasi Rawat Inap HMC dapat diakses dengan cepat ?	P-UAT-03-01	1	0	0	0	0
2	Apakah Sistem Informasi Rawat Inap HMC dapat mempercepat pengguna	P-UAT-03-02	0	1	0	0	0

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
	untuk mendapatkan informasi data obat dan data resep obat dari hasil pemeriksaan pasien ?						
3	Apakah Sistem Informasi Rawat Inap HMC mudah digunakan ?	P-UAT-03-03	0	1	0	0	0
4	Apakah Sistem Informasi Rawat Inap HMC mempermudah pengelolaan data obat dan data resep obat dari hasil pemeriksaan pasien ?	P-UAT-03-04	0	1	0	0	0
5	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam penyajian informasi permintaan resep obat dari hasil pemeriksaan pasien?	P-UAT-03-05	1	0	0	0	0
6	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam mengelola data obat dan data resep obat dari hasil pemeriksaan pasien ?	P-UAT-03-06	1	0	0	0	0
7	Apakah login pengguna pada Sistem Informasi Rawat Inap HMC membantu keamanan informasi data obat dan resep obat dari hasil pemeriksaan pasien ?	P-UAT-03-07	0	1	0	0	0

Hasil dari *User Acceptance Testing* yang dilakukan selanjutnya dianalisis menggunakan skala likert. Teori dan rumus perhitungan skala likert diimplementasikan sesuai dengan perancangan pada pengujian *User Acceptance Testing* ini. Berikut merupakan uraian poin-poin perhitungan hasil *User Acceptance Testing* menggunakan skala likert:

- j. Menghitung total nilai berdasarkan jawaban dari 1 responden petugas Apoteker.



**Tabel 7.36 Hasil Pengujian *User Acceptance Testing* untuk Petugas Apoteker**

Jawaban	Bobot nilai	Jumlah jawaban	Hasil
Sangat Setuju (A)	5	3	15
Setuju (B)	4	4	16
Netral (C)	3	0	0
Tidak Setuju (D)	2	0	0
Sangat Tidak Setuju (E)	1	0	0
Total			31

k. Nilai Y.

$Y = \text{Jumlah Skala} \times \text{Jumlah Responden} \times \text{Jumlah Pertanyaan}$

$$Y = 5 \times 1 \times 7 = 35$$

l. Nilai Presentase.

Rumus Index =  $(\text{Total Nilai} \div Y) \times 100\%$

$$\text{Rumus Index} = (31 \div 35) \times 100\% = 89\% \text{ (Sangat Setuju)}$$

#### 7.4.2.5 Hasil Pengujian *User Acceptance Testing* untuk Petugas Apoteker

Subbab ini merupakan hasil kuisioner *User Acceptance Testing* untuk Petugas Apoteker yang berisi pertanyaan, kode uji, dan jawaban. Responden dalam *User Acceptance Testing* ini adalah dua petugas Apoteker. Hasil kuisioner *User Acceptance Testing* untuk Petugas Apoteker terdapat pada Tabel 7.35.

**Tabel 7.28 Hasil Pengujian *User Acceptance Testing* untuk Perawat**

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
1	Apakah Sistem Informasi Rawat Inap HMC dapat diakses dengan cepat ?	P-UAT-02-01	0	1	0	0	0
2	Apakah Sistem Informasi Rawat Inap HMC dapat mempercepat pengguna untuk mendapatkan informasi pemeriksaan kesehatan fisik dan kejiwaan pasien ?	P-UAT-02-02	1	0	0	0	0
3	Apakah Sistem Informasi Rawat Inap HMC mudah digunakan ?	P-UAT-02-03	1	0	1	0	0
4	Apakah Sistem Informasi Rawat Inap HMC mempermudah pengelolaan data hasil pemeriksaan fisik dan kejiwaan pasien ?	P-UAT-02-04	0	1	0	0	0



No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
5	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam penyajian informasi permintaan pemeriksaan kesehatan fisik dan kejiwaan pasien ?	P-UAT-02-05	0	1	0	0	0
6	Apakah dengan adanya Sistem Informasi Rawat Inap HMC membantu dalam mengelola data hasil pemeriksaan fisik dan kejiwaan pasien ?	P-UAT-02-06	0	0	1	0	0
7	Apakah login pengguna pada Sistem Informasi Rawat Inap HMC membantu keamanan informasi data pemeriksaan fisik dan kejiwaan pasien untuk rawat inap?	P-UAT-02-07	1	0	0	0	0

Hasil dari *User Acceptance Testing* yang dilakukan selanjutnya dianalisis menggunakan skala likert. Teori dan rumus perhitungan skala likert diimplementasikan sesuai dengan perancangan pada pengujian *User Acceptance Testing* ini. Berikut merupakan uraian poin-poin perhitungan hasil *User Acceptance Testing* menggunakan skala likert:

m. Menghitung total nilai berdasarkan jawaban dari 1 responden petugas Perawat.

**Tabel 7.38 Hasil Pengujian *User Acceptance Testing* untuk Perawat**

Jawaban	Bobot nilai	Jumlah jawaban	Hasil
Sangat Setuju (A)	5	3	15
Setuju (B)	4	3	12
Netral (C)	3	1	3
Tidak Setuju (D)	2	0	0
Sangat Tidak Setuju (E)	1	0	0
Total			30

n. Nilai Y.

$$Y = \text{Jumlah Skala} \times \text{Jumlah Responden} \times \text{Jumlah Pertanyaan}$$

$$Y = 5 \times 1 \times 7 = 35$$

o. Nilai Presentase.

Rumus Index = (Total Nilai ÷ Y) × 100%

Rumus Index = (30 ÷ 35) × 100% = 86% (Sangat Setuju)

### 7.4.3 Kesimpulan Pengujian *User Acceptance Testing* (UAT)

Dari hasil analisis menggunakan skala *likert* maka dapat diambil kesimpulan bahwa untuk pengujian *User Acceptance Testing* untuk staf petugas *Admin* memperoleh presentase nilai sebesar 75% yang artinya petugas *Admin* setuju. Kemudian untuk pengujian *User Acceptance Testing* untuk petugas *Front office* memperoleh presentase nilai sebesar 76% yang artinya petugas *Front office* setuju. Selanjutnya untuk pengujian *User Acceptance Testing* untuk Dokter memperoleh presentase nilai sebesar 83% yang artinya Dokter sangat setuju. Pengujian *User Acceptance Testing* untuk petugas Apoteker memperoleh presentase nilai sebesar 89% yang artinya petugas Apoteker sangat setuju. Pengujian *User Acceptance Testing* untuk petugas Perawat memperoleh presentase nilai sebesar 86% yang artinya petugas Perawat Sangat setuju. Berdasarkan presentase pengujian yang dilakukan dengan *User Acceptance Testing* pada beberapa responden menunjukkan sistem informasi rawat inap di rumah sakit HMC Malang dapat diterima baik oleh masing-masing pengguna

### 7.5 Kesimpulan Pengujian Fase Transisi Metode *Rational Unifield Process*

Pengujian fase transisi pada metode *rational unifield process* telah dilakukan. Pengujian dilakukan untuk mengetahui penerimaan pengguna terhadap sistem yang telah dikembangkan. Pengujian menggunakan *user acceptance testing*. Berdasarkan hasil pengujian yang sudah dilakukan sebelumnya menunjukkan bahwa pengguna setuju dengan sistem yang sudah dikembangkan. Syarat yang harus dipenuhi pada fase transisi telah dilakukan maka fase transisi pada RUP terpenuhi.

## 8. PENUTUP

### 8.1 Kesimpulan

Berdasarkan penelitian yang sudah dilakukan, dapat diambil kesimpulan bahwa Sistem Informasi Rawat Inap HMC Malang dapat dikembangkan melalui beberapa kegiatan pengembangan sistem dengan hasil sebagai berikut :

1. Fase inepsi dilakukan analisis proses bisnis dengan melakukan analisis proses bisnis saat ini (*as-is*) yang menghasilkan proses bisnis perbaikan (*to-be*). Diperoleh empat proses bisnis *to-be* yaitu pendaftaran pasien, pemeriksaan pasien, pelayanan obat, dan pelayanan pasien keluar. Empat proses bisnis *to-be* tersebut dipermudah dengan adanya bantuan dari sistem yang dikembangkan. Analisis persyaratan kemudian dilakukan sehingga didapat 7 fitur, 17 kebutuhan persyaratan fungsional dan 1 kebutuhan persyaratan nonfungsional. Dari hasil spesifikasi kebutuhan tersebut akhirnya menghasilkan 8 *use case* yang menggambarkan interaksi antara aktor dan sistem. Kebutuhan analisis persyaratan yang dilakukan telah setuju oleh pemangku kepentingan. *Lifecycle objective milestone* fase inepsi pada metode RUP telah terpenuhi.
2. Fase elaborasi rancangan dilakukan berdasarkan hasil analisis persyaratan yang sudah dilakukan pada fase inepsi. Analisis persyaratan menghasilkan model interaksi objek yang divisualisasikan ke dalam *sequence diagram*, model objek divisualisasikan dalam *class diagram* dan beberapa rancangan yaitu *physical data model*, algoritme, dan antarmuka sistem. Sehingga syarat yang harus terpenuhi pada fase elaborasi yaitu deskripsi aritektur sistem sudah dilakukan, maka *lifecycle architecture milestone* fase elaborasi pada metode RUP telah terpenuhi.
3. Fase konstruksi implementasi sistem dibuat berdasarkan perancangan pada fase elaborasi. Implementasi dikembangkan dengan kerangka kerja Laravel. Pada fase konstruksi pengujian sistem dilakukan dengan pengujian *validation testing* dengan 8 kasus uji yang sudah didefinisikan berasal dari empat *use case scenario* yang berbeda diantaranya tambah data pendaftaran baru, tambah data pemeriksaan pasien, mengubah data hasil rekam medis, dan mengisi data transaksi pembayaran pasien menunjukkan hasil status valid dengan presentase 100%. Pengujian nonfungsional dengan *compatibility testing* didapat hasil 2 *critical issues* disebabkan tidak diaktifkannya fungsi SSL (*secure socket layer*). Berdasarkan hasil pengujian menunjukkan bahwa sistem yang diuji telah sesuai dengan kebutuhan persyaratan fungsional dan non fungsional yang telah diidentifikasi. *Initial operational capability milestone* fase konstruksi pada metode RUP telah terpenuhi.
4. Fase transisi pengujian untuk pengujian *User Acceptance Testing* untuk staf petugas *Admin* memperoleh presentase nilai sebesar 75% yang artinya petugas *Admin* setuju. Kemudian untuk pengujian *User Acceptance Testing*

untuk petugas *Front office* memperoleh presentase nilai sebesar 76% yang artinya petugas *Front office* setuju. Selanjutnya untuk pengujian *User Acceptance Testing* untuk Dokter memperoleh presentase nilai sebesar 83% yang artinya Dokter sangat setuju. Pengujian *User Acceptance Testing* untuk petugas Apoteker memperoleh presentase nilai sebesar 89% yang artinya petugas Apoteker sangat setuju. Pengujian *User Acceptance Testing* untuk petugas Perawat memperoleh presentase nilai sebesar 86% yang artinya petugas Apoteker Sangat setuju. Berdasarkan presentase pengujian yang sudah didapat menunjukkan semua pihak atau aktor yang secara langsung menggunakan sistem informasi rawat inap setuju dengan adanya sistem informasi yang sudah dikembangkan maka *release product milestone* fase transisi pada metode RUP telah terpenuhi.

## 8.2 Saran

Saran yang dapat diberikan sebagai bahan pertimbangan untuk melaksanakan pengembangan lebih lanjut pada Sistem Informasi Rawat inap HMC Malang diantaranya adalah :

1. Bagi peneliti selanjutnya dapat melakukan penelitian tingkat kenyamanan dalam penggunaan sistem informasi rawat inap HMC Malang khususnya antarmuka pengguna.
2. Perlu dilakukan pengembangan sistem informasi layanan lain yang ada di Rumah Sakit Hayunanto *Medical Center* Malang selain layanan Rawat Inap untuk meningkatkan kualitas layanan secara keseluruhan.

## DAFTAR REFERENSI

- A.S, R., & Shalahudin, M. (2015). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- AA, G. M. (2004). *Manajemen Buku Kesehatan*. Jakarta: Penerbit Buku Kedokteran.
- Aggarwal, V. (2002). The Application of the Unified Modelling language in Object-Oriented Analysis of Healthcare Information Systems. *Journal of Medical Systems*, 26(5).
- Anwar, A. (2014). A Review of RUP (Rational Unified Process). *International Journal of Software Engineering(IJSE)*, 5(2).
- Brandenburg, L. (2017). *Bridging The Gap*. Retrieved 03 29, 2018, from <http://www.bridging-the-gap.com/as-is-business-process/>
- Goel, R., & Gupta, D. N. (2014). Survey on Acceptance Testing Technique. *International Journal of Software and Web Sciences (IJSWS)*, 8(1), 20-23.
- IBM. (2007). *The IBM Rational Unified Process for System z* (First Edition ed.). U.S: International Business Machines Corporation.
- Indriati. (2010). *Pengujian Validasi*. Retrieved January 13, 2019, from <http://indryz.lecture.ub.ac.id/files/2010/11/Pengujian-Validasi.docx>
- Kinasih, A. D. (2018). Pengembangan Sistem Evaluasi Pembelajaran PAUD. *j-ptiik*, 2(3), 1027-1035.
- Krol, M., & Reich, D. L. (1999). Object-Oriented Analysis and Design of a Health Care Management Information System. *Journal of Medical Systems*, 23(2).
- Kroll, P. &. (2003). *The Rational Unified Process Made Easy : A Practitioner's Guide to the RUP*. Canada: Addison-Wesley.
- Lan, J., & Xiujuan, L. (2012). System Modeling of Vehicle Management Bases on RUP and UML. *IEEE*.
- Limbong, J. (2010). Pengembangan Sistem Informasi Rawat Inap Pelayanan Penyakit Dalam Guna Mendukung Keputusan Manajemen Pelayanan DI RSUD Dr H Soemarno Sosroadmojo Bulungan Kalimantan Timur. *Universitas Dipenogoro*.
- Menkes, R. (2008). Keputusan Menteri Kesehatan RI Nomor 129/MENKES/SK/II/2008. Jakarta.
- Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN). *Jurnal Ilmiah Teknologi Informasi Terapan*, 1(3), 31-36.
- Naik, K. (2008). *SOFTWARE TESTING AND QUALITY ASSURANCE Theory and Practice*. Canada: John Wiley & Sons, Inc.

- O'brien, J. A. (2008). *Management Information Systems* (Eighth Edition ed.). New York: The McGraw-Hill Companies, Inc.
- Prayitno, W. (2006). *Desain Model Sistem Perangkat Lunak dengan UML. Departemen Teknik Elektro FTI- ITB.*
- Pressman, R. S. (2010). *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman* (7th ed.). New York: McGraw-Hill.
- Ramadhani, M. A. (2015). *Pemodelan Proses Bisnis Sistem Akademik Menggunakan Pendekatan Business Process Modelling Notation (BPMN) (Studi Kasus Institusi Perguruan Tinggi XYZ). Jurnal Informasi, VII(2), 83-92.*
- Sanjay Dahiya, V. P. (2012). *Comprehensive Approach for Cross Compability Testing of Website. International Journal of Computer Applications (IJCA).*
- Sommerville, I. (2011). *Software Engineerring (Rekayasa Perangkat Lunak).* Jakarta: Erlangga.
- Valacich, J. a. (2010). *Information Systems Today: Managing in the Digital World* (4th ed ed.). United States of America : Pearson Education.
- Yoon, i.-C. (2008, July 20-24). *Effective and scalable software compability testing. International symposium on Software testing and Analysis, pp. 63-74.*
- Zamikar , M., & Reshadinezhad, M. R. (2013). *A Comparison Between Two Software Engineering Processes, RUP and Waterfall Models. International Journal of Engineering Research & Technology (IJERT), 2(7).*