

**RANCANG BANGUN SISTEM ENTERPRISE RUMAH SAKIT
MENGUNAKAN *DESIGN PATTERN* MODEL VIEW
CONTROLLER (MVC)**

SKRIPSI

KONSENTRASI REKAYASA PERANGKAT LUNAK

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun oleh :

AFIQIE FADHIHANSAH

NIM 105060802111003

**PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2014

LEMBAR PERSETUJUAN
RANCANG BANGUN SISTEM ENTERPRISE RUMAH SAKIT
MENGUNAKAN *DESIGN PATTERN* MODEL VIEW
CONTROLLER (MVC)

SKRIPSI

KONSENTRASI REKAYASA PERANGKAT LUNAK

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun oleh :

AFIQIE FADHIHANSAH

NIM 105060802111003

Telah diperiksa dan disetujui oleh dosen pembimbing pada tanggal 21 Juli 2014

Dosen Pembimbing I

Dosen Pembimbing II

Denny Sagita Rusdianto, S.Kom, M.Kom

NIK. 851124 06 1 1 0250

Novanto Yudistira, S.Kom, M.Sc

NIK. 831110 16 1 1 0425

LEMBAR PENGESAHAN
RANCANG BANGUN SISTEM ENTERPRISE RUMAH SAKIT
MENGGUNAKAN *DESIGN PATTERN* MODEL VIEW CONTROLLER
(MVC)

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh:

AFIQIE FADHIHANSAH

NIM 105060802111003

Skripsi ini telah diuji dan dinyatakan lulus pada

Tanggal 8 Juli 2014

Penguji 1

Penguji 2

Issa Arwani, S.Kom, M.Sc

NIP. 198309222012121003

Aditya Rachmadi, S.ST, MTI

NIK. 860421 16 1 1 0426

Penguji 3

Budi Darma Setiawan, S.Kom, M.Cs

NIP. 841015 06 1 1 0090

Mengetahui

Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, M.T.

NIP. 19670801 199203 1 001

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Afiqie Fadhihansah

NIM : 105060802111003

Program Studi : Informatika / Ilmu Komputer

Fakultas : Program Teknologi Informasi dan Ilmu Komputer

Universitas : Universitas Brawijaya

Judul skripsi : “RANCANG BANGUN SISTEM ENTERPRISE RUMAH SAKIT MENGGUNAKAN *DESIGN PATTERN* MODEL VIEW CONTROLLER (MVC)”

Dengan ini menyatakan bahwa:

1. Isi dari skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam skripsi ini.
2. Apabila dikemudian hari ternyata skripsi yang saya tulis terbukti hasil jiplakan, maka saya bersedia menanggung segala risiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran, penuh tanggung jawab dan digunakan sebagaimana mestinya.

Malang, 20 Juli 2014

Yang menyatakan,

(Afiqie Fadhihansah)
NIM 105060802111003

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat, taufik dan hidayah-Nya sehingga saya dapat menyelesaikan skripsi ini dengan baik. Adapun skripsi yang saya buat ini yaitu berjudul "RANCANG BANGUN SISTEM ENTERPRISE RUMAH SAKIT MENGGUNAKAN *DESIGN PATTERN* MODEL VIEW CONTROLLER (MVC) (Studi Kasus: Poliklinik Universitas Brawijaya)".

Sejak awal sampai dengan akhir penulisan ini, tidak sedikit bantuan yang saya terima dan karenanya dalam kesempatan ini kami ingin menyampaikan terima kasih kepada :

1. Denny Sagita Rusdianto, S.Kom, M.Kom selaku Dosen Pembimbing I dan Novanto Yudistira, S.Kom, M.Sc selaku Dosen Pembimbing II yang telah menyediakan waktu untuk *focus group discussions* dan berbagai saran dan masukannya.
2. Ridwan Arief dan Siti Rochmah selaku kedua orang tua saya yang telah memberikan *support* dan doa sehingga saya dapat menyelesaikan skripsi ini.
3. drg. Selvi Lesmwati dan Rini Dwi Andarwati, S.KM selaku kakak kandung saya yang telah memberikan dorongan dan nasihat kepada saya.
4. Eliya Zulkarnain, Amd.Keb selaku istri tercinta yang telah memberikan *support*, masukan dan bantuan terhadap pembuatan Sistem Enterprise Rumah Sakit ini, terutama bantuan mengenai database diagnosa penyakit pasien.
5. Tim Netmedis yang terdiri dari Billy Novanta Yudistira, Durrotul Ikrimah, Adwin Setyawan, Romdhan M. Ubaidilah, I Dewa Gedhe Anthasena dan Dwi Fetiria Ningrum yang telah membantu dalam pengumpulan data, uji coba dan implementasi software di Poliklinik Universitas Brawijaya.
6. Dr.dr. Tinny Endang H, Sp.PK(K) selaku Kepala Poliklinik Universitas Brawijaya yang telah meluangkan waktu dalam proses wawancara, pengambilan data serta membantu dalam proses uji coba sistem.
7. Beberapa staf di Poliklinik Universitas Brawijaya yang telah membantu memberi masukan terhadap Sistem Enterprise Rumah Sakit.

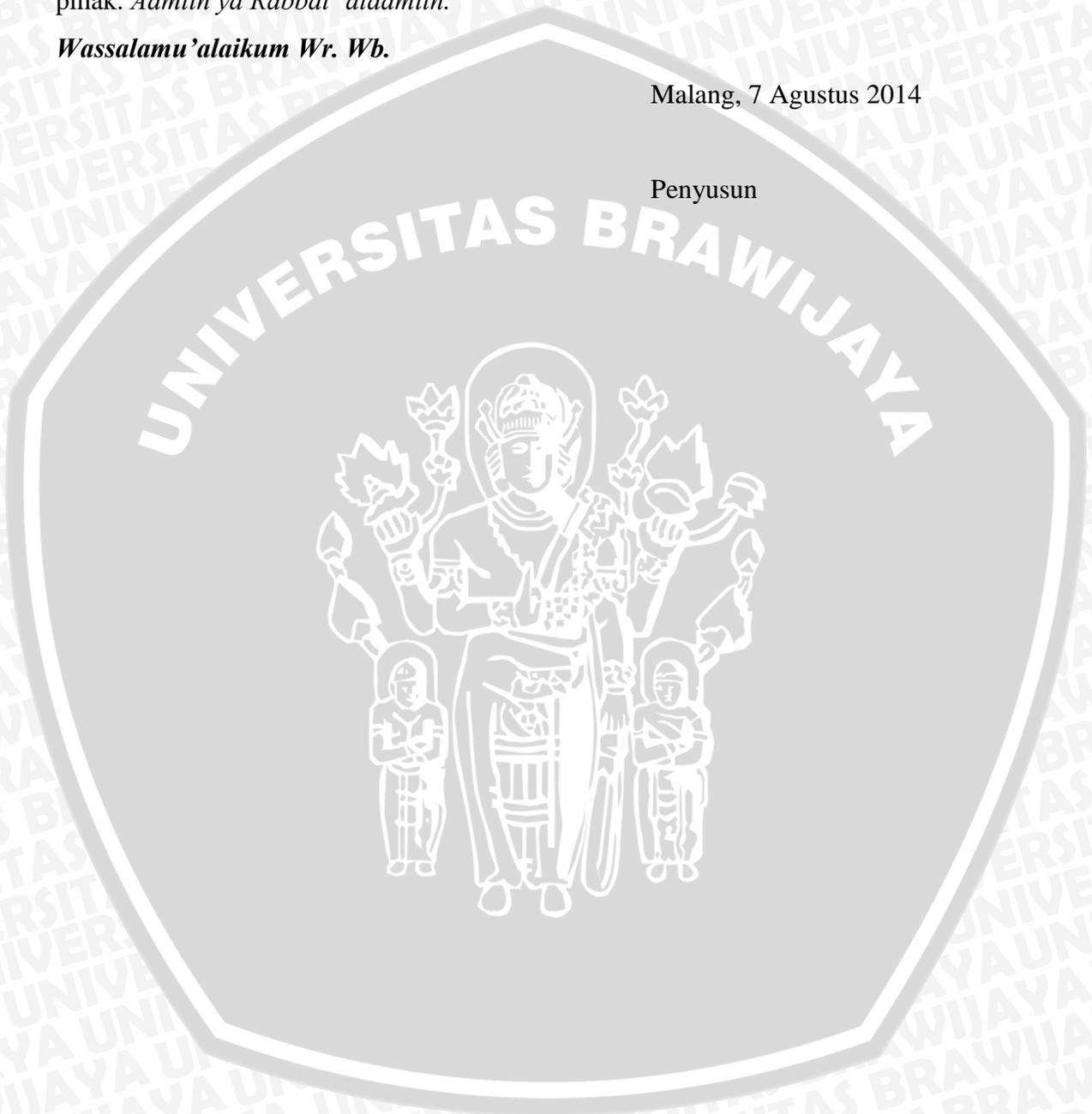
8. Kesatuan Aksi Mahasiswa Muslim Indonesia (KAMMI) Komisariat Brawijaya yang telah memberikan dukungan moril kepada saya.

Demi kesempurnaan penelitian ini, saya mengharapkan kritik dan saran yang bersifat membangun. Harapan saya, skripsi ini dapat bermanfaat bagi semua pihak. *Aamiin ya Rabbal 'alaamiin.*

Wassalamu'alaikum Wr. Wb.

Malang, 7 Agustus 2014

Penyusun



DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	xi
ABSTRAK.....	xiii
ABSTRACT.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB II KAJIAN PUSTAKA.....	5
2.1 <i>Health Information Technology</i>	5
2.2 Rekyasa Perangkat Lunak.....	6
2.3 <i>Enterprise Resource Planning</i>	8
2.4 <i>Unified Modelling Language</i>	10
2.4.1 Usecase Diagram.....	11
2.4.2 <i>Class Diagram</i>	15
2.4.3 <i>Sequence Diagram</i>	17



2.5	Pengujian Perangkat Lunak.....	20
2.5.1	Teknik Pengujian	20
2.5.2	Strategi Pengujian	21
2.6	Software Development Life Cycle	23
2.7	Code Igniter	25
2.7.1	Selayang Pandang Code Igniter	25
2.7.2	Mendapatkan Code Igniter.....	25
2.7.3	Instalasi Code Igniter	25
2.7.4	Konsep Model-View-Controller (MVC)	26
2.8	Basis Data.....	26
2.9	MySQL.....	27
BAB III METODOLOGI PENELITIAN.....		29
3.1	Tahap Studi Literatur.....	29
3.2	Tahap Analisis Kebutuhan	30
3.2.1	Kebutuhan Fungsional	31
3.2.2	Kebutuhan Non Fungsional.....	31
3.3	Tahap Perancangan Sistem.....	31
3.4	Tahap Implementasi	32
3.5	Tahap Pengujian	32
BAB IV PERANCANGAN DAN IMPLEMENTASI.....		33
4.1	Arsitektur Sistem Rumah Sakit berbasis ERP.....	33
4.2	Daftar Kebutuhan Perangkat Lunak	35
4.3	Analisis Kebutuhan	38
4.3.1	Pemodelan Bisnis dengan <i>Usecase</i>	39
4.3.2	Pemodelan Sistem dengan <i>Use Case</i>	53
4.3.3	Pemodelan Sistem dengan <i>Activity Diagram</i>	70

4.3.4	Pemodelan Sistem dengan <i>Class Diagram</i>	76
4.3.5	Pemodelan Sistem dengan <i>Sequence Diagram</i>	76
4.4	Rancang Bangun Aplikasi	78
4.4.1	Human Resource	78
4.4.2	Pharmacy.....	80
4.4.3	Medical Records.....	81
4.4.4	Chasier.....	83
4.4.5	Medicare.....	84
4.4.6	Financial.....	85
4.4.7	Front Office.....	87
BAB V PENGUJIAN DAN ANALISIS		89
5.1	Pengujian	89
5.1.1	Pengujian <i>Functionality</i>	89
5.1.2	Pengujian <i>User Acceptance Test (UAT)</i>	124
5.1.3	Pengujian <i>Integration</i>	127
5.2	Analisis.....	129
5.2.1	Analisis Pengujian <i>Functionality</i>	129
5.2.2	Analisis Pengujian <i>User Acceptance Test (UAT)</i>	129
5.2.3	Analisis Pengujian <i>Integration</i>	129
BAB VI PENUTUP		130
6.1	Kesimpulan.....	130
6.2	Saran.....	130
DAFTAR PUSTAKA		
LAMPIRAN		



DAFTAR TABEL

Tabel 2.2 Keterangan simbol-simbol <i>Usecase Diagram</i>	12
Tabel 2.3 Keterangan simbol-simbol <i>Class Diagram</i>	16
Tabel 2.4 Keterangan simbol - simbol <i>Sequence Diagram</i>	17
Tabel 4.1 Tabel Identifikasi Modul.....	34
Tabel 4.2 Daftar Kebutuhan Fungsional	35
Tabel 4.3 Kebutuhan lingkungan	38
Tabel 4.4 Tabel Skenario Bisnis Registrasi Pasien.....	40
Tabel 4.5 Tabel Skenario Bisnis Ketersediaan Medis	42
Tabel 4.6 Tabel Skenario Bisnis Ketersediaan Medis	44
Tabel 4.7 Tabel Skenario Bisnis Pemeriksaan	46
Tabel 4.8 Tabel Skenario Bisnis Pemeriksaan.....	47
Tabel 4.9 Tabel Skenario Bisnis Pembayaran.....	50
Tabel 4.10 Tabel Skenario Bisnis Layani Obat	52
Tabel 4.11 <i>Use case</i> Menampilkan Daftar Pegawai	55
Tabel 4.12 <i>Use case</i> Tambah data pegawai	55
Tabel 4.13 <i>Use case</i> Ubah Data Pegawai	56
Tabel 4.14 <i>Use case</i> Hapus Data Pegawai.....	56
Tabel 4.15 <i>Use case</i> Menampilkan Daftar Kehadiran	57
Tabel 4.16 <i>Use case</i> Menampilkan Riwayat Penanganan Dokter	57
Tabel 4.17 <i>Use case</i> Register Pasien.....	57
Tabel 4.18 <i>Use case</i> Ubah Daftar Antrian Registrasi Pasien.....	58
Tabel 4.19 <i>Use case</i> Menampilkan dan Menghapus Daftar Registrasi Pasien	58
Tabel 4.20 <i>Use case</i> Pencarian Pasien	59
Tabel 4.21 <i>Use case</i> Menampilkan Statistik Pengunjung Per Bulan.....	59
Tabel 4.22 <i>Use case</i> Menampilkan Daftar Data Registrasi Pasien.....	60
Tabel 4.23 <i>Use case</i> Tambah Rekam Medis Pasien	60
Tabel 4.24 <i>Use case</i> Ubah Rekam Medis Pasien.....	61
Tabel 4.25 <i>Use case</i> Hapus Rekam Medis Pasien	61
Tabel 4.26 <i>Use case</i> Menampilkan Data Pembayaran Pasien	62
Tabel 4.27 <i>Use case</i> Tambah Form Pembayaran Pasien	62

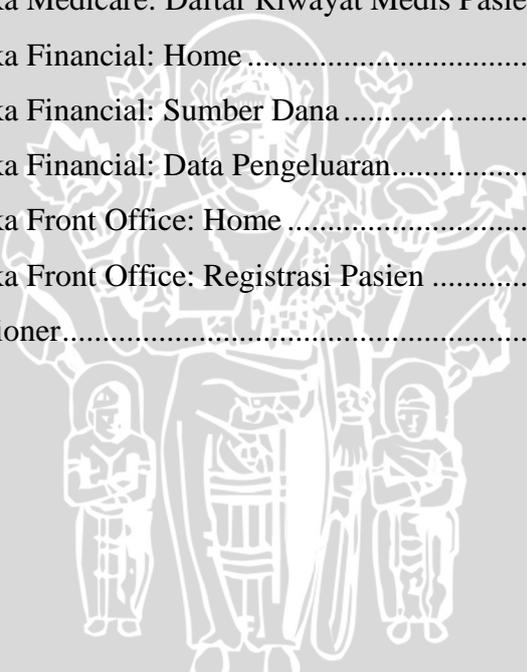


Tabel 4.28 <i>Use case</i> Tambah Form <i>Invoice</i> Pasien	63
Tabel 4.29 <i>Use case</i> Cetak <i>Invoice</i> Pasien	63
Tabel 4.30 <i>Use case</i> Menampilkan Data <i>Warehouse</i> Obat	64
Tabel 4.31 <i>Use case</i> Edit Data Obat	64
Tabel 4.32 <i>Use case</i> Tambah Rekam Medis Pasien	64
Tabel 4.33 <i>Use case</i> Tambah Rekam Medis Pasien	65
Tabel 4.34 <i>Use case</i> Menampilkan Data Sumber Dana	65
Tabel 4.35 <i>Use case</i> Menampilkan Pemasukan Pasien	66
Tabel 4.36 <i>Use case</i> Menampilkan Pengeluaran Keseluruhan	66
Tabel 4.37 <i>Use case</i> Menampilkan Pemasukan dan Pengeluaran	67
Tabel 4.38 <i>Use case</i> Menampilkan Daftar Riwayat Medis Pasien	67
Tabel 4.39 <i>Use case</i> Menampilkan Biodata Pasien	68
Tabel 4.40 <i>Use case</i> Menampilkan Seluruh Rekam Medis Pasien Tertentu	68
Tabel 4.41 <i>Use case</i> Menampilkan Rekam Medis Pasien Tertentu	69
Tabel 4.42 Tabel Skenario Sistem Registrasi Pasien	70
Tabel 4.43 Tabel Skenario Sistem Rawat Pasien	72
Tabel 4.44 Tabel Skenario Sistem Pengaturan Pengguna	74
Tabel 5.1 Human Resource	90
Tabel 5.2 Front Office	96
Tabel 5.3 Medical Records	101
Tabel 5.4 Chasier	105
Tabel 5.5 Pharmacy	110
Tabel 5.6 Financial	115
Tabel 5.7 Medicare	120
Tabel 5.8 Penjabaran Hasil Pengujian User Acceptance Test (UAT)	124
Tabel 5.9 Hasil pengujian <i>Integration</i>	127

DAFTAR GAMBAR

Gambar 2.1 Contoh Alur Informasi dan Layanan dalam Komunitas Di Lingkup Medis yang Diterapkan di Jepang.....	5
Gambar 2.2 <i>Incremental Model</i>	8
Gambar 2.3 Modul ERP.....	10
Gambar 2.4 Diagram UML.....	11
Gambar 2. 5 Contoh penggunaan sequence diagram pada MVC	20
Gambar 2.6 SDLC <i>Cycle</i>	23
Gambar 3.1 Alur Pelaksanaan Kegiatan	29
Gambar 4.1 Arsitektur Sistem.....	33
Gambar 4.2 Komunikasi dengan diagram aplikasi	34
Gambar 4.3 <i>Usecase Model Bisnis (AS-IS)</i>	39
Gambar 4.4 <i>Activity Registrasi Pasien</i>	40
Gambar 4.5 <i>Activity Ketersediaan Medis</i>	42
Gambar 4.6 <i>Activity Data Rekam Medis Pasien</i>	44
Gambar 4.7 <i>Activity Pemeriksaan</i>	45
Gambar 4.8 <i>Activity Uji Laboratorium</i>	47
Gambar 4.9 <i>Activity Pembayaran</i>	49
Gambar 4.10 <i>Activity Layani Obat</i>	51
Gambar 4.11 <i>Usecase Model Sistem (to-be)</i>	54
Gambar 4.12 Model Sistem Registrasi Pasien	70
Gambar 4.13 Model Sistem Rawat Pasien.....	71
Gambar 4.14 Model Sistem Pengaturan Pengguna.....	74
Gambar 4.15 Class Diagram	76
Gambar 4.16 <i>Sequence Registrasi Pasien</i>	76
Gambar 4.17 <i>Sequence Rawat Pasien</i>	77
Gambar 4.18 <i>Sequence Pembayaran Pasien</i>	77
Gambar 4.19 <i>Sequence Pembayaran Pasien</i>	78
Gambar 4.20 <i>Sequence Ubah Data Pengguna</i>	78
Gambar 4.21 Antarmuka Human Resource: Home	79
Gambar 4.22 Antarmuka Human Resource: Data Kepegawaian.....	79

Gambar 4.23 Antarmuka Human Resource: Edit Data Kepegawaian	80
Gambar 4.24 Antarmuka Pharmacy: Home	80
Gambar 4.25 Antarmuka Pharmacy: Data Obat	81
Gambar 4.26 Antarmuka Pharmacy: Data Transaksi Obat	81
Gambar 4.27 Antarmuka Medical Records: Home	82
Gambar 4.28 Antarmuka Medical Records: Register Pasien	82
Gambar 4.29 Antarmuka Medical Records: Biodata Pasien	83
Gambar 4.30 Antarmuka Chasier: Home	83
Gambar 4.31 Antarmuka Chasier: Register Pasien	84
Gambar 4.32 Antarmuka Chasier: Daftar Pembayaran Pasien	84
Gambar 4.33 Antarmuka Medicare: Home	85
Gambar 4.34 Antarmuka Medicare: Daftar Riwayat Medis Pasien	85
Gambar 4.35 Antarmuka Financial: Home	86
Gambar 4.36 Antarmuka Financial: Sumber Dana	86
Gambar 4.37 Antarmuka Financial: Data Pengeluaran	87
Gambar 4.38 Antarmuka Front Office: Home	87
Gambar 4.39 Antarmuka Front Office: Registrasi Pasien	88
Gambar 5.1 Hasil Kuisisioner	124



ABSTRAK

Fadhiansah, Afiqie. 2014. "Rancang Bangun Sistem *Enterprise* Rumah Sakit Menggunakan *Design Pattern* MVC". Pembimbing: Denny Sagita Rusdianto, S.Kom, M.Kom, dan Novanto Yudistira, S.Kom, M.Sc.

Peneliti merancang, mendesain serta mengimplementasikan Sistem *Enterprise* Rumah Sakit dengan Menggunakan *Design pattern* Model View Controller (MVC). Konsep MVC adalah konsep pemisahan antara logic dengan tampilan dan database. Manfaat konsep ini adalah, membuat coding logic lebih simple, karena sudah di pisah dengan code untuk tampilan dan membuat programmer dapat bekerja secara terpisah dengan designer. Programmer mengerjakan logic, sedangkan designer berkuat dengan desain dan tampilan [SOF-07]. Analisis kebutuhan perangkat lunak terdiri dari: pemodelan bisnis (as-is) dan pemodelan sistem (to-be). Tahap implementasi seluruh analisis dan perancangan sistem akan diimplementasikan, yaitu melalui proses pengkodean (coding) menggunakan bahasa pemrograman PHP dan dibangun dengan *design pattern* Model View Controller (MVC). Tahap pengujian akan dilakukan pengujian perangkat lunak. Hal ini dimaksudkan untuk mengetahui kelemahan yang ada di dalam aplikasi yang telah selesai dibuat. Pengujian aplikasi ini dilakukan dengan tiga cara, yakni: Pengujian Functionality; Pengujian User Acceptance Test (UAT); Pengujian Integration. Berdasarkan hasil pengujian Functionality dan pengujian Integration telah menunjukkan bahwa sistem telah layak dan dapat digunakan untuk kepentingan Poliklinik Universitas Brawijaya pada tahap awal. Hasil pengujian User Acceptance Test (UAT) didapatkan bahwa pengguna telah dapat menerima sistem Sistem *Enterprise* Rumah Sakit dengan menggunakan *design pattern* MVC ini. Hasil menunjukkan bahwa sistem 100% valid.

Kata kunci : sistem enterprise, rumah sakit, *design pattern*, MVC

ABSTRACT

Fadhiansah, Afiqie. 2014. “Design of Hospitals Enterprise System Using Design Pattern MVC”. Supervisor: Denny Sagita Rusdianto, S.Kom, M.Kom, dan Novanto Yudistira, S.Kom, M.Sc.

Researchers designing and implementing the Hospital Enterprise System Using Design pattern Model View Controller (MVC). MVC concept is the concept of separation between the display and database logic. The benefits of this concept is, make your code more simple logic, since it has been separated from the code to display and make programmers can work separately with designers. Programmers working on logic, while the designer struggling with the design and look of [SOF-07]. Analysis of software requirements consist of: business modeling (as-is) and modeling system (to-be). Throughout the implementation phase of the analysis and design of the system will be implemented, namely through the process of coding (coding) using PHP programming language and is built with the Model View Controller design pattern (MVC). The testing phase of software testing will be done. It is intended to identify the weaknesses that exist in the application that has been created. Application testing is done in three ways, namely: Testing Functionality; Testing User Acceptance Test (UAT); Integration Testing. Based on the test results Functionality and Integration testing has shown that the system was feasible and could be used for the benefit of the UB clinic in the early stages. Results of testing User Acceptance Test (UAT) found that users have to accept the system of the Hospital Enterprise System using the MVC design pattern. The results show that the system is 100% valid.

Key words : *enterprise systems, hospitals, design pattern, MVC*

BAB I PENDAHULUAN

1.1 Latar Belakang

TIK (Teknologi Informasi dan Komunikasi) tidak dapat dipungkiri lagi saat ini telah menjadi salah satu keunggulan kompetitif dalam bersaing bagi suatu negara. Sebagai produk yang dihasilkan dari kreativitas dan kemampuan ilmu pengetahuan, TIK menjadi sandaran masa depan yang tidak akan habis untuk digali dan dikembangkan.

Salah satu contoh sebagian kecil bidang TIK adalah bagaimana pasar aplikasi berkembang dengan pesat di seluruh dunia. Misalnya dalam hal kesehatan, kemajuan teknologi menuntut setiap rumah sakit berlomba-lomba memaksimalkan pelayanan kepada masyarakat, khususnya untuk pasien, salah satunya dengan menerapkan sistem informasi rumah sakit. Sistem informasi rumah sakit saat ini menjadi kebutuhan di tiap rumah sakit, karena bertujuan untuk penghematan kertas, juga alur informasi yang didapat juga lebih cepat dan efisien. Namun saat ini sistem informasi rumah sakit masih belum banyak diterapkan di tiap rumah sakit, seperti di Kota Malang, Jawa Timur. Rumah Sakit Saiful Anwar yang menjadi rumah sakit terbesar di Kota Malang pun juga belum menerapkan sistem informasi terintegrasi.

Peneliti merancang, mendesain serta mengimplementasikan Sistem Enterprise Rumah Sakit menggunakan *design pattern* Model View Controller (MVC). Konsep MVC adalah konsep pemisahan antara *logic* dengan tampilan dan database. Manfaat konsep ini adalah, membuat *coding logic* lebih simple, karena sudah di pisah dengan *code* untuk tampilan dan membuat *programmer* dapat bekerja secara terpisah dengan *designer*. Programmer mengerjakan *logic*, sedangkan *designer* berkutat dengan desain dan tampilan [SOF-07].

ERP merupakan sebuah kerangka kerja transaksi berkala perusahaan yang meghubungkan antarproses. ERP bekerja sebagai kekuatan lintas fungsional perusahaan yang mengintegrasikan dan mengotomatisasi berbagai proses bisnis internal dan sistem informasi termasuk manufaktur, logistik, distribusi,

akuntansi, keuangan dan sumber daya manusia sebuah perusahaan [ENT-02:35-37].

ERP menjadi sistem yang tak terpisahkan dan sifatnya penting dalam proses bisnis secara keseluruhan dari beberapa organisasi, dan merupakan salah satu investasi terbesar teknologi informasi untuk beberapa perusahaan. ERP merupakan salah satu sektor yang paling cepat berkembang dari industri sistem informasi [LEO-10].

Rumah sakit terintegrasi membutuhkan perencanaan dan sistem kontrol secara terpusat untuk merencanakan proses pada pasien dan kapasitas yang diperlukan. ERP pada rumah sakit dapat sangat berguna untuk perencanaan dan mengendalikan proses deterministik [MER-04]

Pembeda Sistem Enterprise Rumah Sakit dengan sistem yang lain adalah terlihat pada alur validasi, bisnis proses serta pembagian tugas berbasis modul. Studi kasus peneliti dilakukan di Poliklinik Universitas Brawijaya Malang, hal ini dikarenakan belum adanya sistem informasi di Poliklinik Universitas Brawijaya Malang. Sehingga diperlukan sebuah sistem untuk membantu bisnis proses yang ada di Poliklinik Universitas Brawijaya Malang.

Terdapat enam modul yang akan peneliti buat, yaitu modul *Front Office*, *Medicare*, *Human Resource*, *Financial*, *Chasier*, *Pharmacy* dan *Medical Records*. Tiap modul memiliki fungsi dan tujuan yang berbeda berdasarkan aktor yang menjalankannya.

1.2 Perumusan Masalah

1. Bagaimana menganalisis kebutuhan dari pengguna dan merancang aplikasi Sistem Enterprise Rumah Sakit menggunakan *design pattern* Model View Controller (MVC)?
2. Bagaimana mengimplementasikan pembuatan aplikasi Sistem Enterprise Rumah Sakit menggunakan *design pattern* Model View Controller (MVC)?
3. Bagaimana menguji aplikasi Sistem Enterprise Rumah Sakit menggunakan *design pattern* Model View Controller (MVC)?

1.3 Tujuan

Menganalisis dan merancang kebutuhan dari pengguna, mengimplementasikan pembuatan dan menguji aplikasi Sistem Enterprise Rumah Sakit menggunakan *design pattern* Model View Controller (MVC)

1.4 Batasan Masalah

Lingkungan Sistem Enterprise Rumah Sakit menggunakan *Design pattern* Model View Controller (MVC) dibatasi pada Poliklinik Universitas Brawijaya Malang

1.5 Manfaat Penelitian

Pembuatan Sistem Enterprise Rumah Sakit sebagai aplikasi yang mudah digunakan dan dibutuhkan di pelayanan kesehatan seperti rumah sakit

1.6 Sistematika Penulisan

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

BAB I Pendahuluan

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

BAB II Kajian Pustaka

Membahas teori-teori yang mendukung dalam pengembangan dan perancangan aplikasi Sistem Enterprise Rumah Sakit .

BAB III Metodologi Penelitian

Membahas tentang metode yang digunakan dalam penulisan yang terdiri dari studi literatur, perancangan perangkat lunak, implementasi perangkat lunak, pengujian dan analisis.

BAB IV Perancangan dan Implementasi

Membahas tentang analisa kebutuhan dari aplikasi Sistem Enterprise Rumah Sakit dan kemudian merancang serta mengimplementasikan hal-hal yang berhubungan dengan analisa tersebut.

BAB V Pengujian dan Analisis

Memuat tentang hasil pengujian dan analisis terhadap sistem yang telah direalisasikan.

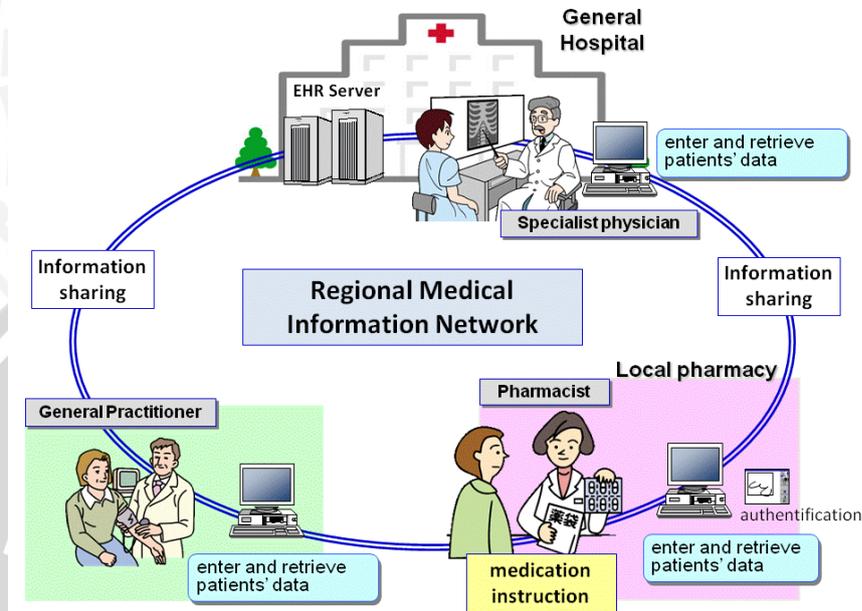
BAB VI Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran-saran untuk pengembangan lebih lanjut.



BAB II KAJIAN PUSTAKA

2.1 Health Information Technology



Gambar 2.1 Contoh Alur Informasi dan Layanan dalam Komunitas Di Lingkup Medis yang Diterapkan di Jepang.

Sumber: [CHO-11:162]

Sebuah penelitian dari Chon Abraham, 2011 dalam *International Journal of Medical Informatics* 80 (2011) halaman 157–170 mengenai upaya Jepang untuk menggunakan *Health Information Technology* (HIT) dalam melakukan reformasi kesehatan. Hasil dari studi kasus masyarakat Yamashina Kyoto bidang medis sebagai bukti konsep untuk kebijakan masa lalu dan sekarang di Jepang yang mendalam untuk proliferasi proyek sukses di Jepang dan adopsi HIT pada umumnya.

Peluncuran sistem yang memungkinkan untuk diterapkan ke setiap rumah sakit menjadi pilihan warga Jepang. Masyarakat membutuhkan penyatuan sumber daya dan menggunakan HIT untuk membantu mengurangi resiko pada rumah sakit utama di daerah pedesaan dan meningkatkan kemampuan perawatan dari masyarakat di seluruh pedesaan.

Selain itu, biaya administrasi perawatan meningkat akibat kenaikan asuransi terkait dengan layanan untuk populasi lansia yang tumbuh pesat. Perawatan umum

sistem di Jepang, di mana pasien biasanya mengasumsikan 30% dari biaya-biaya sedangkan 70% sisanya dibayar dan dibebankan oleh negara dan pemerintah daerah atau asuransi kesehatan masyarakat yang didanai oleh perusahaan yang menaunginya. Semua pilihan asuransi mengoperasikan fasilitas perawatan terpisah dalam masyarakat dengan mayoritas dioperasikan oleh swasta, namun, pasien dapat mengunjungi salah satu fasilitas terlepas dari cakupan asuransi mereka. Klaim processing ditangani untuk setiap fraksi oleh organisasi seperti Asuransi Klaim Ulasan & Layanan Penggantian. Elektronik klaim (e-klaim atau on-line klaim), sebagai jenis HIT, siap untuk membantu mengurangi biaya yang terkait dengan pengumpulan data pasien (*hard copy* atau *soft copy*) dari entitas berbagai kesehatan melayani pasien di seluruh daerah, kurang dari 20% dari seluruh fasilitas kesehatan. Pooling klaim administrasi pengolahan di masyarakat untuk pasien dilayani di berbagai pemerintah dan swasta. Juga, kebijakan penasihat di Jepang percaya bahwa penyatuan data ini lebih baik harus mengaktifkan analisis praktik dengan bukti untuk kualitas dan efektivitas biaya medis, yang pada gilirannya dapat digunakan untuk menentukan penilaian kesehatan masyarakat dan protokol senilai mereplikasi di komunitas medis.

Seperti halnya dalam sistem kesehatan, pasien ditangani oleh para tenaga profesional yang tersedia di daerah, nasional, dan swasta yang dioperasikan layanan dalam perawatan. HIT memungkinkan cara untuk mengkonsolidasikan elektronik medis dan administrasi terhadap pasien dengan harapan kualitas efisiensi dan pelayanan meningkatkan dalam peningkatan informasi. [CHO-11:158-159]

2.2 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah digunakan [SOM-03:7]. Pada definisi ini, ada dua istilah kunci :

1. Disiplin Rekayasa

Perekayasa membuat suatu alat bekerja. Perekayasa menerapkan teori, metode, dan alat bantu yang sesuai. Perekayasa menggunakannya dengan selektif dan selalu mencoba mencari solusi terhadap permasalahan, walaupun tidak ada teori atau metode yang mendukung. Perekayasa juga

menyadari bahwa mereka harus bekerja dalam batasan organisasi dan keuangan, sehingga mereka berusaha mencari solusi dalam batasan-batasan ini.

2. Semua Aspek Produksi Perangkat Lunak

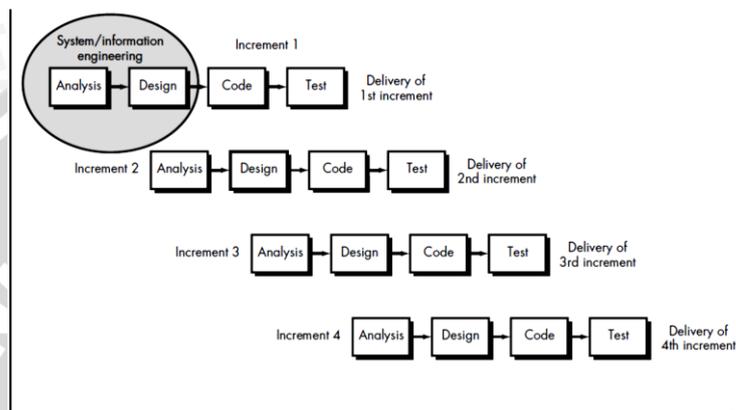
Rekayasa perangkat lunak tidak hanya berhubungan dengan proses teknis dari pengembangan perangkat lunak tetapi juga dengan kegiatan seperti manajemen proyek perangkat lunak dan pengembangan alat bantu, metode dan teori untuk mendukung produksi perangkat lunak.

Perekayasa perangkat lunak memakai pendekatan yang sistematis dan terorganisir terhadap pekerjaan mereka karena cara ini seringkali paling efektif untuk menghasilkan perangkat lunak berkualitas tinggi. Rekayasa ini mencakup masalah pemilihan metode yang paling sesuai untuk satu set keadaan dan pendekatan yang lebih kreatif, informal terhadap pengembangan yang mungkin efektif pada beberapa keadaan [SOM-03:7].

Seorang *software engineer* atau sekumpulan *software engineer* harus menggabungkan strategi pengembangan perangkat lunak yang meliputi proses, metode, dan alat bantu yang digunakan dalam proses pengembangan. Strategi ini disebut sebagai model proses (*process model*) atau paradigma rekayasa perangkat lunak (*software engineering paradigm*). Model proses untuk rekayasa perangkat lunak dipilih sesuai dengan sifat dari proyek dan aplikasi yang akan dibuat. Terdapat beberapa model proses untuk rekayasa perangkat lunak antara lain *linear sequential model (waterfall)*, *prototyping*, *Rapid Application Development (RAD)*, *incremental model* dan *spiral model* [PRE-10:20-42]. Model proses yang digunakan dalam skripsi ini adalah *incremental model*.

Incremental model merupakan gabungan dari elemen *linear sequential model* dan filosofi dari *prototyping* sehingga memudahkan seorang pengembang (*developer*) dalam mengidentifikasi kebutuhan pengguna [PRE-10:35]. *Incremental model* merupakan metodologi pengembangan perangkat lunak yang efisien dan efektif untuk *project* dengan skala kecil hingga skala menengah. *Incremental model* menawarkan strategi pengembangan perangkat lunak yang memungkinkan pengguna (*user*) dapat menggunakan versi awal (*increment pertama*) sebagai *prototype* untuk memperoleh informasi tentang persyaratan

kebutuhan (*requirement*) mereka untuk pengembangan sistem selanjutnya [SOM-11:47]. Kelebihan dari model proses ini antara lain adalah proses *development* yang lebih cepat, lebih mudah dalam mengetahui kebutuhan pengguna, dan *resource* yang dibutuhkan untuk melakukan perubahan terhadap perangkat lunak yang dikembangkan lebih sedikit [SOM-11:33]. Proses pengembangan menggunakan model proses *incremental* ini dapat dilihat pada Gambar 2.1.



Gambar 2.2 Incremental Model

Sumber: [PRE-10:35]

Versi awal dari perangkat lunak (*increment 1*) dievaluasi oleh pengguna sehingga pihak pengembang mengetahui dengan jelas apa yang dibutuhkan oleh pengguna. Pihak pengembang kemudian mengembangkan perangkat lunak untuk *increment* selanjutnya sesuai dengan saran dan permintaan pengguna. Proses ini berulang sampai perangkat lunak selesai atau mencapai tahap yang diinginkan oleh pengguna [PRE-10:35].

Pada skripsi ini digunakan metode analisis kebutuhan, perancangan, implementasi, dan pengujian berorientasi objek menggunakan bahasa pemodelan UML (*Unified Modelling Language*).

2.3 Enterprise Resource Planning

Menjadi organisasi yang kompetitif harus menghasilkan produk secara cepat, berkualitas dan dengan harga yang bersaing. Kumpulan dari bisnis proses yaitu *enterprise resource planning* atau ERP telah terbukti menjadi alat yang efektif untuk mencapai tujuan tersebut. ERP adalah kumpulan alat dan proses yang mengintegrasikan departemen dan fungsi dalam perusahaan menjadi satu sistem

komputer. ERP berjalan dalam database tunggal, memungkinkan bermacam-macam departemen untuk berbagi informasi dan berkomunikasi satu sama lain.

Sistem *enterprise resource planning* (ERP) adalah sistem manajemen bisnis yang meliputi kumpulan software yang terintegrasi, dapat digunakan, dapat diimplementasikan, dikelola dan diintegrasikan ke semua fungsi bisnis dalam organisasi [SSS-04].

ERP sangat dibutuhkan oleh perusahaan karena dapat memperbaiki layanan perusahaan terhadap permintaan konsumen hingga memprosesnya ke dalam *invoice* dan *revenue* yang disebut sebagai proses pemenuhan pemesanan. Itulah salah satu fungsi ERP yang bermanfaat dalam segala sisi di dalam perusahaan. Sebuah alat yang sederhana namun memberikan dampak luar biasa baik terhadap bisnis dan dunia informasi. Berikut merupakan hal yang dapat dilakukan oleh sistem ERP:

- a. Mempengaruhi hampir semua organisasi
- b. Memaksa kompetisi untuk merubah strategi dan proses
- c. Mempengaruhi rekan bisnis untuk menjadi lebih kompetitif
- d. Meningkatkan keuntungan organisasi konsultan
- e. Menjadi alat paling penting dalam bisnis proses *engineering*
- f. Memaksa mengimplementasikan bisnis proses dalam suatu organisasi
- g. Dapat memanfaatkan secara keseluruhan potensi dari *client – server computing* untuk menghasilkan produk *enterprise*.
- h. Merubah kebiasaan dari fungsi sistem informasi dan IT professional
- i. Merubah kebiasaan pekerjaan dalam seluruh area fungsional bisnis
- j. Implementasi ERP sangat mahal

ERP dapat diterapkan dalam segala bidang pekerjaan dan usaha, karena dapat disesuaikan modul sesuai kebutuhan. Seperti dalam bidang bisnis, usaha, maupun pendidikan.

Modul utama dari sistem ERP, dalam lingkungan bisnis, adalah sumber daya manusia, administrasi, keuangan, perencanaan produksi, penjualan dan pemasaran, inventarisasi dan strategi manajemen. Modul ini saling terkait dan ketika terintegrasi, memberikan informasi yang diperlukan digunakan secara luas untuk proses pengambilan keputusan.



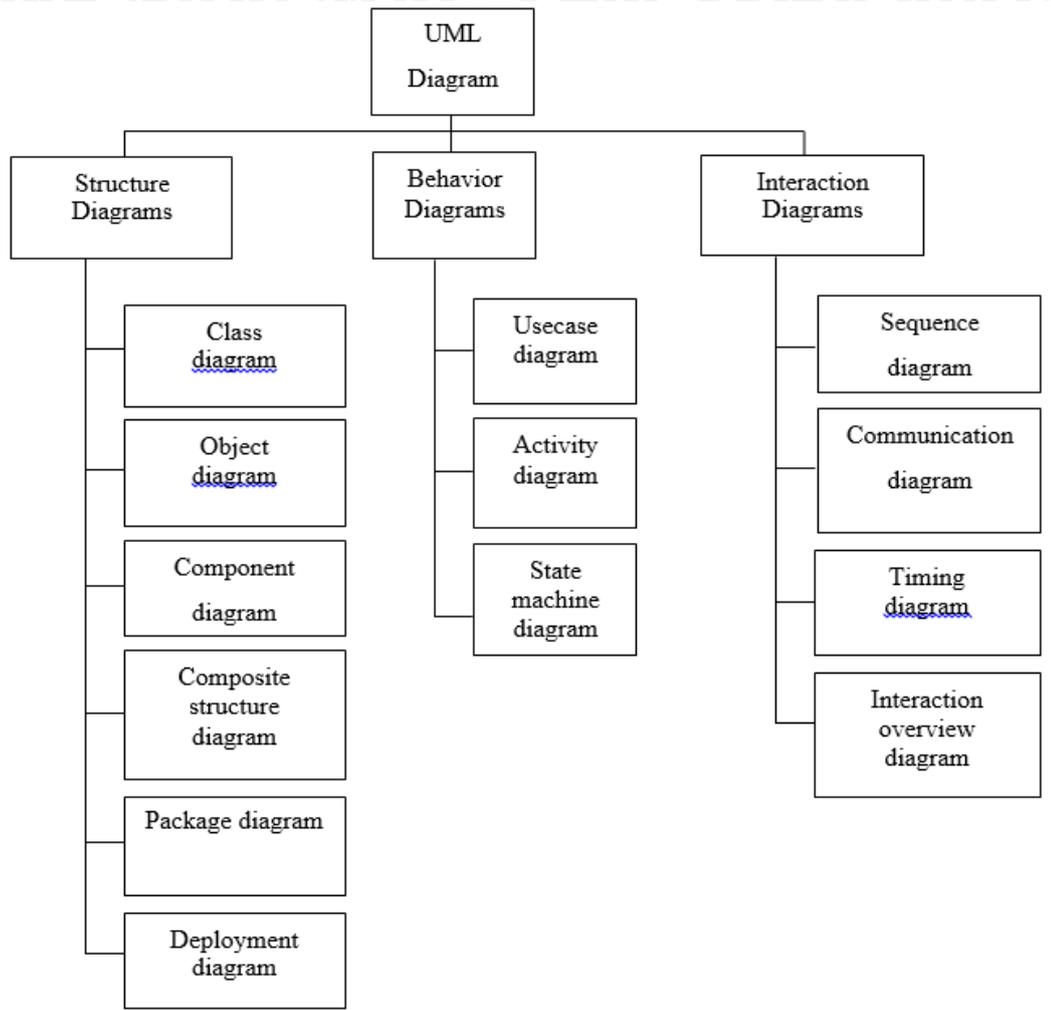
Gambar 2.3 Modul ERP

Sumber: [AGS-09]

Gambar 2.3 menunjukkan modul yang terintegrasi dalam sistem ERP. Mengingat universitas sebagai sistem ERP, modul utama dapat dibandingkan dengan fungsi umum seperti yang diamati dalam lingkungan universitas. Gambar 2.10 menunjukkan kegiatan masing-masing berubah sebagai modul di setup ERP. Dalam sebuah organisasi bisnis dimana sistem ERP diimplementasikan, penerima manfaat (*beneficiary*) adalah 'pelanggan', sedangkan pada universitas, penerima manfaat adalah 'mahasiswa'. Dengan penyesuaian seperti ini, perhatian utama untuk karir masa depan siswa akan dibentuk dengan baik. Dalam perjalanan waktu, perangkat tambahan lebih lanjut dan fitur nilai tambah dapat dimasukkan ke dalam sebuah sistem Perencanaan Sumber Daya Perusahaan berbasis web.

2.4 Unified Modelling Language

Unified Modelling Language (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram teks-teks pendukung. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikkan, menggambarkan, membangun, dan mendokumentasikan dari sistem perangkat lunak [ROS-11:117].



Gambar 2.4 Diagram UML

Sumber: [ROS-11:121]

Structure diagrams yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem [ROS-11 :121].

2.4.1 Usecase Diagram

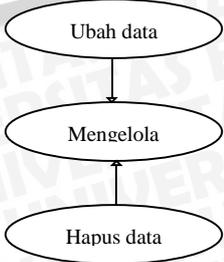
Diagram *Usecase* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Usecase* mendeskripsikan sebuah interaksi antara satu

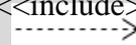
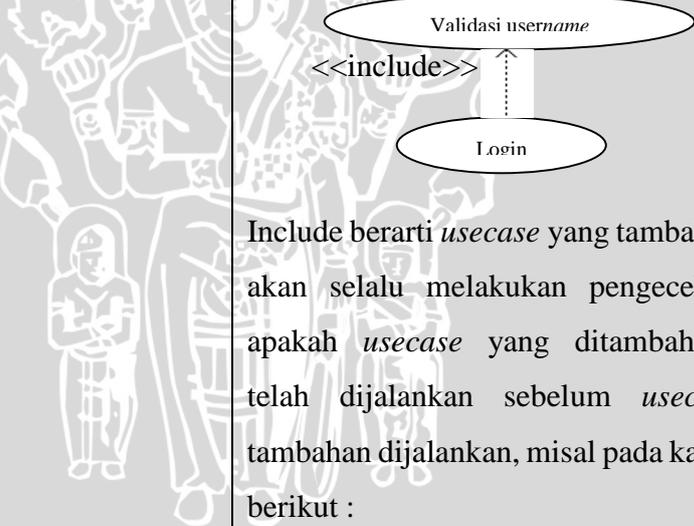
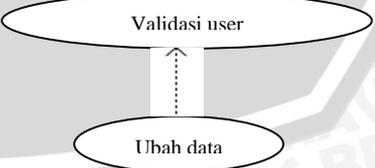
atau lebih aktor dengan sistem informasi yang akan dibuat. Diagram *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu [ROS-11:130].

Syarat penamaan pada diagram *usecase* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *usecase* yaitu pendefinisian apa yang disebut aktor dan *usecase*.

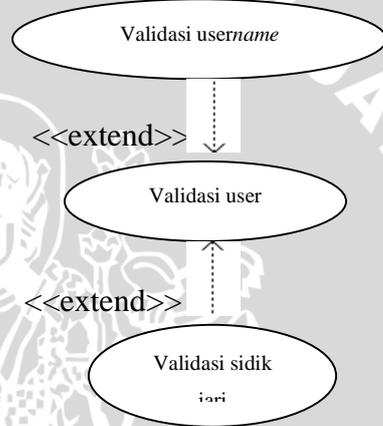
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Simbol aktor adalah g/ambar orang, namun aktor belum tentu merupakan orang.
2. *Usecase* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Tabel 2.1 Keterangan simbol-simbol *Usecase Diagram*

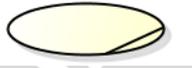
No	Gambar	Nama	Keterangan
1		Aktor / <i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2		Generalisasi/ <i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>usecase</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya : 

			<p>Arah panah mengarah pada <i>usecase</i> yang menjadi generalisasinya (umum).</p>
<p>3</p>	<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p><<include>> </p> <p><<uses>> </p> <p></p>	<p>Fungsinya atau sebagai syarat dijalankan <i>usecase</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di <i>usecase</i> :</p> <p>Include berarti <i>usecase</i> yang ditambahkan akan selalu dipanggil saat <i>usecase</i> tambahan dijalankan, misal pada kasus berikut :</p>  <p>Include berarti <i>usecase</i> yang tambahan akan selalu melakukan pengecekan apakah <i>usecase</i> yang ditambahkan telah dijalankan sebelum <i>usecase</i> tambahan dijalankan, misal pada kasus berikut :</p> 	<p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>



<p>4</p> 	<p>Ekstensi/ <i>Extend</i> <<extend>></p>	<p>Relasi <i>usecase</i> tambahan ke sebuah <i>usecase</i> dimana <i>usecase</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>usecase</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>usecase</i> tambahan memiliki nama depan yang sama dengan <i>usecase</i> yang ditambahkan, misal :</p>  <p>Arah panah mengarah pada <i>usecase</i> yang ditambahkan.</p>
<p>5</p> 	<p>Asosiasi/ <i>Association</i></p>	<p>Komunikasi antara aktor dan <i>usecase</i> yang berpartisipasi pada <i>usecase</i> atau <i>usecase</i> memiliki interksi dengan aktor.</p>
<p>6</p> 	<p><i>Sistem</i></p>	<p>Menspesifikasikan paket yang menampilkan sistem secara terbatas.</p>
<p>7</p> 	<p><i>Usecase</i></p>	<p>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>usecase</i>.</p>



8		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemennya (sinergi).
9		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.
10		<i>Control</i>	Elemen aktor yang melakukan proses bisnis
11		<i>Business</i>	Proses bisnis yang ada dalam <i>use case</i>

Sumber: [ROS-11 :130]

2.4.2 Class Diagram

Class diagram atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas [ROS-11:122].

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 berikut :

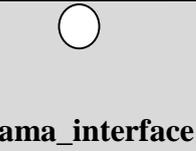
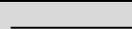
1. Kelas main
kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
2. Kelas yang menangani tampilan sistem
kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
3. Kelas yang diambil dari pendefinisian *usecase*
kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *usecase*
4. Kelas yang diambil dari pendefinisian data
kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data

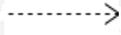
Jenis-jenis kelas dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan [ROS-11:122].

Cohesion dan *coupling* perlu diperhatikan dalam pendefinisian metode yang ada di dalam kelas. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain. *Coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas. Metode yang dibuat harus memiliki kadar *cohesion* yang kuat dan kadar *coupling coupling* yang lemah. Berikut adalah simbol-simbol yang ada pada kelas [ROS-11:123]

Tabel 2.2 Keterangan simbol-simbol *Class Diagram*

Sumber: [ROS-11:123]

No	Gambar	Nama	Keterangan
1		Kelas/ <i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
2		Antarmuka/ <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3		Asosiasi/ <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4		Asosiasi berarah/ <i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

5		Generalisasi/ <i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
6		Kebergantungan <i>/ dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7		Agregasi/ <i>Aggregation</i>	Semua-bagian (<i>whole-part</i>).

2.4.3 Sequence Diagram

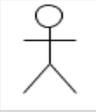
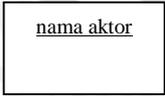
Sequence diagram atau diagram sekuen menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Penggambaran diagram sekuen harus diketahui objek-objek yang terlibat dalam sebuah *usecase* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu [ROS-11:137].

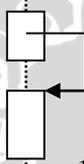
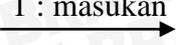
Diagram sekuen yang harus digambar adalah sebanyak pendefinisian *usecase* yang memiliki proses sendiri. Banyaknya *usecase* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

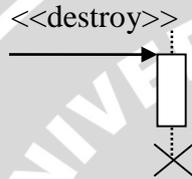
Berikut adalah simbol-simbol yang ada pada diagram sekuen :

Tabel 2.3 Keterangan simbol - simbol *Sequence Diagram*

Sumber: [ROS-11:138]

No	Gambar	Nama	Keterangan
1	 	Aktor/ <i>Actor</i> atau tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
2		Garis hidup/ <i>Lifeline</i>	Menyatakan kehidupan suatu objek.

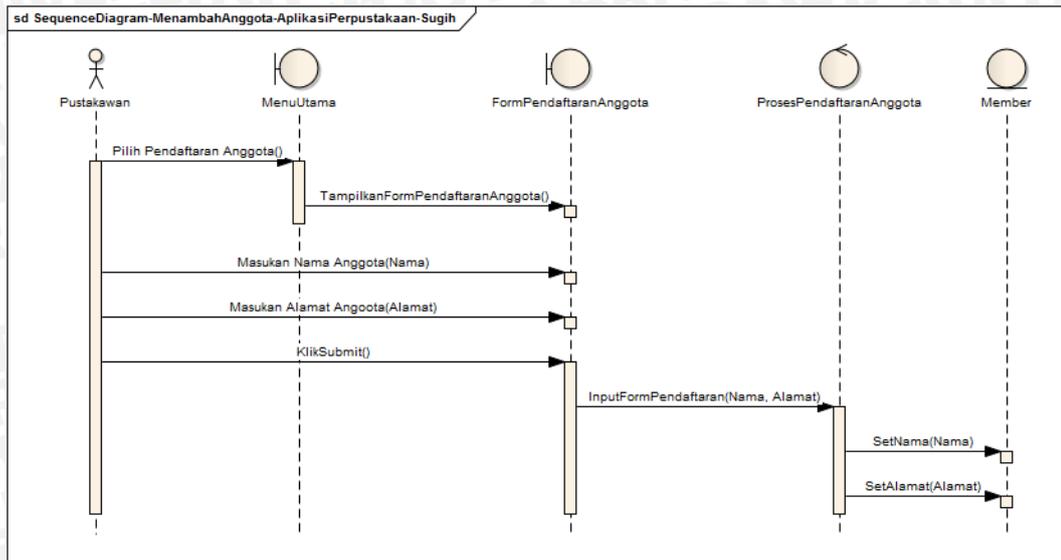
3	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> nama objek : nama kelas </div>	Objek/ <i>Object</i>	Menyatakan objek yang berinteraksi pesan.
4		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
5		Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.
6	1 : nama_metode() 	Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lin atau dirinya sendiri. 
7	1 : masukan 	Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data masukan atau informasi ke objek lainnya, arah

			panah mengarah pada objek yang akan dikirim.
8	1 : keluaran 	Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9		Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Penomoran pesan berdasarkan urutan interaksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu [ROS-11 :137].

Sequence Diagram menggambarkan urutan interaksi antar objek ketika satu Use Case dieksekusi/dilakukan, sehingga Sequence Diagram (sebaiknya) sebanyak Use Case nya. Dalam membuat Sequence Diagram ini ada yang menggunakan metode programming MVC (Model-View-Controller) atau dalam istilah lain Model=Entity, View=Boundary, dan Controller=Control.

View/Boundary adalah class yang berinteraksi langsung dengan Actor. Controller/Control adalah class interaksi perantara antara View/Boundary dan Mode/Entity. sedang Model/Entity adalah class yang menyimpan data. Berikut adalah contoh penggunaan sequence diagram dengan menggunakan metode MVC terdapat pada Gambar 2.5.



Gambar 2. 5 Contoh penggunaan sequence diagram pada MVC

2.5 Pengujian Perangkat Lunak

Pengujian (*Testing*) Arsitektur dari perangkat lunak berorientasi objek menghasilkan sekumpulan *layered subsystems* yang mengenkapsulasi kelas-kelas yang berkolaborasi. Setiap elemen sistem (subsistem dan *class*) melakukan fungsi yang membantu untuk mencapai kebutuhan sistem. Setiap elemen sistem sangat penting untuk menguji sebuah *object oriented sistem* pada berbagai macam level yang berbeda dalam sebuah usaha untuk menemukan kesalahan-kesalahan yang mungkin terjadi dari kolaborasi kelas-kelas dan komunikasi subsistem melewati *architetural layer* [PRE-01:631].

2.5.1 Teknik Pengujian

Pengujian perangkat lunak memerlukan perancangan kasus uji (*test case*) agar dapat menemukan kesalahan dalam waktu singkat dan usaha minimum. Berbagai macam metode perancangan kasus uji telah berevolusi. Metode-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 [PRE-01:443]. Teknik atau metode perancangan kasus uji yang digunakan adalah *black-box testing*.

Black box testing atau *behavioral testing* berfokus pada persyaratan fungsional perangkat lunak [PRE-01:459]. Dengan demikian, pengujian *black-box*

memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian *black-box* merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan.

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *basis data* eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

Pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Pengujian *black-box* memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi.

2.5.2 Strategi Pengujian

Strategi untuk pengujian perangkat lunak mengintegrasikan metode desain *test case* perangkat lunak ke dalam sederetan langkah yang direncanakan dengan baik, dan hasilnya adalah konstruksi perangkat lunak yang berhasil [PRE-01:477]. Sejumlah strategi pengujian perangkat lunak telah diusulkan di dalam literatur. Strategi pengujian harus mengakomodasi pengujian tingkat rendah yang diperlukan untuk membuktikan bahwa segmen kode sumber yang kecil telah diimplementasikan dengan tepat, demikian juga pengujian tingkat tinggi yang memvalidasi fungsi-fungsi sistem mayor yang berlawanan dengan kebutuhan pelanggan [PRE-01:481].

2.5.2.1 Usability Testing

Proses pembelajaran mengenai pengguna dengan mengobservasi mereka menggunakan produk untuk mencapai tujuan tertentu yang membuat mereka tertarik [JDU-08]. Usability testing dapat meliputi:

1. Mudah dipelajari
2. Mudah Digunakan
3. Menyenangkan
4. Intuitif

Pengujian ini berfokus dalam mengobservasi pengguna dengan menguji menggunakan tugas yang sebenarnya dan dapat dilakukan dengan dua cara yaitu:

1. Pengujian formatif : dilakukan pada saat program sedang dibuat.
2. Pengujian summatif: dilakukan pada saat program selesai dibuat.

2.5.2.2 Functional Testing

Pengujian fungsional memastikan bahwa fungsi individu bekerja dengan baik. Pengujian bagaimana fungsi aplikasi, atau, dengan kata lain, kaitannya dengan pengguna dan terutama ke seluruh sistem. Secara tradisional, pengujian fungsional dilaksanakan oleh tim penguji, independen dari pengembang [AGS-09].

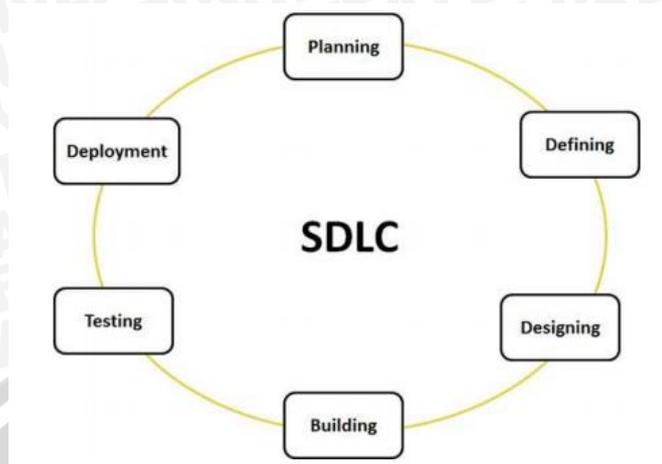
Sebuah tes fungsional tidak peduli dengan rincian internal sebuah aplikasi. Cara lain untuk menyatakan ini adalah bahwa pengujian fungsional adalah "tes pelanggan" [FTT-13]. Pengembang perlu patokan selama semua tahap pembangunan untuk memberitahu mereka apa yang mereka miliki dan belum dicapai. Pengujian fungsional dimulai segera setelah ada fungsi untuk menguji dan terus melalui penyelesaian aplikasi dan kontak pelanggan pertama. Beberapa orang berpikir pengujian fungsional sebagai meniru pengguna dan memeriksa output yang diharapkan. Tapi pelanggan nyata tidak hanya memberi perintah sederhana ke dalam aplikasi. Mereka menjalankan aplikasi pada sistem, bersamaan dengan aplikasi lain, dengan fluktuasi konstan dalam beban pengguna.

Aplikasi harus tahan benturan dalam menghadapi kondisi ini. Semakin banyak, user interface aplikasi kita diberikan oleh komponen pra-diuji, yang menghasilkan beberapa kejutan setelah mereka terintegrasi dengan benar. Di sisi lain, kita terus-menerus diminta untuk menulis aplikasi kustom untuk sistem yang terus meningkat kompleksitasnya. Oleh karena itu, fungsi utama diuji melalui uji fungsional yang berfokus kepada sistem yang terkait daripada pengguna terkait.

Tahap Pengambilan Kesimpulan dan Saran

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

2.6 Software Development Life Cycle



Gambar 2.6 SDLC Cycle

Sumber: [SDL-13: 2]

Sebuah Siklus Pengembangan Perangkat Lunak dalam gambar 2.7 terdiri dari tahapan sebagai berikut:

Tahap 1: Perencanaan dan Analisis Kebutuhan

Analisis Kebutuhan merupakan tahap yang paling penting dan mendasar dalam SDLC. Hal ini dilakukan oleh anggota senior dari tim dengan masukan dari pelanggan, departemen penjualan, survei pasar dan pakar domain dalam industri. Informasi ini kemudian digunakan untuk merencanakan pendekatan proyek dasar dan untuk melakukan studi kelayakan produk dalam ekonomi, operasional, dan teknis daerah. Perencanaan untuk persyaratan jaminan kualitas dan identifikasi risiko yang terkait dengan proyek ini juga dilakukan dalam tahap perencanaan. Hasil dari studi kelayakan teknis adalah untuk menentukan berbagai pendekatan teknis yang dapat diikuti untuk melaksanakan proyek tersebut berhasil dengan resiko minimal.

Tahap 2: Mendefinisikan Persyaratan

Setelah analisis kebutuhan dilakukan langkah berikutnya adalah untuk secara jelas mendefinisikan dan mendokumentasikan persyaratan produk dan membuat mereka disetujui dari pelanggan atau analis pasar. Hal ini dilakukan melalui 'SRS' - Dokumen Spesifikasi Kebutuhan Software yang terdiri dari semua

persyaratan produk yang akan dirancang dan dikembangkan selama siklus hidup proyek.

Tahap 3: Merancang arsitektur produk

SRS adalah referensi bagi arsitek produk untuk keluar dengan arsitektur terbaik bagi produk yang akan dikembangkan. Berdasarkan persyaratan yang ditentukan dalam SRS, biasanya lebih dari satu pendekatan desain untuk arsitektur produk diusulkan dan didokumentasikan dalam DDS - Dokumen Desain Spesifikasi. DDS ini ditinjau oleh semua pemangku kepentingan terkait dan berdasarkan berbagai parameter sebagai risiko.

Tahap 4: Membangun atau Mengembangkan Produk

Pada tahap ini SDLC perkembangan sebenarnya dimulai dan produk dibangun. Kode pemrograman yang dihasilkan sesuai DDS selama tahap ini. Jika desain dilakukan dengan cara yang rinci dan terorganisir, generasi kode dapat dicapai tanpa banyak kerumitan. Pengembang harus mengikuti pedoman pengkodean yang ditetapkan oleh organisasi mereka dan tool pemrograman seperti kompiler, interpreter, debugger dll digunakan untuk menghasilkan kode. Bahasa pemrograman tingkat tinggi yang berbeda seperti C, C++, Pascal, Java, dan PHP digunakan untuk pemrograman. Bahasa pemrograman yang dipilih sehubungan dengan jenis perangkat lunak yang dikembangkan.

Tahap 5: Pengujian Produk

Tahap ini biasanya subset dari semua tahapan seperti dalam model SDLC modern, kegiatan pengujian sebagian besar terlibat dalam semua tahapan SDLC. Namun tahap ini mengacu pada tahap -satunya pengujian produk di mana cacat produk dilaporkan, dilacak, tetap dan diuji ulang, sampai produk mencapai standar kualitas yang ditetapkan di SRS.

Tahap 6: Penyebaran di Pasar dan Pemeliharaan

Setelah produk diuji dan siap untuk digunakan dilepaskan secara resmi di pasar yang tepat. Kadang penyebaran produk terjadi secara bertahap sesuai strategi

bisnis organisasi. Produk ini mungkin pertama kali dirilis dalam segmen terbatas dan diuji dalam lingkungan bisnis yang nyata. Kemudian berdasarkan umpan balik, produk mungkin akan dirilis seperti itu atau dengan perangkat tambahan yang disarankan dalam segmen pasar sasaran. Setelah produk ini dirilis di pasar, pemeliharaan dilakukan untuk basis pelanggan yang sudah ada.

Model SDLC

Ada pengembangan perangkat lunak berbagai model siklus hidup didefinisikan dan dirancang yang diikuti selama proses pengembangan perangkat lunak. Model ini juga disebut sebagai "Proses Pengembangan Perangkat Lunak Model". Setiap model proses mengikuti Seri langkah unik untuk jenisnya, untuk memastikan keberhasilan dalam proses pengembangan perangkat lunak. Ada banyak model yang termasuk SDLC, salah satunya adalah *Software Prototyping* yang akan digunakan dalam penelitian ini.

2.7 Code Igniter

2.7.1 Selayang Pandang Code Igniter

Code Igniter adalah sebuah framework PHP. Framework itu sendiri adalah suatu kerangka kerja yang berupa sekumpulan folder yang memuat file-file php yang menyediakan class libraries, helpers, plugins dan lainnya. Framework menyediakan konfigurasi dan teknik coding tertentu.

2.7.2 Mendapatkan Code Igniter

Code Igniter dapat di peroleh secara gratis, dengan mendownloadnya di www.codeigniter.com.

2.7.3 Instalasi Code Igniter

Karena berupa folder php, maka instalasi Code Igniter, cukup dengan mengekstrak file kompresi yang di dapat dari situs www.codeigniter.com dan menempatkannya di dalam folder *directory web* server Anda. Kemudian edit file config.php yang ada di application/config/config.php dan set base URL anda, misal : <http://localhost/ci>. Jika anda hendak menggunakan database, edit file database.php yang ada di application/config/database.php. Setting database anda di file tsb.

Setelah di lakukan instalasi dan setting, test lah Code Igniter anda dengan membuka browser anda dan menyetikkan url yang anda taruh di file config.php di atas, misal : `http://localhost/ci`.

Jika muncul halaman welcome ala Code Igniter, maka Code Igniter telah siap di gunakan.

2.7.4 Konsep Model-View-Controller (MVC)

Konsep MVC adalah konsep pemisahan antara logic dengan tampilan dan database. Manfaat konsep ini adalah, membuat coding logic lebih simple, karena sudah di pisah dengan code untuk tampilan dan membuat programmer dapat bekerja secara terpisah dengan designer. Programmer mengerjakan logic, sedangkan designer berkuat dengan design dan tampilan.

Model merupakan code struktur data. Model berisi fungsi di dalam pengolahan database. Script Sql masuk di model. View merupakan code untuk menampilkan tampilan suatu program. Tampilan dapat berupa web page, header, footer dan apa saja yang berjenis tampilan. Sedangkan controller merupakan code untuk logic, algoritma dan sebagai penghubung antara model, view, dan sumber lain yang di perlukan untuk mengolah HTTP request dan generate web page.

CI menerapkan pola MVC yang flexible, karena model dapat tidak di gunakan. Anda dapat hanya menggunakan Controller dan View saja dalam menggunakan CI tanpa Model. Jika anda tidak memerlukan pemisahan di dalam struktur data dan database atau menganggap penggunaan model hanya menambah kompleks aplikasi dengan keuntungan yang kurang sebanding, maka anda dapat tidak menggunakan model. [SOF-07]

2.8 Basis Data

Basis data dapat didefinisikan dalam sejumlah sudut pandang, seperti menurut Connolly (2002), definisi basis data adalah kumpulan data yang dihubungkan secara bersama-sama, dan gambaran dari data yang dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi. Berbeda dengan sistem file yang menyimpan data secara terpisah, pada basis data data tersimpan secara terintegrasi. Basis data bukan menjadi milik dari suatu departemen tetapi sebagai sumber daya perusahaan yang dapat digunakan bersama.

Menurut Date (1990), definisi dari basis data adalah kumpulan terintegrasi dari file yang merupakan representasi data dari suatu model enterprise.

Sedangkan menurut Fathansyah (1999), basis data adalah :

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/ tabel/ arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Data dalam basis data disimpan dalam tiga struktur, yaitu file, tabel atau objek. File terdiri dari record dan field, tabel terdiri dari baris dan kolom. Objek terdiri dari data dan instruksi program yang memfungsikan data. Tabel terdiri dari kolom-kolom yang saling terkait, seperti file yang terdiri dari record yang saling terkait. File didalam basis data dapat terhubung kepada beberapa tabel. Dalam sebuah tabel, data pada tiap kolom terdiri dari ukuran dan tipe yang sejenis (char/ numeric).

2.9 MySQL

MySQL merupakan perangkat lunak yang tergolong basis data server dan bersifat Open Source, bisa diperoleh dengan cara mengunduh di Internet secara gratis. MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*).

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas, MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka, MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.

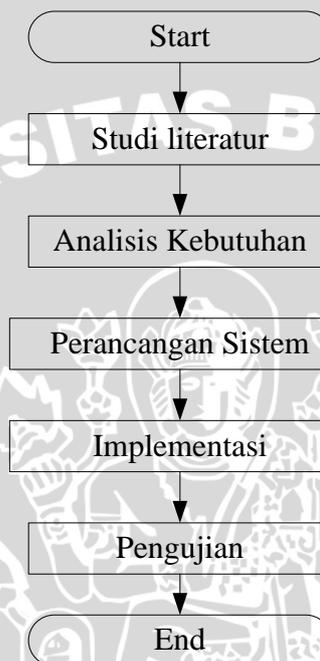
3. Multi-user, MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data, MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. Perintah dan Fungsi, MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
7. Keamanan, MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan, MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

Antar Muka, MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*). Klien dan Peralatan, MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.

BAB III

METODOLOGI PENELITIAN

Dalam bab ini membahas metode yang digunakan dalam penelitian yang terdiri dari studi literatur, metode analisis, metode perancangan sistem, metode implementasi, metode pengujian serta pengambilan kesimpulan dan saran. Berikut adalah diagram alir dari metodologi penelitian yang dilakukan



Gambar 3.1 Alur Pelaksanaan Kegiatan
Sumber: Perancangan

Pelaksanaan skripsi ini, dibagi menjadi 5 tahapan, yaitu tahap studi literatur, tahap analisis kebutuhan, tahap perancangan sistem, tahap implementasi dan tahap pengujian.

3.1 Tahap Studi Literatur

Studi literature menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut meliputi:

- Health Information Technology
- ERP
- Rekayasa Perangkat Lunak

- d. Unified Modeling Language
- e. Pengujian Perangkat Lunak
- f. Code Igniter
- g. Basis Data
- h. MySQL

3.2 Tahap Analisis Kebutuhan

Analisis sistem merupakan tahap yang paling penting dalam suatu pengembangan sebuah aplikasi, karena kesalahan pada tahap analisis sistem akan menyebabkan kesalahan pada tahap selanjutnya. Dengan adanya proses ini, diharapkan dapat menentukan sejauh mana aplikasi yang dibuat tersebut dapat mencapai target. Dari proses tersebut akan dihasilkan suatu gambaran sistem yang kemungkinan memiliki kesalahan-kesalahan ataupun kelemahan-kelemahan sehingga dimungkinkan dilakukan perbaikan.

Dalam sebuah sistem pasti sering terjadi suatu kendala untuk mencapai suatu tujuan dan sering kali kita menyadari bahwa masalah itu terjadi setelah sistem yang berjalan tidak benar. Jika terjadi masalah kita harus bisa menemukan penyebabnya, karena masalah timbul tidak dengan sendirinya melainkan ada sebab yang menimbulkan masalah tersebut. Tujuan dilakukannya analisis sistem yaitu untuk memahami sistem yang sedang berjalan.

Analisis kebutuhan dilakukan dengan menentukan kebutuhan apa saja yang dibutuhkan untuk membangun Sistem Enterprise Rumah Sakit menggunakan *design pattern* Model View Controller (MVC). Proses Analisis ini meliputi:

- a. Pemodelan *usecase* model dengan *as - is* dan *to - be*
- b. Pemodelan *activity* diagram model bisnis
- c. Pemodelan *activity* diagram model sistem

Metode analisis yang digunakan adalah *Object Oriented Analysis* dengan menggunakan bahasa UML (*Unified Modelling Language*). Fungsi-fungsi yang dapat dilakukan oleh aplikasi ini antara lain:

1. CRUD proses bisnis Sistem Enterprise Rumah Sakit
2. Melakukan pelaporan mengenai *Pharmacy, Medical Records, Medicare, Financial, Chasier, Front Office, Human Resource*.

Kebutuhan yang digunakan dalam pembuatan aplikasi ini meliputi:

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsionalistas berisi proses-proses apa saja yang nantinya dapat dilakukan oleh sistem, dibutuhkan sebuah sistem yang mampu melakukan fungsi-fungsi seperti :

1. Sistem mampu berjalan dengan baik waktu running.
2. Sistem mampu menjalankan fungsi–fungsi yang ada pada program
3. Aplikasi dapat berjalan dengan baik sesuai dengan kontrol dari user.
4. Sistem dapat melakukan proses ketika sedang dijalankan, sesuai dengan keinginan dan harapan Poliklinik Universitas Brawijaya.

3.2.2 Kebutuhan Non Fungsional

1. Kebutuhan perangkat keras

Perangkat keras yang dibutuhkan sebagai sarana penunjang dalam pembuatan Sistem Enterprise Rumah Sakit ini, yaitu berupa seperangkat PC (*Personal Computer*).

2. Kebutuhan perangkat lunak, meliputi:
 - a. Microsoft Windows 8 Proffesional sebagai sistem operasi
 - b. Adobe Dreamweaver CS5 sebagai platform pengembangan
 - c. XAMPP sebagai *tool* pembuatan server local
 - d. Astah UML sebagai *tool* pembuatan diagram pemodelan sistem

3.3 Tahap Perancangan Sistem

Aplikasi ini dirancang dengan menggunakan proses model *incremental*. Dalam model ini *client* mengidentifikasi secara garis besar layanan yang diberikan sistem. Model proses *incremental* mempunyai fase-fase perancangan *software*, yaitu: penentuan kebutuhan, spesifikasi kebutuhan, perancangan arsitektur *software*, desain *software* dan pengkodean serta pengujian [IAN-06].

Kelebihan dari model proses ini antara lain adalah proses *development* yang lebih cepat, lebih mudah dalam mengetahui kebutuhan pengguna, dan *resource* yang dibutuhkan untuk melakukan perubahan terhadap perangkat lunak yang dikembangkan lebih sedikit [IAN-11:33].

3.4 Tahap Implementasi

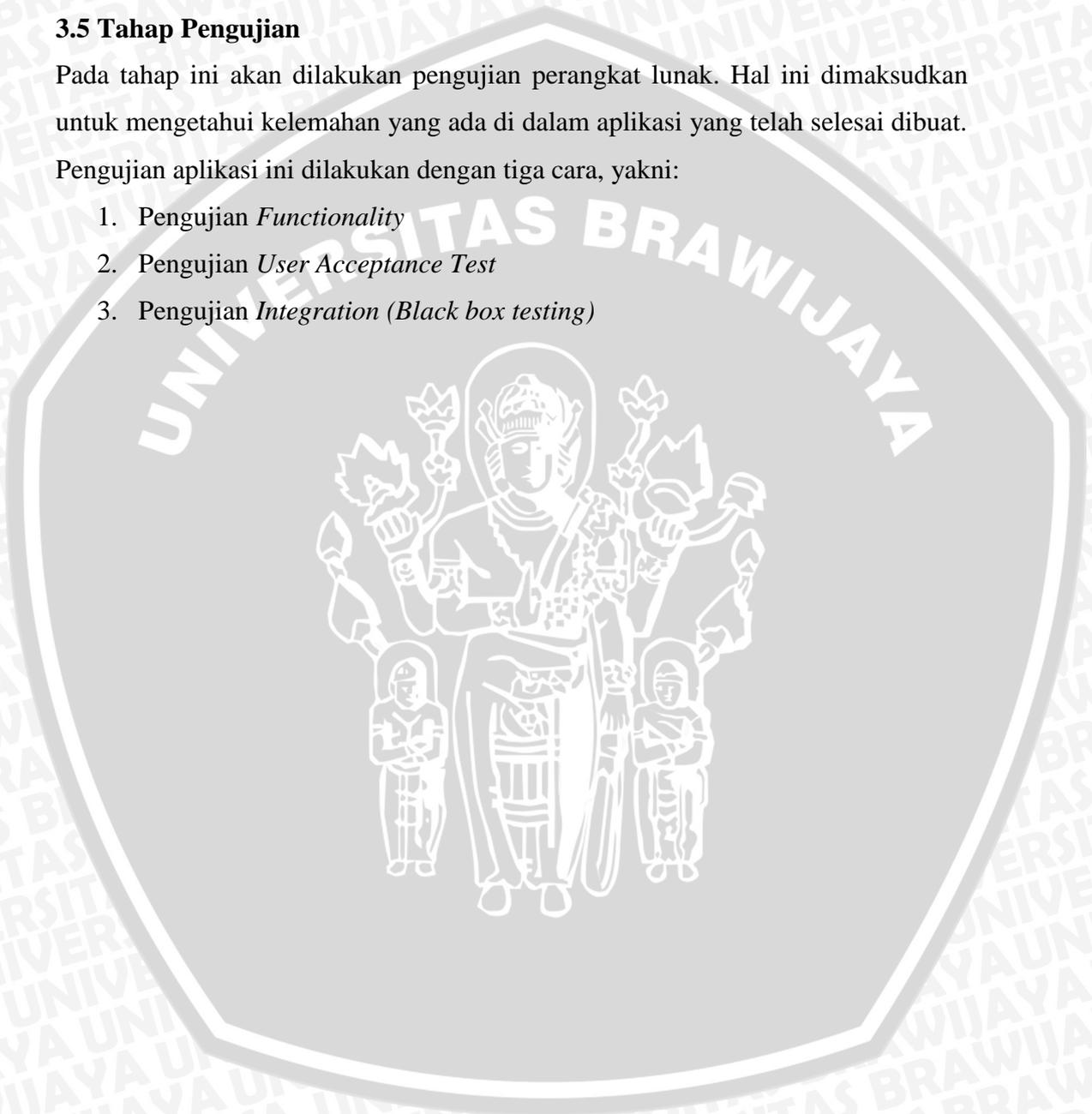
Pada tahap ini, seluruh analisis dan perancangan sistem akan diimplementasikan, yaitu melalui proses pengkodean (*coding*) menggunakan bahasa pemrograman PHP dan dibangun dengan *design pattern Model View Controller* (MVC).

3.5 Tahap Pengujian

Pada tahap ini akan dilakukan pengujian perangkat lunak. Hal ini dimaksudkan untuk mengetahui kelemahan yang ada di dalam aplikasi yang telah selesai dibuat.

Pengujian aplikasi ini dilakukan dengan tiga cara, yakni:

1. Pengujian *Functionality*
2. Pengujian *User Acceptance Test*
3. Pengujian *Integration (Black box testing)*



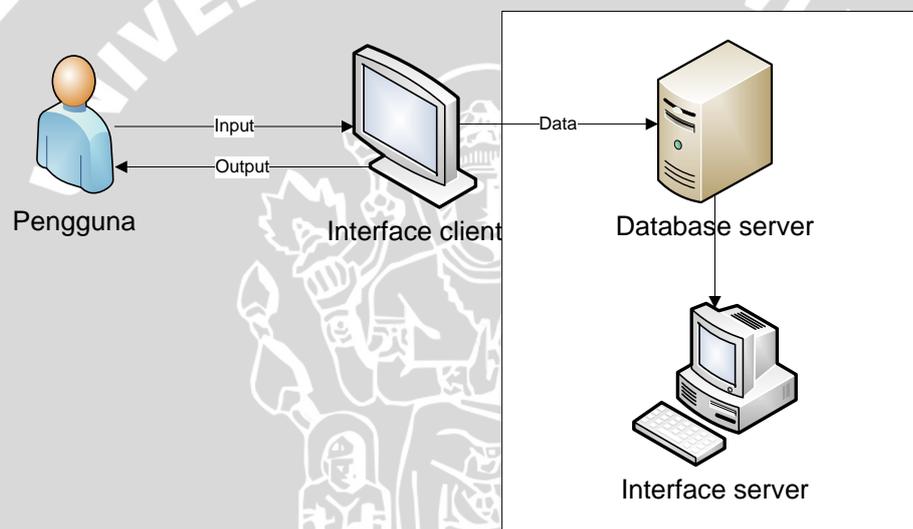
BAB IV

PERANCANGAN DAN IMPLEMENTASI

Pada tahap ini akan dibahas mengenai arsitektur, kerangka kerja dan analisis kebutuhan yang akan digunakan dalam pengembangan sistem ini.

4.1 Arsitektur Sistem Rumah Sakit berbasis ERP

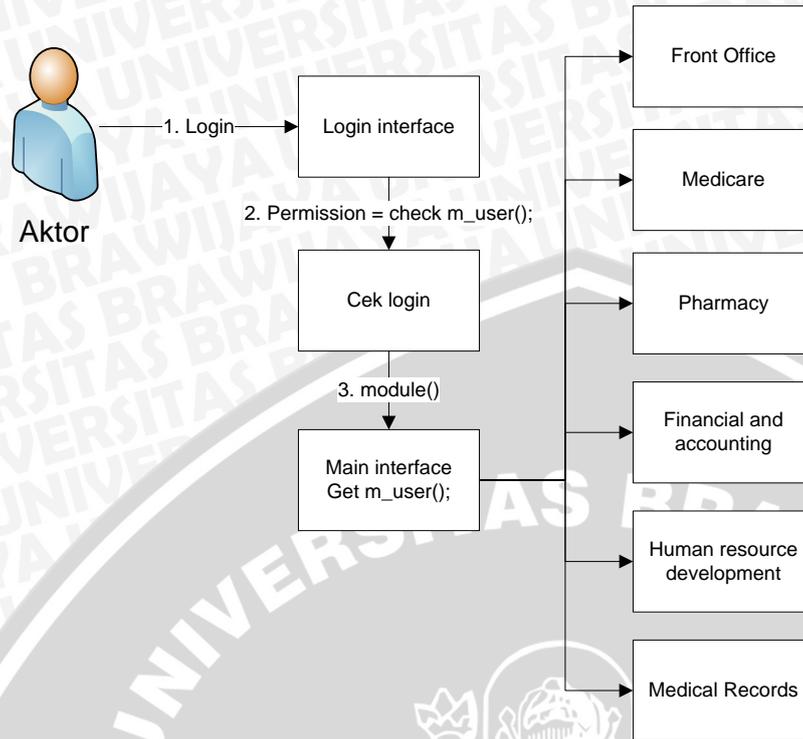
Aplikasi Sistem Enterprise Rumah Sakit dapat melakukan kegiatan registrasi pasien, rawat pasien, penggajian pegawai, rekrutmen pegawai dan pengaturan pengguna. Gambar 4.15 berikut menunjukkan arsitektur sistem secara umum:



Gambar 4.1 Arsitektur Sistem

Sumber: Perancangan

Aplikasi yang dikembangkan bersifat *thin client* sehingga kinerja operasi aplikasi ringan di sisi *client*. Basis data aplikasi diletakkan di server oleh karenanya dalam menggunakan aplikasi ini, *client* hanya mengirim dan menerima data yang sudah diproses di server. Dari arsitektur tersebut, aplikasi yang akan dikembangkan memiliki struktur kelas seperti pada gambar 4.2



Gambar 4.2 Komunikasi dengan diagram aplikasi

Sumber: Perancangan

Pada tahap (1) aplikasi menerima data masukan berupa *username* dan *password* di antarmuka login. Sistem melakukan pengecekan login dari user berupa hak akses (2). Setelah hak akses pengguna yang login diketahui maka aplikasi akan menampilkan antarmuka aplikasi (3) menurut hak akses pengguna apakah *front office*, *medicare*, *pharmacy*, *medical records*, *financial and accounting*, *human resoruce*. Masing-masing modul memiliki kegunaan. Berikut identifikasi dari tiap-tiap modul:

Tabel 4.1 Tabel Identifikasi Modul

Modul	Deskripsi Modul
Front Office	Front Office merupakan aktor yang berperan untuk menangani administrasi pasien pada rumah sakit
HR (Human resources)	<i>Human resources and development</i> (HRD) adalah aktor yang berperan untuk menangani kepegawaian yang ada di lingkungan rumah sakit

Financial	<i>Financial</i> adalah aktor yang berperan untuk menangani keuangan yang ada di rumah sakit
Medicare	Tenaga medis adalah aktor yang berperan menangani pemeriksaan dan pengobatan kepada pasien
Pharmacy	Apoteker adalah aktor yang berperan mengenai transaksi obat di rumah sakit
Chasier	<i>Chasier</i> adalah aktor yang berperan untuk menangani pembayaran pasien di rumah sakit
Medical Records	<i>Medcal Records</i> adalah aktor yang berperan untuk data rekam medis pasien di rumah sakit

4.2 Daftar Kebutuhan Perangkat Lunak

Daftar kebutuhan sistem terdiri dari kebutuhan fungsional dan kebutuhan non fungsional. Pada daftar kebutuhan akan dispesifikasikan menjadi dua yaitu spesifikasi kebutuhan fungsional *client* dan spesifikasi kebutuhan non fungsional. Spesifikasi kebutuhan fungsional ditunjukkan pada tabel 4.2.

Tabel 4.2 Daftar Kebutuhan Fungsional

Nama use case	Aktor	Requirements
Menampilkan daftar pegawai	<i>Human Resources</i>	Sistem menyediakan antarmuka untuk menampilkan daftar data pegawai pada staf kepegawaian
Tambah data pegawai	<i>Human Resources</i>	Sistem menyediakan fitur tambah data pegawai untuk staff kepegawaian
Ubah data pegawai	<i>Human Resources</i>	Sistem menyediakan fitur ubah data pegawai untuk staff kepegawaian.
Hapus daftar pegawai	<i>Human Resources</i>	Sistem menyediakan fitur hapus daftar pegawai untuk staff kepegawaian.

Menampilkan daftar kehadiran	<i>Human Resources</i>	Sistem menyediakan antarmuka untuk menampilkan daftar kehadiran untuk staff kepegawaian
Menampilkan riwayat penanganan dokter	<i>Front Office</i>	Sistem menyediakan antarmuka untuk menampilkan riwayat penanganan dokter
Register Pasien	<i>Front Office</i>	Sistem menyediakan fitur untuk Register Pasien
Ubah daftar antrian registrasi pasien	<i>Front Office</i>	Sistem menyediakan fitur untuk ubah daftar antrian registrasi pasien
Menampilkan dan menghapus daftar registrasi pasien	<i>Front Office</i>	Sistem menyediakan fitur untuk menampilkan dan menghapus daftar registrasi pasien
Pencarian Pasien	<i>Front Office dan Medical Records</i>	Sistem menyediakan fitur untuk Pencarian Pasien
Menampilkan statistik pengunjung per bulan	<i>Front Office</i>	Sistem menyediakan antarmuka untuk menampilkan statistik pengunjung per bulan
Menampilkan daftar data registrasi pasien	<i>Medical Records</i>	Sistem menyediakan antarmuka untuk menampilkan daftar data registrasi pasien
Tambah rekam medis pasien	<i>Medical Records dan Medicare</i>	Sistem menyediakan fitur untuk tambah rekam medis pasien
Ubah rekam medis pasien	<i>Medical Records</i>	Sistem menyediakan fitur untuk ubah rekam medis pasien
Hapus rekam medis pasien	<i>Medical Records</i>	Sistem menyediakan fitur untuk hapus rekam medis pasien
Menampilkan data pembayaran pasien	<i>Chasier</i>	Sistem menyediakan antarmuka untuk menampilkan daftar data pembayaran pasien

Tambah form pembayaran pasien	<i>Chasier</i>	Sistem menyediakan fitur untuk tambah form pembayaran pasien
Tambah form <i>invoice</i> pasien	<i>Chasier</i>	Sistem menyediakan fitur untuk tambah form <i>invoice</i> pasien
Cetak <i>invoice</i> pasien	<i>Chasier</i>	Sistem menyediakan fitur untuk cetak <i>invoice</i> pasien
Menampilkan data <i>warehouse</i> obat	<i>Pharmacy</i>	Sistem menyediakan antarmuka untuk menampilkan data <i>warehouse</i> obat
Edit data obat	<i>Pharmacy</i>	Sistem menyediakan fitur untuk edit data obat
Menampilkan transaksi obat pasien	<i>Pharmacy</i>	Sistem menyediakan antarmuka untuk menampilkan transaksi obat pasien
Validasi obat pasien	<i>Pharmacy</i>	Sistem menyediakan fitur untuk validasi obat pasien
Menampilkan data sumber dana	<i>Financial</i>	Sistem menyediakan antarmuka untuk menampilkan data sumber dana
Menampilkan pemasukan pasien	<i>Financial</i>	Sistem menyediakan antarmuka untuk menampilkan pemasukan pasien
Menampilkan pengeluaran keseluruhan	<i>Financial</i>	Sistem menyediakan antarmuka untuk menampilkan pengeluaran keseluruhan
Menampilkan pemasukan dan pengeluaran dalam bentuk <i>chart</i>	<i>Financial</i>	Sistem menyediakan antarmuka untuk menampilkan pemasukan dan pengeluaran dalam bentuk <i>chart</i>
Menampilkan daftar riwayat medis pasien	<i>Medicare</i>	Sistem menyediakan antarmuka untuk menampilkan daftar riwayat medis pasien

Menampilkan biodata pasien	<i>Medicare</i>	Sistem menyediakan antarmuka untuk menampilkan biodata pasien
Menampilkan seluruh rekam medis pasien tertentu	<i>Medicare</i>	Sistem menyediakan antarmuka untuk menampilkan seluruh rekam medis pasien tertentu
Menampilkan rekam medis pasien tertentu	<i>Medicare</i>	Sistem menyediakan antarmuka untuk menampilkan rekam medis pasien tertentu

Sumber: Perancangan

Berikut adalah kebutuhan lingkungan sistem ditunjukkan pada Tabel 4.3

Tabel 4.3 Kebutuhan lingkungan

Kebutuhan Lingkungan	
Hardware	<ul style="list-style-type: none"> - Personal Computer minimal processor Dual Core - RAM minimal 2 GB
Software	<ul style="list-style-type: none"> - Microsoft Windows 8 sebagai sistem operasi. - NetBeans IDE 7.4 sebagai pendukung pembuatan website. - AstahUML sebagai pembuat model perancangan aplikasi. - Microsoft Visio sebagai pembuat model perancangan aplikasi. - XAMPP sebagai alat pendukung basis data. - Code Igniter sebagai framework pembuatan web. - MooTools sebagai framework JavaScript.

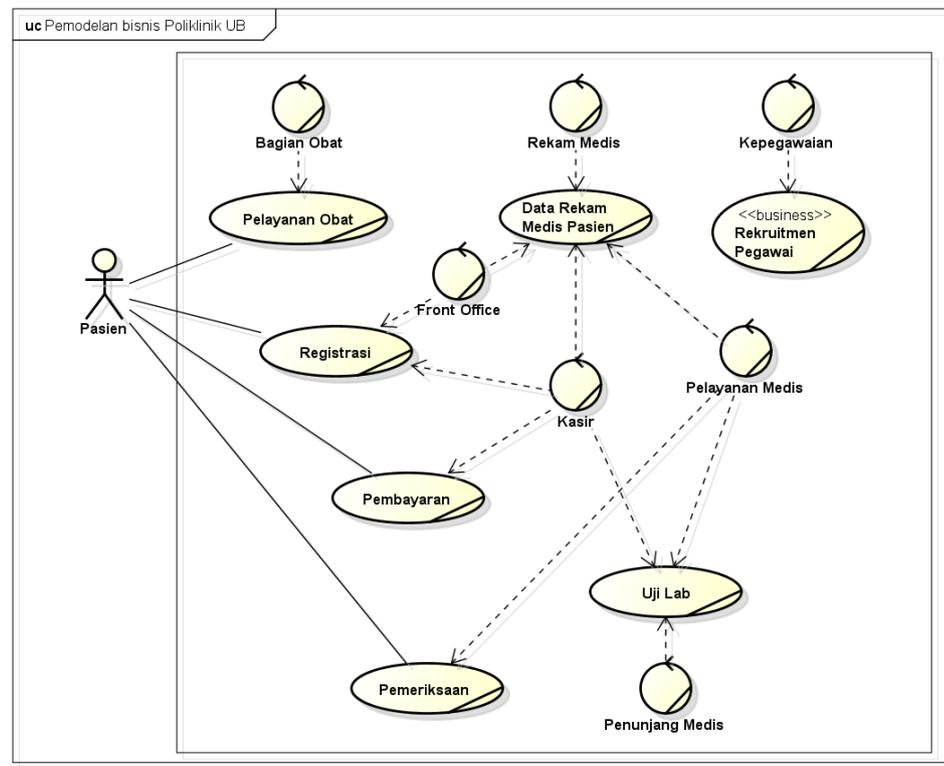
Sumber: Perancangan

4.3 Analisis Kebutuhan

Pada tahap ini proses analisis kebutuhan dibagi menjadi dua, yakni pemodelan bisnis dengan usecase (*as-is*) dan pemodelan sistem (*to-be*). Aplikasi yang akan dibuat pada penelitian ini adalah aplikasi yang digunakan untuk mengotomatisasi proses bisnis pada rumah sakit. Analisis ini akan membandingkan proses bisnis sebelum adanya sistem informasi berbasis komputer (*as-is*) dan proses bisnis yang berbasis komputer (*to-be*).

4.3.1 Pemodelan Bisnis dengan Usecase

Pada bagian ini akan dibahas mengenai proses-proses yang saat ini terjadi pada Poliklinik Universitas Brawijaya Malang sebelum menggunakan Sistem Enterprise Rumah Sakit. Adapun setiap kegiatan dapat dijelaskan oleh Gambar 4.3:

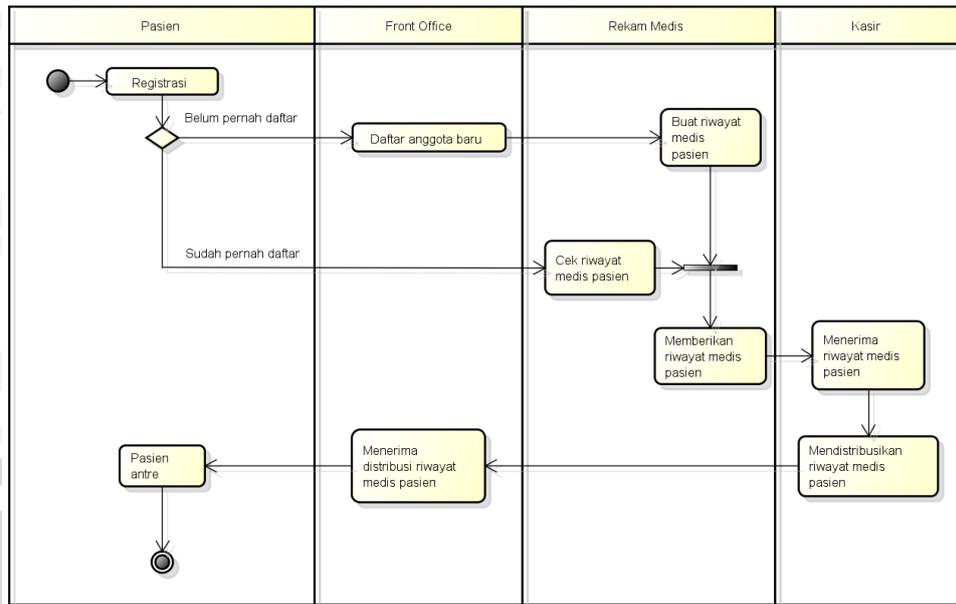


Gambar 4.3 Usecase Model Bisnis (AS-IS)

Sumber: Perancangan

Setiap usecase menggambarkan proses bisnis yang terjadi pada Poliklinik Universitas Brawijaya Malang. Adapun setiap usecase pada gambar 4.4 tersebut dijelaskan sebagai berikut:

1. Registrasi Pasien



Gambar 4.4 Activity Registrasi Pasien

Sumber: Perancangan

Model bisnis registrasi pasien pada gambar 4.4 di atas dapat diperjelas dengan runtutan kegiatan seperti tabel 4.4 berikut:

Tabel 4.4 Tabel Skenario Bisnis Registrasi Pasien

Flow Bisnis Registrasi Pasien
<p>1. <i>Basic Flow</i></p> <p>1.1 Registrasi</p> <p>Pada proses ini, pasien melakukan registrasi ke bagian Front Office. Namun pasien di sini dibagi menjadi dua, yaitu pasien yang telah terdaftar maupun pasien yang belum terdaftar</p> <p>1.2 Daftar anggota baru</p> <p>Proses ini hanya terjadi bila pasien belum terdaftar menjadi anggota di rumah sakit</p> <p>1.3 Buat riwayat medis pasien</p> <p>Proses ini melanjutkan dari proses sebelumnya, yakni setelah melalui proses Daftar Anggota Baru, pasien akan dibuatkan daftar riwayat medis yang berisi seluruh informasi mengenai catatan medis pasien</p> <p>1.4 Cek riwayat medis pasien</p>



Proses ini terjadi bila pasien sebelumnya telah terdaftar menjadi anggota

1.5 Memberikan riwayat medis pasien

Proses ini dilakukan oleh bagian rekam medis untuk diserahkan menuju Chasier

1.6 Menerima riwayat medis pasien

Proses ini bagian Chasier menerima riwayat medis pasien dari bagian rekam medis

1.7 Mendistribusikan rekam medis

Proses ini Chasier melakukan distribusi ke bagian *front office* sebelum didistribusikan ke bagian yang dibutuhkan oleh pasien

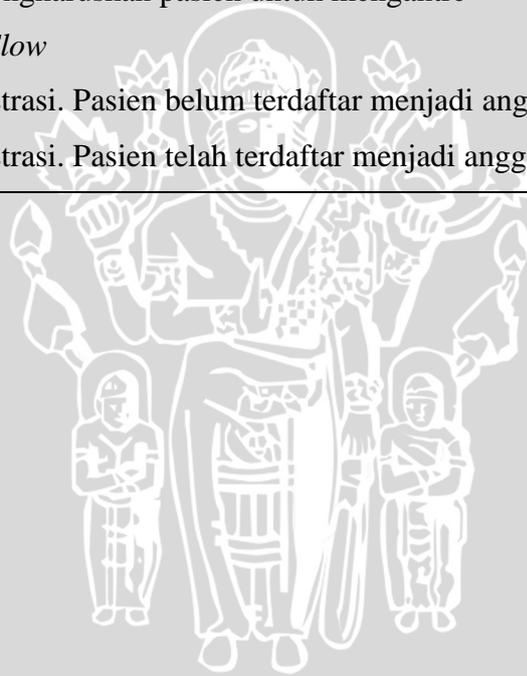
1.8 Pasien antre

Proses ini mengharuskan pasien untuk mengantre

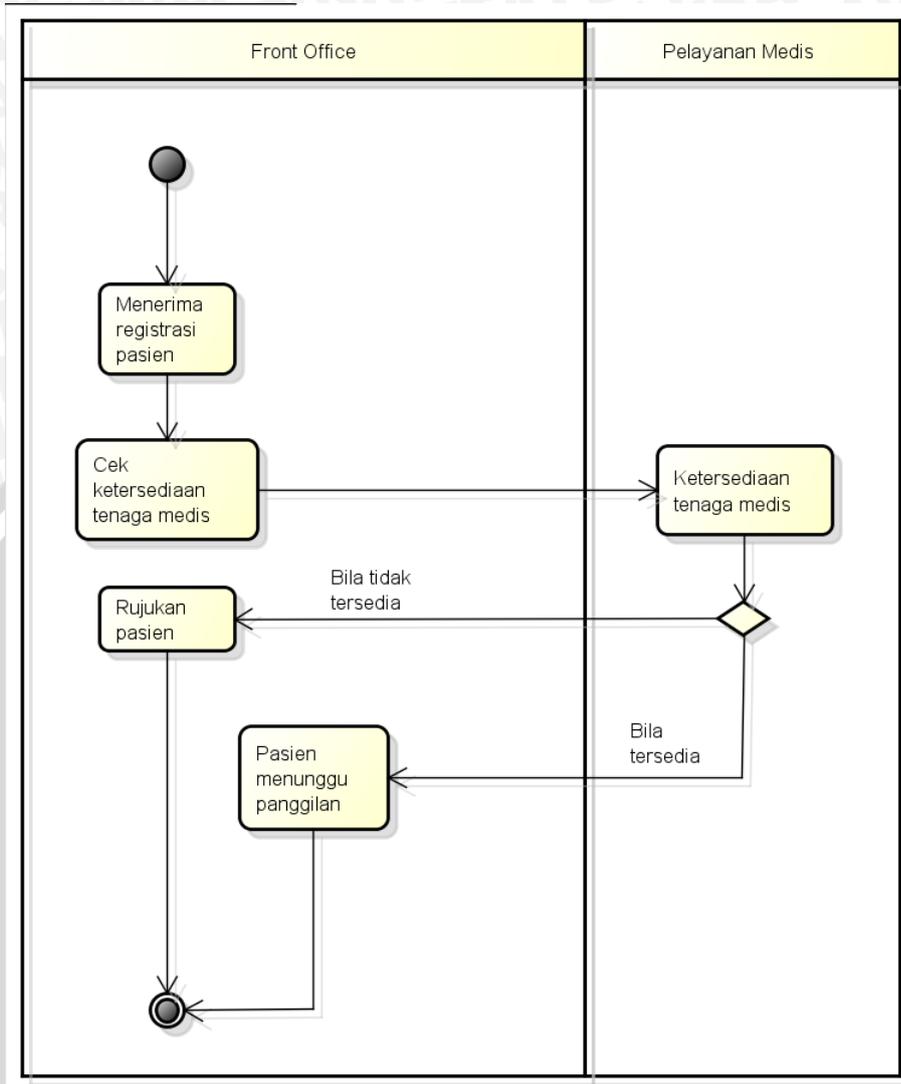
2. *Alternative Flow*

2.1 AT BF Registrasi. Pasien belum terdaftar menjadi anggota

2.2 AT BF Registrasi. Pasien telah terdaftar menjadi anggota



2. Ketersediaan Medis



Gambar 4.5 Activity Ketersediaan Medis

Sumber: Perancangan

Model bisnis ketersediaan medis pada Gambar 4.5 di atas dapat diperjelas dengan runtutan kegiatan seperti Tabel 4.5 berikut:

Tabel 4.5 Tabel Skenario Bisnis Ketersediaan Medis

<p><i>Flow</i> Bisnis Registrasi Pasien</p> <ol style="list-style-type: none"> 1. <i>Basic Flow</i> <ol style="list-style-type: none"> 1.1 Menerima registrasi pasien Pada proses ini, bagian <i>Front Office</i> menerima registrasi pasien 1.2 Cek Ketersediaan Medis

Proses ini *Front Office* melakukan pengecekan ketersediaan medis

1.3 Ketersediaan tenaga medis

Tenaga medis memberikan tanda bahwa saat itu sedang hadir atau tidak

1.4 Rujukan pasien

Proses ini terjadi bila pasien tenaga medis di rumah sakit yang dibutuhkan oleh pasien tidak tersedia

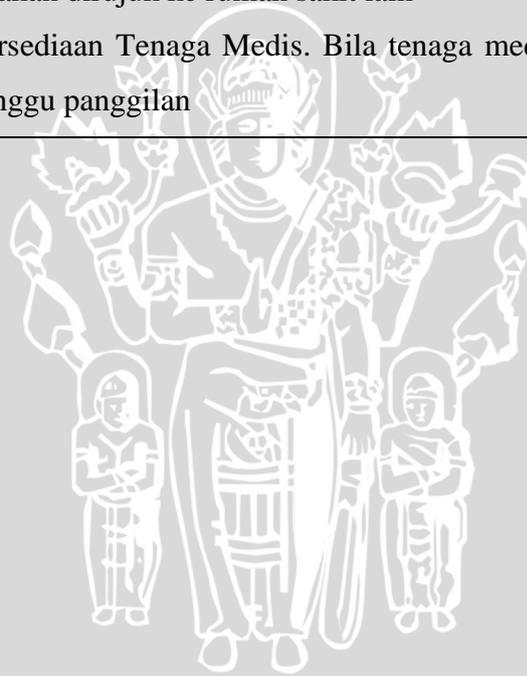
1.5 Pasien menunggu panggilan

Proses ini dilakukan oleh bagian *Fornt Office* untuk memanggil pasien bila tiba giliran

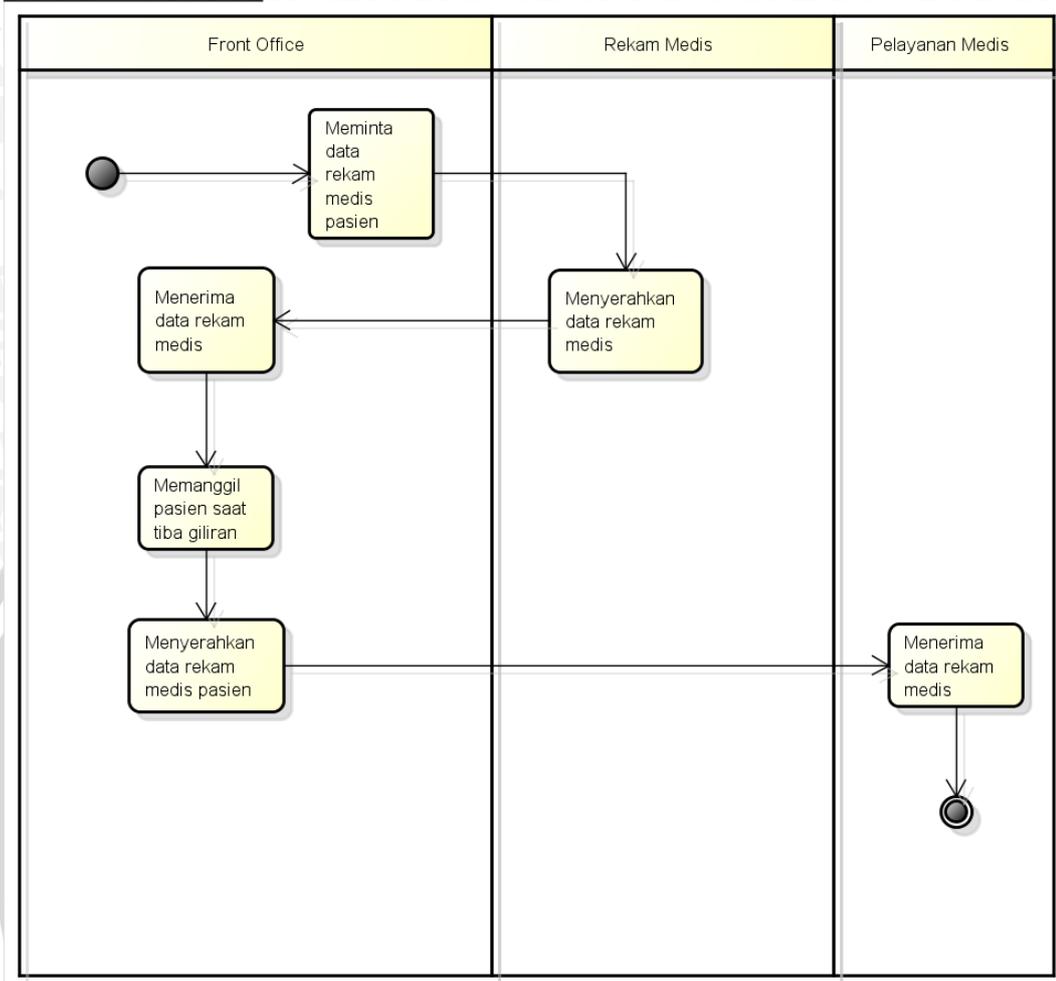
2. *Alternative Flow*

2.1 AT BF Ketersediaan Tenaga Medis. Bila tenaga medis tidak tersedia maka pasien akan dirujuk ke rumah sakit lain

2.2 AT BF Ketersediaan Tenaga Medis. Bila tenaga medis tersedia maka pasien menunggu panggilan



3. Data Rekam Medis Pasien



Gambar 4.6 Activity Data Rekam Medis Pasien

Sumber: Perancangan

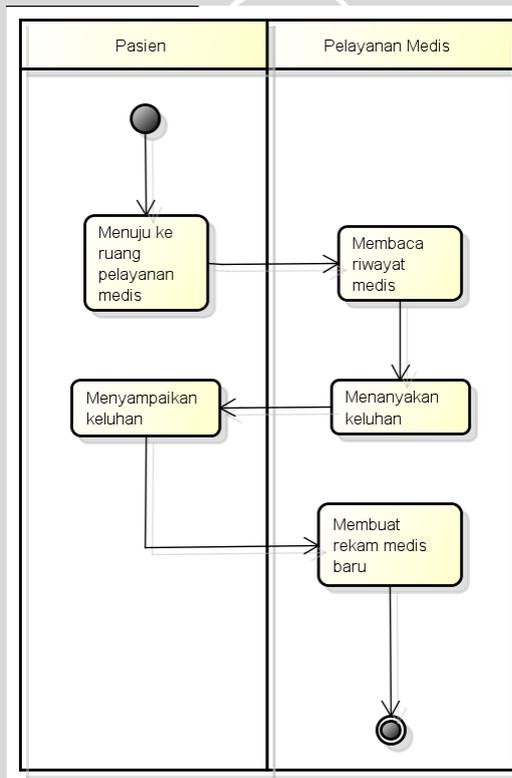
Model bisnis data rekam medis pasien pada Gambar 4.6 di atas dapat diperjelas dengan runtutan kegiatan seperti Tabel 4.6 berikut:

Tabel 4.6 Tabel Skenario Bisnis Ketersediaan Medis

<p><i>Flow</i> Bisnis Registrasi Pasien</p> <ol style="list-style-type: none"> 1. <i>Basic Flow</i> <ol style="list-style-type: none"> 1.1 Meminta data rekam medis pasien Pada proses ini, bagian <i>Front Office</i> meminta data rekam medis pasien 1.2 Menyerahkan data rekam medis pasien Bagian rekam medis menyerahkan data rekam medis ke bagian <i>Front Office</i>
--

- 1.3 Menerima data rekam medis
 - Bagian *Front Office* menerima data rekam medis pasien dari bagian rekam medis
- 1.4 Memanggil pasien saat tiba giliran
 - Bagian *Front Office* memanggil pasien saat tiba giliran
- 1.5 Menyerahkan data rekam medis pasien
 - Bagian *Front Office* mendistribusikan data rekam medis pasien ke bagian medis
- 1.6 Menerima data rekam medis pasien
 - Bagian medis menerima data rekam medis oleh bagian *Front Office*

4. Pemeriksaan



Gambar 4.7 Activity Pemeriksaan

Sumber: Perancangan

Model bisnis pemeriksaan pada Gambar 4.7 di atas dapat diperjelas dengan runtutan kegiatan seperti Tabel 4.7 berikut:



Tabel 4.7 Tabel Skenario Bisnis Pemeriksaan*Flow* Bisnis Registrasi Pasien1. *Basic Flow*

1.1 Menuju ke ruang pelayanan medis

Pada proses ini, pasien yang namanya dipanggil menuju ke ruang pelayanan medis

1.2 Membaca riwayat medis

Bagian tenaga medis membaca dan melihat catatan medis sebelumnya

1.3 Menanyakan keluhan

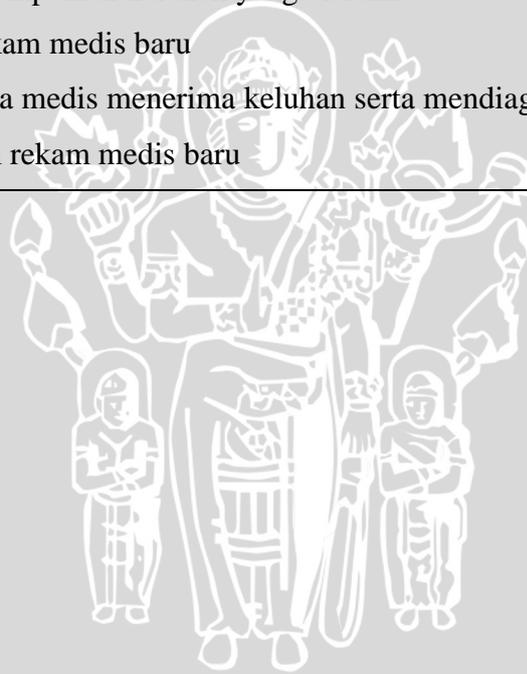
Bagian tenaga medis menanyakan keluhan kepada pasien

1.4 Menyampaikan keluhan

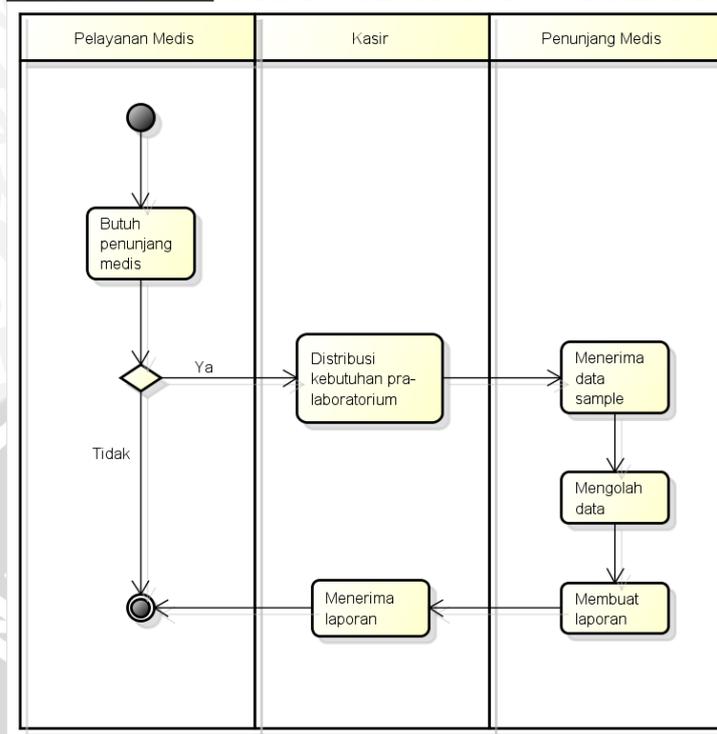
Pasien menyampaikan keluhan yang dialami

1.5 Membuat rekam medis baru

Bagian tenaga medis menerima keluhan serta mendiagnosa pasien yang dicatat dalam rekam medis baru



5. Uji Laboratorium



Gambar 4.8 Activity Uji Laboratorium

Sumber: Perancangan

Model bisnis uji laboratorium pada Gambar 4.8 di atas dapat diperjelas dengan runtutan kegiatan seperti Tabel 4.8 berikut:

Tabel 4.8 Tabel Skenario Bisnis Pemeriksaan

<p><i>Flow</i> Bisnis Registrasi Pasien</p> <ol style="list-style-type: none"> 1. <i>Basic Flow</i> <ol style="list-style-type: none"> 1.1 Butuh penunjang medis <p>Pada proses ini, bagian medis memiliki pilihan untuk memanfaatkan penunjang medis atau tidak</p> 1.2 Distribusi kebutuhan pra-labortorium <p>Bagian Chasier menerima sampel dari bagian medis</p> 1.3 Menerima data sampel <p>Bagian penunjang medis menerima data sampel dari bagian Chasier</p> 1.4 Mengolah data
--

Bagian penunjang medis melakukan uji laboratorium sehingga dihasilkan data olahan

1.5 Membuat laporan

Bagian penunjang medis membuat laporan hasil uji laboratorium

1.6 Menerima laporan

Bagian Chasier menerima laporan hasil uji laboratorium

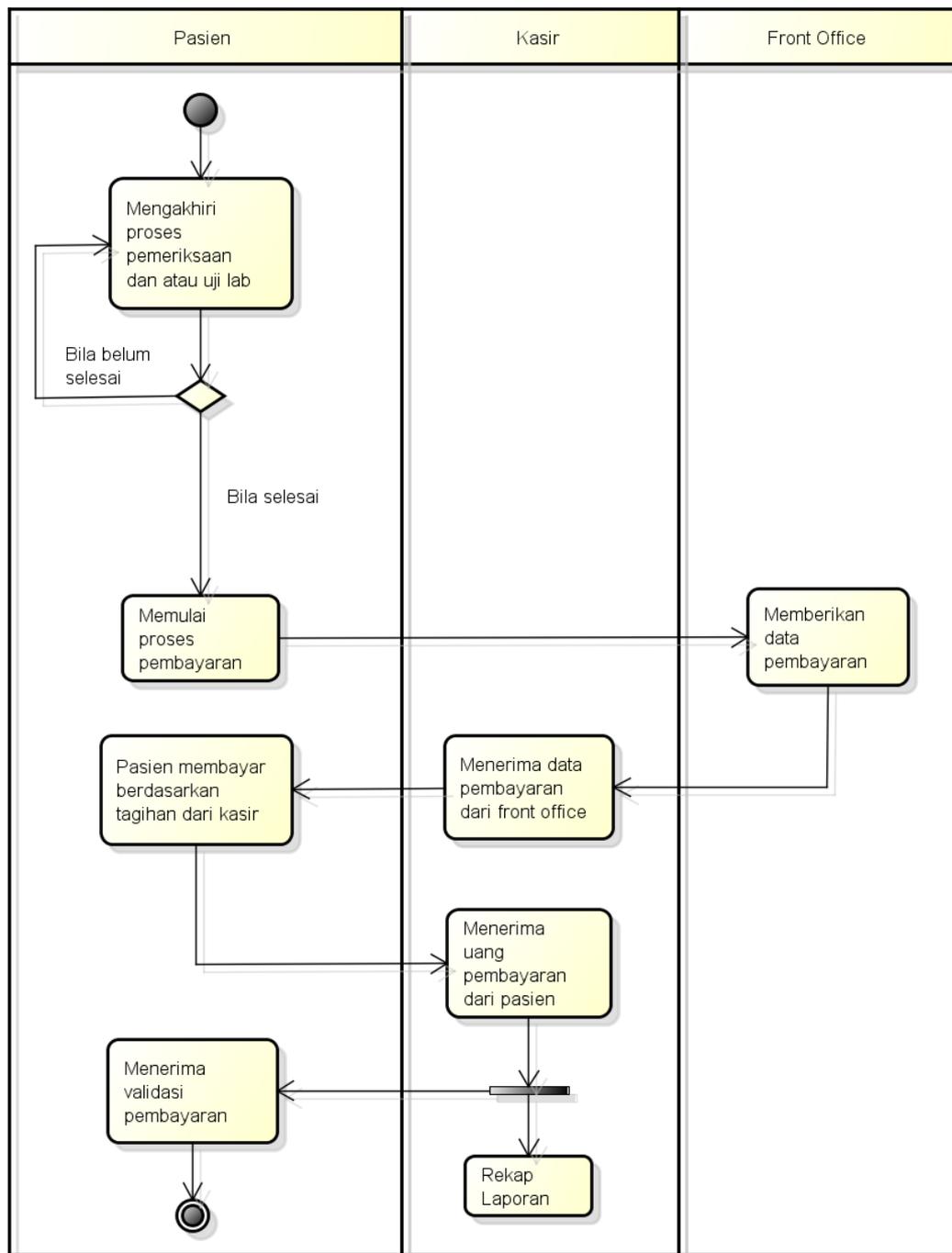
2. *Alternative Flow*

2.1 AT BF Butuh penunjang medis. Bila pasien berdasarkan rekomendasi bagian tim medis membutuhkan penunjang medis maka akan dilakukan uji laboratorium

2.2 AT BF Butuh penunjang medis. Bila pasien berdasarkan rekomendasi bagian tim medis tidak membutuhkan penunjang medis maka proses ini akan berakhir (tidak dilakukan uji laboratorium)



6. Pembayaran



Gambar 4.9 Activity Pembayaran

Sumber: Perancangan

Model bisnis pembayaran pada Gambar 4.9 di atas dapat diperjelas dengan runtutan kegiatan seperti Tabel 4.9 berikut:

Tabel 4.9 Tabel Skenario Bisnis Pembayaran*Flow* Bisnis Registrasi Pasien1. *Basic Flow*

1.1 Mengakhiri proses pemeriksaan dan atau uji lab

Proses ini pasien mengakhiri proses pemeriksaan dan atau uji lab

1.2 Memulai proses pembayaran

Bila proses pemeriksaan dan atau uji lab sudah selesai, maka proses pembayaran sudah bisa dimulai

1.3 Memberikan data pembayaran

Bagian *front office* memberikan data pembayaran kepada bagian Chasier

1.4 Menerima data pembayaran *front office*

Bagian Chasier menerima data pembayaran pasien dari bagian *front office*

1.5 Pasien membayar berdasarkan tagihan dari Chasier

Pasien melakukan pembayaran ke bagian Chasier berdasarkan tagihan dari Chasier

1.6 Menerima uang pembayaran dari pasien

Bagian Chasier menerima uang pembayaran dari pasien

1.7 Rekap laporan

Bagian Chasier merekap laporan harian pemasukan dari pembayaran pasien

1.8 Menerima validasi pembayaran

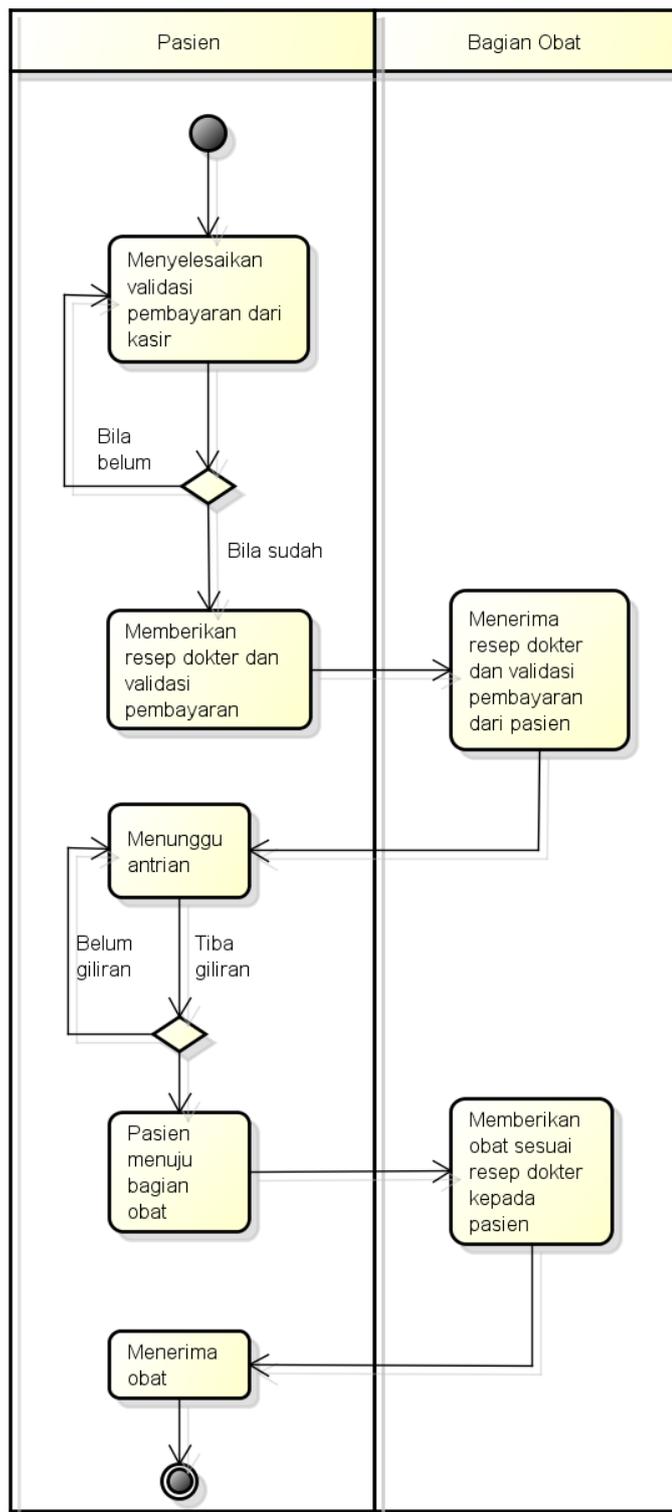
Pasien menerima validasi pembayaran

2. *Alternative Flow*

2.1 AT BF Mengakhiri proses pemeriksaan dan atau uji lab. Bila pasien telah mengakhiri proses pemeriksaan dan atau uji lab maka pasien dapat memulai proses pembayaran

2.2 AT BF Mengakhiri proses pemeriksaan dan atau uji lab. Bila pasien belum mengakhiri proses pemeriksaan dan atau uji lab maka pasien belum dapat memulai proses pembayaran

7. Layani obat



Gambar 4.10 Activity Layani Obat

Sumber: Perancangan

Model bisnis layani obat pada Gambar 4.10 di atas dapat diperjelas dengan runtutan kegiatan seperti tabel 4.10 berikut:

Tabel 4.10 Tabel Skenario Bisnis Layani Obat

Flow Bisnis Registrasi Pasien

1. *Basic Flow*

1.1 Menyelesaikan validasi pembayaran dari Chasier

Proses ini pasien mengakhiri proses pemeriksaan dan atau uji lab

1.2 Memulai proses pembayaran

Bila proses pemeriksaan dan atau uji lab sudah selesai, maka proses pembayaran sudah bisa dimulai

1.3 Memberikan data pembayaran

Bagian *front office* memberikan data pembayaran kepada bagian Chasier

1.4 Menerima data pembayaran *front office*

Bagian Chasier menerima data pembayaran pasien dari bagian *front office*

1.5 Pasien membayar berdasarkan tagihan dari Chasier

Pasien melakukan pembayaran ke bagian Chasier berdasarkan tagihan dari Chasier

1.6 Menerima uang pembayaran dari pasien

Bagian Chasier menerima uang pembayaran dari pasien

1.7 Rekap laporan

Bagian Chasier merekap laporan harian pemasukan dari pembayaran pasien

1.8 Menerima validasi pembayaran

Pasien menerima validasi pembayaran

2. *Alternative Flow*

2.1 AT BF Mengakhiri proses pemeriksaan dan atau uji lab. Bila pasien telah mengakhiri proses pemeriksaan dan atau uji lab maka pasien dapat memulai proses pembayaran

2.2 AT BF Mengakhiri proses pemeriksaan dan atau uji lab. Bila pasien belum mengakhiri proses pemeriksaan dan atau uji lab maka pasien belum dapat memulai proses pembayaran

4.3.2 Pemodelan Sistem dengan *Use Case*

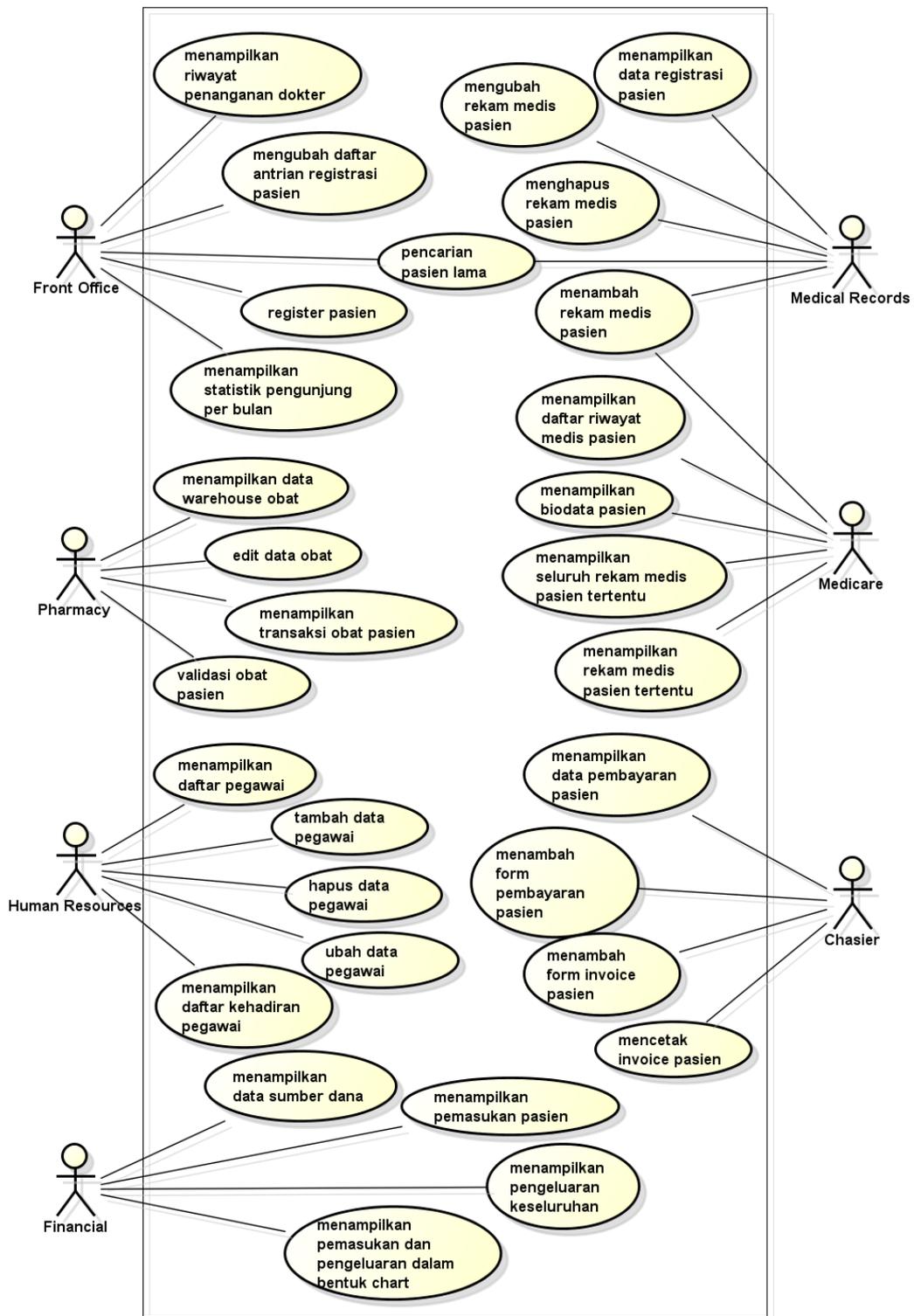
Dari penjelasan proses bisnis pada sub bab 4.3.1 maka dikembangkan suatu aplikasi berbasis ERP yang dapat mengurangi aktivitas manual menjadi otomatisasi. Dalam pengembangan sistem ini perubahan analisis kebutuhan dari sistem menjadi sebagai berikut:

4.3.2.1 Diagram *Use case*

Diagram *use case* digunakan untuk mendeskripsikan kebutuhan-kebutuhan dan fungsionalitas sistem dari perspektif end-user dan menunjukkan aksi-aksi yang dapat dilakukan oleh user.

Gambar 4.11 menunjukkan spesifikasi fungsi-fungsi yang disediakan sistem dari *Medicare, Financial, Chasier, Human Resource, Pharmacy, Medical Records, Front Office*.





Gambar 4.11 Usecase Model Sistem (to-be)

Sumber: Perancangan



4.3.2.2 Skenario *Use case*

Setiap *usecase* pada Gambar 4.11 akan dijelaskan lebih detail dalam skenario *use case*. Sebuah skenario *use case* mejabarkan uraian nama *use case*, aktor yang berhubungan dengan *use case* tersebut, tujuan *use case* tersebut dalam sistem, deskripsi global tentang *use case*, kondisi awal yang harus dipenuhi dan kondisi akhir yang diharapkan setelah berjalannya fungsional *use case*.

1.) *Use case* Menampilkan Daftar Pegawai

Use case menampilkan daftar pegawai akan diterangkan pada Tabel 4.11.

Tabel 4.11 *Use case* Menampilkan Daftar Pegawai

Use case name	Menampilkan Daftar Pegawai
Actor	<i>Human Resources</i>
Description	Aktor melihat informasi data kepegawaian
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah melihat informasi data kepegawaian
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman kepegawaian
Alternate course	

2.) *Use case* Tambah Data Pegawai

Use case menambah data pegawai akan diterangkan pada Tabel 4.12.

Tabel 4.12 *Use case* Tambah data pegawai

Use case name	Tambah Data Pegawai
Actor	<i>Human Resources</i>
Description	Aktor melakukan tambah data pegawai
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor mendapat pemberitahuan pegawai baru atau pegawai lama yang belum diinputkan ke dalam database kepegawaian
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka form tambah pegawai 3. Aktor mengisi data pegawai
Alternate course	

3.) *Use case* Ubah Data Pegawai

Use case mengubah data pegawai akan diterangkan pada Tabel 4.13.

Tabel 4.13 *Use case* Ubah Data Pegawai

Use case name	Ubah Data Pegawai
Actor	<i>Human Resources</i>
Description	Aktor melakukan ubah data pegawai
Pre-codition	1. Aktor telah login ke dalam sistem 2. Data pegawai telah ada
Post condition	Aktor telah melihat notifikasi dari pegawai tentang permintaan ubah biodata pegawai
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman edit pegawai 3. Aktor mengubah data pegawai
Alternate course	

4.) *Use case* Hapus Data Pegawai

Use case menghapus data pegawai akan diterangkan pada Tabel 4.14.

Tabel 4.14 *Use case* Hapus Data Pegawai

Use case name	Hapus Data Pegawai
Actor	<i>Human Resources</i>
Description	Aktor melakukan penghapusan data pegawai
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah melihat informasi kesalahan data pegawai
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman registrasi pasien
Alternate course	

5.) *Use case* Menampilkan Daftar Kehadiran

Use case menampilkan daftar kehadiran pegawai akan diterangkan pada Tabel 4.15.

Tabel 4.15 *Use case* Menampilkan Daftar Kehadiran

Use case name	Menampilkan Daftar Kehadiran
Actor	<i>Human Resources</i>
Description	Aktor melihat informasi daftar kehadiran pegawai
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah melihat informasi informasi daftar kehadiran pegawai
Flow of Events	<ol style="list-style-type: none"> 1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman daftar kehadiran pegawai
Alternate course	

6.) *Use case* Menampilkan Riwayat Penanganan Dokter

Use case menampilkan riwayat penanganan dokter akan diterangkan pada Tabel 4.16.

Tabel 4.16 *Use case* Menampilkan Riwayat Penanganan Dokter

Use case name	Menampilkan Riwayat Penanganan Dokter
Actor	<i>Front Office</i>
Description	Aktor melihat informasi riwayat penanganan dokter dan jumlah penanganannya
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah melihat informasi riwayat penanganan dokter dan jumlah penanganannya
Flow of Events	<ol style="list-style-type: none"> 1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman registrasi pasien
Alternate course	

7.) *Use case* Register Pasien

Use case register pasien akan diterangkan pada Tabel 4.17.

Tabel 4.17 *Use case* Register Pasien

Use case name	Melakukan Register Pasien
Actor	<i>Front Office</i>

Description	Aktor melakukan register pasien yang berkunjung untuk periksa dan berobat
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor mengisi form registrasi pasien
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman registrasi pasien 3. Aktor mengisi form registrasi pasien
Alternate course	

8.) *Use case* Ubah Daftar Antrian Registrasi Pasien

Use case ubah daftar antrian registrasi pasien akan diterangkan pada Tabel 4.18.

Tabel 4.18 *Use case* Ubah Daftar Antrian Registrasi Pasien

Use case name	Ubah Daftar Antrian Registrasi Pasien
Actor	<i>Front Office</i>
Description	Aktor mengubah daftar antrian registrasi pasien bila terjadi hal-hal seperti pergantian nomor urut pasien
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah mengubah daftar antrian registrasi pasien
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman registrasi pasien 3. Aktor mengubah daftar antrian registrasi pasien
Alternate course	

9.) *Use case* Menampilkan dan Menghapus Daftar Registrasi Pasien

Use case menampilkan dan menghapus daftar registrasi pasien akan diterangkan pada Tabel 4.19.

Tabel 4.19 *Use case* Menampilkan dan Menghapus Daftar Registrasi Pasien

Use case name	Menampilkan dan Menghapus Daftar Registrasi Pasien
Actor	<i>Front Office</i>

Description	Aktor menampilkan dan menghapus daftar registrasi pasien bila pasien tidak menginginkan untuk melanjutkan proses di dalam rumah sakit
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah menampilkan dan menghapus daftar registrasi pasien
Flow of Events	<ol style="list-style-type: none"> 1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman registrasi pasien 3. Aktor menampilkan dan menghapus daftar registrasi pasien
Alternate course	

10.) *Use case* Pencarian Pasien

Use case Pencarian Pasien akan diterangkan pada Tabel 4.20.

Tabel 4.20 *Use case* Pencarian Pasien

Use case name	Pencarian Pasien
Actor	<i>Front Office</i> dan <i>Medical Records</i>
Description	Aktor melakukan pencarian terhadap pasien dengan parameter nama, nomor rekam medis, alamat, dsb
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah melihat informasi riwayat penanganan dokter dan jumlah penanganannya
Flow of Events	<ol style="list-style-type: none"> 1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman registrasi pasien
Alternate course	

11.) *Use case* Menampilkan Statistik Pengunjung Per Bulan

Use case menampilkan statistik pengunjung akan diterangkan pada Tabel 4.21.

Tabel 4.21 *Use case* Menampilkan Statistik Pengunjung Per Bulan

Use case name	Menampilkan Statistik Pengunjung Per Bulan
Actor	<i>Front Office</i>

Description	Aktor melakukan pencarian terhadap pasien dengan parameter nama, nomor rekam medis, alamat, dsb
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah melihat informasi riwayat penanganan dokter dan jumlah penanganannya
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman registrasi pasien
Alternate course	

12.) *Use case* Menampilkan Daftar Data Registrasi Pasien

Use case menampilkan daftar data registrasi pasien akan diterangkan pada Tabel 4.22.

Tabel 4.22 *Use case* Menampilkan Daftar Data Registrasi Pasien

<i>Use case name</i>	Menampilkan Daftar Data Registrasi Pasien
Actor	<i>Medical Records</i>
Description	Aktor menampilkan daftar data registrasi pasien
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah melihat informasi daftar data registrasi pasien
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman registrasi pasien
Alternate course	

13.) *Use case* Tambah Rekam Medis Pasien

Use case tambah rekam medis pasien akan diterangkan pada Tabel 4.23.

Tabel 4.23 *Use case* Tambah Rekam Medis Pasien

<i>Use case name</i>	Tambah Rekam Medis Pasien
Actor	<i>Medical Records</i>
Description	Aktor menambah rekam medis setelah pasien melakukan pemeriksaan
Pre-codition	Aktor telah login ke dalam sistem

Post condition	Aktor telah menambah rekam medis pasien
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman rekam medis pasien
Alternate course	

14.) *Use case* Ubah Rekam Medis Pasien

Use case ubah rekam medis pasien akan diterangkan pada Tabel 4.24.

Tabel 4.24 *Use case* Ubah Rekam Medis Pasien

Use case name	Ubah Rekam Medis Pasien
Actor	<i>Medical Records</i>
Description	Aktor mengubah rekam medis setelah pasien melakukan pemeriksaan
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah menambah rekam medis pasien
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman rekam medis pasien 3. Aktor mengubah rekam medis pasien
Alternate course	

15.) *Use case* Hapus Rekam Medis Pasien

Use case hapus rekam medis pasien akan diterangkan pada Tabel 4.25.

Tabel 4.25 *Use case* Hapus Rekam Medis Pasien

Use case name	Hapus Rekam Medis Pasien
Actor	<i>Medical Records</i>
Description	Aktor menghapus rekam medis setelah pasien melakukan pemeriksaan
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah menambah rekam medis pasien
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman rekam medis pasien 3. Aktor menghapus rekam medis pasien

Alternate course	
-------------------------	--

16.) *Use case* Menampilkan Data Pembayaran Pasien

Use case menampilkan data pembayaran pasien akan diterangkan pada Tabel 4.26.

Tabel 4.26 *Use case* Menampilkan Data Pembayaran Pasien

Use case name	Menampilkan Data Pembayaran Pasien
Actor	<i>Chasier</i>
Description	Aktor menampilkan data pembayaran pasien
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah menampilkan data pembayaran pasien
Flow of Events	<ol style="list-style-type: none"> 1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman pembayaran pasien 3. Aktor menampilkan data pembayaran pasien
Alternate course	

17.) *Use case* Tambah Form Pembayaran Pasien

Use case tambah form pembayaran pasien akan diterangkan pada Tabel 4.27.

Tabel 4.27 *Use case* Tambah Form Pembayaran Pasien

Use case name	Tambah Form Pembayaran Pasien
Actor	<i>Chasier</i>
Description	Aktor menambah form pembayaran pasien yang berisi tentang biaya tindakan dan obat
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah menambah form pembayaran pasien
Flow of Events	<ol style="list-style-type: none"> 1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman form pembayaran pasien 3. Aktor mengisi form pembayaran pasien
Alternate course	

18.) *Use case* Tambah Form *Invoice* Pasien

Use case tambah form *invoice* pasien akan diterangkan pada Tabel 4.28.

Tabel 4.28 *Use case* Tambah Form *Invoice* Pasien

Use case name	Tambah Form <i>Invoice</i> Pasien
Actor	<i>Chasier</i>
Description	Aktor mengisi jumlah uang yang dibayarkan pasien setelah pasien diperiksa oleh tenaga medis
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor mengisi jumlah uang yang dibayarkan pasien
Flow of Events	<ol style="list-style-type: none"> 1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman <i>invoice</i> pasien 3. Aktor mengisi jumlah uang yang dibayarkan pasien
Alternate course	

19.) *Use case* Cetak *Invoice* Pasien

Use case cetak *invoice* pasien akan diterangkan pada Tabel 4.29.

Tabel 4.29 *Use case* Cetak *Invoice* Pasien

Use case name	Cetak <i>Invoice</i> Pasien
Actor	<i>Chasier</i>
Description	Aktor mencetak <i>invoice</i> pasien
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah mencetak <i>invoice</i> pasien
Flow of Events	<ol style="list-style-type: none"> 1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman <i>invoice</i> pasien 3. Aktor mencetak <i>invoice</i> pasien
Alternate course	

20.) *Use case* Menampilkan Data *Warehouse* Obat

Use case menampilkan data *warehouse* obat akan diterangkan pada Tabel 4.30.

Tabel 4.30 *Use case* Menampilkan Data Warehouse Obat

Use case name	Menampilkan Data Warehouse Obat
Actor	<i>Pharmacy</i>
Description	Aktor menampilkan data <i>warehouse</i> obat
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah menampilkan data <i>warehouse</i> obat
Flow of Events	<ol style="list-style-type: none"> 1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman data obat 3. Aktor menampilkan data <i>warehouse</i> obat
Alternate course	

21.) *Use case* Edit Data Obat

Use case edit data obat akan diterangkan pada Tabel 4.31.

Tabel 4.31 *Use case* Edit Data Obat

Use case name	Edit Data Obat
Actor	<i>Pharmacy</i>
Description	Aktor melakukan edit data obat
Pre-codition	<ol style="list-style-type: none"> 1. Aktor telah login ke dalam sistem 2. Data obat tersedia
Post condition	Aktor telah melakukan edit data obat
Flow of Events	<ol style="list-style-type: none"> 1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman detail obat 3. Aktor mengubah data obat
Alternate course	

22.) *Use case* Menampilkan Transaksi Obat Pasien

Use case menampilkan transaksi obat pasien akan diterangkan pada Tabel 4.32.

Tabel 4.32 *Use case* Tambah Rekam Medis Pasien

Use case name	Menampilkan Transaksi Obat Pasien
Actor	<i>Pharmacy</i>

Description	Aktor menampilkan transaksi obat pasien
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah menampilkan transaksi obat pasien
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman transaksi obat pasien
Alternate course	

23.) *Use case* Validasi Obat Pasien

Use case validasi obat pasien akan diterangkan pada Tabel 4.33.

Tabel 4.33 *Use case* Tambah Rekam Medis Pasien

Use case name	Validasi Obat Pasien
Actor	<i>Pharmacy</i>
Description	Aktor melakukan validasi obat pasien
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah validasi obat pasien
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman transaksi obat 3. Aktor mengurangi stok obat sesuai dengan jumlah di transaksi obat pasien
Alternate course	

24.) *Use case* Menampilkan Data Sumber Dana

Use case menampilkan data sumber dana akan diterangkan pada Tabel 4.34.

Tabel 4.34 *Use case* Menampilkan Data Sumber Dana

Use case name	Menampilkan Data Sumber Dana
Actor	<i>Financial</i>
Description	Aktor menampilkan data sumber dana
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah menampilkan data sumber dana
Flow of Events	1. Aktor masuk ke dalam sistem

	2. Aktor membuka halaman sumber dana 3. Aktor menampilkan data sumber dana berdasarkan waktu
Alternate course	

25.) *Use case* Menampilkan Pemasukan Pasien

Use case menampilkan pemasukan pasien akan diterangkan pada Tabel 4.35.

Tabel 4.35 *Use case* Menampilkan Pemasukan Pasien

Use case name	Menampilkan Pemasukan Pasien
Actor	<i>Financial</i>
Description	Aktor menampilkan pemasukan pasien secara keseluruhan
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah menampilkan pemasukan pasien
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman pemasukan pasien 3. Aktor menampilkan pemasukan pasien berdasarkan waktu
Alternate course	

26.) *Use case* Menampilkan Pengeluaran Keseluruhan

Use case menampilkan pengeluaran keseluruhan akan diterangkan pada Tabel 4.36.

Tabel 4.36 *Use case* Menampilkan Pengeluaran Keseluruhan

Use case name	Menampilkan Pengeluaran Keseluruhan
Actor	<i>Financial</i>
Description	Aktor menampilkan pengeluaran keseluruhan
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah menampilkan pengeluaran keseluruhan
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman pengeluaran

	3. Aktor menampilkan pengeluaran berdasarkan waktu
Alternate course	

27.) *Use case* Menampilkan Pemasukan Dan Pengeluaran Dalam Bentuk *Chart*
Use case menampilkan pemasukan dan pengeluaran dalam bentuk *chart* akan diterangkan pada Tabel 4.37.

Tabel 4.37 *Use case* Menampilkan Pemasukan dan Pengeluaran dalam Bentuk *Chart*

Use case name	Menampilkan Pemasukan Dan Pengeluaran Dalam Bentuk <i>Chart</i>
Actor	<i>Financial</i>
Description	Aktor menampilkan pemasukan dan pengeluaran dalam bentuk <i>chart</i>
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah menampilkan pemasukan dan pengeluaran dalam bentuk <i>chart</i>
Flow of Events	<ol style="list-style-type: none"> 1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman pemasukan dan pengeluaran 3. Aktor mengekspor pemasukan dan pengeluaran dalam bentuk <i>chart</i>
Alternate course	

28.) *Use case* Menampilkan Daftar Riwayat Medis Pasien
Use case menampilkan daftar riwayat medis pasien akan diterangkan pada Tabel 4.38.

Tabel 4.38 *Use case* Menampilkan Daftar Riwayat Medis Pasien

Use case name	Menampilkan Daftar Riwayat Medis Pasien
Actor	<i>Medicare</i>
Description	Aktor menampilkan daftar riwayat medis pasien
Pre-codition	Aktor telah login ke dalam sistem

Post condition	Aktor telah menampilkan daftar riwayat medis pasien
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka daftar riwayat medis pasien
Alternate course	

29.) *Use case* Menampilkan Biodata Pasien

Use case menampilkan biodata pasien akan diterangkan pada Tabel 4.39.

Tabel 4.39 *Use case* Menampilkan Biodata Pasien

Use case name	Menampilkan Biodata Pasien
Actor	<i>Medicare</i>
Description	Aktor menampilkan biodata pasien
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah menampilkan biodata pasien
Flow of Events	1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman pasien 3. Aktor menampilkan biodata pasien
Alternate course	

30.) *Use case* Menampilkan Seluruh Rekam Medis Pasien Tertentu

Use case menampilkan seluruh rekam medis pasien tertentu akan diterangkan pada Tabel 4.40.

Tabel 4.40 *Use case* Menampilkan Seluruh Rekam Medis Pasien Tertentu

Use case name	Menampilkan Seluruh Rekam Medis Pasien Tertentu
Actor	<i>Medicare</i>
Description	Aktor menampilkan seluruh rekam medis pasien tertentu
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah menampilkan seluruh rekam medis pasien tertentu
Flow of Events	1. Aktor masuk ke dalam sistem

	<ol style="list-style-type: none"> 2. Aktor membuka halaman rekam medis pasien 3. Aktor menampilkan seluruh rekam medis pasien tertentu
Alternate course	

31.) *Use case* Menampilkan Rekam Medis Pasien Tertentu

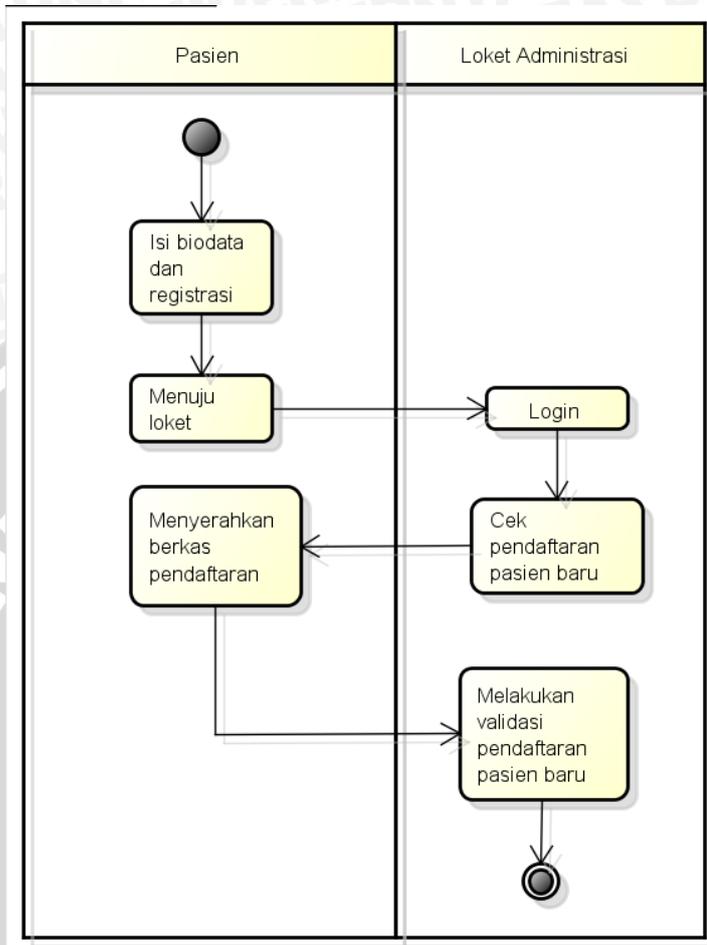
Use case menampilkan rekam medis pasien tertentu akan diterangkan pada Tabel 4.41.

Tabel 4.41 *Use case* Menampilkan Rekam Medis Pasien Tertentu

Use case name	Menampilkan Rekam Medis Pasien Tertentu
Actor	<i>Medicare</i>
Description	Aktor menampilkan rekam medis pasien tertentu
Pre-codition	Aktor telah login ke dalam sistem
Post condition	Aktor telah menampilkan rekam medis pasien tertentu
Flow of Events	<ol style="list-style-type: none"> 1. Aktor masuk ke dalam sistem 2. Aktor membuka halaman rekam medis pasien 3. Aktor menampilkan rekam medis pasien tertentu
Alternate course	

4.3.3 Pemodelan Sistem dengan Activity Diagram

1. Registrasi Pasien



Gambar 4.12 Model Sistem Registrasi Pasien

Sumber: Perancangan

Model sistem registrasi pasien pada gambar 4.12 di atas dapat diperjelas dengan runtutan kegiatan seperti tabel 4.42 berikut:

Tabel 4.42 Tabel Skenario Sistem Registrasi Pasien

<p><i>Flow</i> Sistem Registrasi Pasien</p> <ol style="list-style-type: none"> 1. <i>Basic Flow</i> <ol style="list-style-type: none"> 1.1 Isi biodata dan registrasi Pasien melakukan registrasi dengan cara melakukan input biodata pada form yang telah ditampilkan 1.2 Menuju Front Office
--

Setelah pasien melakukan registrasi Pasien menuju Front Office administrasi

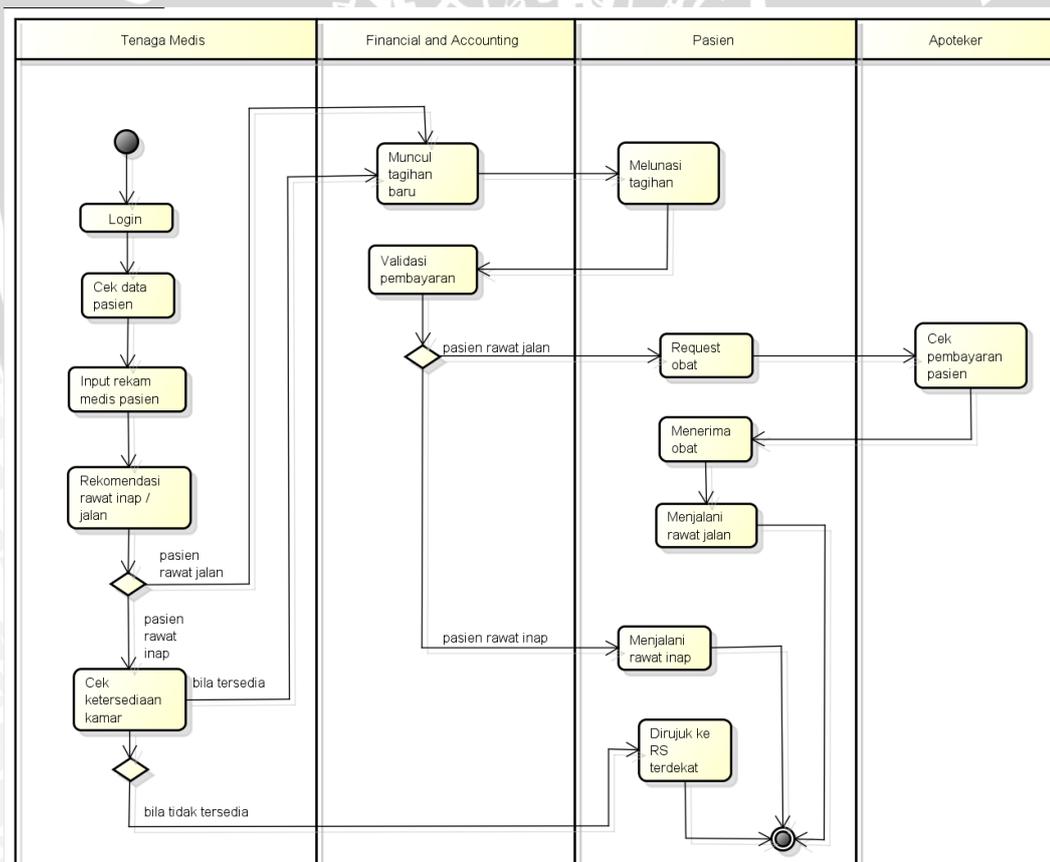
1.3 Login
Bagian Front Office administrasi melakukan login

1.4 Cek pendaftaran pasien baru
Bagian Front Office administrasi melakukan cek pendaftaran pasien baru

1.5 Menyerahkan berkas pendaftaran
Pasien menuju Front Office dengan menyerahkan berkas pendaftaran

1.6 Melakukan validasi pendaftaran pasien baru
Bagian Front Office administrasi melakukan validasi dan penambahan pasien baru

2. Rawat Pasien



Gambar 4.13 Model Sistem Rawat Pasien

Sumber: Perancangan

Model sistem rawat pasien pada gambar 4.13 di atas dapat diperjelas dengan runtutan kegiatan seperti tabel 4.43 berikut:

Tabel 4.43 Tabel Skenario Sistem Rawat Pasien

Flow Sistem Rawat Pasien

1. *Basic Flow*

1.1 Login

Tenaga medis melakukan login ke sistem terlebih dahulu

1.2 Cek data pasien

Tenaga medis melakukan cek data pasien yang akan diperiksa. Data pasien meliputi data rekam medis

1.3 Input rekam medis pasien

Proses ini dilakukan untuk memasukkan data rekam medis baru setelah menangani pemeriksaan dan diagnosa

1.4 Rekomendasi rawat inap / jalan

Tenaga medis merekomendasikan untuk rawat inap atau rawat jalan

1.5 Cek ketersediaan kamar

Bila pasien dimungkinkan untuk menjalani rawat inap maka dokter langsung bisa cek ketersediaan ruangan

1.6 Dirujuk ke RS terdekat

Proses ini dilakukan setelah tenaga medis melakukan cek ketersediaan kamar. Bila ruangan tidak tersedia maka dokter dapat merekomendasikan agar pasien dirujuk ke rumah sakit terdekat

1.7 Muncul tagihan baru

Muncul tagihan baru di modul *financial and accounting*

1.8 Melunasi tagihan

Pasien diharuskan melunasi tagihan

1.9 Validasi pembayaran

Setelah pasien melunasi tagihan, bagian *financial and accounting* melakukan validasi pembayaran pasien

1.10 Request obat

Bila pasien rawat jalan, maka proses ini dilakukan untuk meminta obat sesuai resep dokter

1.11 Cek pembayaran pasien

Apoteker melakukan pengecekan terhadap pembayaran pasien

1.12 Menerima obat

Pasien menerima obat dari apoteker

1.13 Menjalani rawat jalan

Pasien menjalani rawat jalan

1.14 Menjalani rawat inap

Pasien menjalani rawat inap

2. *Alternative Flow*

2.1 AT BF Rekomendasi rawat inap / jalan. Bila pasien direkomendasikan untuk melakukan rawat jalan maka akan muncul tagihan baru pada modul *financial and accounting* sesuai hasil diagnosa (tagihan pemeriksaan dan resep dokter)

2.2 AT BF Rekomendasi rawat inap / jalan. Bila pasien direkomendasikan untuk melakukan rawat inap maka bagian

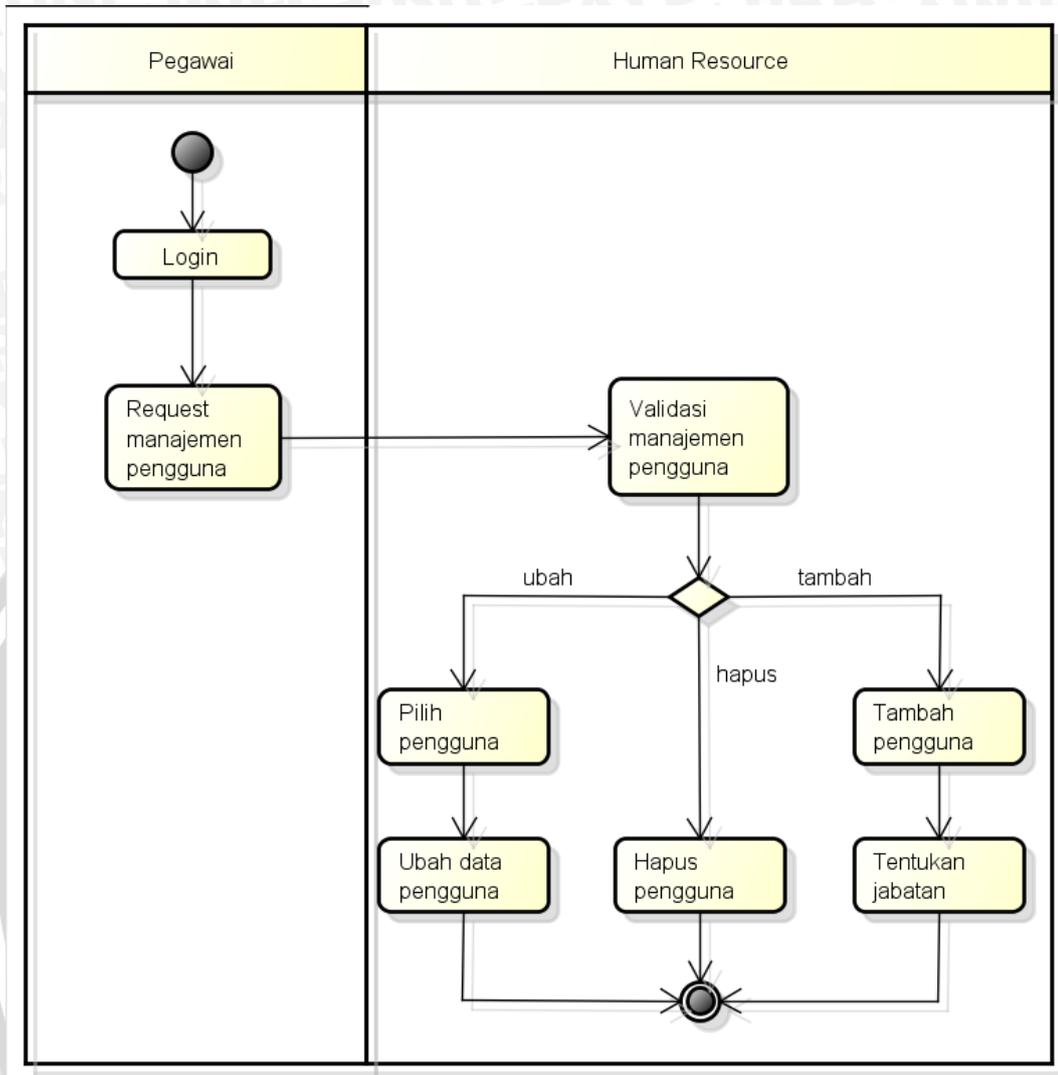
2.3 AT BF Cek ketersediaan kamar. Bila kamar tersedia maka akan muncul tagihan baru pada modul *financial and accounting* sesuai dengan kebutuhan pasien

2.4 AT BF Cek ketersediaan kamar. Bila kamar tidak tersedia maka tenaga medis akan merekomendasikan pasien untuk melakukan rujukan ke rumah sakit terdekat

2.5 AT BF Validasi pembayaran. Bila pasien rawat jalan maka setelah validasi pembayaran akan langsung melakukan pengambilan obat

2.6 AT BF Validasi pembayaran. Bila pasien rawat inap maka setelah validasi pembayaran akan langsung menjalani rawat inap

3. Pengaturan Pengguna



Gambar 4.14 Model Sistem Pengaturan Pengguna

Sumber: Perancangan

Model sistem pengaturan pengguna pada Gambar 4.14 di atas dapat diperjelas dengan runtutan kegiatan seperti tabel 4.44 berikut:

Tabel 4.44 Tabel Skenario Sistem Pengaturan Pengguna

Flow Sistem Pengaturan Pengguna	
1.	<i>Basic Flow</i>
1.1	Login
	Bagian HRD melakukan login
1.2	Request manajemen pengguna
	Bagian HRD melakukan <i>request</i> manajemen pengguna

1.3 Validasi manajemen pengguna

Bagian *Human Resources* melakukan validasi manajemen pengguna

1.4 Pilih pengguna

Bila pegawai melakukan pengubahan jabatan pengguna, maka *Human Resources* akan melakukan validasi untuk mengubah jabatan

1.5 Tambah pengguna

Bila *Human Resources* melakukan tambah pengguna, maka *Human Resources* akan melakukan validasi untuk menambah pengguna

1.6 Ubah data pengguna

Ubah jabatan adalah proses setelah hasil request pegawai (ubah jabatan pengguna) divalidasi oleh *Human Resources*

1.7 Hapus pengguna

Hapus pengguna adalah proses setelah hasil request pegawai (hapus pengguna) divalidasi oleh *Human Resources*

1.8 Tentukan jabatan

Tentukan jabatan adalah proses setelah hasil request pegawai (tambah pengguna) divalidasi oleh *Human Resources*

2. *Alternative Flow*

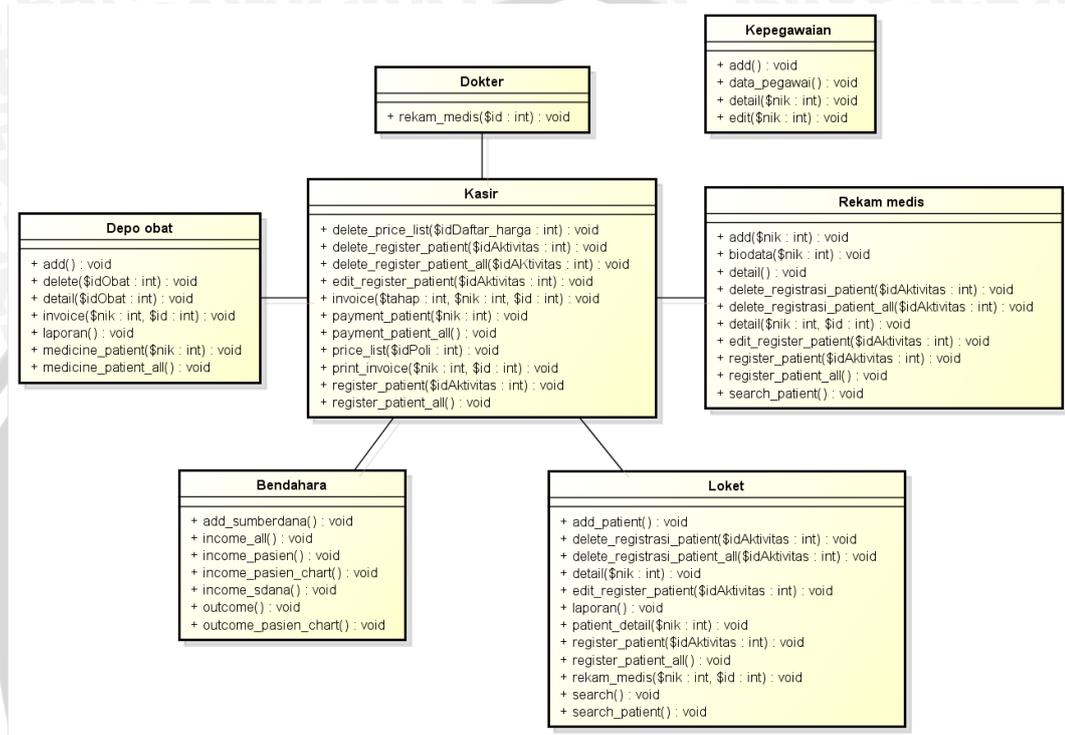
2.1 AT BF Validasi manajemen pengguna. Bila telah divalidasi manajemen pengguna untuk menguubah jabatan, maka *Human Resources* melakukan ubah jabatan

2.2 AT BF Validasi manajemen pengguna. Bila telah divalidasi manajemen pengguna untuk menghapus pengguna, maka *Human Resources* melakukan hapus pengguna

2.3 AT BF Validasi manajemen pengguna. Bila telah divalidasi manajemen pengguna untuk menambah pengguna, maka *Human Resources* melakukan tambah pengguna

4.3.4 Pemodelan Sistem dengan Class Diagram

Class diagram pada sistem menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem enterprise rumah sakit menggunakan design pattern MVC. Berikut adalah class diagram dari sistem ini terdapat pada Gambar 4.15.

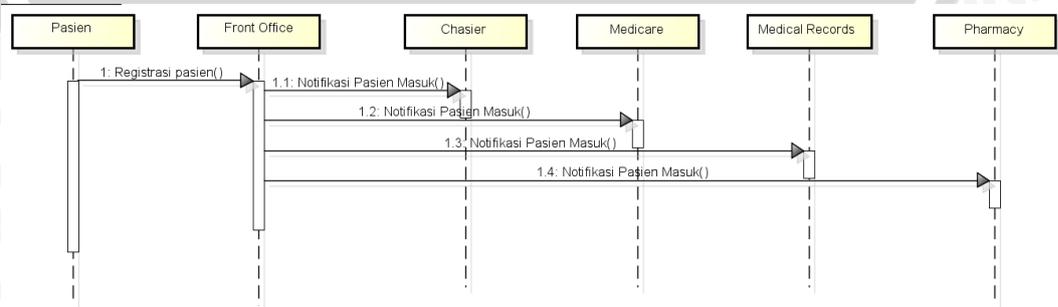


Gambar 4. 15 Class Diagram

4.3.5 Pemodelan Sistem dengan Sequence Diagram

1. Registrasi Pasien

Pemodelan sistem dengan sequence diagram pada proses registrasi pasien dapat digambarkan pada Gambar 4.16.



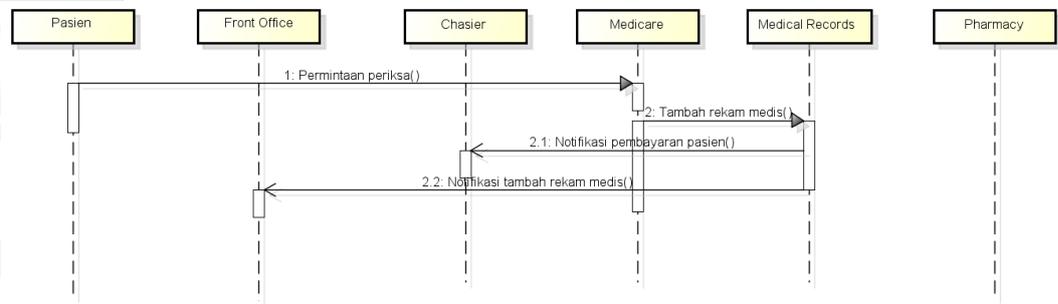
Gambar 4.16 Sequence Registrasi Pasien

Sumber: Perancangan



2. Pemeriksaan Pasien

Pemodelan sistem dengan sequence diagram pada proses pemeriksaan pasien dapat digambarkan pada Gambar 4.17.

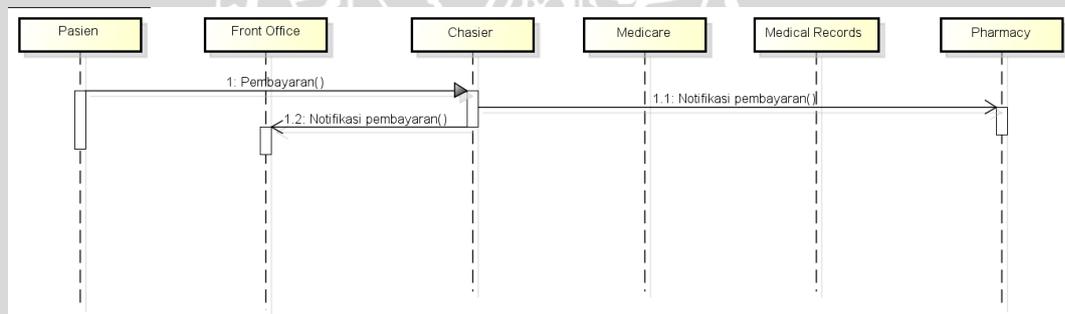


Gambar 4.17 Sequence Rawat Pasien

Sumber: Perancangan

3. Pembayaran pasien

Pemodelan sistem dengan sequence diagram pada proses pembayaran pasien dapat digambarkan pada Gambar 4.18.

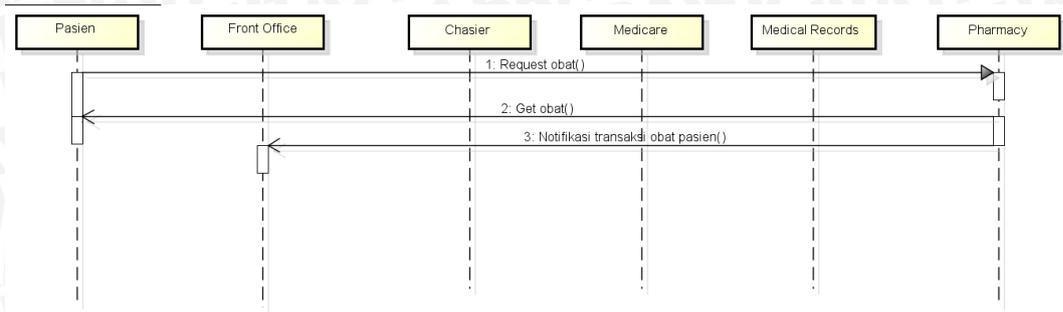


Gambar 4.18 Sequence Pembayaran Pasien

Sumber: Perancangan

4. Transaksi obat

Pemodelan sistem dengan sequence diagram pada proses transaksi obat pasien dapat digambarkan pada Gambar 4.19.

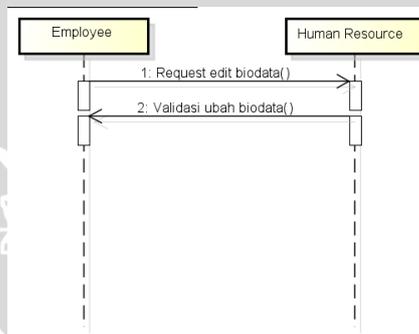


Gambar 4.19 Sequence Pembayaran Pasien

Sumber: Perancangan

5. Ubah data pengguna

Pemodelan sistem dengan sequence diagram pada proses ubah data pengguna (pegawai) dapat digambarkan pada Gambar 4.20.



Gambar 4.20 Sequence Ubah Data Pengguna

Sumber: Perancangan

4.4 Rancang Bangun Aplikasi

4.4.1 Human Resource

Gambar dibawah ini merupakan gambar dari antar muka Human Resource pada Sistem Enterprise Rumah Sakit menggunakan Design Pattern MVC.

The screenshot shows the 'Main' page of the Human Resource system. It includes a navigation sidebar with 'Human Resources' and sub-items like 'Data Kepegawaian' and 'Tambah Pegawai'. The main content area displays a welcome message and a 'Menu Panel' with 'Informasi Akun' selected. The account information is as follows:

Informasi	Data Anda
Nama	Afiqie Fadhihansah
NIK	3571031607910006
Instansi	Universitas Brawijaya
Email	afiqie.fadhihansah@yahoo.com
Jenis pegawai	Kepegawaian
Status Layanan	Aktif

Gambar 4.21 Antarmuka Human Resource: Home

Sumber: Implementasi

Fitur yang terdapat pada Human Resource berikut adalah lihat, edit dan tambah data kepegawaian terdapat pada Gambar 4.22.

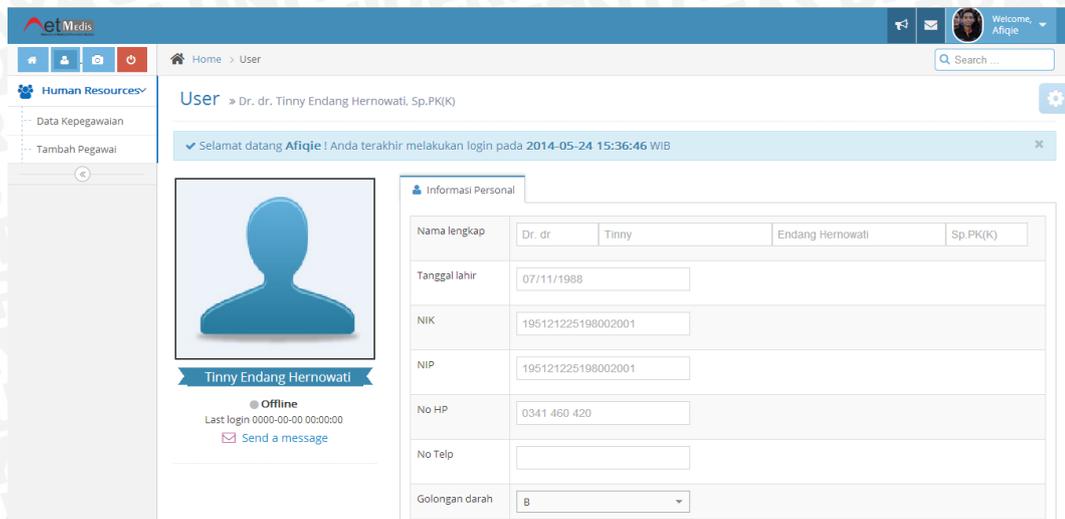
The screenshot shows the 'Data Pegawai' page. It features a search bar and a table of employee records. The table has the following columns: Jenis Pegawai, Nama Lengkap, Bagian, Login terakhir, Status, and Opsi. The data is as follows:

Jenis Pegawai	Nama Lengkap	Bagian	Login terakhir	Status	Opsi
Apoteker	Billy Novanta Yudistira	Pharmacy	Online	Aktif	[Cari] [Edit]
Bendahara	Syahri Dwi Andarwati, S.KM	Financial	2014-05-14 13:43:02	Aktif	[Cari] [Edit]
Dokter	Dr. dr. Timmy Endang Hernowati, Sp.PK(K)	Medicare	0000-00-00 00:00:00	Aktif	[Cari] [Edit]
Dokter	dr. Luqman Hakim Hidayat	Medicare	0000-00-00 00:00:00	Aktif	[Cari] [Edit]
Dokter	dr. Billy Novanta Yudistira	Medicare	2014-05-16 21:19:55	Tidak Aktif	[Cari] [Edit]
Dokter	dr. Durotul Ikrimah	Medicare	2014-05-16 20:49:21	Tidak Aktif	[Cari] [Edit]
Dokter	dr. Winda Sabtikarisa, S. Ked	Medicare	0000-00-00 00:00:00	Tidak Aktif	[Cari] [Edit]
Dokter	dr. Wachidah Wahyuningsih	Medicare	2014-05-17 06:26:56	Tidak Aktif	[Cari] [Edit]

Gambar 4.22 Antarmuka Human Resource: Data Kepegawaian

Sumber: Implementasi

Fitur yang terdapat pada Human Resource berikut adalah edit data kepegawaian terdapat pada Gambar 4.23.

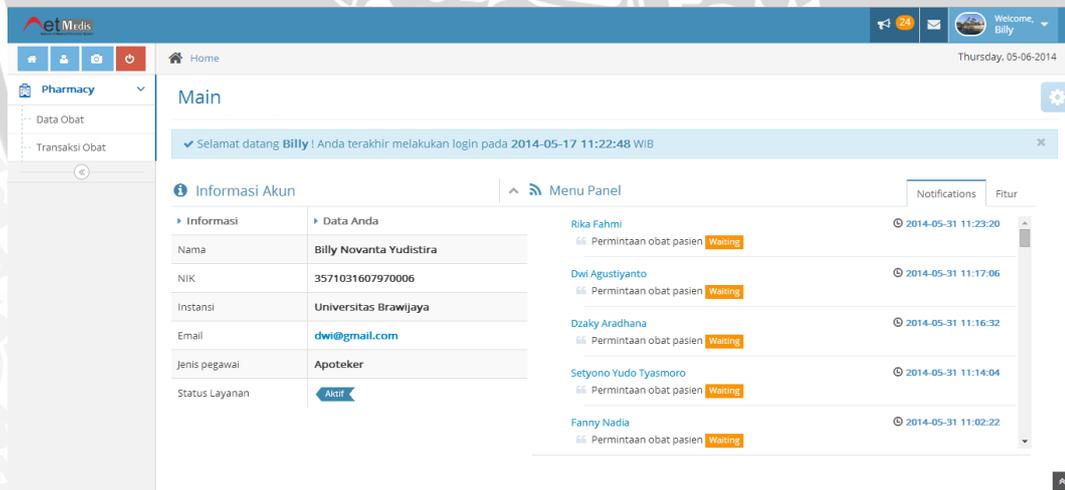


Gambar 4.23 Antarmuka Human Resource: Edit Data Kepegawaian

Sumber: Implementasi

4.4.2 Pharmacy

Gambar dibawah ini merupakan gambar dari antar muka Pharmacy pada Sistem *Enterprise* Rumah Sakit menggunakan *Design Pattern* MVC.



Gambar 4.24 Antarmuka Pharmacy: Home

Sumber: Implementasi

Fitur yang terdapat pada Pharmacy berikut adalah menampilkan data obat terdapat pada Gambar 4.24.



Nama	Golongan	Satuan	Jumlah	Harga	Expired	Opsi
Abocat No. 18		Pcs	10	0	0000-00-00	[Search] [Edit]
Abocat No. 20		Pcs	10	0	0000-00-00	[Search] [Edit]
Abocat No. 22		Pcs	10	0	0000-00-00	[Search] [Edit]
Acetosol 80mg		Tab	10	0	0000-00-00	[Search] [Edit]
Acifar 400mg		Tab	300	0	0000-00-00	[Search] [Edit]
Acyclovir 400mg		Tab	500	0	0000-00-00	[Search] [Edit]
Acyclovir 5K		Fls	10	3000	0000-00-00	[Search] [Edit]
Acyclovir zalf		Tube	5	0	0000-00-00	[Search] [Edit]

Gambar 4.25 Antarmuka Pharmacy: Data Obat

Sumber: Implementasi

Fitur yang terdapat pada Pharmacy berikut adalah menampilkan data transaksi obat terdapat pada Gambar 4.25.

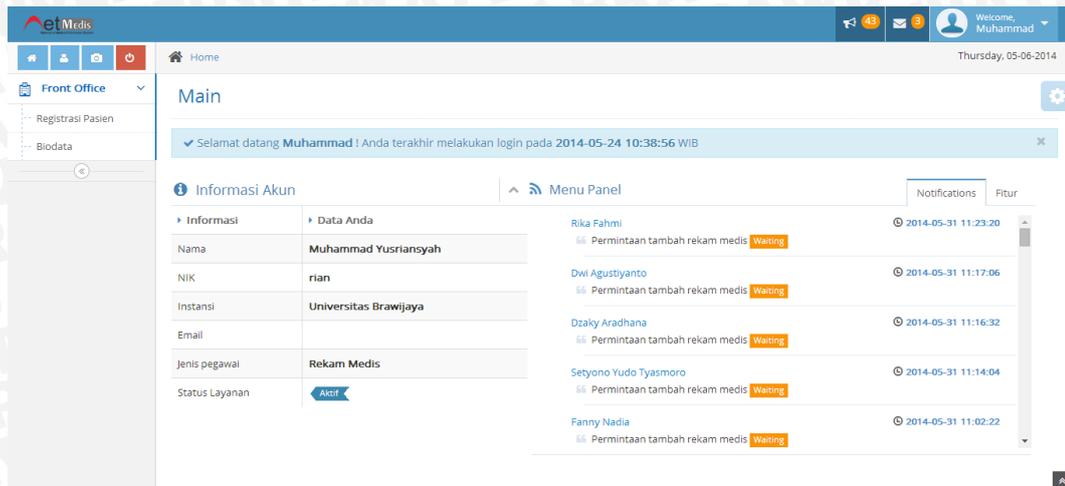
No	Waktu	Nama Pasien	NIK	Payment	Medicine	Apoteker	Opsi
1	2014-05-31 08:10:32	Arfiq Isyah R	04890611	Done	Warning	null	[Detail]

Gambar 4.26 Antarmuka Pharmacy: Data Transaksi Obat

Sumber: Implementasi

4.4.3 Medical Records

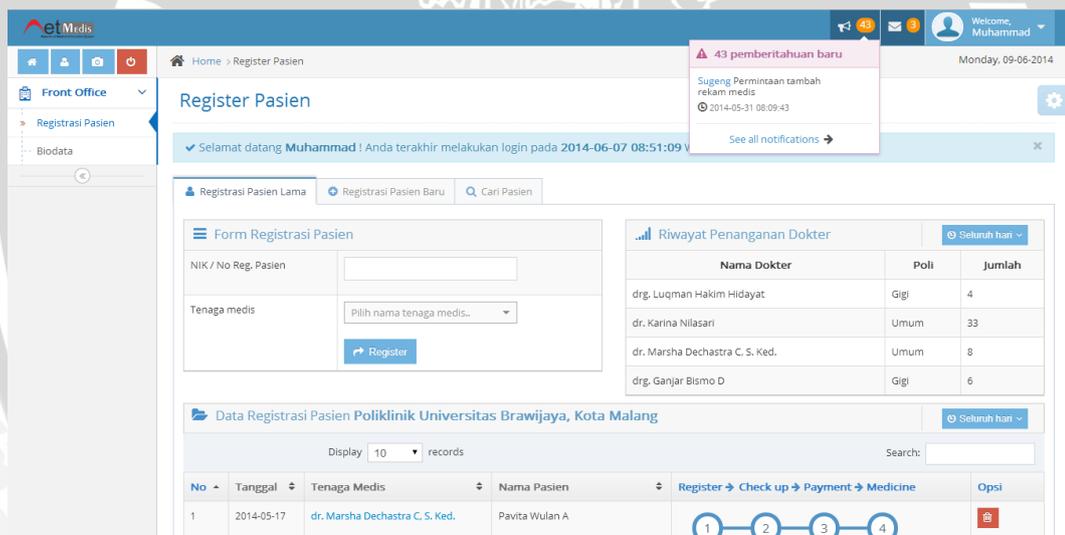
Gambar dibawah ini merupakan gambar dari antar muka Medical Records pada Sistem *Enterprise* Rumah Sakit menggunakan *Design Pattern* MVC terdapat pada Gambar 4.27.



Gambar 4.27 Antarmuka Medical Records: Home

Sumber: Implementasi

Fitur yang terdapat pada Medical Records berikut adalah melakukan register pasien terdapat pada Gambar 4.28.



Gambar 4.28 Antarmuka Medical Records: Register Pasien

Sumber: Implementasi

Fitur yang terdapat pada Medical Records berikut adalah menampilkan data pasien terdapat pada Gambar 4.29.



etMedis Home > Biodata Pasien Thursday, 05-06-2014

Front Office

- Registrasi Pasien
- Biodata

Biodata Pasien

✓ Selamat datang **Muhammad** ! Anda terakhir melakukan login pada 2014-05-24 10:38:56 WIB

Biodata Pasien yang Belum Lengkap

Display 10 records Search:

Jenis Pasien	Nama	No Rekam Medis	Opsl
Honorer	Henry Kurniawati	01580514	Edit
Honorer	Halifatuz Zar Zarah	01590514	Edit
Honorer	M. Syamsul Arif H	01600514	Edit
Honorer	Budi Wicaksono	FK00170413	Edit
Honorer	Anna Widyahsih	07600713	Edit
Mahasiswa	Romdhan Ubaidillah Ubaidillah	3571031607920007	Edit
Mahasiswa	Ubaidillah Muhamad Romdan	20000514	Edit
Mahasiswa	Arfiq Isyah R	04890611	Edit

Gambar 4.29 Antarmuka Medical Records: Biodata Pasien

Sumber: Implementasi

4.4.4 Chasier

Gambar dibawah ini merupakan gambar dari antar muka Chasier pada Sistem Enterprise Rumah Sakit menggunakan Design Pattern MVC terdapat pada Gambar 4.30.

etMedis Home Thursday, 05-06-2014

Financial

- Registrasi Pasien
- Pembayaran Pasien
- Daftar Harga

Main

✓ Selamat datang **SITI** ! Anda terakhir melakukan login pada 2014-05-31 10:51:31 WIB

Informasi Akun Menu Panel Notifications Fitur

Informasi Data Anda

Nama	Siti Rochmah
NIK	3571031607940006
Instansi	Universitas Brawijaya
Email	siti.rochmah@yahoo.com
Jenis pegawai	Kasir
Status Layanan	Aktif

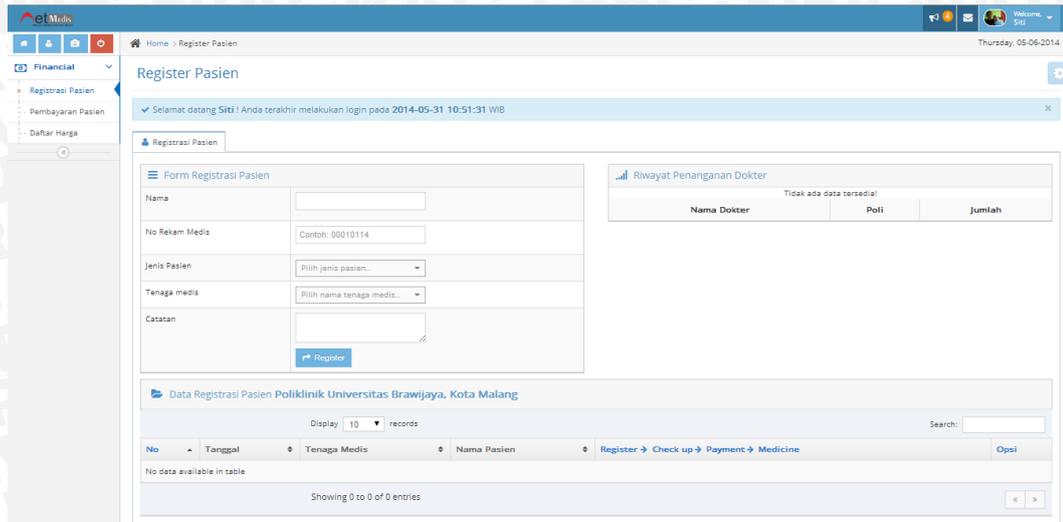
Notifications

Rika Fahmi	Permintaan pembayaran pasien	Waiting	2014-05-31 11:23:20
Dwi Agustiyanto	Permintaan pembayaran pasien	Done	2014-05-31 11:39:02
Dzaky Aradhana	Permintaan pembayaran pasien	Done	2014-05-31 11:35:56
Setyono Yudo Tyasmoro	Permintaan pembayaran pasien	Done	2014-05-31 11:37:10
Fanny Nadia	Permintaan pembayaran pasien	Done	2014-05-31 11:26:17

Gambar 4.30 Antarmuka Chasier: Home

Sumber: Implementasi

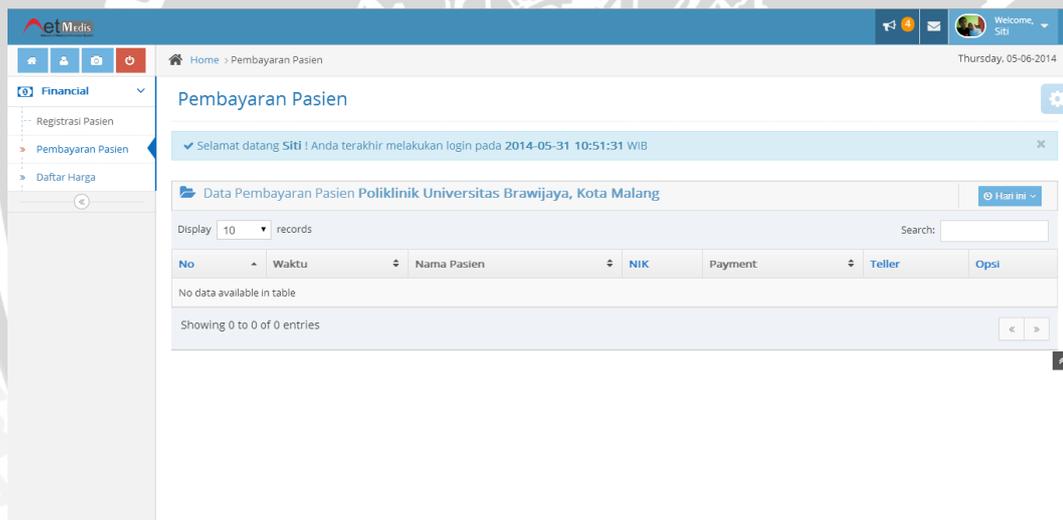
Fitur yang terdapat pada Chasier berikut adalah melakukan register pasien terdapat pada Gambar 4.31.



Gambar 4.31 Antarmuka Chasier: Register Pasien

Sumber: Implementasi

Fitur yang terdapat pada Chasier berikut adalah melakukan daftar pembayaran pasien terdapat pada Gambar 4.32.

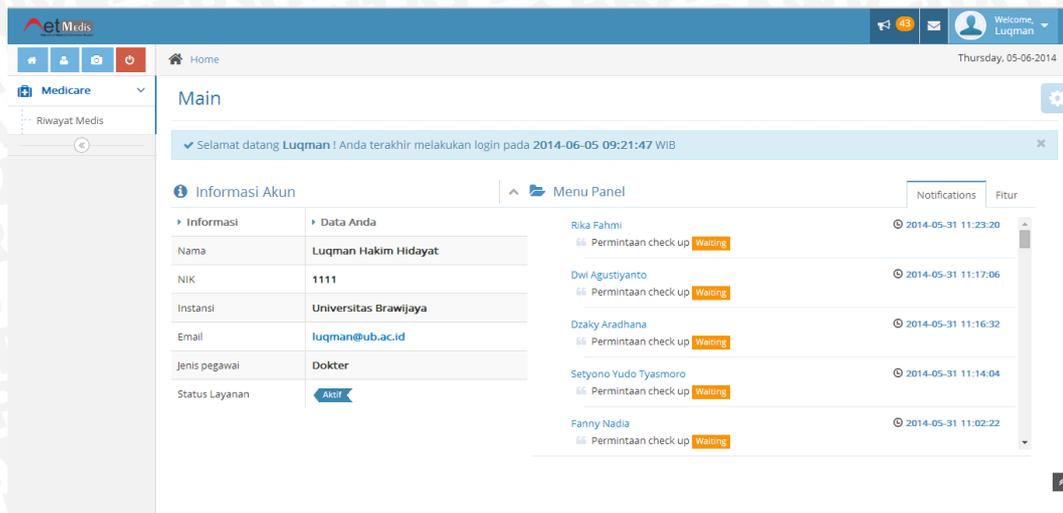


Gambar 4.32 Antarmuka Chasier: Daftar Pembayaran Pasien

Sumber: Implementasi

4.4.5 Medicare

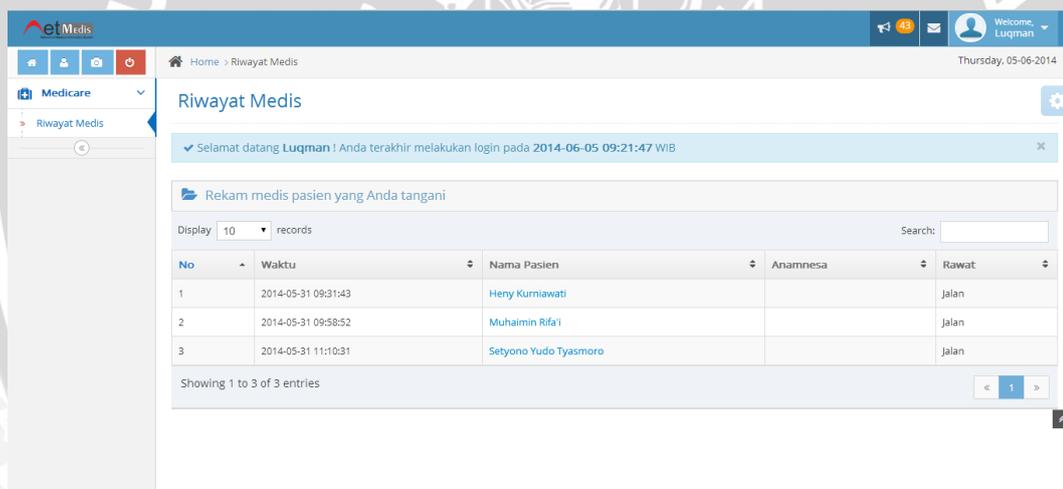
Gambar dibawah ini merupakan gambar dari antar muka Medicare pada Sistem Enterprise Rumah Sakit menggunakan Design Pattern MVC terdapat pada Gambar 4.33.



Gambar 4.33 Antarmuka Medicare: Home

Sumber: Implementasi

Fitur yang terdapat pada Medicare berikut adalah menampilkan daftar riwayat pasien terdapat pada Gambar 4.34.

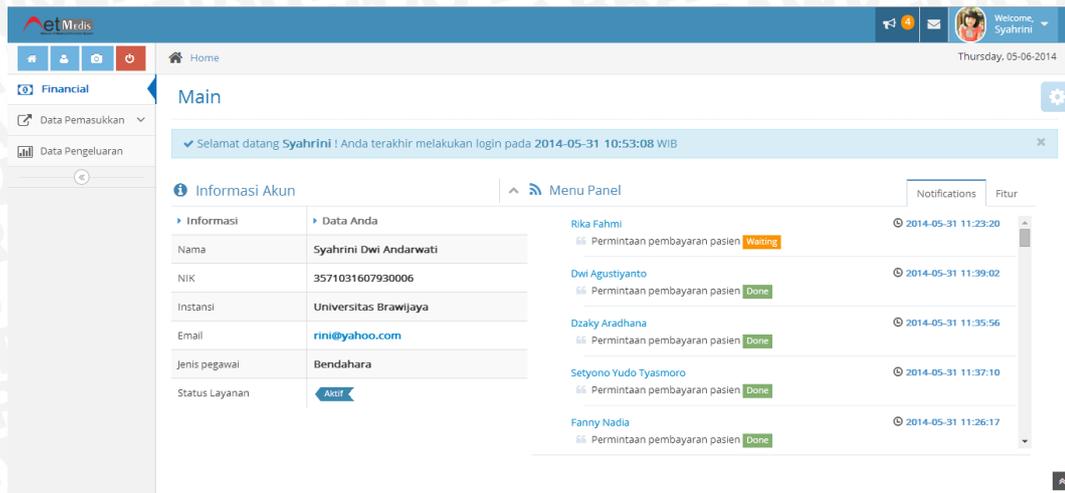


Gambar 4.34 Antarmuka Medicare: Daftar Riwayat Medis Pasien

Sumber: Implementasi

4.4.6 Financial

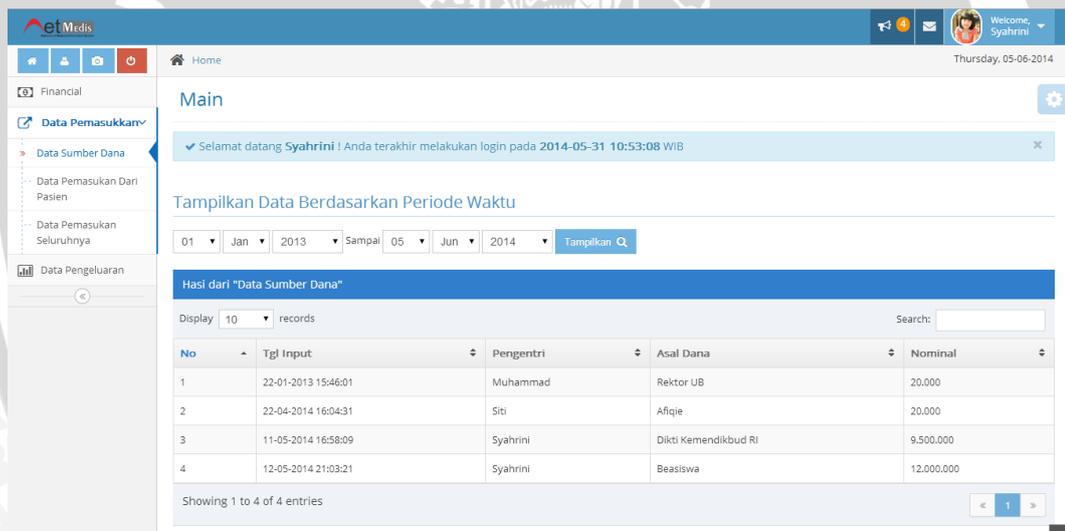
Gambar dibawah ini merupakan gambar dari antar muka Financial pada Sistem Enterprise Rumah Sakit menggunakan Design Pattern MVC terdapat pada Gambar 4.35.



Gambar 4.35 Antarmuka Financial: Home

Sumber: Implementasi

Fitur yang terdapat pada Financial berikut adalah menampilkan sumber dana terdapat pada Gambar 4.36.



Gambar 4.36 Antarmuka Financial: Sumber Dana

Sumber: Implementasi

Fitur yang terdapat pada Financial berikut adalah menampilkan data pengeluaran terdapat pada Gambar 4.37.



etMedis

Home

Financial

Data Pemasukkan

Data Pengeluaran

Main

✓ Selamat datang Syahrini ! Anda terakhir melakukan login pada 2014-05-31 10:53:08 WIB

Tampilkan Data Berdasarkan Periode Waktu

01 Jan 2013 Sampai 05 Jun 2014

Hasi dari "Data Pengeluaran"

Display 10 records

No	Tanggal Input	Tanggal Nota	Penanggung Jawab	Pengentri	Subjek	Jumlah
1	2013-12-12 10:34:59	02-04-2014 10:31:45	Parjos	Billy	Pembelian Laptop Dell Kasir	80.000
2	2014-03-12 10:34:59	02-04-2014 10:31:45	Parjos	Afiqie	Pembelian Laptop Dell Kasir	78.000
3	2014-04-24 16:05:56	2014/04/24 16:05:26	Sudirman	Siti	Beli komputer	280.000

Showing 1 to 3 of 3 entries

Gambar 4.37 Antarmuka Financial: Data Pengeluaran

Sumber: Implementasi

4.4.7 Front Office

Gambar dibawah ini merupakan gambar dari antar muka Front Office pada Sistem Enterprise Rumah Sakit menggunakan Design Pattern MVC terdapat pada Gambar 4.38.

etMedis

Home

Front Office

Registrasi Pasien

Laporan

Main

✓ Selamat datang Muhammad ! Anda terakhir melakukan login pada 2014-05-31 07:45:47 WIB

Informasi Akun

Menu Panel

Notifications

Fitur

Informasi

Data Anda

✓ Tidak ada pemberitahuan

Nama	Muhammad Al-Fatih
NIK	3571031607950006
Instansi	Universitas Brawijaya
Email	corbuzier@yahoo.com
Jenis pegawai	Loket
Status Layanan	Aktif

Gambar 4.38 Antarmuka Front Office: Home

Sumber: Implementasi

Fitur yang terdapat pada Front Office berikut adalah menampilkan data registrasi pasien terdapat pada Gambar 4.39.

No	Tanggal	Tenaga Medis	Nama Pasien	Register → Check up → Payment → Medicine	Opsl
1	2014-05-17	dr. Marsha Dechastra C, S. Ked.	Pavita Wulan A	 [09:19:05] [09:52:52] [09:55:03] [10:00:08]	
2	2014-05-17	dr. Marsha Dechastra C, S. Ked.	Sakunda STP	 [09:47:08] [10:21:08] [10:21:30] [10:22:51]	
3	2014-05-17	drg. Ganjar Bismo D	Evian Eka Kusuma W	 [09:48:13] [10:07:10] [10:10:51] [09:48:13]	
4	2014-05-17	dr. Marsha Dechastra C. S. Ked.	Retno Kusuma N	 [09:52:56] [10:45:26] [10:47:03] [10:48:30]	

Gambar 4.39 Antarmuka Front Office: Registrasi Pasien

Sumber: Implementasi



BAB V

PENGUJIAN DAN ANALISIS

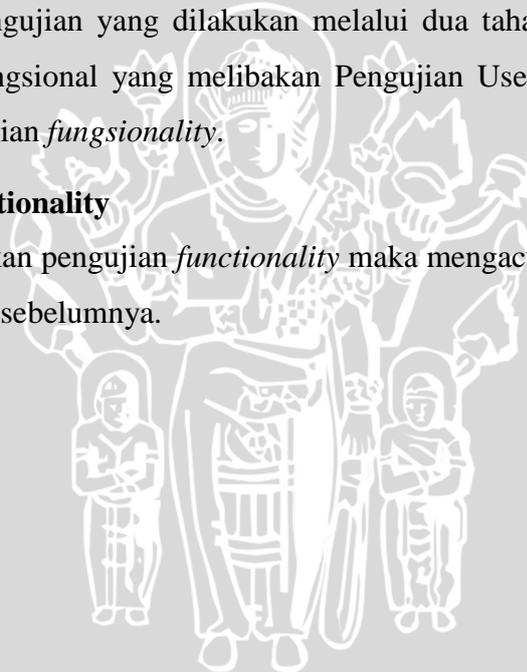
Bab ini membahas mengenai tahapan pengujian dan analisis perangkat lunak Sistem Enterprise Rumah Sakit menggunakan *design pattern* Model View Controller (MVC) yang penulis kembangkan. Proses pengujian melalui tiga tahapan pengujian, yaitu Pengujian User Acceptance Test (UAT) dan pengujian *functionality*. Pada Pengujian User Acceptance Test (UAT) dilakukan wawancara kepada staff dan pengguna sistem ini. Selain itu juga terdapat pengujian non-fungsionalitas yang terdapat dalam perancangan sebelumnya.

5.1 Pengujian

Proses pengujian yang dilakukan melalui dua tahap pengujian yaitu pengujian non-fungsional yang melibatkan Pengujian User Acceptance Test (UAT) dan pengujian *functionality*.

5.1.1 Pengujian Functionality

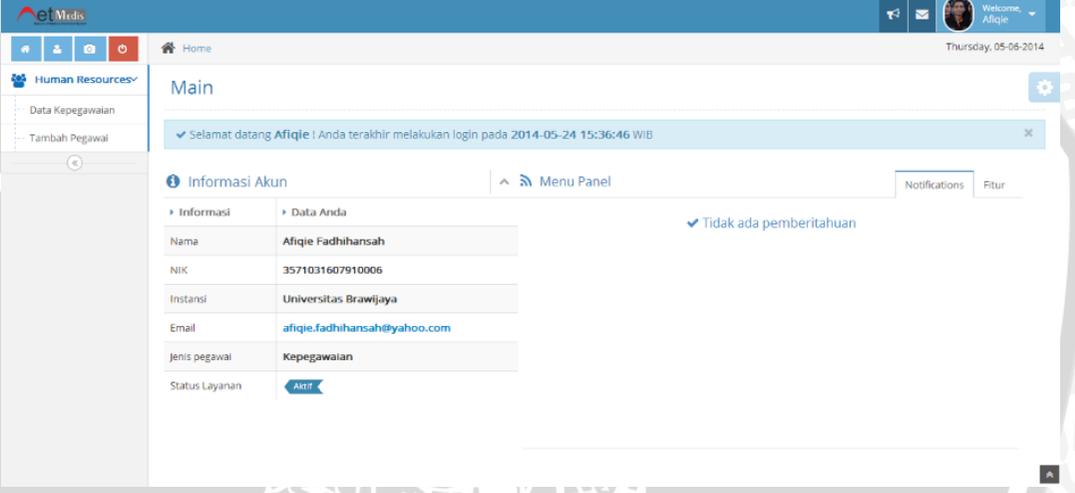
Untuk melakukan pengujian *functionality* maka mengacu kepada dokumen SRS yang telah dibuat sebelumnya.



5.1.1.1 Human Resource

Seperti pada tabel 5.1 menunjukkan fungsionalitas sistem sebagai Human Resource yang dapat mengakses modul.

Tabel 5.1 Human Resource

No SRS	Kebutuhan	Tampilan	Status
SRS_1_01	Sistem harus menyediakan login staf HRD	 <p>The screenshot shows the 'Main' page of the 'elMedis' system. It features a navigation menu on the left with 'Human Resources' selected. A central panel displays 'Informasi Akun' (Account Information) for the user 'Afiqie Fadhihansah'. The information includes: <ul style="list-style-type: none"> Nama: Afiqie Fadhihansah NIK: 3571031607910006 Instansi: Universitas Brawijaya Email: afiqie.fadhihansah@yahoo.com Jenis pegawai: Kepegawaian Status Layanan: Aktif A notification bar at the top indicates a successful login on 2014-05-24 at 15:36:46 WIB. </p>	Valid

SRS_1_02

Sistem harus menyediakan fitur menampilkan daftar data pegawai untuk staf kepegawaian

Data Pegawai Poliklinik Universitas Brawijaya, Kota Malang

Display 10 records Search:

Jenis Pegawai	Nama Lengkap	Bagian	Login terakhir	Status	Opsi
Apoteker	Billy Novanta Yudistira	Pharmacy	Online	Aktif	Cari Edit
Bendahara	Syahrini Dwi Andarwati, SKM	Financial	Online	Aktif	Cari Edit
Dokter	Dr. dr. Tinny Endang Hernowati, Sp.PK(K)	Medicare	0000-00-00 00:00:00	Aktif	Cari Edit
Dokter	drg. Luqman Hakim Hidayat	Medicare	Online	Aktif	Cari Edit
Dokter	dr. Billy Novanta Yudistira	Medicare	2014-05-16 21:19:55	Tidak Aktif	Cari Edit
Dokter	dr. Durrotul Ikrimah	Medicare	Online	Tidak Aktif	Cari Edit
Dokter	dr. Winda Sabtikarisa, S. Ked	Medicare	0000-00-00 00:00:00	Aktif	Cari Edit
Dokter	dr. Wachidah Wahyuningsih	Medicare	2014-05-17 06:26:56	Aktif	Cari Edit
Dokter	dr. Fida Rahmayanti	Medicare	2014-05-17 06:26:31	Aktif	Cari Edit
Dokter	dr. Chamidatus Sa'idah	Medicare	0000-00-00 00:00:00	Aktif	Cari Edit

Showing 1 to 10 of 38 entries

<< 1 2 3 4 >>

Valid

SRS_1_03

Sistem harus menyediakan fitur tambah data pegawai untuk staff kepegawaian.

Tambah Pegawai

Nama	<input type="text" value="Gelar depan"/> <input type="text" value="Nama Awal"/> <input type="text" value="Nama Akhir"/> <input type="text" value="Gelar belakan"/>
Gelar diisi tanpa penghubung titik (.)	
Nomor induk	<input type="text" value="Nomor Induk Kependudukan"/> <input type="text" value="Nomor Induk Pegawai"/>
Tanggal lahir	<input type="text" value="hh/bb/tttt"/>
Jenis kelamin	<input type="text" value="Pilih jenis kelamin"/>
No HP	<input type="text"/>
Golongan darah	<input type="text" value="-"/>
Email	<input type="text"/>
Jenis pegawai	<input type="text" value="Pilih jenis pegawai"/>
Jabatan struktural	<input type="text" value="Pilih jabatan struktural"/>
Pendidikan terakhir	<input type="text" value="Pilih pendidikan terakhir"/>
Alamat	<input type="text" value="Isi alamat"/> <input type="text" value="Pilih negara"/> <input type="text" value="Pilih Negara"/> <input type="text" value="Pilih Propinsi"/>
Bio	<input type="text" value="Profil pegawai"/>
Password	<input type="text"/>
Status Layanan	<input type="text" value="Pilih status layanan"/>

Valid

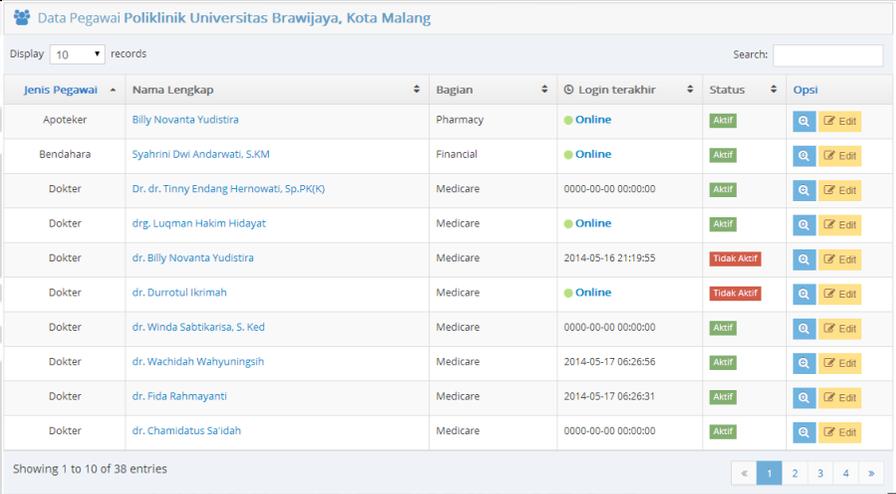
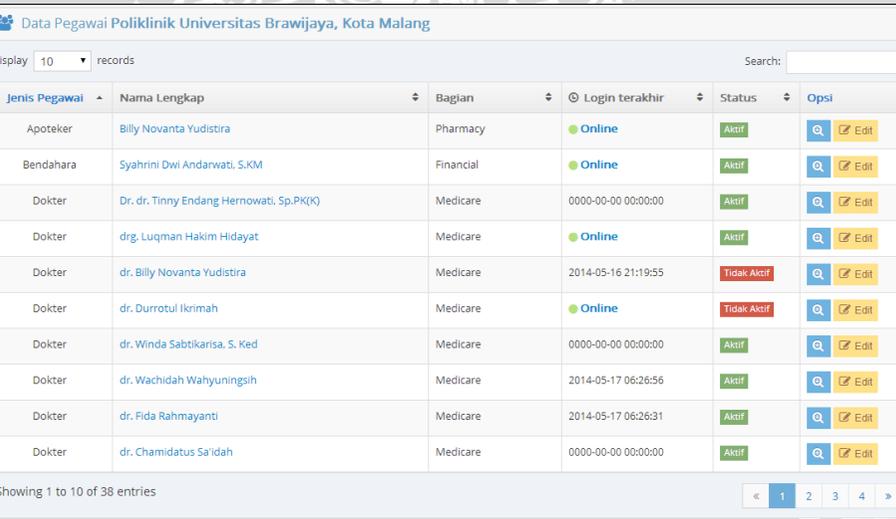
SRS_1_04

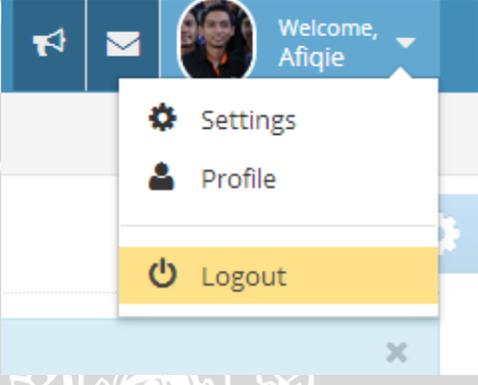
Sistem harus menyediakan fitur ubah data pegawai untuk staff kepegawaian.

The screenshot displays the user profile interface for Dr. dr. Tinny Endang Hernowati, Sp.PK(K). The page includes a navigation menu on the left with options like 'Human Resources', 'Data Kepegawaian', and 'Tambah Pegawai'. The main content area shows the user's name, a placeholder profile picture, and a status of 'Offline' with the last login time '0000-00-00 00:00:00'. Below this, there is a 'Send a message' button. The 'Informasi Personal' section contains a table with the following data:

Informasi Personal			
Nama lengkap	Dr. dr.	Tinny	Endang Hernowati Sp.PK(K)
Tanggal lahir	07/11/1988		
NIK	195121225198002001		
NIP	195121225198002001		
No HP	0341 460 420		
No Telp			
Golongan darah	B		
Jenis kelamin	Perempuan		

Valid

<p>SRS_1_05</p>	<p>Sistem harus menyediakan fitur hapus daftar pegawai untuk staff kepegawaian.</p>	 <p>The screenshot shows a table titled 'Data Pegawai Poliklinik Universitas Brawijaya, Kota Malang'. The table has columns for 'Jenis Pegawai', 'Nama Lengkap', 'Bagian', 'Login terakhir', 'Status', and 'Ops'. The 'Status' column shows 'Online' for most entries, and the 'Ops' column contains a magnifying glass icon and an 'Edit' button. The 'Login terakhir' column shows various timestamps, including '0000-00-00 00:00:00' and '2014-05-16 21:19:55'.</p>	<p>Valid</p>
<p>SRS_1_06</p>	<p>Sistem harus menyediakan fitur menampilkan daftar kehadiran untuk staff kepegawaian.</p>	 <p>This screenshot is identical to the one above, showing the same table of staff members. The 'Status' column shows 'Online' for most entries, and the 'Ops' column contains a magnifying glass icon and an 'Edit' button. The 'Login terakhir' column shows various timestamps, including '0000-00-00 00:00:00' and '2014-05-16 21:19:55'.</p>	<p>Valid</p>

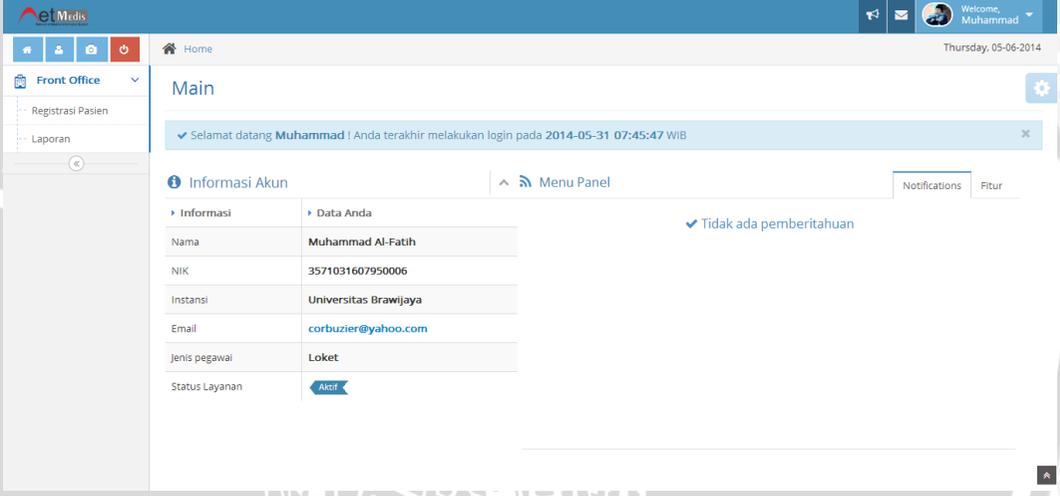
SRS_1_07	Sistem harus menyediakan fitur logout		Valid
----------	---------------------------------------	---	-------

Sumber: Perancangan

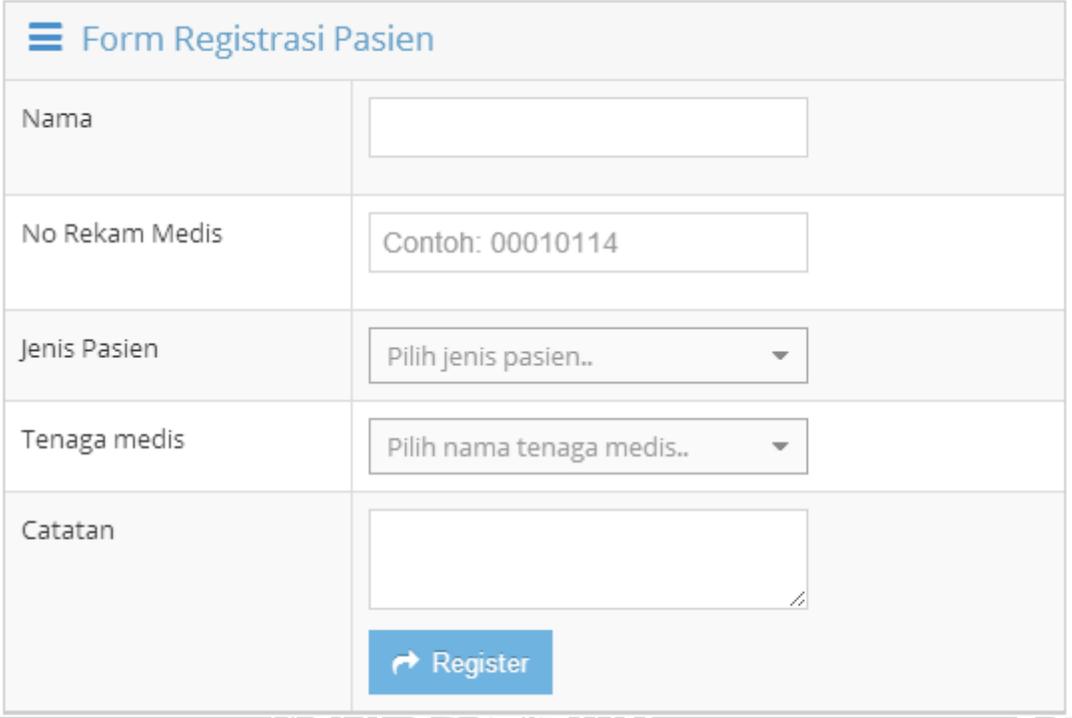
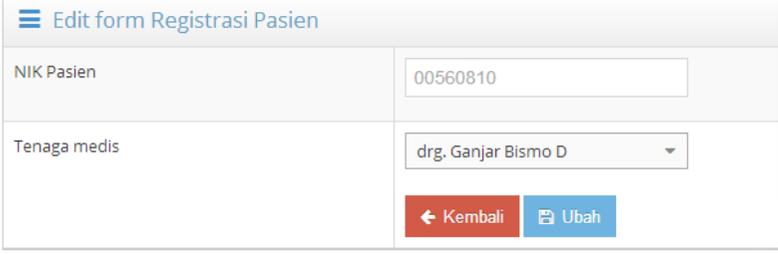
5.1.1.2 Front Office

Seperti pada tabel 5.2 menunjukkan fungsionalitas sistem sebagai Front Office yang dapat mengakses modul

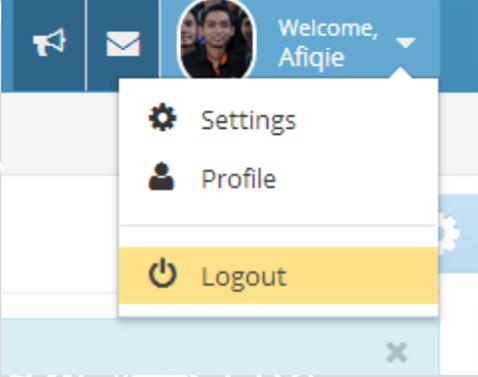
Tabel 5.2 Front Office

No SRS	Kebutuhan	Tampilan	Status
SRS_2_01	Sistem harus menyediakan login staf Front Office		Valid

SRS_2_02	Sistem harus menyediakan fitur menampilkan riwayat penanganan dokter	 <table border="1"> <thead> <tr> <th data-bbox="728 308 1391 379">Nama Dokter</th> <th data-bbox="1391 308 1581 379">Poli</th> <th data-bbox="1581 308 1800 379">Jumlah</th> </tr> </thead> <tbody> <tr> <td data-bbox="728 379 1391 451">drg. Luqman Hakim Hidayat</td> <td data-bbox="1391 379 1581 451">Gigi</td> <td data-bbox="1581 379 1800 451">3</td> </tr> <tr> <td data-bbox="728 451 1391 528">dr. Karina Nilasari</td> <td data-bbox="1391 451 1581 528">Umum</td> <td data-bbox="1581 451 1800 528">33</td> </tr> <tr> <td data-bbox="728 528 1391 604">dr. Marsha Dechastra C, S. Ked.</td> <td data-bbox="1391 528 1581 604">Umum</td> <td data-bbox="1581 528 1800 604">8</td> </tr> <tr> <td data-bbox="728 604 1391 683">drg. Ganjar Bismo D</td> <td data-bbox="1391 604 1581 683">Gigi</td> <td data-bbox="1581 604 1800 683">6</td> </tr> </tbody> </table>	Nama Dokter	Poli	Jumlah	drg. Luqman Hakim Hidayat	Gigi	3	dr. Karina Nilasari	Umum	33	dr. Marsha Dechastra C, S. Ked.	Umum	8	drg. Ganjar Bismo D	Gigi	6	Valid
Nama Dokter	Poli	Jumlah																
drg. Luqman Hakim Hidayat	Gigi	3																
dr. Karina Nilasari	Umum	33																
dr. Marsha Dechastra C, S. Ked.	Umum	8																
drg. Ganjar Bismo D	Gigi	6																

SRS_2_03	Sistem harus menyediakan fitur Register Pasien		
SRS_2_04	Sistem harus menyediakan fitur ubah daftar antrian registrasi pasien		Valid

SRS_2_05	Sistem harus menyediakan fitur menampilkan dan menghapus daftar registrasi pasien	<table border="1"> <thead> <tr> <th>No</th> <th>Tanggal</th> <th>Tenaga Medis</th> <th>Nama Pasien</th> <th>Register → Check up → Payment → Medicine</th> <th>Ops</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2014-05-17</td> <td>dr. Marsha Dechastra C. S. Ked.</td> <td>Pavita Wulan A</td> <td> [09:19:05] [09:52:52] [09:55:03] [10:00:08] </td> <td></td> </tr> <tr> <td>2</td> <td>2014-05-17</td> <td>dr. Marsha Dechastra C. S. Ked.</td> <td>Sakunda STP</td> <td> [09:47:08] [10:21:08] [10:21:30] [10:22:51] </td> <td></td> </tr> <tr> <td>3</td> <td>2014-05-17</td> <td>drg. Ganjar Bismo D</td> <td>Evian Eka Kusuma W</td> <td> [09:48:13] [10:07:10] [10:10:51] [09:48:13] </td> <td></td> </tr> <tr> <td>4</td> <td>2014-05-17</td> <td>dr. Marsha Dechastra C. S. Ked.</td> <td>Retno Kusuma N</td> <td> [09:52:56] [10:45:26] [10:47:03] [10:48:30] </td> <td></td> </tr> </tbody> </table>	No	Tanggal	Tenaga Medis	Nama Pasien	Register → Check up → Payment → Medicine	Ops	1	2014-05-17	dr. Marsha Dechastra C. S. Ked.	Pavita Wulan A	 [09:19:05] [09:52:52] [09:55:03] [10:00:08]		2	2014-05-17	dr. Marsha Dechastra C. S. Ked.	Sakunda STP	 [09:47:08] [10:21:08] [10:21:30] [10:22:51]		3	2014-05-17	drg. Ganjar Bismo D	Evian Eka Kusuma W	 [09:48:13] [10:07:10] [10:10:51] [09:48:13]		4	2014-05-17	dr. Marsha Dechastra C. S. Ked.	Retno Kusuma N	 [09:52:56] [10:45:26] [10:47:03] [10:48:30]		Valid
No	Tanggal	Tenaga Medis	Nama Pasien	Register → Check up → Payment → Medicine	Ops																												
1	2014-05-17	dr. Marsha Dechastra C. S. Ked.	Pavita Wulan A	 [09:19:05] [09:52:52] [09:55:03] [10:00:08]																													
2	2014-05-17	dr. Marsha Dechastra C. S. Ked.	Sakunda STP	 [09:47:08] [10:21:08] [10:21:30] [10:22:51]																													
3	2014-05-17	drg. Ganjar Bismo D	Evian Eka Kusuma W	 [09:48:13] [10:07:10] [10:10:51] [09:48:13]																													
4	2014-05-17	dr. Marsha Dechastra C. S. Ked.	Retno Kusuma N	 [09:52:56] [10:45:26] [10:47:03] [10:48:30]																													
SRS_2_06	Sistem harus menyediakan fitur Pencarian Pasien		Valid																														
SRS_2_07	Sistem harus menyediakan fitur menampilkan statistik pengunjung per bulan	<table border="1"> <thead> <tr> <th colspan="2">Statistik Pengunjung</th> </tr> </thead> <tbody> <tr> <td>Jumlah pengunjung bulan ini</td> <td>51</td> </tr> <tr> <td>Jumlah kunjungan bulan ini</td> <td>52</td> </tr> </tbody> </table>	Statistik Pengunjung		Jumlah pengunjung bulan ini	51	Jumlah kunjungan bulan ini	52	Valid																								
Statistik Pengunjung																																	
Jumlah pengunjung bulan ini	51																																
Jumlah kunjungan bulan ini	52																																

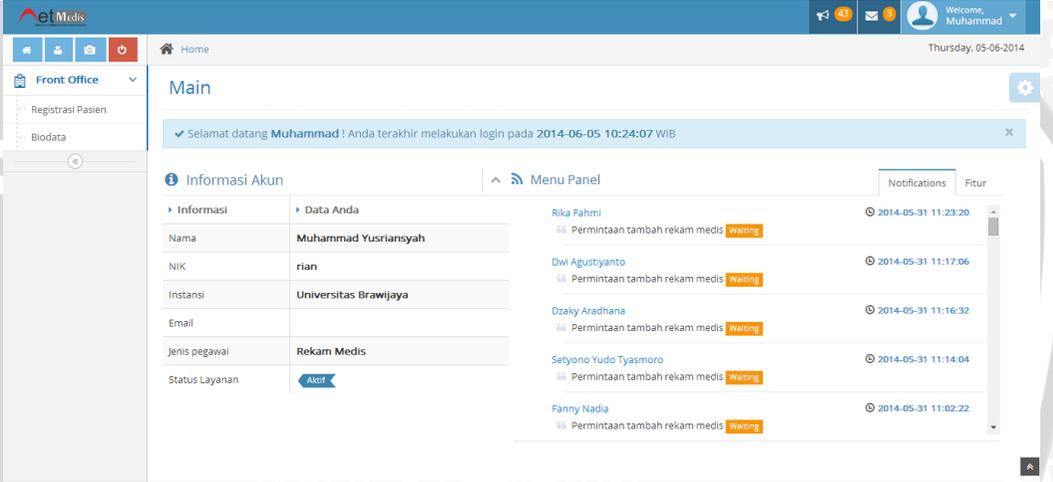
SRS_2_08	Sistem harus menyediakan fitur logout		Valid
----------	---------------------------------------	---	-------

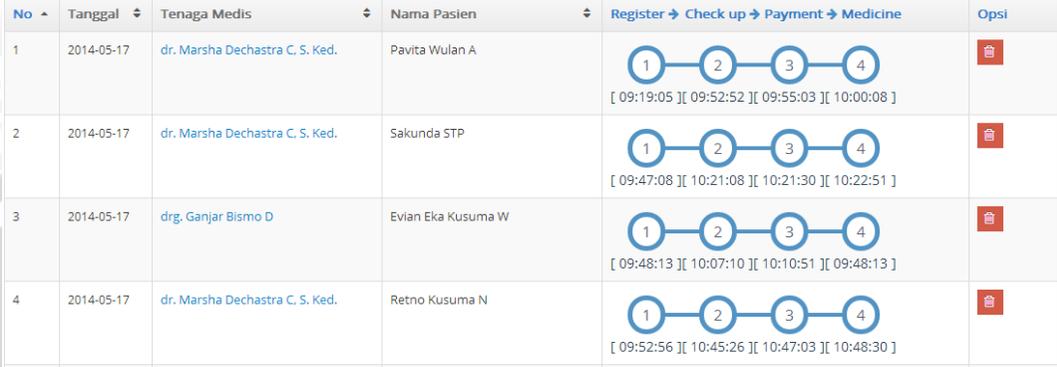
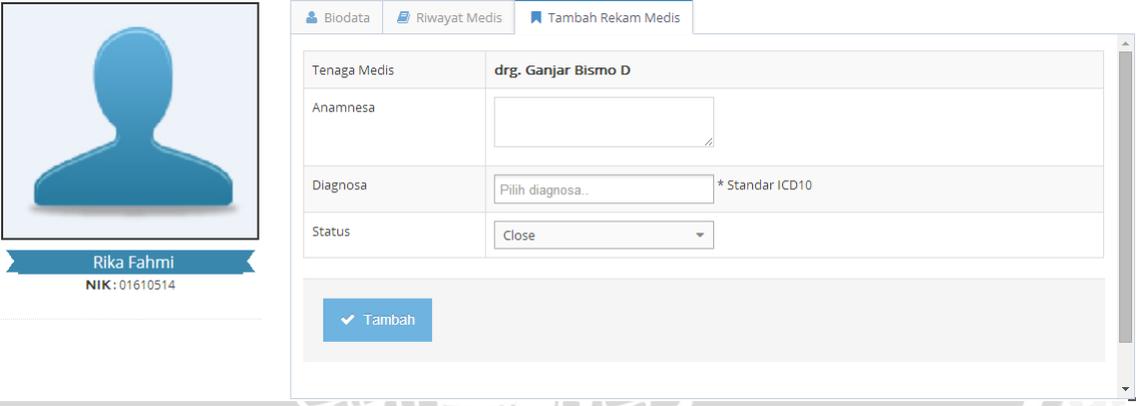
Sumber: Perancangan

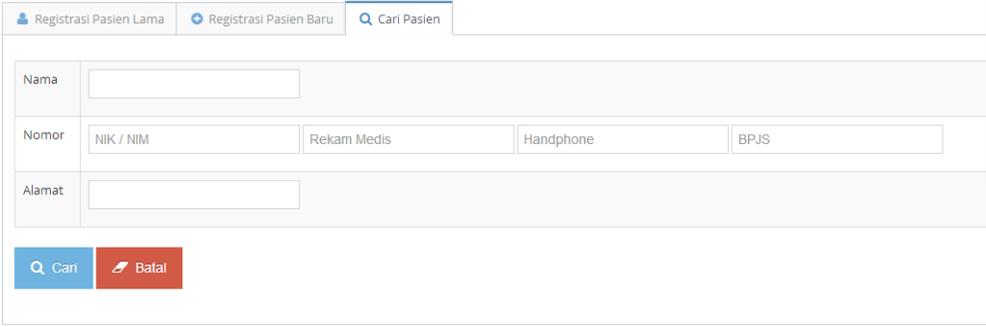
5.1.1.3 Medical Records

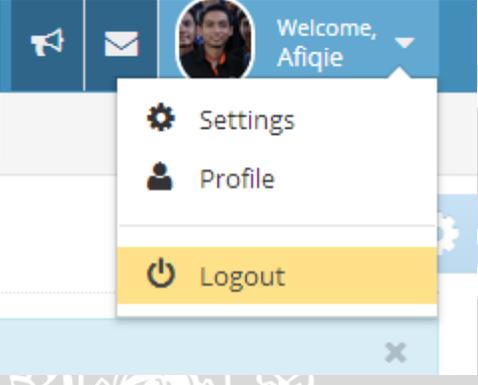
Seperti pada tabel 5.3 menunjukkan fungsionalitas sistem sebagai Medical Records yang dapat mengakses modul

Tabel 5.3 Medical Records

No SRS	Kebutuhan	Tampilan	Status
SRS_3_01	Sistem harus menyediakan login rekam medis	 <p>The screenshot shows the 'etMedis' web application interface. The user is logged in as Muhammad Yusransyah. The interface displays a welcome message and a list of pending medical record requests (Permintaan tambah rekam medis) with a 'Waiting' status. The requests are listed with names like Rika Fahmi, Dwi Agustyanto, Dzaky Aradhana, Setyono Yudo Tyasmoro, and Fanny Nadia, along with their respective timestamps.</p>	Valid

<p>SRS_3_02</p>	<p>Sistem harus menyediakan fitur menampilkan daftar data registrasi pasien</p>	 <table border="1"> <thead> <tr> <th>No</th> <th>Tanggal</th> <th>Tenaga Medis</th> <th>Nama Pasien</th> <th>Register</th> <th>Check up</th> <th>Payment</th> <th>Medicine</th> <th>Opsi</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2014-05-17</td> <td>dr. Marsha Dechastra C. S. Ked.</td> <td>Pavita Wulan A</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td></td> </tr> <tr> <td>2</td> <td>2014-05-17</td> <td>dr. Marsha Dechastra C. S. Ked.</td> <td>Sakunda STP</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td></td> </tr> <tr> <td>3</td> <td>2014-05-17</td> <td>drg. Ganjar Bismo D</td> <td>Evian Eka Kusuma W</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td></td> </tr> <tr> <td>4</td> <td>2014-05-17</td> <td>dr. Marsha Dechastra C. S. Ked.</td> <td>Retno Kusuma N</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td></td> </tr> </tbody> </table>	No	Tanggal	Tenaga Medis	Nama Pasien	Register	Check up	Payment	Medicine	Opsi	1	2014-05-17	dr. Marsha Dechastra C. S. Ked.	Pavita Wulan A	1	2	3	4		2	2014-05-17	dr. Marsha Dechastra C. S. Ked.	Sakunda STP	1	2	3	4		3	2014-05-17	drg. Ganjar Bismo D	Evian Eka Kusuma W	1	2	3	4		4	2014-05-17	dr. Marsha Dechastra C. S. Ked.	Retno Kusuma N	1	2	3	4		<p>Valid</p>
No	Tanggal	Tenaga Medis	Nama Pasien	Register	Check up	Payment	Medicine	Opsi																																								
1	2014-05-17	dr. Marsha Dechastra C. S. Ked.	Pavita Wulan A	1	2	3	4																																									
2	2014-05-17	dr. Marsha Dechastra C. S. Ked.	Sakunda STP	1	2	3	4																																									
3	2014-05-17	drg. Ganjar Bismo D	Evian Eka Kusuma W	1	2	3	4																																									
4	2014-05-17	dr. Marsha Dechastra C. S. Ked.	Retno Kusuma N	1	2	3	4																																									
<p>SRS_3_03</p>	<p>Sistem harus menyediakan fitur tambah rekam medis pasien</p>	 <p>Rika Fahmi NIK: 01610514</p> <p>Tenaga Medis: drg. Ganjar Bismo D</p> <p>Anamnesa: <input type="text"/></p> <p>Diagnosa: <input type="text"/> * Standar ICD10</p> <p>Status: <input type="text"/></p> <p><input type="button" value="Tambah"/></p>	<p>Valid</p>																																													

<p>SRS_3_04</p>	<p>Sistem harus menyediakan fitur ubah rekam medis pasien</p>		<p>Valid</p>
<p>SRS_3_05</p>	<p>Sistem harus menyediakan fitur hapus rekam medis pasien</p>		<p>Valid</p>
<p>SRS_3_06</p>	<p>Sistem harus menyediakan fitur Pencarian Pasien</p>		<p>Valid</p>

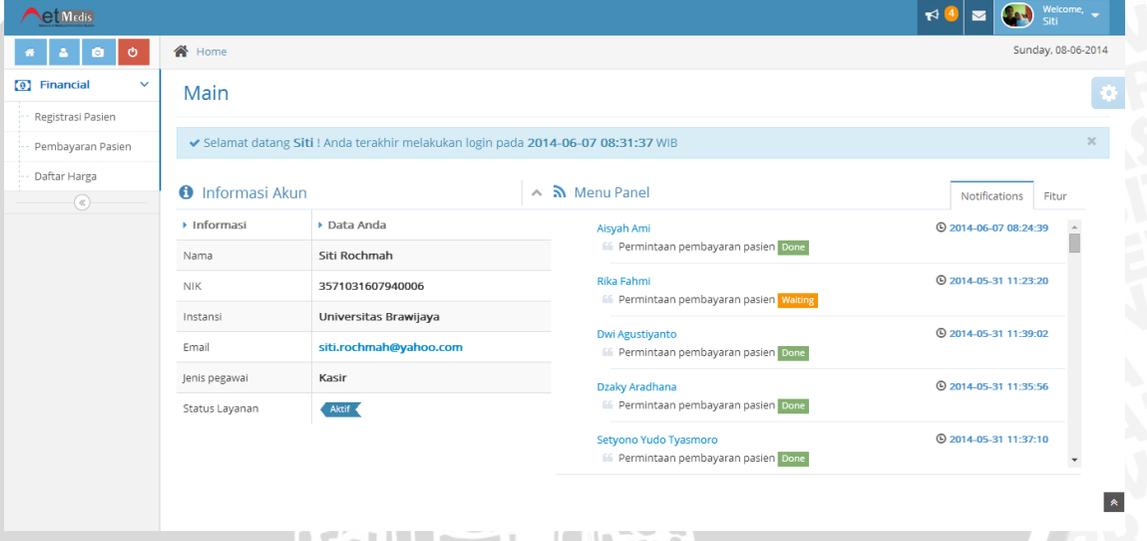
SRS_3_07	Sistem harus menyediakan fitur logout	 A screenshot of a user profile menu. The menu is open, showing options: Settings (gear icon), Profile (person icon), and Logout (power icon). The 'Logout' option is highlighted in yellow. The background shows a blue header with a notification bell, an envelope icon, and a user profile picture with the text 'Welcome, Afiqie'. There is also a close button (X) in the bottom right corner of the menu.	Valid
----------	---------------------------------------	---	-------

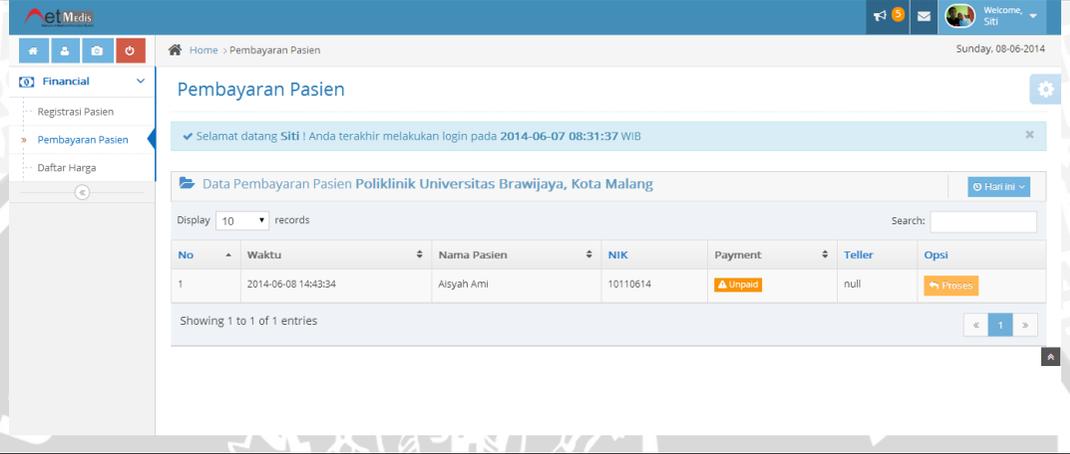
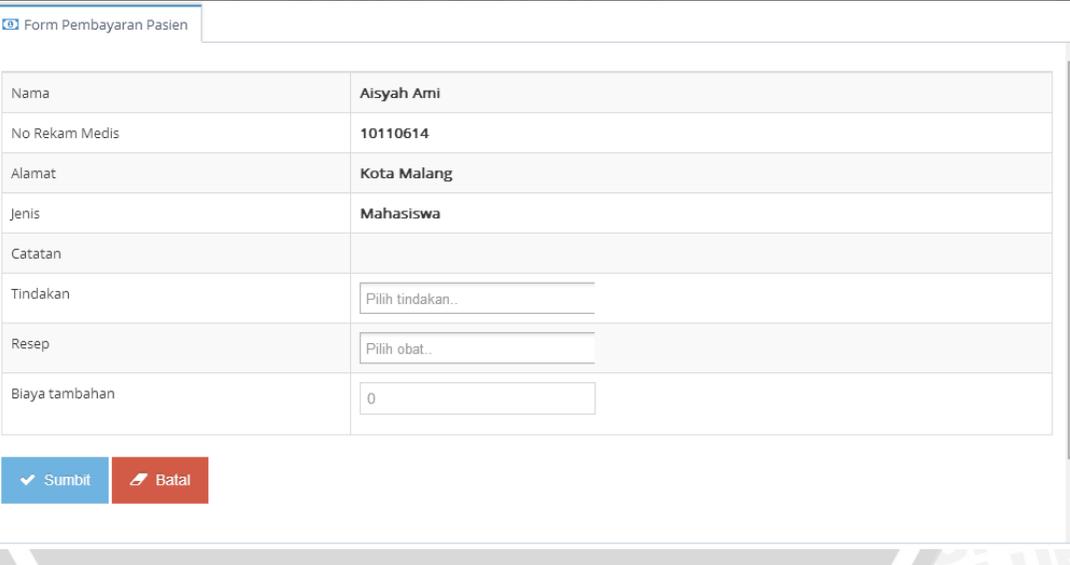
Sumber: Perancangan

5.1.1.4 Chasier

Seperti pada tabel 5.4 menunjukkan fungsionalitas sistem sebagai Chasier yang dapat mengakses modul

Tabel 5.4 Chasier

No SRS	Kebutuhan	Tampilan	Status
SRS_4_01	Sistem harus menyediakan login Chasier		Valid

SRS_4_02	Sistem harus menyediakan fitur menampilkan daftar data pembayaran pasien	 <p>The screenshot shows the 'Pembayaran Pasien' page in the etMedis system. The page title is 'Pembayaran Pasien' and it displays a table of patient payment records. The table has the following columns: No, Waktu, Nama Pasien, NIK, Payment, Teller, and Opsi. There is one record for 'Aisyah Ami' with NIK 10110614, a payment status of 'Unpaid', and a 'Proses' button. The page also includes a search bar and a 'Filter' button.</p>	Valid																
SRS_4_03	Sistem harus menyediakan fitur tambah form pembayaran pasien	 <p>The screenshot shows the 'Form Pembayaran Pasien' in the etMedis system. The form contains the following fields and values:</p> <table border="1" data-bbox="728 758 1780 1117"> <tr> <td>Nama</td> <td>Aisyah Ami</td> </tr> <tr> <td>No Rekam Medis</td> <td>10110614</td> </tr> <tr> <td>Alamat</td> <td>Kota Malang</td> </tr> <tr> <td>Jenis</td> <td>Mahasiswa</td> </tr> <tr> <td>Catatan</td> <td></td> </tr> <tr> <td>Tindakan</td> <td>Pilih tindakan...</td> </tr> <tr> <td>Resep</td> <td>Pilih obat...</td> </tr> <tr> <td>Biaya tambahan</td> <td>0</td> </tr> </table> <p>At the bottom of the form, there are two buttons: 'Sumbit' (Submit) and 'Batal' (Cancel).</p>	Nama	Aisyah Ami	No Rekam Medis	10110614	Alamat	Kota Malang	Jenis	Mahasiswa	Catatan		Tindakan	Pilih tindakan...	Resep	Pilih obat...	Biaya tambahan	0	Valid
Nama	Aisyah Ami																		
No Rekam Medis	10110614																		
Alamat	Kota Malang																		
Jenis	Mahasiswa																		
Catatan																			
Tindakan	Pilih tindakan...																		
Resep	Pilih obat...																		
Biaya tambahan	0																		

SRS_4_04

Sistem harus menyediakan fitur tambah form invoice pasien

Invoice #178


Status: UNPAID
Date: 2014-06-08 14:43:34

Pay to:
Poliklinik Universitas Brawijaya
 Jl. Soekarno Hatta no 5, Kota Malang
 +623415992032

Invoice to:
Aisyah Ami
 No Rekam Medis : 10110614
 Jenis Pasien : Mahasiswa

Check up Info:
 Poli Umum
 Tenaga medis: dr. Winda Sabtikarisa S. Ked

Payment Info:

Deskripsi	Jumlah
Layanan: Crown Sementara	Rp 100.000
Resep: Acetosal 80mg	Rp 0
Lain-lain	Rp 0

Total : Rp **100.000**

Bayar : Rp Enter

Valid

SRS_4_05

Sistem harus menyediakan fitur cetak *invoice* pasien

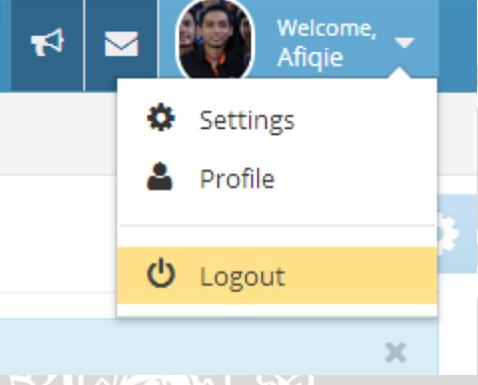
The screenshot shows a web browser window titled "Hospital ERP System - Google Chrome" displaying a print invoice page. The URL is "financial/hes/financial/kasir/print_invoice/10110614/178". The invoice details are as follows:

- Invoice #178** | Status: PAID | Date: 2014-06-08 14:43:34
- Pay to:** Poliklinik Universitas Brawijaya, Jl. Soekarno Hatta no 5, Kota Malang, +623415992032
- Invoice to:** Aisyah Ami
- No Rekam Medis : 10110614
Jenis Pasien : Mahasiswa
- Check up Info:** Poli Umum
- Tenaga medis:** dr. Winda Sabtikarisa S. Ked
- Payment Info:**

Deskripsi	Jumlah
Layanan: Crown Sementara	Rp 100.000
Resep: Acetosal 80mg	Rp 0
Lain-lain	Rp 0
Total : Rp 100.000	
Bayar : Rp 100.000	
Kembali : Rp 0	

Terima kasih atas kepercayaan Anda

Valid

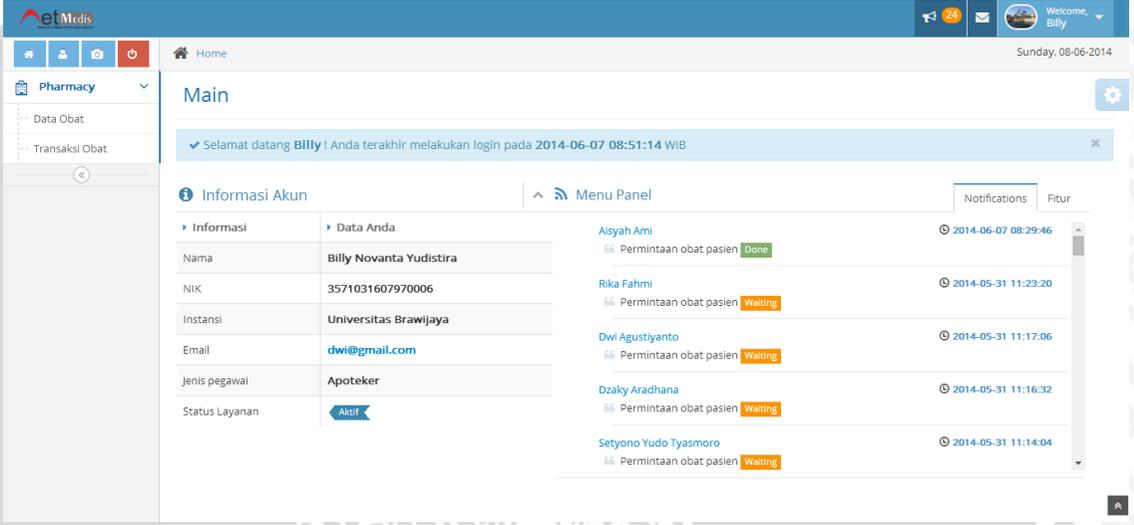
SRS_4_06	Sistem harus menyediakan fitur logout	 A screenshot of a user interface showing a profile menu. The menu is open, displaying options: 'Settings', 'Profile', and 'Logout'. The 'Logout' option is highlighted in yellow. The background shows a blurred user profile header with a 'Welcome, Afiqie' message and a close button (X) in the bottom right corner.	Valid
----------	---------------------------------------	--	-------

Sumber: Perancangan

5.1.1.5 Pharmacy

Seperti pada tabel 5.5 menunjukkan fungsionalitas sistem sebagai Pharmacy yang dapat mengakses modul

Tabel 5.5 Pharmacy

No SRS	Kebutuhan	Tampilan	Status																																
SRS_5_01	Sistem harus menyediakan login Apotek	 <p>The screenshot shows the 'elMedis' Pharmacy system interface. The user is logged in as Billy Novanta Yudistira, an Apoteker at Universitas Brawijaya. The interface displays a welcome message, account information, and a list of pending medication requests for various patients.</p> <table border="1"> <thead> <tr> <th>Informasi</th> <th>Data Anda</th> </tr> </thead> <tbody> <tr> <td>Nama</td> <td>Billy Novanta Yudistira</td> </tr> <tr> <td>NIK</td> <td>3571031607970006</td> </tr> <tr> <td>Instansi</td> <td>Universitas Brawijaya</td> </tr> <tr> <td>Email</td> <td>dwi@gmail.com</td> </tr> <tr> <td>Jenis pegawai</td> <td>Apoteker</td> </tr> <tr> <td>Status Layanan</td> <td>Aktif</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Patient Name</th> <th>Request Status</th> <th>Timestamp</th> </tr> </thead> <tbody> <tr> <td>Aisyah Ami</td> <td>Done</td> <td>2014-06-07 08:29:46</td> </tr> <tr> <td>Rika Fahmi</td> <td>Waiting</td> <td>2014-05-31 11:23:20</td> </tr> <tr> <td>Dwi Agustyanto</td> <td>Waiting</td> <td>2014-05-31 11:17:06</td> </tr> <tr> <td>Dzaky Aradhana</td> <td>Waiting</td> <td>2014-05-31 11:16:32</td> </tr> <tr> <td>Setyono Yudo Tyasmoro</td> <td>Waiting</td> <td>2014-05-31 11:14:04</td> </tr> </tbody> </table>	Informasi	Data Anda	Nama	Billy Novanta Yudistira	NIK	3571031607970006	Instansi	Universitas Brawijaya	Email	dwi@gmail.com	Jenis pegawai	Apoteker	Status Layanan	Aktif	Patient Name	Request Status	Timestamp	Aisyah Ami	Done	2014-06-07 08:29:46	Rika Fahmi	Waiting	2014-05-31 11:23:20	Dwi Agustyanto	Waiting	2014-05-31 11:17:06	Dzaky Aradhana	Waiting	2014-05-31 11:16:32	Setyono Yudo Tyasmoro	Waiting	2014-05-31 11:14:04	Valid
Informasi	Data Anda																																		
Nama	Billy Novanta Yudistira																																		
NIK	3571031607970006																																		
Instansi	Universitas Brawijaya																																		
Email	dwi@gmail.com																																		
Jenis pegawai	Apoteker																																		
Status Layanan	Aktif																																		
Patient Name	Request Status	Timestamp																																	
Aisyah Ami	Done	2014-06-07 08:29:46																																	
Rika Fahmi	Waiting	2014-05-31 11:23:20																																	
Dwi Agustyanto	Waiting	2014-05-31 11:17:06																																	
Dzaky Aradhana	Waiting	2014-05-31 11:16:32																																	
Setyono Yudo Tyasmoro	Waiting	2014-05-31 11:14:04																																	

SRS_5_02

Sistem harus menyediakan fitur menampilkan data *warehouse* obat

Data Obat Poliklinik Universitas Brawijaya, Kota Malang More ▾

Display 10 records Search:

Nama	Golongan	Satuan	Jumlah	Harga	Expired	Opsi
Abocat No. 18		Pcs	10	0	0000-00-00	
Abocat No. 20		Pcs	10	0	0000-00-00	
Abocat No. 22		Pcs	10	0	0000-00-00	
Acetosal 80mg		Tab	9	0	0000-00-00	
Acifar 400mg		Tab	300	0	0000-00-00	
Acyclovir 400mg		Tab	500	0	0000-00-00	
Acyclovir SK		Fls	10	3000	0000-00-00	
Acyclovir zalf		Tube	4	0	0000-00-00	
Alkohol 70% 1ltr		Fls	10	0	0000-00-00	
Alkohol swab		Pcs	10	0	0000-00-00	

Showing 1 to 10 of 377 entries < 1 2 3 4 5 >

Valid

SRS_5_03

Sistem harus menyediakan fitur edit data obat



Abocat No. 18

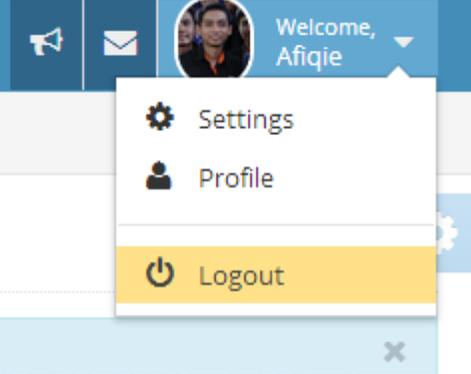
Informasi Obat Edit

Nama Obat	Abocat No. 18
Golongan	
Satuan	Box
Jumlah	10
Harga	0
Expired	hh/bb/tttt

Update

Valid

SRS_5_04	Sistem harus menyediakan fitur menampilkan transaksi obat pasien	<p>Data Transaksi Obat Pasien Poliklinik Universitas Brawijaya, Kota Malang Seluruh hari ▾</p> <p>Display <input type="text" value="10"/> records Search: <input type="text"/></p> <table border="1"> <thead> <tr> <th>No</th> <th>Waktu</th> <th>Nama Pasien</th> <th>NIK</th> <th>Payment</th> <th>Medicine</th> <th>Apoteker</th> <th>Opsi</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2014-05-17 10:00:08</td> <td>Pavita Wulan A</td> <td>62112</td> <td>✓ Done</td> <td>✓ Done</td> <td>3571031607970006</td> <td>Detail</td> </tr> <tr> <td>10</td> <td>2014-05-31 08:16:28</td> <td>Nela Marinamuti</td> <td>00670314</td> <td>✓ Done</td> <td>⚠ Waiting</td> <td>null</td> <td>Detail</td> </tr> <tr> <td>11</td> <td>2014-05-31 09:51:15</td> <td>Heny Kurniawati</td> <td>01580514</td> <td>✓ Done</td> <td>✓ Done</td> <td>3571031607970006</td> <td>Detail</td> </tr> <tr> <td>12</td> <td>2014-05-31 11:36:11</td> <td>Rachmad Sustyo W</td> <td>07730514</td> <td>✓ Done</td> <td>✓ Done</td> <td>3571031607970006</td> <td>Detail</td> </tr> <tr> <td>13</td> <td>2014-05-31 09:07:48</td> <td>Siti Rochani</td> <td>A00320514</td> <td>✓ Done</td> <td>✓ Done</td> <td>3571031607970006</td> <td>Detail</td> </tr> <tr> <td>14</td> <td>2014-05-31 09:17:10</td> <td>Anindya Puri P</td> <td>00820514</td> <td>✓ Done</td> <td>✓ Done</td> <td>3571031607970006</td> <td>Detail</td> </tr> <tr> <td>15</td> <td>2014-05-31 10:05:48</td> <td>Muhaimin Rifa'i</td> <td>04020610</td> <td>✓ Done</td> <td>✓ Done</td> <td>3571031607970006</td> <td>Detail</td> </tr> <tr> <td>16</td> <td>2014-05-31 09:32:32</td> <td>Fatchur Rachman</td> <td>07720514</td> <td>✓ Done</td> <td>✓ Done</td> <td>3571031607970006</td> <td>Detail</td> </tr> <tr> <td>17</td> <td>2014-05-31 09:37:04</td> <td>Dona Oktavia A</td> <td>09890913</td> <td>⚠ Waiting</td> <td>⚠ Waiting</td> <td>null</td> <td>Detail</td> </tr> <tr> <td>18</td> <td>2014-05-31 09:43:46</td> <td>Sugeng</td> <td>00560810</td> <td>✓ Done</td> <td>✓ Done</td> <td>3571031607970006</td> <td>Detail</td> </tr> </tbody> </table> <p>Showing 1 to 10 of 55 entries << 1 2 3 4 5 >></p>	No	Waktu	Nama Pasien	NIK	Payment	Medicine	Apoteker	Opsi	1	2014-05-17 10:00:08	Pavita Wulan A	62112	✓ Done	✓ Done	3571031607970006	Detail	10	2014-05-31 08:16:28	Nela Marinamuti	00670314	✓ Done	⚠ Waiting	null	Detail	11	2014-05-31 09:51:15	Heny Kurniawati	01580514	✓ Done	✓ Done	3571031607970006	Detail	12	2014-05-31 11:36:11	Rachmad Sustyo W	07730514	✓ Done	✓ Done	3571031607970006	Detail	13	2014-05-31 09:07:48	Siti Rochani	A00320514	✓ Done	✓ Done	3571031607970006	Detail	14	2014-05-31 09:17:10	Anindya Puri P	00820514	✓ Done	✓ Done	3571031607970006	Detail	15	2014-05-31 10:05:48	Muhaimin Rifa'i	04020610	✓ Done	✓ Done	3571031607970006	Detail	16	2014-05-31 09:32:32	Fatchur Rachman	07720514	✓ Done	✓ Done	3571031607970006	Detail	17	2014-05-31 09:37:04	Dona Oktavia A	09890913	⚠ Waiting	⚠ Waiting	null	Detail	18	2014-05-31 09:43:46	Sugeng	00560810	✓ Done	✓ Done	3571031607970006	Detail	Valid
No	Waktu	Nama Pasien	NIK	Payment	Medicine	Apoteker	Opsi																																																																																				
1	2014-05-17 10:00:08	Pavita Wulan A	62112	✓ Done	✓ Done	3571031607970006	Detail																																																																																				
10	2014-05-31 08:16:28	Nela Marinamuti	00670314	✓ Done	⚠ Waiting	null	Detail																																																																																				
11	2014-05-31 09:51:15	Heny Kurniawati	01580514	✓ Done	✓ Done	3571031607970006	Detail																																																																																				
12	2014-05-31 11:36:11	Rachmad Sustyo W	07730514	✓ Done	✓ Done	3571031607970006	Detail																																																																																				
13	2014-05-31 09:07:48	Siti Rochani	A00320514	✓ Done	✓ Done	3571031607970006	Detail																																																																																				
14	2014-05-31 09:17:10	Anindya Puri P	00820514	✓ Done	✓ Done	3571031607970006	Detail																																																																																				
15	2014-05-31 10:05:48	Muhaimin Rifa'i	04020610	✓ Done	✓ Done	3571031607970006	Detail																																																																																				
16	2014-05-31 09:32:32	Fatchur Rachman	07720514	✓ Done	✓ Done	3571031607970006	Detail																																																																																				
17	2014-05-31 09:37:04	Dona Oktavia A	09890913	⚠ Waiting	⚠ Waiting	null	Detail																																																																																				
18	2014-05-31 09:43:46	Sugeng	00560810	✓ Done	✓ Done	3571031607970006	Detail																																																																																				

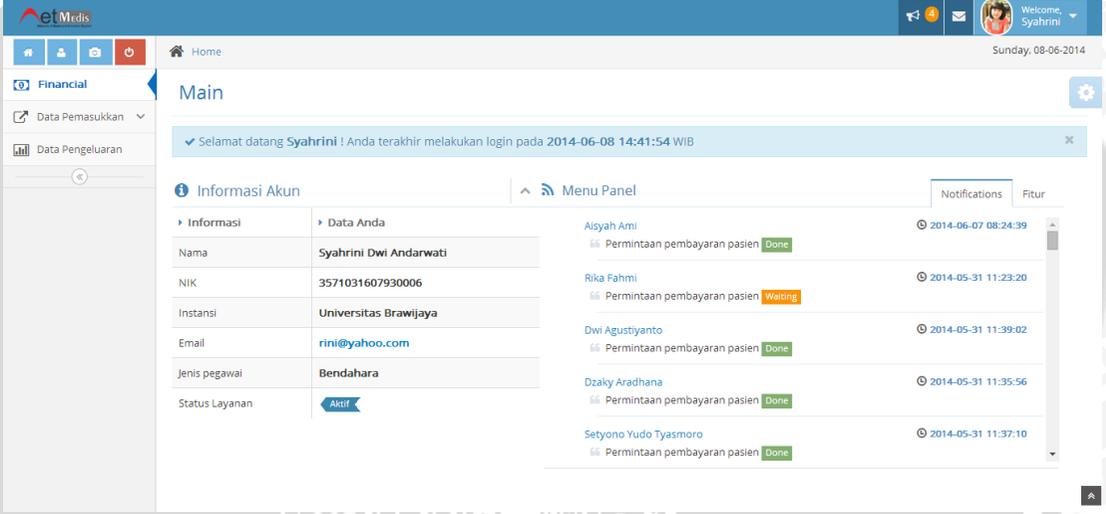
SRS_5_05	Sistem harus menyediakan fitur validasi obat pasien		Valid
SRS_5_06	Sistem harus menyediakan fitur logout		Valid

Sumber: Perancangan

5.1.1.6 Financial

Seperti pada tabel 5.6 menunjukkan fungsionalitas sistem sebagai Financial yang dapat mengakses modul

Tabel 5.6 Financial

No SRS	Kebutuhan	Tampilan	Status
SRS_6_01	Sistem harus menyediakan login bendahara	 <p>The screenshot shows the 'etMedis' Financial module interface. The user is logged in as 'Syahrini' (Syahrini Dwi Andarwati), a Bendahara (Finance Officer) at Universitas Brawijaya. The interface displays a 'Main' dashboard with a 'Selamat datang Syahrini!' message and a notification panel listing several 'Permintaan pembayaran pasien' (Patient payment requests) with status indicators like 'Done' or 'Waiting'.</p>	Valid

SRS_6_02

Sistem harus menyediakan fitur menampilkan data sumber dana

Tampilkan Data Berdasarkan Periode Waktu

01 Jan 2013 Sampai 08 Jun 2014 Tampilkan

Hasi dari "Data Sumber Dana"

Display 10 records

Search:

No	Tgl Input	Pengentri	Asal Dana	Nominal
1	22-01-2013 15:46:01	Muhammad	Rektor UB	20.000
2	22-04-2014 16:04:31	Siti	Afrique	20.000
3	11-05-2014 16:58:09	Syahrini	Dikti Kemendikbud RI	9.500.000
4	12-05-2014 21:03:21	Syahrini	Beasiswa	12.000.000

Showing 1 to 4 of 4 entries

< 1 >

Valid

SRS_6_03

Sistem harus menyediakan fitur menampilkan pemasukan pasien

Tampilkan Data Berdasarkan Periode Waktu View CHART

01 Jan 2013 Sampai 08 Jun 2014 Tampilkan

Hasil dari "Data Pemasukkan"

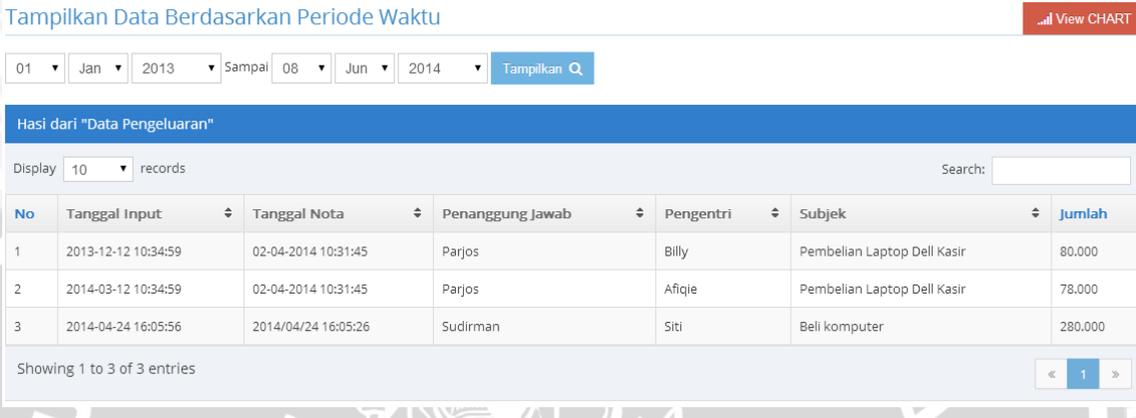
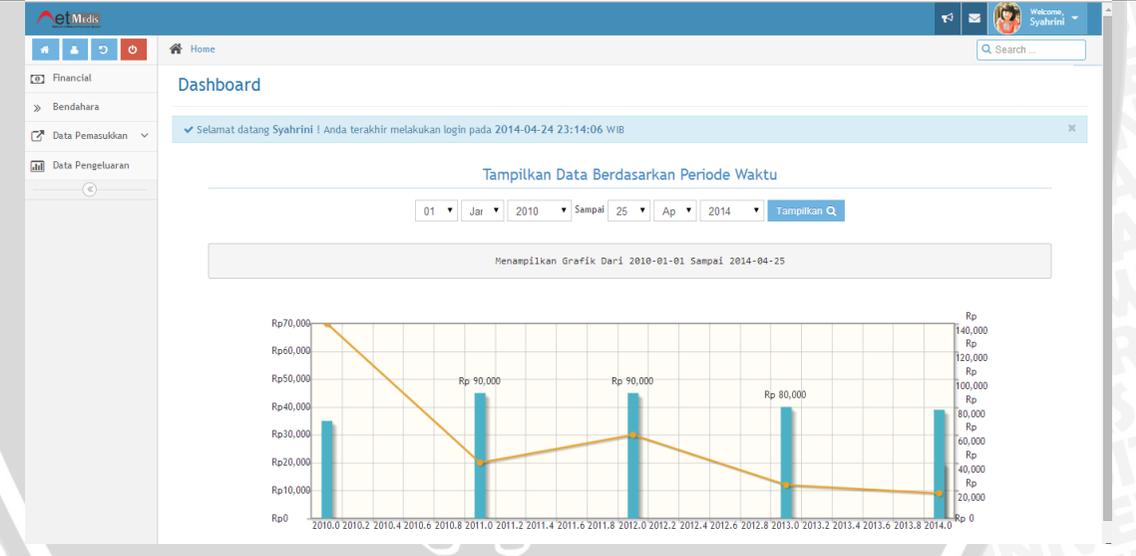
Display 10 records Search:

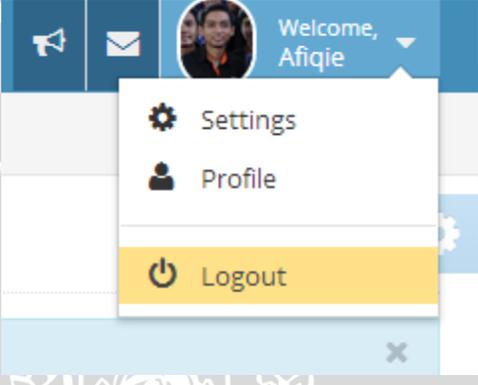
No	Tanggal	NIK	Nominal
21	31-05-2014 11:05:48	3571031607930007	140.000
22	31-05-2014 09:41:15	4380414	45.000
23	31-05-2014 09:41:15	10000114	0
24	31-05-2014 08:31:31	10010114	175.000
25	31-05-2014 08:35:47	10010114	400.000
26	31-05-2014 09:06:19	A00320514	10.000
27	31-05-2014 09:15:10	00820514	70.000
28	31-05-2014 09:29:26	07720514	30.000
29	31-05-2014 09:41:15	00560810	0
30	31-05-2014 09:46:16	01590514	25.000

Showing 21 to 30 of 63 entries

←
1
2
3
4
5
→

Valid

<p>SRS_6_04</p>	<p>Sistem harus menyediakan fitur menampilkan pengeluaran keseluruhan</p>		<p>Valid</p>
<p>SRS_6_05</p>	<p>Sistem harus menyediakan fitur pemasukan dan pengeluaran dalam bentuk chart</p>		<p>Valid</p>

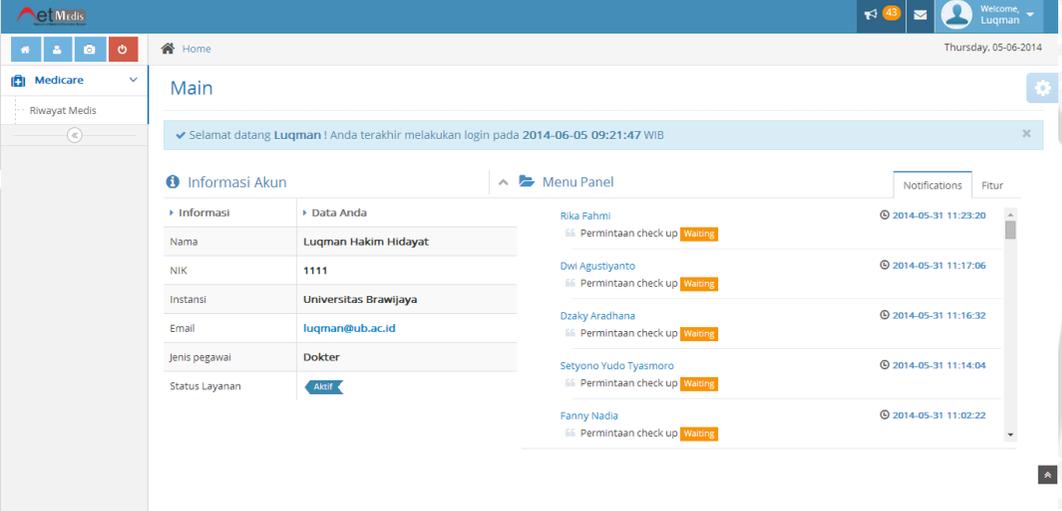
SRS_6_06	Sistem harus menyediakan fitur logout		Valid
----------	---------------------------------------	---	-------

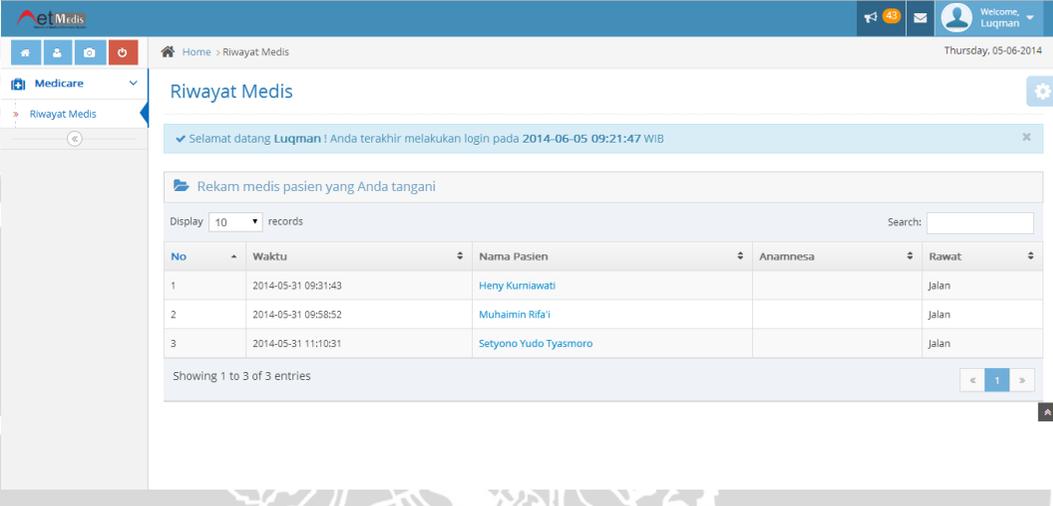
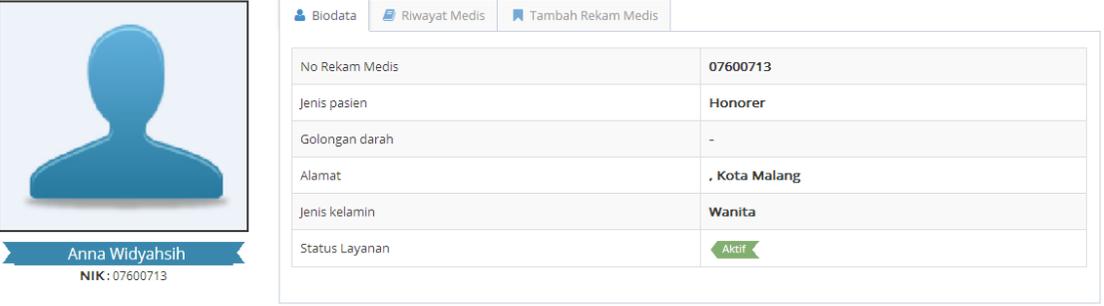
Sumber: Perancangan

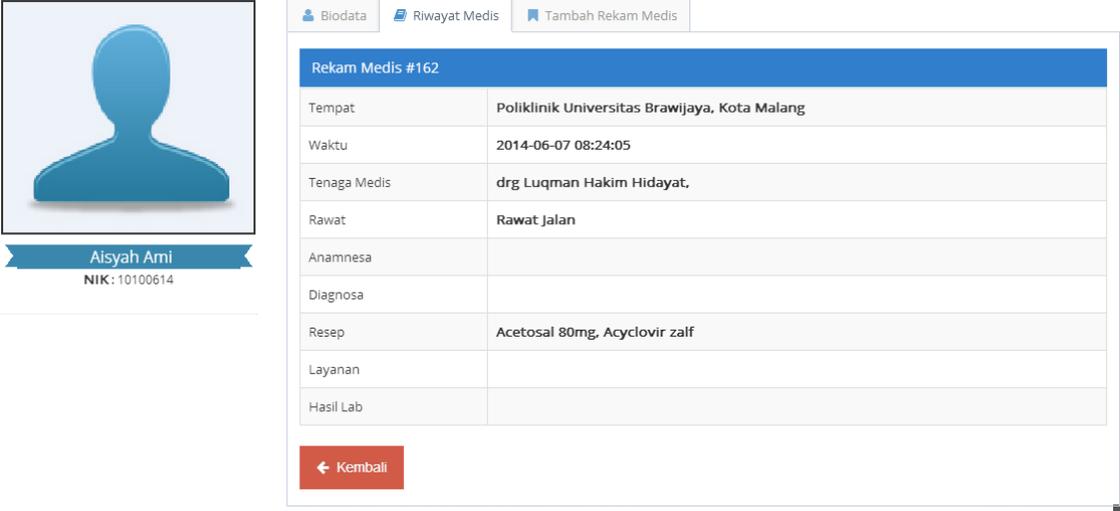
5.1.1.7 Medicare

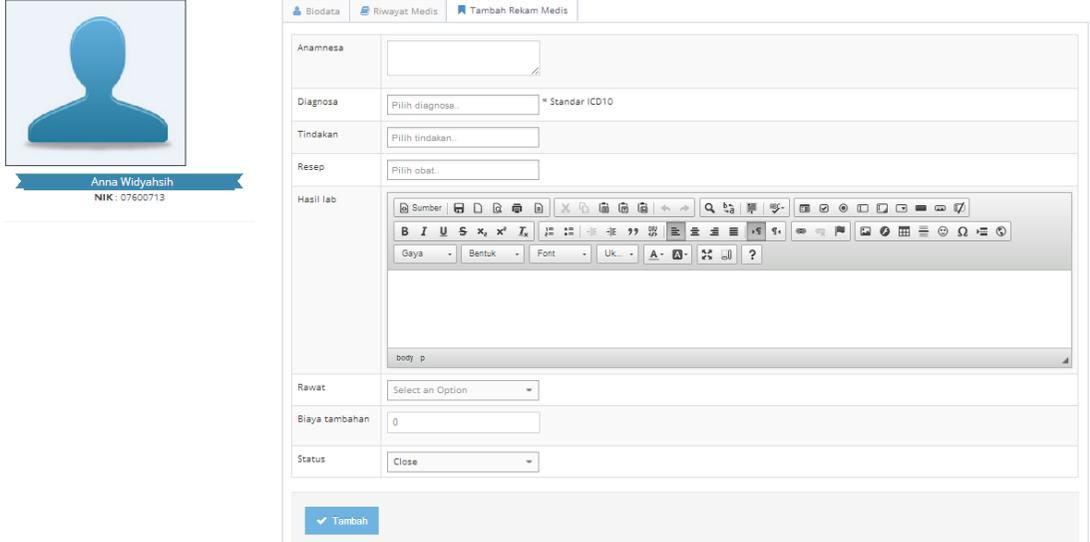
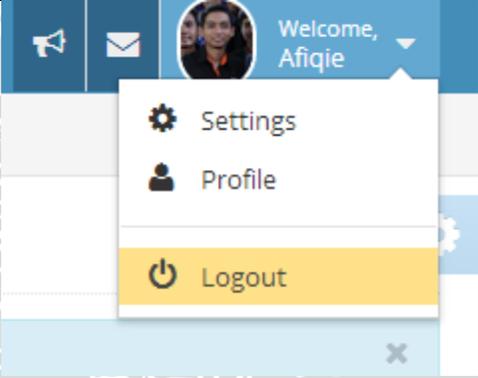
Seperti pada tabel 5.7 menunjukkan fungsionalitas sistem sebagai Medicare yang dapat mengakses modul

Tabel 5.7 Medicare

No SRS	Kebutuhan	Tampilan	Status
SRS_7_01	Sistem harus menyediakan login medicare		Valid

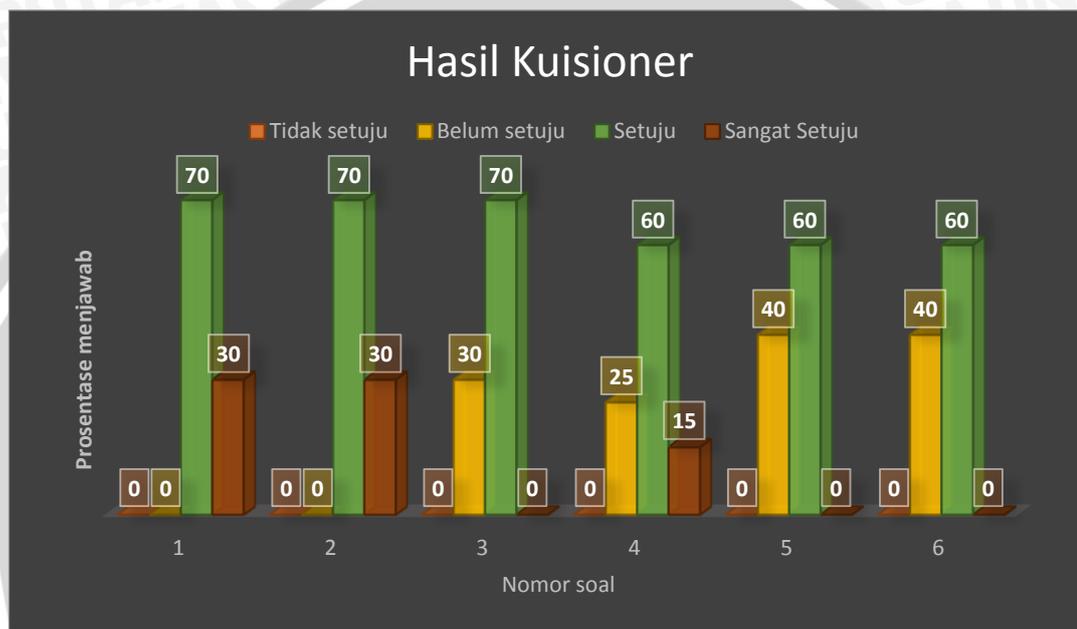
SRS_7_02	Sistem harus menyediakan fitur menampilkan daftar riwayat medis pasien		Valid												
SRS_7_03	Sistem harus menyediakan fitur menampilkan biodata pasien	 <table border="1" data-bbox="992 802 1798 1026"> <thead> <tr> <th>No Rekam Medis</th> <th>07600713</th> </tr> </thead> <tbody> <tr> <td>Jenis pasien</td> <td>Honoror</td> </tr> <tr> <td>Golongan darah</td> <td>-</td> </tr> <tr> <td>Alamat</td> <td>. Kota Malang</td> </tr> <tr> <td>Jenis kelamin</td> <td>Wanita</td> </tr> <tr> <td>Status Layanan</td> <td>Aktif</td> </tr> </tbody> </table>	No Rekam Medis	07600713	Jenis pasien	Honoror	Golongan darah	-	Alamat	. Kota Malang	Jenis kelamin	Wanita	Status Layanan	Aktif	Valid
No Rekam Medis	07600713														
Jenis pasien	Honoror														
Golongan darah	-														
Alamat	. Kota Malang														
Jenis kelamin	Wanita														
Status Layanan	Aktif														

SRS_7_04	Sistem harus menyediakan fitur menampilkan seluruh rekam medis pasien tertentu		Valid
SRS_7_05	Sistem harus menyediakan fitur menampilkan rekam medis pasien tertentu		Valid

<p>SRS_7_06</p>	<p>Sistem harus menyediakan fitur tambah rekam medis pasien</p>		<p>Valid</p>
<p>SRS_7_07</p>	<p>Sistem harus menyediakan fitur logout</p>		<p>Valid</p>

5.1.2 Pengujian *User Acceptance Test* (UAT)

Pengujian *User Acceptance Test* melibatkan penyebaran kuisiener terhadap pengguna dimana dalam kasus ini adalah dokter, perawat, staff rekam medis, staff Chasier, staff kepegawaian, staff depo obat, dan staff front office masing-masing sebanyak satu orang untuk melakukan persetujuan terhadap sistem yang telah dibuat.



Gambar 5.1 Hasil Kuisiener

Sumber: Pengujian

Berikut hasil deskripsi kuisiener dari Gambar 5.1 tersebut yang dijabarkan dalam Tabel 5.8 di bawah ini:

Tabel 5.8 Penjabaran Hasil Pengujian *User Acceptance Test* (UAT)

No.	Pertanyaan	Hasil
1.	Tampilan aplikasi mudah dipahami	Sebanyak 70 persen responden menyatakan setuju bahwa tampilan aplikasi mudah dipahami, sedangkan 30 persen lainnya menyatakan sangat setuju bahwa tampilan aplikasi mudah dipahami.

2.	Aplikasi turut membantu dalam mempermudah kinerja pengguna	Sebanyak 70 persen responden menyatakan setuju bahwa aplikasi turut membantu dalam mempermudah kinerja pengguna, sedangkan 30 persen lainnya menyatakan sangat setuju bahwa aplikasi turut membantu dalam mempermudah kinerja pengguna.
3.	Tidak terjadi bug/error/kesalahan pada aplikasi	Sebanyak 70 persen responden menyatakan setuju bahwa aplikasi tidak terjadi bug/error/kesalahan pada aplikasi, sedangkan 30 persen lainnya menyatakan belum setuju bahwa tidak terjadi bug/error/kesalahan pada aplikasi.
4.	Aplikasi mudah digunakan atau dijalankan	Sebanyak 60 persen responden menyatakan setuju bahwa aplikasi mudah digunakan atau dijalankan, sedangkan 25 persen lainnya menyatakan belum setuju bahwa aplikasi mudah digunakan atau dijalankan, sisanya 15 persen responden menyatakan sangat setuju bahwa aplikasi mudah digunakan atau dijalankan.
5.	Menu-menu pada aplikasi sudah sesuai dengan kebutuhan	Sebanyak 60 persen responden menyatakan setuju bahwa menu-menu pada aplikasi sudah sesuai dengan kebutuhan, sedangkan 40 persen lainnya menyatakan belum setuju bahwa menu-menu pada aplikasi sudah sesuai dengan kebutuhan.
6.	Aplikasi berjalan dengan baik tanpa kendala	Sebanyak 60 persen responden menyatakan setuju bahwa aplikasi berjalan dengan baik tanpa kendala,

	sedangkan 40 persen lainnya menyatakan belum setuju bahwa aplikasi berjalan dengan baik tanpa kendala.
--	--

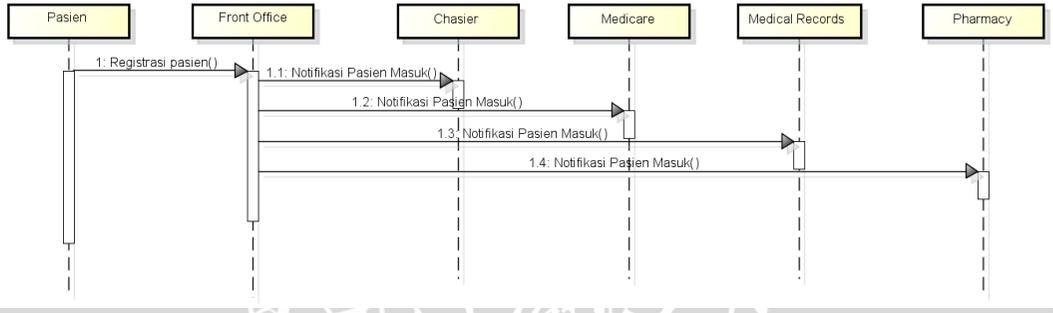
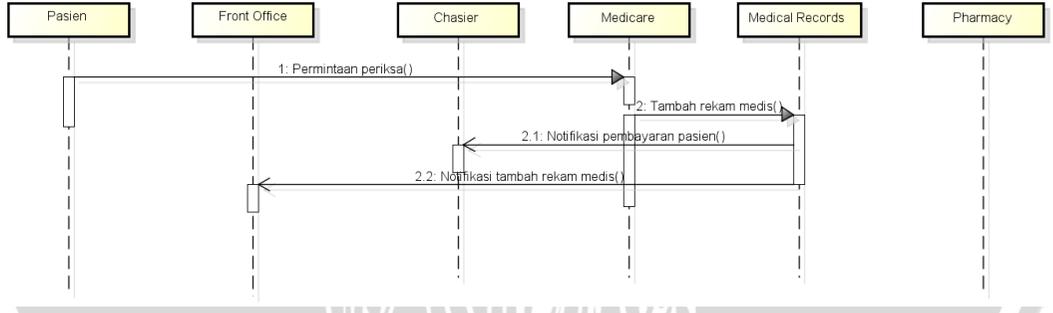
Sumber: Pengujian



5.1.3 Pengujian *Integration*

Pengujian ini menggambarkan hasil pengujian selain pengujian fungsional seperti yang tercantum pada table 5.9 berikut ini.

Tabel 5.9 Hasil pengujian *Integration*

Proses bisnis	Tampilan	Hasil
Registrasi Pasien	 <pre> sequenceDiagram participant P as Pasien participant FO as Front Office participant C as Chasier participant M as Medicare participant MR as Medical Records participant Ph as Pharmacy P->>FO: 1: Registrasi pasien() activate FO FO->>C: 1.1: Notifikasi Pasien Masuk() deactivate FO activate C C->>M: 1.2: Notifikasi Pasien Masuk() deactivate C activate M M->>MR: 1.3: Notifikasi Pasien Masuk() deactivate M activate MR MR->>Ph: 1.4: Notifikasi Pasien Masuk() deactivate MR deactivate Ph deactivate MR deactivate M deactivate C deactivate FO </pre>	Valid
Pemeriksaan pasien	 <pre> sequenceDiagram participant P as Pasien participant FO as Front Office participant C as Chasier participant M as Medicare participant MR as Medical Records participant Ph as Pharmacy P->>FO: 1: Permintaan periksa() activate FO FO->>M: 1: Permintaan periksa() deactivate FO activate M M->>MR: 2: Tambah rekam medis() deactivate M activate MR MR->>C: 2.1: Notifikasi pembayaran pasien() deactivate MR activate C C->>FO: 2.2: Notifikasi tambah rekam medis() deactivate C deactivate FO deactivate MR deactivate M </pre>	Valid

<p>Pembayaran pasien</p>	<pre> sequenceDiagram participant Pasien participant Front Office participant Chasier participant Medicare participant Medical Records participant Pharmacy Pasien->>Chasier: 1: Pembayaran() Chasier->>Front Office: 1.2: Notifikasi pembayaran() Chasier->>Pharmacy: 1.1: Notifikasi pembayaran() </pre>	<p>Valid</p>
<p>Transaksi obat</p>	<pre> sequenceDiagram participant Pasien participant Front Office participant Chasier participant Medicare participant Medical Records participant Pharmacy Chasier->>Pharmacy: 1: Request obat() Pharmacy->>Chasier: 2: Get obat() Pharmacy->>Front Office: 3: Notifikasi transaksi obat pasien() </pre>	<p>Valid</p>
<p>Ubah data pengguna</p>	<pre> sequenceDiagram participant Employee participant Human Resource Employee->>Human Resource: 1: Request edit biodata() Human Resource->>Employee: 2: Validasi ubah biodata() </pre>	<p>Valid</p>

Sumber: Pengujian

5.2 Analisis

Proses Analisis bertujuan untuk mendapatkan kesimpulan dari hasil pengujian perangkat lunak Sistem Enterprise Rumah Sakit menggunakan *design pattern* Model View Controller (MVC). Proses analisis mengacu pada dasar teori sesuai dengan hasil pengujian di setiap tahap pengujian. Proses analisis yang dilakukan meliputi analisa Pengujian User Acceptance Test (UAT), analisa pengujian *functionality*, dan analisa pengujian integrasi.

5.2.1 Analisis Pengujian *Functionality*

Proses analisis terhadap hasil pengujian *Functionality* dilakukan dengan melihat kinerja aplikasi. Berdasarkan hasil pengujian tersebut dapat disimpulkan bahwa Sistem Enterprise Rumah Sakit menggunakan *design pattern* Model View Controller (MVC) telah berjalan sesuai dengan kebutuhan sementara bisnis proses Poliklinik Universitas Brawijaya.

5.2.2 Analisis Pengujian *User Acceptance Test* (UAT)

Proses analisis terhadap hasil pengujian *User Acceptance Test* (UAT) dilakukan dengan melihat hasil penyebaran kuisisioner yang diberikan kepada staff Poliklinik Universitas Brawijaya. Berdasarkan hasil pengujian tersebut dapat disimpulkan bahwa Sistem Enterprise Rumah Sakit menggunakan *design pattern* Model View Controller (MVC) dapat diterima dan diterapkan di Poliklinik Universitas Brawijaya dengan memberikan pelatihan terlebih dahulu kepada setiap pengguna.

5.2.3 Analisis Pengujian *Integration*

Proses analisis terhadap hasil pengujian *Integration* dilakukan dengan melihat integrasi aplikasi antar modul dilihat dari proses bisnis yang ada. Berdasarkan hasil pengujian tersebut dapat disimpulkan bahwa Sistem Enterprise Rumah Sakit menggunakan *design pattern* Model View Controller (MVC) telah terintegrasi sesuai dengan pemodelan sistemn dengan *sequence diagram*.

BAB VI PENUTUP

6.1 Kesimpulan

1. Untuk menganalisis kebutuhan sistem, dilakukan survei langsung pada Poliklinik Universitas Brawijaya dalam menyusun gambaran proses bisnis yang ada pada saat itu (as-is). Dari hasil analisis tersebut dibuat desain perancangan untuk proses rancang bangun Sistem Enterprise Rumah Sakit menggunakan *design pattern* Model View Controller (MVC).
2. Sistem Enterprise Rumah Sakit dibangun menggunakan aplikasi berbasis web yang dibangun di atas *design pattern* MVC (Model View Controller) dan *framework codeigniter*.
3. Berdasarkan hasil Pengujian User Acceptance Test (UAT), pengujian *user acceptance test* (UAT) dan pengujian integrasi telah menunjukkan bahwa sistem telah layak dan dapat digunakan untuk kepentingan poliklinik pada tahap awal. Hasil kuisioner didapatkan bahwa pengguna telah dapat menerima sistem Sistem Enterprise Rumah Sakit menggunakan *design pattern* Model View Controller (MVC) ini. Hasil pengujian menunjukkan bahwa sistem 100% valid

6.2 Saran

1. Untuk pengembangan selanjutnya disarankan menambah fitur yang terdapat dalam setiap modul serta menambah modul yang belum tercakup secara keseluruhan.
2. Untuk pengembangan mobilitas aplikasi dapat juga dikembangkan versi *mobile*.

DAFTAR PUSTAKA

- [MIK-11] Akiyama, Miki. 2011. *“Impact of information sharing through electric medical network between pharmacist and physician on patient care in Japan”*. Journal of Keio University.
- [CHO-11] Chon Abraham. 2011. *“Transforming healthcare with information technology in Japan: A review of policy, people, and progress”*. International Journal of Medical Informatics, Vol 80, hal. 157-170.
- [FAU-08] Fauci AS, et al. 2008. *Harrison’s Principles of Internal Medicine 17th edition*. New York : The McGraw-Hill Companies, Inc.
- [GDS-11] Goods. 2011. *Interest Your Child in Educational Computer Aplikasi*. <http://www.edmontonnaoshweek.com> (Online 5 Maret 2013).
- [LEO-10] Jackson, Leonard A. 2010. *“Enterprise resource planning systems: revolutionizing lodging human resources management”*. Kemmons Wilson School of Hospitality & Resort Management, University of Memphis, Memphis, Tennessee, USA.
- [MER-04] Merode, Godefridus G Van, etc. 2004. *Enterprise Resource Planning for Hospitals*. International journal of Medical Informatics, Vol 73, hal. 493-501.
- [PRE-01] Pressman, Roger S. 2001. *Software Engineering : A Practitioner’s Approach, Fifth Edition*. McGraw Hill.
- [ROS-11] A. S., Rosa & Shalahudin, M. 2011. *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.
- [SEG-07] S. Egenfeldt-Nielsen. 2007. *Overview of research on the educational use of video aplikasis*. Digital kompetensi. Nordic Journal of Digital Literacy.
- [SOF-07] Sofwan, Akhmad. 2007. *Belajar PHP dengan Framework Code Igniter*. Komunitas eLearning IlmuKomputer.Com
- [SOM-03] Sommerville, Ian. 2003. *Software Engineering (Rekayasa Perangkat Lunak) / Edisi 6 / Jilid 1*. Jakarta : Penerbit Erlangga
- [SOM-06] Sommerville, Ian. 2006. *Software Engineering (Rekayasa Perangkat Lunak) / Edisi / Jilid*. Jakarta : Penerbit Erlangga
- [MAR-09] Graban, Mark. 2009. *“Statistics on Healthcare Quality and Patient Safety Problems – Errors & Harm (Updated April 10, 2013)”*. <http://www.leanblog.org/2009/08/statistics-on-healthcare-quality-and/> . Diakses pada tanggal 22 Oktober 2013
- [JDU-08] Rubin, Jeff dan Cihsnell Dana. 2008, *“Handbook of Usability Testing”*, Second Edition, Willey Publishing, Indianapolis.
- [AGS-09] Mudaliar, Garde, dan Sharma, D.P. 2009, "Educational Resource Planning - A Framework for Educational Institutions", Vol.09, No. 978-0-7695-3884-6, Hal. 488 - 493.
- [SSS-04] Shehab, E.M, dkk. 2004, "Enterprise Resource Planning An Integrated Review", Emerald Group Publishing Limited
- [SDL-13] *“SDLC Tutorial”*, Tutorialspoint.com.