



INSTITUT TEKNOLOGI DEL

Rancang Bangun Sistem Informasi Rumah Sakit

**(Studi Kasus : Rumah Sakit Umum Daerah Doloksanggul
Kabupaten Humbang Hasundutan Provinsi Sumatera Utara)**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Diploma III

11315006
11315017
11315027

Triana C. Baringbing
Leo Parhaposan Pakpahan
Berta Novalin Nainggolan

**FAKULTAS TEKNIK INFORMATIKA DAN ELEKTRO
PROGRAM STUDI DIII TEKNIK INFORMATIKA
LAGUBOTI
SEPTEMBER 2018**

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Triana C. Baringbing

NIM : 11315006

Tanda Tangan :

Tanggal : 03 September 2018

Nama : Leo Parhaposan Pakpahan

NIM : 11315017

Tanda Tangan :

Tanggal : 03 September 2018

Nama : Berta Novalin Nainggolan

NIM : 11315027

Tanda Tangan :

Tanggal : 03 September 2018

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh

1. Nama	:	Triana C. Baringbing
NIM	:	11315006
Program Studi	:	DIII Teknik Informatika
2. Nama	:	Leo Parhaposan Pakpahan
NIM	:	11315017
Program Studi	:	DIII Teknik Informatika
3. Nama	:	Berta Novalin Nainggolan
NIM	:	11315027
Program Studi	:	DIII Teknik Informatika
Judul Tugas Akhir	:	Rancang Bangun Sistem Informasi Rumah Sakit (Studi Kasus : Rumah Sakit Umum Daerah Doloksanggul Kabupaten Humbang Hasundutan Provinsi Sumatera Utara)

Telah berhasil dipertahankan dihadapan Dewan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Diploma III, pada program studi Diploma III Teknik Informatika, Fakultas Teknik Informatika dan Elektro, Institut Teknologi Del.

DEWAN PENGUJI

Pembimbing : Roy Deddy Hasiholan Lumban Tobing, S.T, M.ICT ()

Penguji : Inte Christinawati Bu'ulolo, ST., M.T.I ()

Penguji : Lit Malem Ginting, S.Si, MT ()

Ditetapkan di : Laguboti

Tanggal : 03 September 2018

KATA PENGANTAR

Puji dan syukur kepada Tuhan Yang Maha Esa atas berkat dan rahmat yang diberikan kepada penulis selama pengerjaan Tugas Akhir serta penyusunan Laporan Tugas Akhir yang dapat berjalan dengan baik. Laporan Tugas Akhir ini ditulis sebagai syarat kelulusan Diploma III Institut Teknologi Del. Laporan Tugas Akhir ini bertujuan untuk mendokumentasikan hasil Tugas Akhir mengenai Rancang Bangun Sistem Informasi Rumah Sakit (Studi Kasus : Rumah Sakit Umum Daerah Doloksanggul Kabupaten Humbang Hasundutan Provinsi Sumatera Utara).

Penulis mengucapkan terima kasih kepada Bapak Roy Deddy Hasiholan Lumban Tobing, S.T, M.ICT. selaku pembimbing atas masukan, bimbingan dan arahan selama pengerjaan Tugas Akhir. Penulis juga mengucapkan terima kasih kepada penguji TA 1 (Ibu Inte Christinawati Bu'ulolo, S.T., M.T.I. dan Ibu Yuniarta Basani, S.Si., M.Si.) yang telah memberikan review dan memberikan masukan selama pengerjaan Tugas Akhir serta orang tua dan semua pihak yang sudah memberikan dukungan, masukan serta saran kepada penulis dalam pengerjaan Tugas Akhir ini.

Penulis berharap laporan Tugas Akhir ini dapat bermanfaat bagi semua pihak yang memerlukannya. Penulis juga menyadari bahwa laporan Tugas Akhir ini masih memiliki banyak kekurangan, sehingga penulis mengharapkan saran dan kritik yang membangun.

Laguboti, 03 September 2018

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Institut Teknologi Del, penulis yang bertanda tangan di bawah ini:

1. Nama : Triana C. Baringbing
NIM : 11315006
Program Studi : DIII Teknik Informatika
2. Nama : Leo Parhaposan Pakpahan
NIM : 11315017
Program Studi : DIII Teknik Informatika
3. Nama : Berta Novalin Nainggolan
NIM : 11315027
Program Studi : DIII Teknik Informatika
Fakultas : Teknik Informatika dan Elektro
Jenis Karya : Tugas Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Del **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah penulis yang berjudul:

Rancang Bangun Sistem Informasi Rumah Sakit (Studi Kasus : Rumah Sakit Umum Daerah Doloksanggul Kabupaten Humbang Hasundutan Provinsi Sumatera Utara). Dengan Hak Bebas Royalti Noneksklusif ini Institut Teknologi Del berhak menyimpan, mengalih/media-format dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir penulis selama tetap mencantumkan nama penulis sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini penulis buat dengan sebenarnya.

Dibuat di : Laguboti

Pada tanggal : 03 September 2018

Yang menyatakan

(Triana C. Baringbing)

(Leo P. Pakpahan)

(Berta N. Nainggolan)

ABSTRAK

Sistem informasi rumah sakit adalah suatu rangkaian yang berhubungan dengan pengumpulan dan pengelolaan data, penyajian informasi, analisis, dan juga penyampaian informasi yang dibutuhkan untuk kegiatan di rumah sakit. Saat ini masih ada subsistem di rumah sakit yang belum terintegrasi antara satu dengan yang lain, sehingga bila diperlukan informasi yang menyeluruh diperlukan waktu yang cukup lama.

Tujuan dibangunnya sistem informasi rumah sakit ini adalah untuk melakukan perancangan dan pembangunan sistem informasi rumah sakit yang terintegrasi. Metode yang digunakan dalam rancang bangun sistem informasi rumah sakit adalah *Model Driven Development* (MDD) yang dapat membantu *developer* seperti *code engineering*. MDD juga menerapkan *Software Development Life Cycle* (SDLC) yang kritis terhadap proses pengembangan aplikasi yang terdiri dari 4 (empat) tahap, yaitu *inception*, *elaboration*, *construction*, dan *transition*.

Pengerjaan Tugas Akhir ini telah menghasilkan sistem informasi rumah sakit berbasis *desktop*, *mobile*, dan *web*. Aplikasi *desktop* digunakan oleh *admin*, pegawai, dan kasir. Aplikasi *mobile* digunakan oleh pegawai dan dokter, sedangkan *web* digunakan oleh direktur.

Keywords: Sistem informasi, rumah sakit, MDD, and SDLC

ABSTRACT

Hospital information system is a connection that relates in data collecting and data management, presentation of information, analysis, and also information's delivery that needed for activity in hospital. In this time, there is subsystem in hospital that has not been integrated with one another. In this time, there is subsystem in hospital that has not been integrated with one another, so it takes a long time for the comprehensive information. The purpose of building hospital information system is to design and build integrated hospital information system. The method that used to design hospital information system is Model Driven Development (MDD) that can help developer like code engineering. MDD also apply Software Development Life Cycle (SDLC) that critical for application development's process that consist of 4 phase, that are inception, elaboration, construction, and transition. This final project has delivered hospital information system desktop based, mobile based, and web based. Desktop application is used by admin, employees, and cashier. Mobile application is used by employees and doctor, while web application is used by director.

Keywords: Information system, hospital, MDD, and SDLC

DAFTAR ISI

KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN	xiv
Bab I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	3
1.3 Lingkup	3
1.4 Pendekatan	4
1.5 Sistematika Penyajian	5
Bab II TINJAUAN PUSTAKA	6
2.1 Software Development Life Cycle (SDLC).....	6
2.2 Model Driven Development (MDD).....	10
2.3 Sistem Informasi	12
2.3.1.1 Rawat Jalan	18
2.3.2 Rawat Inap	18
2.3.3 Rekam Medis	18
2.3.4 Billing System.....	20
2.4 Mobile Application	20
2.4.1 Mobile Web Application.....	20
2.4.2 Mobile Native Application.....	21
2.4.3 Mobile Hybrid Application	21
2.5 Desktop Application.....	22
2.6 Web Application	22
2.7 Platform Android Mobile	24
2.7.1 Fitur-fitur Android	24
2.7.2 Arsitektur Android	25
2.8 Android Studio vs. Eclips	27
2.9 Microsoft SQL Server vs. MySQL	28
2.10 Framework Laravel	31
2.11 JavaFX.....	34
Bab III PERENCANAAN DAN ANALISIS.....	34
3.1 Inisialisasi Proyek	34
3.2 Project Plan	34
3.3 Pengumpulan Data	34
3.3.1 Persiapan Survei.....	34
3.3.2 Pelaksanaan Survei.....	35
3.3.3 Hasil Survei.....	36
3.4 Analisis Registrasi, Billing System, dan Rekam Medis.....	36
3.4.1 Analisis Registrasi.....	36
3.4.2 Analisis Billing System.....	37
3.4.2.3 Analisis Rekam Medis	40

3.5	Current System.....	44
3.6	Target System.....	48
3.6.1	User Characteristics	51
3.6.2	Use Case Diagram.....	54
Bab IV	DESAIN	55
4.1	Rancangan Diagram Integrasi	55
4.2	Data Requirement.....	55
4.3	Desain Aplikasi Sistem Informasi Rumah Sakit.....	58
4.3.1	Domain Model	59
4.3.2	Conceptual Data Model (CDM).....	60
4.3.3	Physical Data Model (PDM)	62
4.3.4	Class Diagram	64
4.3.5	Sequence Diagram	65
4.4	Desain Antarmuka Aplikasi	75
4.5	Functional Requirement.....	75
Bab V	IMPLEMENTASI	77
5.1	Kebutuhan Implementasi	77
5.2	Batasan Implementasi	77
5.3	Implementasi Aplikasi	78
Bab VI	HASIL DAN PEMBAHASAN	77
6.1	Hasil Aplikasi.....	77
6.2	Penerapan MDD pada Tugas Akhir	77
6.3	Pengaruh MDD pada Tugas Akhir.....	78
Bab VII	PENGUJIAN.....	77
7.1	Tujuan Pengujian	77
7.2	Skenario Pengujian.....	77
Bab VIII	KESIMPULAN DAN SARAN.....	99
8.1	Kesimpulan	99
8.2	Saran.....	99
DAFTAR PUSTAKA		100

DAFTAR GAMBAR

Gambar 1. Distribusi rumah sakit di Indonesia yang memiliki SIMRS fungsional (sumber : Olahan dari SIRS November 2016)	2
Gambar 2. Software Development Life Cycle	6
Gambar 3. Definisi Sistem Informasi.....	12
Gambar 4. Sistem Informasi yang sederhana.....	13
Gambar 5. Sistem Informasi Rumah Sakit.....	15
Gambar 6. Diagram Aplikasi Web.....	23
Gambar 7. Arsitektur Android	25
Gambar 8. Perbandingan Framework PHP	32
Gambar 9. Current System Registrasi Rawat Jalan	44
Gambar 10. Current System Registrasi Rawat Inap.....	45
Gambar 11. Current System Proses Rekam Medis	46
Gambar 12. Current System Proses Billing System	47
Gambar 13. Target System Registrasi Rawat Jalan	48
Gambar 14. Target System Registrasi Rawat Inap	49
Gambar 15. Target System Proses Rekam Medis	50
Gambar 16. Target System Proses Billing System	51
Gambar 17. Use Case Diagram	54
Gambar 18. Rancangan Diagram Integrasi	55
Gambar 19. E-R Diagram Desktop	56
Gambar 20. Domain Model.....	59
Gambar 21. Conceptual Data Model (CDM)	61
Gambar 22. Physical Data Model (PDM)	63
Gambar 23. Class Diagram	64
Gambar 24. Sequence Diagram Login	65
Gambar 25. Sequence Diagram Melihat Data Dokter	65
Gambar 26. Sequence Diagram Melihat Data Pasien	66
Gambar 27. Sequence Diagram Mengelola Ruangan	66
Gambar 28. Sequence Diagram Melihat Rekam Medis Pasien	67
Gambar 29. Sequence Diagram Membuat Data Cuti Dokter.....	68
Gambar 30. Sequence Diagram Membuat Rekam Medis Pasien	68
Gambar 31. Sequence Diagram Mengelola Hasil Pemeriksaan.....	69
Gambar 32. Sequence Diagram Mengelola ICD.....	70
Gambar 33. Sequence Diagram Mengelola Dokter	70
Gambar 34. Sequence Diagram Mengelola Jadwal Praktek Dokter	71
Gambar 35. Sequence Diagram Mengelola Obat.....	71
Gambar 36. Sequence Diagram Mengelola Pasien Rawat Inap.....	72
Gambar 37. Sequence Diagram Mengelola Pasien Rawat Jalan.....	72
Gambar 38. Sequence Diagram Mengelola Perawat.....	73
Gambar 39. Sequence Diagram Mengelola Pembayaran.....	73
Gambar 40. Sequence Diagram Mengelola Tindakan	74
Gambar 41. Sequence Diagram Mengelola Tarif Dokter	74
Gambar 42. Desain Aplikasi Desktop – Home Login untuk Pegawai Registrasi	126
Gambar 43. Desain Aplikasi Desktop – Form Pasien.....	126
Gambar 44. Desain Aplikasi Desktop – PopUp Simpan Data Pasien	126

Gambar 45. Desain Aplikasi Desktop - PopUp Data Pasien Berhasil Disimpan	126
Gambar 46. Desain Aplikasi Desktop – Edit Data Pasien	127
Gambar 47. Desain Aplikasi Desktop - PopUp Hapus Data Pasien	127
Gambar 48. Desain Aplikasi Desktop – PopUp Gagal Hapus Data Pasien	127
Gambar 49. Desain Aplikasi Desktop - PopUp Berhasil Hapus Data Pasien	127
Gambar 50. Desain Aplikasi Desktop - Home sebelum login untuk Pegawai Rekam Medis	128
Gambar 51. Desain Aplikasi Desktop - Login untuk Pegawai Rekam Medis	128
Gambar 52. Desain Aplikasi Desktop - Home Setelah Login untuk Pegawai Rekam Medis	128
Gambar 53. Desain Aplikasi Desktop - Form Rawat Inap	128
Gambar 54. Desain Aplikasi Desktop - Form Rawat Jalan	129
Gambar 55. Desain Aplikasi Desktop - Data Rekam Medis.....	129
Gambar 56. Desain Aplikasi Desktop - Home Sebelum Login untuk Perawat dan Dokter.....	129
Gambar 57. Desain Aplikasi Desktop - Login untuk Perawat dan Dokter	129
Gambar 58. Desain Aplikasi Desktop - Home Setelah Login untuk Perawat dan Dokter.....	130
Gambar 59. Desain Aplikasi Desktop - Antrian Pasien untuk Rawat Jalan	130
Gambar 60. Desain Aplikasi Desktop - Home Sebelum Login untuk Admin	130
Gambar 61. Desain Aplikasi Desktop - Login untuk Admin.....	130
Gambar 62. Desain Aplikasi Desktop - Home Setelah Login untuk Admin	131
Gambar 63. Desain Aplikasi Desktop - Form Obat	131
Gambar 64. Desain Aplikasi Desktop - Form ICD	131
Gambar 65. Desain Aplikasi Desktop - Form Ruangan.....	131
Gambar 66. Desain Aplikasi Desktop - Form Perawat.....	132
Gambar 67. Desain Aplikasi Desktop - Form Akun	132
Gambar 68. Desain Aplikasi Desktop - Form Data Dokter	132
Gambar 69. Desain Aplikasi Desktop - Form Tarif Dokter.....	132
Gambar 70. Desain Aplikasi Desktop - Form Jadwal Praktek Dokter	133
Gambar 71. Desain Aplikasi Desktop - Form Cuti Dokter.....	133
Gambar 72. Desain Aplikasi Desktop - Form Tindakan.....	133
Gambar 73. Desain Aplikasi Desktop - Home Login untuk Kasir	133
Gambar 74. Desain Aplikasi Desktop - Login Kasir	134
Gambar 75. Desain Aplikasi Desktop - Form Pembayaran	134
Gambar 76. Desain Aplikasi Mobile - Home.....	135
Gambar 77. Desain Aplikasi Mobile – Beranda	135
Gambar 78. Desain Aplikasi Mobile - Beranda Hasil Pemeriksaan	135
Gambar 79. Desain Aplikasi Mobile – Pasien	136
Gambar 80. Desain Aplikasi Mobile - Search Pasien	136
Gambar 81. Desain Aplikasi Mobile – Lihat Detail Pasien	136
Gambar 82. Desain Aplikasi Mobile - Detail Pasien	136
Gambar 83. Desain Aplikasi Mobile - Pemeriksaan.....	137
Gambar 84. Desain Aplikasi Mobile - Hasil Pemeriksaan	137
Gambar 85. Desain Aplikasi Mobile - Edit Data Pemeriksaan.....	137
Gambar 86. Desain Aplikasi Mobile - Notifikasi Data Pemeriksaan Berhasil Diubah	137
Gambar 87. Desain Aplikasi Mobile - Hapus Hasil Pemeriksaan	138

Gambar 88. Desain Aplikasi Mobile - Notifikasi Hasil Pemeriksaan Berhasil Dihapus	138
Gambar 89. Desain Aplikasi Mobile - Search Dokter	138
Gambar 90. Desain Aplikasi Mobile - Detail Dokter	138
Gambar 91. Tampilan Rawat Inap pada Desktop	140
Gambar 92. Tampilan Rawat Jalan pada Desktop	141
Gambar 93. Tampilan Rekam Medis pada Desktop	141
Gambar 94. Tampilan Mengelola Ruangan pada Desktop	142
Gambar 95. Tampilan Mengelola Pasien pada Desktop	142
Gambar 96. Tampilan Mengelola Pembayaran pada Desktop.....	143
Gambar 97. Tampilan Melihat Daftar Antrian pada Desktop.....	143
Gambar 98. Tampilan Mengelola Dokter	144
Gambar 99. Tampilan Mengelola Tarif Dokter	144
Gambar 100. Tampilan Mengelola Jadwal Praktek Dokter	145
Gambar 101. Tampilan Mengelola Cuti Dokter	145
Gambar 102. Tampilan Mengelola Perawat.....	146
Gambar 103. Tampilan Mengelola Data Obat	146
Gambar 104. Tampilan Mengelola Tindakan	147
Gambar 105. Tampilan Mengelola ICD.....	147
Gambar 106. Tampilan Login	148
Gambar 107. Tampilan Halaman Utama Perawat.....	149
Gambar 108. Tampilan Halaman Utama Dokter	149
Gambar 109. Tampilan Hasil Pemeriksaan.....	149
Gambar 110. Tampilan Mengelola Dokter	150
Gambar 111. Tampilan Login pada Web.....	151
Gambar 112. Tampilan dashboard direktur	151

DAFTAR TABEL

DAFTAR LAMPIRAN

Lampiran 1 – Use Case Scenario	102
Use Case Scenario Login	102
Use Case Scenario Mengelola Data Pasien.....	102
Use Case Scenario Melihat Data Pasien	103
Use Case Scenario Mengelola Data ICD	104
Use Case Scenario Mengelola Data Dokter	105
Use Case Scenario Mengelola Tarif Dokter.....	106
Use Case Scenario Mengelola Jadwal Praktek Dokter	107
Use Case Scenario Mengelola Cuti Dokter.....	109
Use Case Scenario Melihat Data Dokter.....	110
Use Case Scenario Mengelola Obat	110
Use Case Scenario Mengelola Data Perawat	112
Use Case Scenario Mengelola Data Tindakan	113
Use Case Scenario Mengelola Ruangan	114
Use Case Scenario Mengelola Rawat Jalan	115
Use Case Scenario Mengelola Rawat Inap	117
Use Case Scenario Mengelola Rekam Medis	118
Use Case Scenario Mengelola Hasil Pemeriksaan.....	119
Use Case Scenario Mengelola Pembayaran	120
Lampiran 2 – List Pertanyaan	121
Lampiran 3 – Desain Antarmuka Aplikasi.....	126
Lampiran 4 – Hasil Aplikasi	140
Antarmuka Desktop	140
Antarmuka Mobile	148
Antarmuka Web	151
Lampiran 5 – Hasil UAT	152

BAB 1 PENDAHULUAN

Pada Bab I Pendahuluan ini dimuat mengenai latar belakang pemilihan topik, tujuan pelaksanaan Tugas Akhir, yang menjadi lingkup kajian dalam penelitian, pendekatan yang dilakukan dalam pelaksanaan kajian, serta sistematika penyajian materi yang disediakan dalam laporan ini.

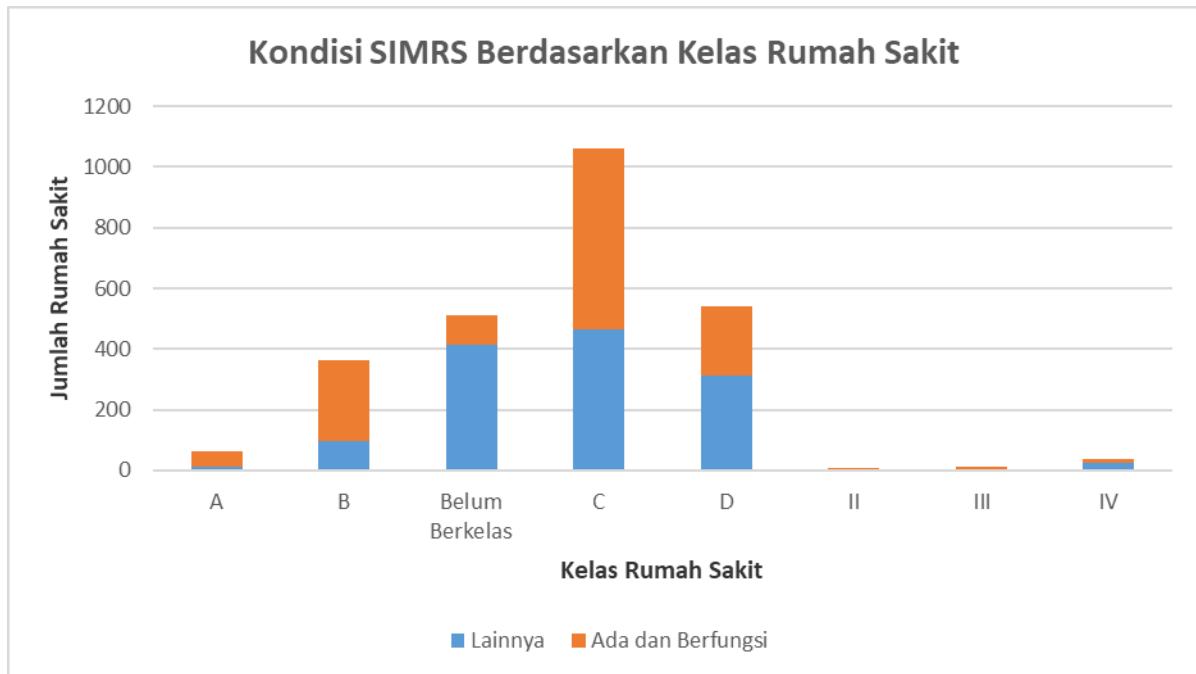
1.1 Latar Belakang

Teknologi informasi merupakan teknologi yang sedang berkembang pesat saat ini. Teknologi informasi yang semakin maju membuat pengaksesan terhadap data atau informasi yang ada berlangsung dengan cepat, akurat, dan efisien[1].

Rumah sakit merupakan suatu organisasi yang menghasilkan data yang sangat beragam. Beragamnya informasi yang dihasilkan menuntut pengelolaan, dimulai dari data yang diperoleh, diproses hingga menghasilkan suatu informasi. Sebagian besar rumah sakit masih melakukan pengolahan data secara manual, mulai dari pencatatan identitas pasien hingga pembuatan laporan. Data kegiatan pelayanan di setiap unit rumah sakit diarsipkan dalam bentuk *paper-based* dan membutuhkan ruang penyimpanan yang luas sehingga terjadi kesulitan dalam pencarian data dan informasi yang dibutuhkan. Namun demikian, pengolahan data secara manual dianggap sudah tidak optimal dalam pelayanan kesehatan yang semakin kompleks, sehingga beberapa Rumah Sakit atau Rumah Sakit Khusus mulai mengadopsi teknologi informasi untuk mengatasi kendala yang ada dan untuk meningkatkan efektivitas dan efisiensi pelayanan kesehatan[2].

Sistem informasi mempunyai peranan penting dalam hal mendukung proses pelayanan kesehatan, yaitu mendukung proses dan operasi pelayanan kesehatan, pengambilan keputusan staf dan manajemen, serta strategi untuk memperoleh keunggulan kompetitif. Selain itu, sistem informasi yang baik dapat mendukung alur kerja klinis yang akan memberikan kontribusi untuk perawatan pasien yang lebih baik[33].

Sesuai dengan data yang diperoleh oleh Kemenkes melalui Sistem Informasi Rumah Sakit bahwa sampai dengan akhir November 2016 dilaporkan bahwa terdapat 1257 dari 2588 rumah sakit (sekitar 48%) di Indonesia telah memiliki Sistem Informasi Manajemen Rumah Sakit yang fungsional. Selain itu, terdapat 128 rumah sakit (sekitar 5%) yang melaporkan bahwa sudah memiliki SIMRS namun tidak berjalan secara fungsional, 425 rumah sakit (16%) yang belum memiliki SIMRS, dan 745 rumah sakit (28%) yang tidak memberikan laporan apakah sudah memiliki SIMRS atau belum[3].



Gambar 1. Distribusi rumah sakit di Indonesia yang memiliki SIMRS fungsional (sumber : Olahan dari SIRS November 2016)

Fakta menunjukkan bahwa beberapa rumah sakit sudah menggunakan atau mulai menggunakan teknologi informasi dalam pelayanan kesehatan, baik pengembangan sendiri, bekerja sama dengan pihak lain, maupun *outsourcing*[1]. Selain itu, Menteri Kesehatan Republik Indonesia Nomor 1171/Menkes/Per/VI/2011 juga telah menetapkan bahwa setiap rumah sakit wajib melakukan pencatatan dan pelaporan tentang semua kegiatan penyelenggaraan rumah sakit dalam bentuk sistem informasi rumah sakit. Sistem informasi rumah sakit secara umum bertujuan untuk mengintegrasikan sistem informasi dari berbagai subsistem, mengumpulkan, menyajikan dan mengolah data rumah sakit sehingga menghasilkan informasi yang dibutuhkan. Namun, tujuan tersebut belum tercapai dikarenakan sistem informasi yang dimiliki oleh masing-masing program belum terintegrasi, sehingga bila diperlukan informasi yang menyeluruh diperlukan waktu yang cukup lama[4].

Rumah Sakit Umum Daerah Doloksanggul adalah sebuah rumah sakit yang berada di Doloksanggul yang merupakan salah satu layanan kesehatan milik PEMKAB (Pemerintah Kabupaten) Humbang Hasundutan, yang tercatat kedalam Rumah Sakit Tipe C dan memiliki beberapa layanan dalam meningkatkan kesehatan masyarakat. Rumah Sakit Umum Daerah Doloksanggul ini telah memiliki sistem informasi dalam bentuk web.

RSUD Doloksanggul memiliki kendala dalam pengembangan aplikasi yaitu dikarenakan terdapat aturan bahwa Departemen Komunikasi dan Informasi (KOMINFO) mewajibkan

untuk menyerahkan segala tanggung jawab pembangunan sistem informasi rumah sakit kepada mereka sepenuhnya tanpa adanya campur tangan dari pihak manapun. Pembangunan sistem informasi ini berjalan sangat lamban sehingga pihak rumah sakit doloksanggul sendiri yang akhirnya membangun dan hingga pada saat ini pihak rumah sakit tetap melakukan pengembangan.

Oleh karena itu, peneliti akan membangun sistem informasi rumah sakit yang sudah terintegrasi antara satu subsistem dengan subsistem yang lain dimana setiap subsistem diimplementasikan dalam tiga aplikasi berbeda, yaitu aplikasi *desktop*, *mobile*, dan *web*. Pembangunan aplikasi *mobile* didukung oleh tingginya minat masyarakat dunia dalam penggunaan teknologi ini, dikarenakan kemudahan yang didapat dalam mengakses konten yang terdapat di dalamnya[34]. Aplikasi *desktop* memiliki kelebihan dalam hal banyaknya operasi yang dapat dilakukan pada sisi *client* dan kehandalan eksekusi *real-time*. Selain itu jika menggunakan *desktop* cukup menginstal aplikasi hanya sekali, tidak seperti *web* yang selalu membuka alamat *website*. Sedangkan kelebihan aplikasi *web* adalah dalam hal pengaksesan data yang hampir tanpa batas, fleksibilitas, dan portabilitasnya[35]. Integrasi dalam sistem ini dicapai dengan adanya arsitektur aplikasi, dimana data dari ketiga aplikasi akan dihubungkan dalam satu server sehingga dapat diakses dan dikelola dalam sistem sehingga pengisian data di salah satu subsistem akan mempengaruhi subsistem yang lain sebab data di subsistem yang lain akan bertambah juga. Sebagai contoh, data yang diisi dari aplikasi *mobile* dapat diakses dari aplikasi *desktop* dan setelah itu dapat diolah di aplikasi *desktop* itu sendiri.

1.2 Tujuan

Adapun tujuan yang akan ditempuh pada Tugas Akhir ini adalah untuk melakukan perancangan dan pembangunan sistem informasi rumah sakit yang terintegrasi.

1.3 Lingkup

Pada penggeraan aplikasi sistem informasi ini akan mencakup beberapa hal, yaitu:

1. Aplikasi sistem informasi yang akan dibangun berbasis *web*, *desktop* dan *mobile*, dimana ketiganya akan dihubungkan oleh web service.
2. Modul yang akan dikerjakan mencakup modul registrasi, modul pembayaran, dan modul rekam medik.
3. Pihak yang menjadi narasumber dalam penelitian adalah pihak Rumah Sakit Umum Daerah Doloksanggul.

1.4 Pendekatan

Metode penelitian yang akan digunakan selama pengerjaan tugas akhir ini adalah penelitian terapan (*applied research*). Penelitian terapan adalah penelitian yang dilakukan berkenaan dengan pemecahan masalah[5]. Tujuan dari penelitian terapan ini selain untuk mengembangkan wawasan keilmuan dan digunakan juga untuk pemecahan masalah praktis, sehingga hasil penelitian yang dilakukan dapat dimanfaatkan.

Pendekatan yang akan digunakan dalam rancang bangun sistem informasi ini adalah SDLC (*System Development Life Cycle*) yaitu siklus hidup pengembangan sistem perangkat lunak yang terdiri dari beberapa fase berdasarkan analisa kebutuhan yang ada.

Tahap-tahap dari SDLC adalah sebagai berikut[6].

1. Perencanaan

Tahap perencanaan adalah proses dasar dalam memahami mengapa sistem informasi harus dibangun dan menentukan bagaimana tim proyek akan membangun sistem tersebut.

2. Analisis

Tahap analisis menjawab pertanyaan tentang siapa yang akan menggunakan sistem, apa yang akan dilakukan sistem, dan kapan akan digunakan. Selama tahap ini, tim proyek akan mengidentifikasi sistem apa yang sedang berjalan saat ini, peluang peningkatan, dan mengembangkan sebuah konsep pada sistem yang baru.

3. Desain

Tahap desain membahas mengenai bagaimana sistem nantinya akan beroperasi, baik dalam hal perangkat keras, perangkat lunak, dan infrastruktur jaringan, *user interface*, *form* dan laporan, dan program khusus lainnya, database, dan file-file yang diperlukan.

4. Implementasi

Tahap terakhir dari SDLC adalah tahap implementasi. Pada tahap implementasi, sistem akan mulai dibangun. Tahap ini biasanya merupakan tahap yang paling penting dikarenakan dalam pembangunan sistem dibutuhkan waktu yang lama dan bagian paling berharga dalam proses pembangunan.

1.5 Sistematika Penyajian

Secara garis besar laporan tugas akhir ini terdiri dari tujuh bab. Sistematika dalam penulisan dokumen ini adalah sebagai berikut :

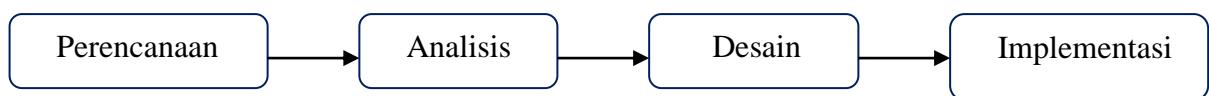
1. Bab I merupakan Pendahuluan yang menjelaskan mengenai Latar Belakang, Tujuan, Lingkup, Pendekatan, dan Sistematika Penyajian.
2. Bab II merupakan Tinjauan Pustaka yang di dalamnya diuraikan dasar-dasar teori yang relevan dengan topik tugas akhir ini.
3. Bab III merupakan Perencanaan dan Analisis yang menjelaskan mengenai inisialisasi proyek, *project plan* dan analisis yang dilakukan untuk mendapatkan rancangan aplikasi yang akan dikembangkan.
4. Bab IV merupakan Desain yang mendeskripsikan perancangan dari sistem dan *stackholder* yang berperan serta *role* atau tanggung jawab masing-masing *stackholder* terhadap sistem, yang keseluruhannya digambarkan dalam bentuk diagram UML.
5. Bab V merupakan Implementasi yang menjelaskan deskripsi umum sistem yang meliputi kebutuhan implementasi, batasan implementasi, dan implementasi aplikasi. Pada bab ini juga dijelaskan mengenai langkah-langkah awal implementasi mulai dari mempersiapkan perangkat keras dan perangkat lunak yang diperlukan.
6. Bab VI merupakan Hasil dan Pembahasan yang menjelaskan hasil yang didapat setelah melakukan tahap implementasi sistem, kendala yang dihadapi saat pembangunan sistem, serta kekurangan pada sistem setelah diimplementasi.
7. Bab VII merupakan Pengujian yang menjelaskan skenario pengujian yang akan dilakukan terhadap aplikasi yang akan dibangun.
8. Bab VIII merupakan Kesimpulan dan Saran yang menjelaskan kesimpulan yang didapat setelah melalui tahap hasil dan implementasi dan memberikan saran apabila memungkinkan sistem ini untuk dikembangkan lebih lagi.

BAB 2 TINJAUAN PUSTAKA

Bab II Tinjauan Pustaka berisi uraian mengenai informasi yang dikumpulkan dari berbagai pustaka yang berhubungan dengan bahan kajian dari Tugas Akhir untuk memperluas pemahaman dalam objek kajian.

2.1 Software Development Life Cycle (SDLC)

SDLC adalah sistem informasi berbasis komputer yang digunakan untuk merancang dan mengembangkan sistem yang akan dibangun yang setiap prosesnya dilakukan secara bertahap[7]. SDLC terdiri dari beberapa tahapan, yaitu: 1) Perencanaan, 2) Analisis, 3) Desain, 4) Implementasi.



Gambar 2. Software Development Life Cycle

Dari gambar diatas menjelaskan mengenai tahapan-tahapan yang dilakukan pada SDLC.

Berikut akan dijelaskan dari setiap tahap SDLC tersebut[7].

1. Perencanaan

Tahap perancanaan adalah proses dasar dalam memahami mengapa sistem informasi harus dibangun dan menentukan bagaimana tim proyek akan membangun sistem tersebut.

Terdapat dua langkah yang dapat dilakukan:

- a. *Project initiation* (inisialisasi projek), proses ini menentukan bagaimana nilai bisnis yang dihasilkan dari sistem yang dibangun? Bagaimana ide – ide baru dapat muncul dalam proses pengembangan sistem.

Sistem request menampilkan sebuah rangkuman dari apa yang menjadi kebutuhan dari sistem, dan menjelaskan bagaimana sistem dapat mendukung nilai bisnis dari sistem yang dibangun. Ada beberapa aspek yang menjadi bahan pemikiran pada proses *project initiation*, yaitu: 1) *technical feasibility* (dapatkah tim pengembang membangun sistem yang direncanakan?); 2) *economic feasibility* (dapatkah sistem membuktikan bahwa sistem yang dibangun dapat menghasilkan nilai bisnis yang menguntungkan?); 3) *organizational feasibility* (jika sistem yang dibangun selesai, apakah sistem tersebut digunakan?).

- b. Jika projek disetujui, maka proses selanjutnya adalah *project management*. Selama *project management*, *project manager* membuat sebuah *work plan*, susunan tim pada projek, dan mengambil keputusan untuk membuat teknik dalam mengontrol anggota tim dan *deliverable* dari *project management* adalah sebuah *project plan*. *Project plan* merupakan pendeskripsian tentang bagaimana tim akan membangun sistem.
2. Analisis
- Pada proses analisis merupakan proses yang akan menjawab pertanyaan tentang siapa yang akan menggunakan sistem, apa saja yang dapat dilakukan sistem, dan dimana dan kapan sistem tersebut digunakan. Selama proses analisis, tim projek menganalisis atau mengamati bagaimana sistem yang sedang berjalan atau disebut dengan *current system*, mengidentifikasi peluang yang mungkin menjadi perbaikan dari *current system*, dan membuat konsep untuk sistem yang baru. Proses analisis ini juga terdiri dari beberapa proses, yaitu sebagai berikut:
- a. Analisis strategi, merupakan proses untuk membuat usaha – usaha apa saja yang mungkin menjadi panduan dalam pengerjaan sistem. Contohnya, tim boleh melihat kekurangan dan kelebihan *current system* lalu memikirkan cara untuk mendesain sebuah sistem yang baru.
 - b. *Requirement gathering*, merupakan proses untuk mengumpulkan informasi – informasi yang berkaitan dengan sistem yang akan dibangun. Contohnya adalah melakukan *interview* dan menyebarkan *questioner*. Dari proses interview maupun *questioner* tersebut akan dihasilkan informasi yang membantu tim untuk memikirkan bagaimana konsep sistem yang baru tersebut dibangun atau dikembangkan. Konsep sistem tersebut akan digunakan sebagai dasar untuk membangun atau mengembangkan bisnis model yang menggambarkan bagaimana bisnis tersebut akan berjalan jika sistem yang baru dibangun dan dikembangkan.
 - c. Analisis, konsep sistem dan model akan dikombinasikan dalam sebuah dokumen yang disebut dengan *system proposal*, yang akan ditunjukkan kepada tim dan *stakeholder*. *System proposal* akan menginisialisasi *deliverable* yang mendeskripsikan apa saja kebutuhan bisnis dari sistem yang baru karena hal tersebut merupakan hal yang menjadi langkah pertama dalam melakukan desain untuk sistem yang baru.

3. Desain

Proses ini akan menceritakan bagaimana sistem akan berjalan pada perangkat keras (*hardware*), perangkat lunak (*software*) dan sarana jaringan (*network*), tampilan *user interface*, laporan yang digunakan, spesifik program, database, dan file – file yang dibutuhkan. Tahapan proses desain terdiri dari 4 (empat) langkah, yaitu:

- a. Strategi desain yang digunakan. Langkah ini merupakan langkah untuk menentukan apakah sistem akan dibangun oleh tim projek sendiri atau bahkan menggunakan tim dari luar yang disebut dengan vendor atau apakah tim projek membeli sebuah perangkat lunak (*software*) untuk membangun sistem.
- b. Membuat dasar desain arsitektur (*architecture design*) untuk sistem yang menjelaskan *hardware*, *software* dan jaringan yang digunakan. Dalam kasus yang sering muncul, sistem akan menambahkan atau mengubah sarana yang telah digunakan sebelumnya. Desain *interface* (tampilan) menspesifikasi bagaimana pengguna (*user*) akan menggunakan sistem, dan kebutuhan akan bentuk – bentuk dan laporan – laporan yang digunakan sistem.
- c. Membuat *database and file specifications*. Proses ini merupakan proses untuk memastikan data – data apa saja yang akan disimpan dan dimana data – data tersebut akan disimpan.

4. Implementasi

Tahapan akhir dan merupakan tahapan final dari SDLC adalah tahap implementasi, dimana selama proses ini sistem yang telah direncanakan, dianalisis, dan didesain akan dibangun menjadi sebuah sistem yang nyata. Pada proses ini, tim akan lebih memperhatikan setiap proses–proses yang akan dilakukan. Proses implementasi ini juga memiliki beberapa sub proses yang harus dilakukan oleh tim pembangun, yaitu:

- a. *System construction*, merupakan proses membangun sistem dan menguji sistem untuk memastikan bahwa sistem yang dibangun telah sesuai dengan desain yang telah dibuat sebelumnya. Karena untuk biaya perbaikan dari kesalahan dapat menghabiskan biaya yang sangat besar, maka pada proses ini dilakukan pengujian dari sistem yang dibuat. Proses pengujian akan menghabiskan waktu yang panjang dan mengambil perhatian yang lebih dari tim pengembang kemudian menuliskan program yang telah dibuat.

- b. *Installation*, merupakan proses dimana sistem yang lama akan di non-aktifkan atau tidak digunakan lagi dan digantikan dengan yang baru. Ada beberapa pendekatan yang boleh digunakan untuk memperbarui sistem yang lama menjadi sistem baru. Satu aspek yang paling penting adalah *training plan*, digunakan untuk melatih pengguna (*user*) untuk menggunakan sistem yang baru.
- c. Tim analis menyusun sebuah *support plan* untuk sistem yang dibangun. *Support plan* biasanya berisi tentang bagaimana cara untuk mengidentifikasi hal-hal utama dan perubahan – perubahan yang butuhkan untuk sistem.

Proses SDLC diatas dapat dilihat secara singkat pada Tabel 1 berikut.

Tabel 1. Tahapan - Tahapan SDLC

Tahapan SDLC	Fokus	Langkah	Teknik	Deliverable
Perencanaan (Planning)	1. Mengapa sistem dibangun atau dikembangkan ?	<i>Identify opportunity</i>	<i>Project Identification</i>	<i>System Request</i>
	2. Bagaimana struktur projek yang dibangun atau dikembangkan ?	<i>Analyze feasibility</i>	1. <i>Technical feasibility</i> 2. <i>Economic feasibility</i> 3. <i>Organizational feasibility</i>	<i>Feasibility study</i>
		<i>Develop workplan</i>	1. <i>Time estimation</i> 2. <i>Task identification</i> 3. <i>Work breakdown structure</i> 4. <i>PERT chart</i> 5. <i>Gantt chart</i> 6. <i>Scope management</i>	<i>Project plan – work plan</i>
		<i>Staff Project</i>	1. <i>Project staffing</i> 2. <i>Project charter</i>	<i>Staffing plan</i>
		<i>Control and direct project</i>	1. <i>CASE repository</i> 2. <i>Standards</i> 3. <i>Documentation</i> 4. <i>Timeboxing</i> 5. <i>Risk management</i>	1. <i>Standards list</i> 2. <i>Risk management</i>
Analisis (Analysis)	Siapa, apa, dimana dan kapan sistem ini dilakukan?	<i>Develop analysis strategy</i>	1. <i>Business process automation</i> 2. <i>Business process improvement</i> 3. <i>Business process reengineering</i>	<i>System proposal</i>
		<i>Determine business requirement</i>	1. <i>Interview</i> 2. <i>JAD session</i> 3. <i>Questionnaire</i> 4. <i>Document analysis</i> 5. <i>Observation</i>	<i>Requirement definition</i>
		<i>Create use cases</i>	<i>Use case analysis</i>	<i>Use cases</i>
		<i>Model processes</i>	<i>Data flow diagramming</i>	<i>Process models</i>
		<i>Model data</i>	1. <i>Entity relationship</i>	<i>Data model</i>

Tahapan SDLC	Fokus	Langkah	Teknik	Deliverable
			<i>modeling 2. normalization</i>	
Desain	Bagaimana sistem dapat bekerja?	<i>Design physical system</i>	<i>Design strategy</i>	<i>1. Alternative Matrix 2. System Specification</i>
		<i>Design architecture</i>	<i>1. Architecture design 2. Hardware & software selection</i>	<i>1. Architecture report 2. Hardware & software specification</i>
		<i>Design interface</i>	<i>1. Use scenario 2. Interface structure 3. Interface standards 4. Interface prototype 5. Interface evaluation</i>	<i>Interface design</i>
		<i>Design programs</i>	<i>1. Data flow diagramming 2. Program structure chart 3. Program specification</i>	<i>1. Physical process model 2. Program design</i>
		<i>Design databases and files</i>	<i>1. Data format selection 2. Entity relationship modeling 3. Denormalization 4. Performance tuning 5. Size estimation</i>	<i>1. Database & file specification 2. Physical data model</i>
Implementation	Penyampaian sistem yang komplit	<i>Construct system</i>	<i>1. Programming 2. Software testing 3. Performance testing</i>	<i>1. Test plan 2. Programs 3. Documentation 4. Migration plan</i>
		<i>Install system</i>	<i>Conversion strategy selection</i>	<i>1. Conversion plan 2. Business contingency plan 3. Training plan</i>
		<i>Maintain system</i>	<i>1. Support selection 2. Sistem maintenance 3. Project assessment</i>	<i>1. Support plan 2. Problem report 3. Change request</i>
		<i>Post - implementation</i>	<i>Post implementation audit</i>	<i>Post implementation audit report</i>

2.2 Model Driven Development (MDD)

Model Driven Development merupakan ide yang digunakan untuk dapat membangun sebuah model dari sistem yang akan dibangun[29]. MDD memiliki ciri khas yaitu berfokus pada pengembangan perangkat lunak dari program komputer[30]. MDD merupakan sebuah metodelogi pengembangan yang bertujuan untuk menunjukkan langkah-langkah pengembangan secara otomatis. MDD menggunakan model pengembangan dimana masing-masing model dipetakan berdasarkan masalahnya masing-masing. Penggunaan MDD tidak terlepas dari pendekatan OOAD, sehingga *developer* membutuhkan orang-orang yang mampu untuk menganalisa atau memodelkan setiap detail dari implementasi sistem yang dibangun menggunakan sebuah bahasa

pemrograman, dan memungkinkan untuk menganalisa model fungsi seperti apa yang dibutuhkan dan bagaimana arsitektur yang dimiliki sistem secara keseluruhan[31]. Maka, MDD bertujuan untuk mengotomatisasikan tugas pemrograman yang kompleks. Hal yang mendasari MDD adalah untuk meningkatkan produktivitas produk. Hal ini memberikan manfaat dengan dua cara yang mendasar, yaitu meningkatkan produktivitas dalam jangka pendek dan meningkatkan produktivitas dalam jangka panjang. Meningkatkan produktivitas produk dalam jangka pendek maksudnya adalah dengan meningkatkan sebuah nilai artifak perangkat lunak yang utama dengan menunjukkan berapa banyak fungsi yang akan diberikan sistem sedangkan untuk meningkatkan produktivitas dalam jangka panjang adalah dengan mengurangi tingkat dimana nilai artifak yang dihasilkan menjadi usang. Dengan kata lain, bahwa sistem yang dihasilkan nantinya dapat berjalan dalam jangka waktu yang lama. Dengan kata lain, ada dua konsep inti yang terkait dengan pengembangan MDD, yaitu abstraksi dan otomasi. Model aplikasi perangkat lunak didefinisikan pada tingkat abstraksi yang lebih tinggi dan kemudian diubah menjadi aplikasi dengan menggunakan transformasi otomatis atau interpretasi. Pendekatan pengembangan model yang tepat dapat memanfaatkan pencapaian model, dimana model secara otomatis berubah menjadi aplikasi perangkat lunak yang bekerja dengan menafsirkan dan menjalankan model yang telah dibuat.

MDD adalah sebuah ide yang mampu mentransformasikan model ke dalam sebuah sistem komputer. Model yang dapat dibentuk tersebut akan memiliki berbagai macam bentuk seperti *parametrics for controllers*, *control diagrams*, program, dan UML. Langkah yang dilakukan terlebih dahulu untuk dapat menggambarkan model-model tersebut adalah melakukan pengumpulan kebutuhan prasyarat (*requirements*) dengan tujuan membuat sebuah model yang independen dan spesifik, dan selanjutnya mengimplementasikan kebutuhan-kebutuhan tersebut dalam bentuk nyata pada sebuah sistem komputer yang dapat dikompilasi dengan menguji fungsi-fungsi yang telah dibuat sesuai dengan kebutuhan yang telah dikumpulkan. Oleh karena itu, pada proses pengembangan MDD, seluruh tim pengembang harus dapat berorientasi dan melakukan banyak aktifitas seperti mampu menganalisis perancangan awal sampai mampu merancang kebutuhan pada proses pengujian seperti membuat kasus uji (*test case*) yang semuanya akan didasarkan pada model yang dibuat sebelumnya menggunakan UML.

Pada proses pengembangan sistem komputer dengan menggunakan metode MDD dibutuhkan banyak *tools* sesuai dengan tahapan pengembangannya. Pada tahapan analisis

dan perancangan misalnya, dibutuhkan *Enterprise Architect* atau *visual paradigm* yang dapat menggambarkan model-model dengan menggunakan notasi UML. Sedangkan pada tahapan implementasi (*code program*) yang dibutuhkan adalah seperti *Laravel*, *Yii Framework*, *Android Studio*, dan lain sebagainya[32].

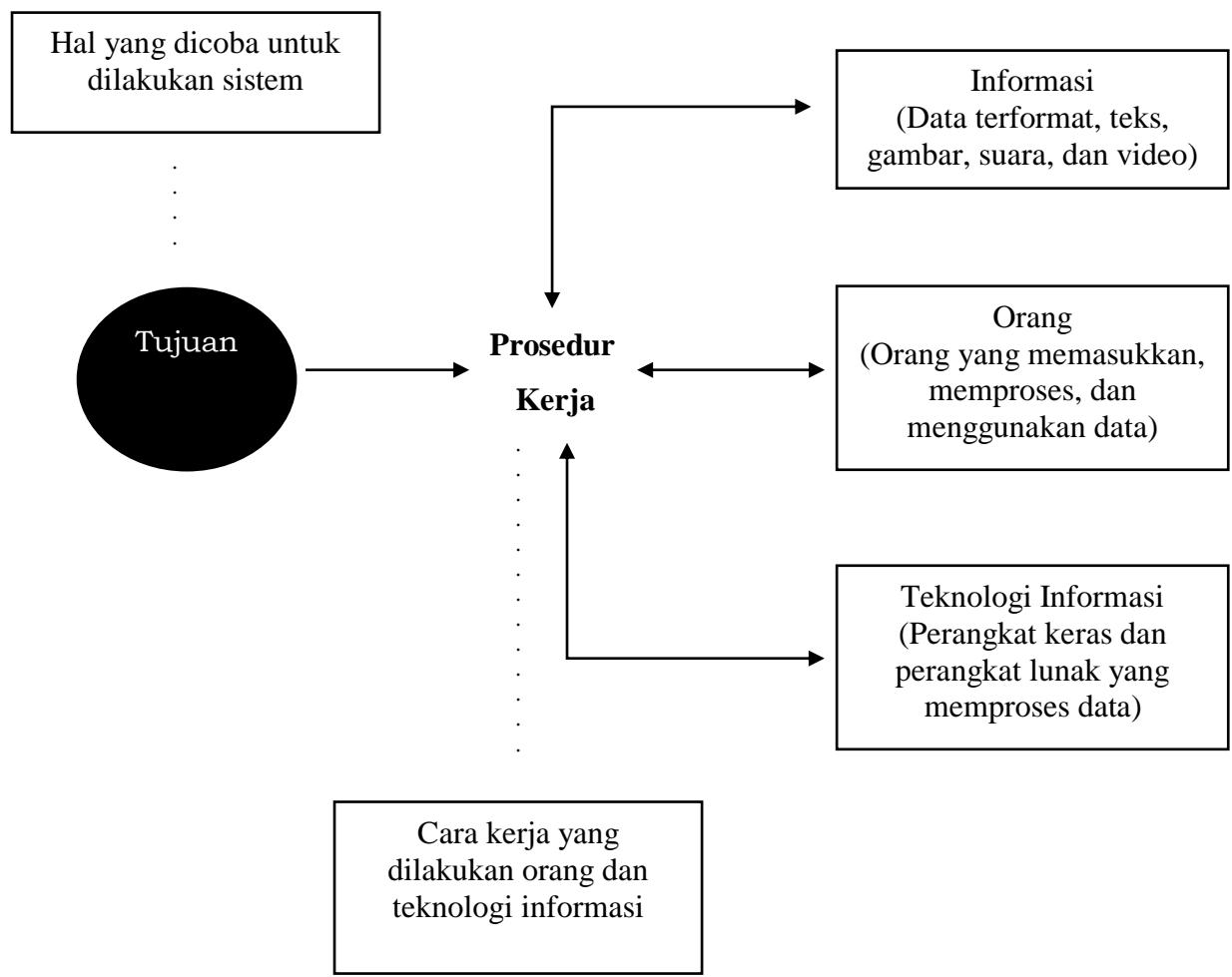
2.3 Sistem Informasi

Sistem informasi merupakan paduan antara orang, prosedur kerja, informasi, dan teknologi informasi untuk mencapai tujuan dalam sebuah organisasi tertentu[9].

Sistem informasi merupakan sekumpulan komponen yang terdiri dari manusia, komputer, teknologi informasi dan prosedur kerja, memproses data menjadi informasi untuk mencapai tujuan tertentu[10].

Dalam sistem informasi terjadi pengumpulan, pemrosesan, penyimpanan, penganalisisan, dan penyebaran informasi untuk tujuan yang spesifik.

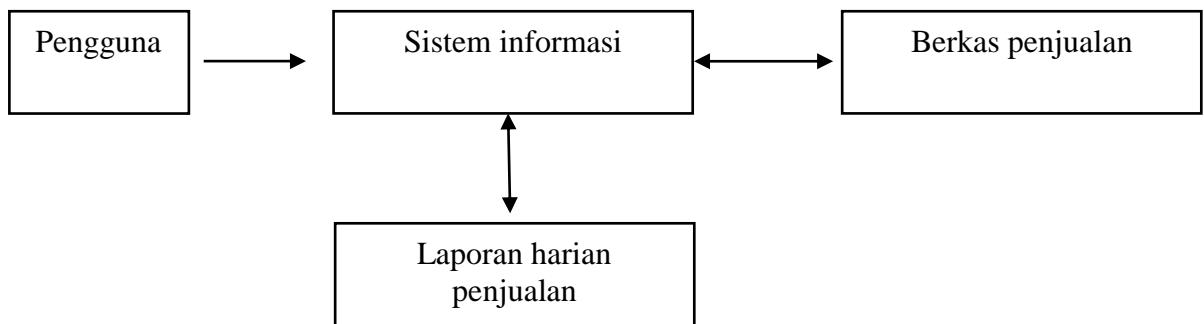
Berikut akan digambarkan definisi sistem informasi[9].



Gambar 3. Definisi Sistem Informasi

Sistem informasi tidak harus selalu berbentuk kompleks. Gambar dibawah memperlihatkan suatu sistem informasi yang sangat sederhana. Sistem hanya digunakan untuk mencatat transaksi penjualan dan melibatkan satu orang saja. Melalui sebuah komputer, pemakai memasukkan data penjualan dan mencetak laporan harian penjualan. Selanjutnya, laporan digunakan untuk melakukan analisis tentang barang-barang yang laku, yang berguna untuk pengambilan keputusan.

Berikut merupakan gambaran sistem informasi yang sederhana[9].



Gambar 4. Sistem Informasi yang sederhana

Dalam bentuk yang lebih kompleks, sistem informasi melibatkan banyak pemakai dan memerlukan sarana jaringan yang memungkinkan pemakai yang tidak berada pada tempat yang sama dapat saling berbagi informasi.

Sistem Informasi Rumah Sakit

Menurut Depkes RI, (2009) rumah sakit adalah sarana kesehatan yang menyelenggarakan pelayanan kesehatan secara menyeluruh yang memprioritaskan upaya penyembuhan penyakit dan pemulihan kesehatan. Selain itu, rumah sakit juga merupakan institusi yang mampu memberikan contoh bagaimana budaya hidup bersih dan menjaga kebersihan lingkungan.

Sistem informasi rumah sakit adalah suatu rangkaian yang berhubungan dengan pengumpulan dan pengelolaan data, penyajian informasi, analisis, dan juga penyampaian informasi yang dibutuhkan untuk kegiatan di rumah sakit[1].

Menurut Undang-Undang Republik Indonesia No 44 Tahun 2009 tentang rumah sakit, rumah sakit mempunyai fungsi yaitu:

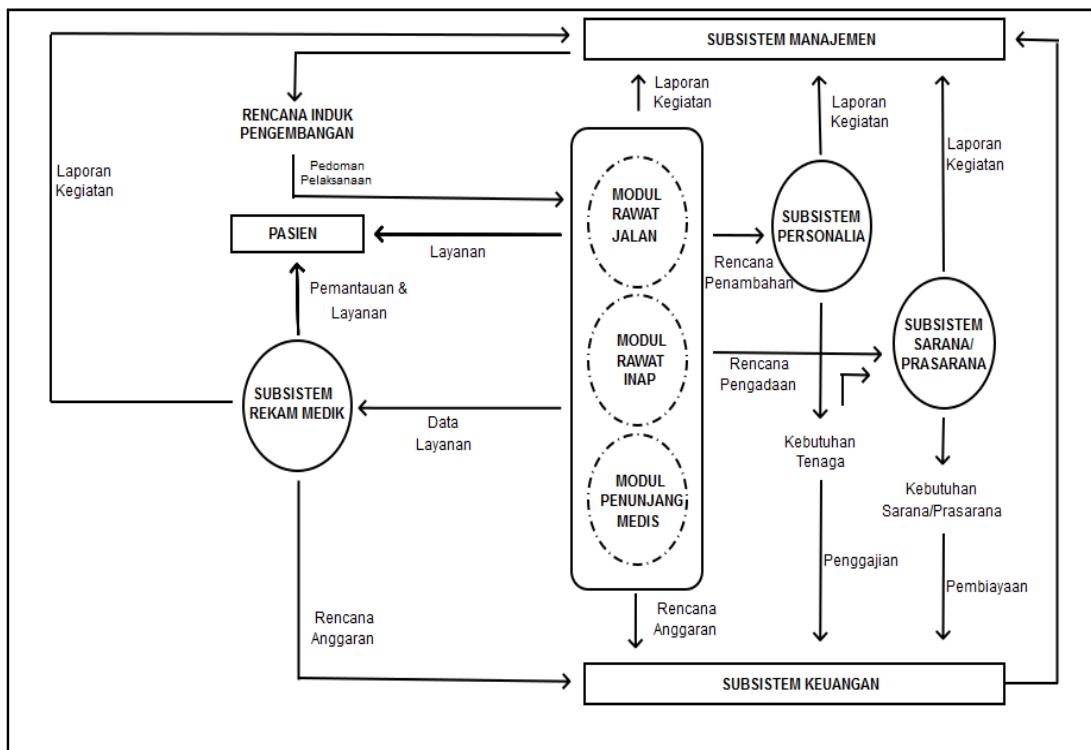
1. Penyelenggaraan pelayanan pengobatan dan pemulihan kesehatan sesuai dengan standar pelayanan rumah sakit.
2. Pemeliharaan dan peningkatan kesehatan perorangan melalui pelayanan kesehatan yang paripurna tingkat kedua dan ketiga sesuai kebutuhan medis.

3. Penyelenggaraan pendidikan dan pelatihan sumber daya manusia dalam rangka peningkatan kemampuan dalam pemberian pelayanan kesehatan.
4. Penyelenggaraan penelitian dan pengembangan serta penapisan teknologi bidang kesehatan dalam rangka peningkatan pelayanan kesehatan dengan memperhatikan etika ilmu pengetahuan bidang kesehatan.

Adapun unit yang terdapat pada rumah sakit, yaitu[8]:

1. Unit Gawat Darurat (UGD)
2. Unit Rawat Jalan
3. Unit Rawat Inap
4. Unit Medical Check Up (MCU)
5. Unit Intensive Cardiac Care Unit (ICCU)
6. Unit Kamar Bedah dan Kamar Persalinan
7. Unit Rehabilitasi Medik (Fisioterapi)
8. Unit Hemodialisa
9. Unit Laboratorium
10. Unit Radiologi
11. Unit Apotek

Berikut adalah skema rancang bangun SIRS secara global. Gambar ini menjelaskan hubungan antara subsistem yang satu dengan yang lainnya. Rancangan global SIRS terbagi menjadi subsistem, modul, submodul, dan aplikasi[1].



Gambar 5. Sistem Informasi Rumah Sakit

Atas dasar pemakaian, sistem informasi rumah sakit dibagi menjadi tiga bagian, yaitu[11]:

a) Sistem informasi klinik

Sistem informasi klinik merupakan sistem informasi yang secara langsung membantu pasien dalam hal pelayanan medis.

b) Sistem informasi administratif

Sistem informasi administratif merupakan sistem informasi yang membantu pelaksanaan administrasi di rumah sakit.

c) Sistem informasi manajemen

Sistem informasi manajemen merupakan sistem informasi yang membantu manajemen rumah sakit dalam hal pengambilan keputusan.

Berdasarkan jaringan sistem yang digunakan, sistem informasi rumah sakit dibagi menjadi tiga bagian, yaitu[11]:

a) Individual

Individual berarti bahwa hanya terdapat sistem tersebut, tanpa terlihat sistem yang lain.

b) Modular

Modular berarti bahwa beberapa sistem dikaitkan sebagai satu kelompok.

c) Sistem informasi terpadu

Sistem informasi terpadu berarti bahwa beberapa sistem digabung menjadi satu kesatuan.

Berikut merupakan manfaat dari sistem informasi rumah sakit, yaitu[11]:

- a) Permintaan tujuan dan target
- b) Memperhatikan kebutuhan pelayanan
- c) Alokasi sumber daya
- d) Pengendalian mutu pelayanan
- e) Evaluasi program

Sistem informasi rumah sakit berperan dalam mendukung beberapa hal, yaitu[11]:

- a) Pengendalian mutu pelayanan medis
- b) Pengendalian mutu dan penilaian produktivitas
- c) Analisis pemanfaatan dan perkiraan kebutuhan
- d) Perencanaan dan evaluasi program
- e) Menyederhanakan pelayanan
- f) Penelitian klinis
- g) Pendidikan

Sistem informasi rumah sakit tersusun atas beberapa sistem dan modul, yaitu sebagai berikut[11]:

- a) Admin Sistem
- b) Sistem Pelayanan Pasien/Billing System
 - Modul Loket/Registrasi Pasien
 - Modul Pelayanan Rawat Jalan
 - Modul Pelayanan Rawat Inap
 - Modul UGD
 - Modul IRD (Instalasi Rawat Inap)
 - Modul Kamar Operasi
 - Modul Persalinan
 - Modul Paviliun
 - Modul Rawat Intensive (ICU/NICU/PICU)
 - Modul Instalasi Hemodelisia
 - Modul Medical Check Up
 - Modul Billing System

- c) Sistem Farmasi
 - Modul Pengendalian Stok
 - Modul Gudang Obat
 - Modul Floor Stock
 - Modul Produksi Obat
 - Modul Apotek (multi apotek)
- d) Sistem Penunjang Medis
 - Modul Laboratorium
 - Modul Radiologi
 - Modul Bank Darah
 - Modul Fisioterapi
 - Modul Rehab Medis
 - Modul Kamar Jenazah
 - Modul Manajemen Dapur
 - Modul Gizi
 - Modul Rekam Medik
- e) Sistem Aset/Inventori
- f) Sistem Keuangan dan Akuntansi
 - Modul Hutang
 - Modul Piutang
 - Modul Kas-Bank
 - Modul Budgeting
 - Modul Akuntansi
- g) Sistem Human Resources Development (HRD)
 - Modul Personalia
 - Modul Penggajian/Payroll
 - Modul RSU Pendidikan
- h) Sistem Manajemen
 - Modul Manajemen Pelayanan
 - Modul Manajemen Farmasi
 - Modul Manajemen Keuangan
 - Modul Manajemen Aset

- Modul Pemasaran dan Publikasi/PR

Pada tugas akhir ini modul yang akan dikerjakan adalah modul utama dari rumah sakit yaitu sebagai berikut.

2.3.1.1 Rawat Jalan

Pelayanan rawat jalan adalah bentuk dari pelayanan yang sederhana yang diberikan dokter. Pelayanan ini dilakukan jika pasien tidak ingin rawat inap[12].

Pada Keputusan Kesehatan No.66 / Menkes / 11 / 1987 yang dimaksud rawat jalan adalah pelayanan untuk keperluan observasi, diagnosa, pengobatan, rehabilitasi medik, dan pelayanan kesehatan lainnya yang diberikan terhadap orang yang masuk rumah sakit namun tidak tinggal (menginap) di rumah sakit.

2.3.2 Rawat Inap

Rawat inap adalah salah satu jenis pelayanan kesehatan bagi pasien yang memerlukan tindakan medis, keperawatan dan *non* medis lebih lanjut (dalam kurun waktu tertentu), yang membutuhkan perawatan rawat inap di rumah sakit (*hospitalization*), hal ini dikarenakan penyakit yang diderita oleh pasien dianggap memerlukan perawatan yang intensif oleh tenaga medis, keperawatan dan *non* medis untuk mencapai kesehatan yang optimal (Undang-undang RI No.44 Tahun 2009). Pasien yang dirawat inap memerlukan penanganan dalam hal pencatatan tindakan medik yang setiap saat diberikan kepada pasien.

2.3.3 Rekam Medis

Data riwayat kesehatan pasien merupakan hal yang sangat penting dalam dunia kesehatan, oleh sebab itu setiap instansi medis selalu melakukan pencatatan data riwayat pasien. Istilah ini disebut juga dengan rekam medis. Pencatatan data riwayat pasien (rekam medis) dilakukan jika setiap kali pasien melakukan pemeriksaan atau menjalani perawatan oleh dokter maupun instansi medis. Data rekam medis yang dimiliki pasien akan dijadikan sebagai acuan jika pasien melakukan pemeriksaan selanjutnya dan juga sebagai bukti tertulis mengenai diagnosis penyakit pasien dan pelayanan medis yang diperoleh pasien[10].

Menurut Peraturan Menteri Kesehatan Nomor 749a/Menkes/Per/XII/1989 yang berisi tentang Rekam Medis dijelaskan bahwa rekam medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain kepada pasien pada sarana pelayanan kesehatan.

Sedangkan menurut Gemala Hatta (2008), rekam medis merupakan kumpulan fakta tentang kehidupan seseorang dan riwayat penyakitnya, termasuk keadaan sakit, pengobatan saat ini dan saat lampau yang ditulis oleh para praktisi kesehatan dalam upaya mereka memberikan pelayanan kesehatan kepada pasien. Latar belakang perlunya dibuat rekam medis adalah untuk mendokumentasikan semua kejadian yang berkaitan dengan kesehatan pasien serta menyediakan media komunikasi di antara tenaga kesehatan bagi kepentingan perawatan penyakitnya yang sekarang maupun yang akan datang[10].

Berdasarkan dari kedua pengertian tersebut, dapat dikatakan bahwa rekam medis bukan hanya sekedar catatan biasa. Namun, dengan adanya catatan medis yang dimiliki maka dokter dapat menentukan tindakan lebih lanjut kepada pasien karena di dalam rekam medis memuat segala informasi yang berhubungan dengan kesehatan pasien.

Menurut Depkes RI, (1993) Dirjen Pelayanan Medis dalam buku Pedoman Pengolahan rekam medis rumah sakit di Indonesia, terdapat beberapa kegunaan dari rekam medis yaitu:

a. Administrasi

Rekam medis berisi berkas yang memiliki nilai administrasi, dikarenakan pada isinya menyangkut tindakan dan tanggung jawab sebagai tenaga medis dan perawatan dalam mencapai tujuan pelayanan kesehatan.

b. Medis

Digunakan untuk perencanaan terapi pengobatan dan perawat yang akan diberikan kepada pasien.

c. Hukum

Digunakan sebagai tanda bukti untuk menegakkan keadilan.

d. Penelitian

Rekam medis berisi data atau informasi yang dapat digunakan sebagai bahan penelitian.

e. Dokumentasi

Rekam medis akan menjadi sumber dokumentasi dikarenakan akan diingat dan akan digunakan sebagai bahan pertanggungjawaban dan laporan kesehatan.

2.3.4 Billing System

Billing System merupakan sistem yang sangat penting dalam pertumbuhan suatu rumah sakit, khususnya dalam stabilisasi keuangan dalam sebuah rumah sakit. *Billing System* diharapkan dapat mempercepat dan mempermudah sistem penagihan biaya perawatan yang dilakukan dilakukan pasien mulai pasien mendaftar, menerima pelayanan kesehatan, hingga pasien pulang. Dengan adanya *billing system* suatu rumah sakit akan mengetahui jumlah pendapatan yang diperoleh dari pelayanan yang diberikan.

2.4 Mobile Application

Mobile Application merupakan program perangkat lunak yang dapat diunduh dan diakses dengan menggunakan perangkat *mobile* dan dikembangkan untuk perangkat *mobile* seperti *smartphone* dan *tablet*. *Mobile Application* menyediakan *interface* yang lebih menarik dan populer dalam berinteraksi dengan sistem daripada berinteraksi melalui mesin penjelajah lainnya[14].

Pengembangan aplikasi *mobile* dibagi dalam tiga pendekatan, yaitu *mobile web application*, *mobile native application*, dan *mobile hybrid application*. Setiap pendekatan memiliki kelebihan dan kekurangan satu sama lain. Perbedaan tersebut akan dijelaskan sebagai berikut.

2.4.1 Mobile Web Application

Mobile web application adalah sebuah aplikasi *website* yang diformat untuk *smartphone* dan *tablet* yang dapat digunakan atau berjalan pada semua *browser* pada *platform mobile* sehingga aplikasi ini disebut aplikasi *multi platform*[15].

Pada aplikasi *web* ini menggunakan teknologi yang tidak asing lagi bagi pengembang ataupun *programmer* yaitu HTML5, CSS3, dan JavaScript sehingga tergolong mudah. Aplikasi *web* tidak dapat mengakses fitur-fitur perangkat keras yang terdapat pada *smartphone* karena aplikasi *web* hanya dapat berjalan pada *browser smartphone*. Aplikasi *web* juga hanya dapat diakses jika terhubung dengan jaringan internet dan bergantung pada konektivitas yang menyebabkan performanya menjadi kurang stabil[15].

2.4.2 Mobile Native Application

Mobile native application adalah aplikasi yang dibangun secara khusus untuk *platform mobile* tertentu dan bahasa pemrograman serta perangkat pengembangan yang digunakan harus sesuai dengan *platform* yang dipakai.

Mobile native application dibangun dengan menggunakan bahasa pemrograman Java untuk Eclipse IDE, Objective C untuk iOS (iphone), atau beberapa bahasa pemrograman lainnya.

Tampilan antarmuka yang dihasilkan *mobile native application* adalah alami serta sesuai untuk *platform* yang dikembangkan. Dengan menggunakan bahasa pemrograman yang sesuai dengan sistem operasi yang dipakai menyebabkan performa dari aplikasi *native* sangat baik. Selain itu *native* juga mendukung akses ke semua fitur perangkat keras *smartphone* misalnya info *device*, kamera, *compass*, file, dan GPS. Pengembangan aplikasi terbatas yaitu hanya untuk satu sistem operasi saja. Hal tersebut dikarenakan aplikasi *native* hanya dapat berjalan pada *platform* yang telah di spesifikasikan sejak pada tahap awal pengembangan aplikasi[15].

2.4.3 Mobile Hybrid Application

Mobile Hybrid Application adalah penggabungan dari aplikasi *web* dengan aplikasi *native*. Aplikasi *hybrid* mengubah aplikasi *web mobile* HTML5 ke aplikasi *native smartphone* yang akan ditargetkan. Pengubahan aplikasi *web mobile* ke dalam aplikasi *native* diperlukan semacam *framework* pengembangan aplikasi *mobile* sebagai dukungan perangkat lunak spesifik.

Aplikasi *hybrid* memiliki tahap pengembangan yang relatif mudah dikarenakan menggunakan standar teknologi *web*. Aplikasi *hybrid* dapat berjalan pada hampir semua *platform mobile* yang dituju, yang berarti aplikasi *hybrid* dapat didistribusikan secara aplikasi *native* menuju ke *platform mobile* yang akan ditargetkan oleh pengembang aplikasi. Aplikasi *hybrid* mungkin saja dapat mengakses fitur-fitur perangkat keras misalnya kamera, *wifi*, kalender, dan file jika dilakukan distribusi aplikasi secara *native*[15].

2.5 Desktop Application

Menurut Konixbam (2009), *Desktop Application* adalah suatu aplikasi yang dapat berjalan sendiri atau independen tanpa menggunakan *browser* atau koneksi internet di suatu komputer otonom dengan *operating system* atau *platform* tertentu[13]. *Desktop Application* difokuskan kepada aplikasi yang lebih independen. Hal tersebut bertujuan untuk mempermudah para pengguna aplikasi dalam hal memodifikasi pengaturan aplikasi sehingga efektifitas, efisiensi waktu, dana, dan tenaga lebih ditekankan semaksimal mungkin.

Secara garis besar pada pemrograman terutama pada aplikasi yang berbasis *desktop* dapat dibedakan menjadi dua macam yaitu pemrograman konvensional dan pemrograman visual.

- a. Pemrograman konvensional merupakan metode mendesain suatu aplikasi. Pemrograman dituntut untuk bisa menerapkan baris demi baris kode program agar bisa menghasilkan sebuah bentuk tampilan aplikasi yang dibuat dan akan memakan waktu lama.
- b. Pemrograman visual merupakan metode pembuatan program untuk membuat koneksi antar objek dengan cara menggambar, menunjuk, dan mengklik pada diagram dan ikon dengan berinteraksi dengan diagram jalur.

2.6 Web Application

Web Application adalah kumpulan halaman *web* yang mampu melakukan interaksi dengan sesama halaman *web*, pengunjung, dan sumber data lain yang ada di *server web*. Dalam *web application*, permintaan pengunjung akan diproses oleh *server* dan kemudian dikirimkan ke *browser*. Teknologi ini sering disebut sebagai *web dinamis berbasis server* karena sebagian besar halaman di *web application* dikerjakan di sisi *server*[16].

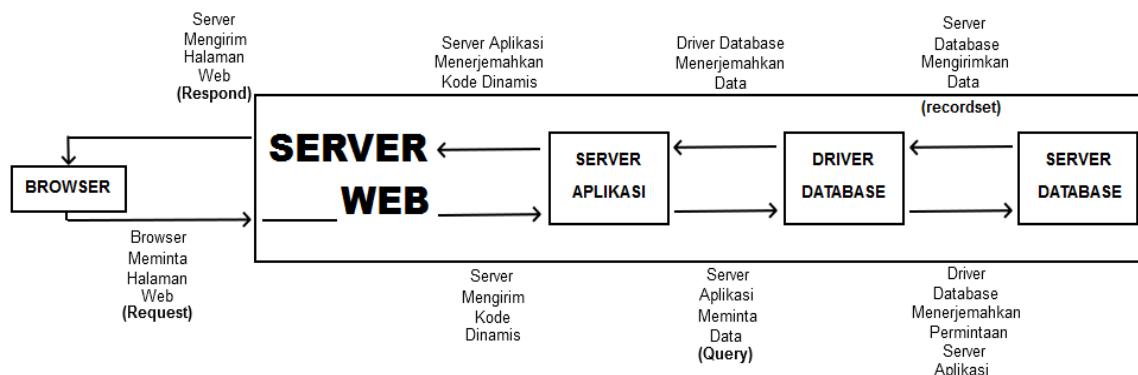
Web application mempunyai banyak kegunaan untuk pengunjung dan pengembang, diantaranya adalah[16]:

- a. Memungkinkan pengunjung melakukan pencarian informasi dengan cepat dan mudah pada situs *web*. *Web application* seperti ini biasanya menyediakan fasilitas pencarian, pengelolaan, dan navigasi isi sesuai kehendak pengunjung. Contoh perusahaan yang menggunakan sistem seperti ini adalah Microsoft MSDN dan Amazon.
- b. Mengumpulkan, menyimpan, dan menganalisis data yang disediakan oleh pengunjung *web*. Data yang ada di halaman *web* dimasukkan melalui *form* dan

dikirimkan sebagai *email* untuk diproses oleh aplikasi CGI atau penerima. Namun sekarang penyimpanan data dapat langsung ke *database* untuk kemudian diolah menjadi laporan dengan menggunakan *web application*. Aplikasi ini bisa digunakan untuk pembuatan situs *online banking* dan *e-commerce*.

- c. Memperbarui situs *web* yang isinya tidak konsisten. Pengembang *web* memiliki kemudahan untuk memperbarui isi *web* sebab pengembang *web* hanya menyediakan isi berita dan *web application*-lah yang akan memperbarui isi halaman dari situs tersebut. Sistem ini banyak digunakan oleh situs penyedia berita seperti CNN.

Berikut adalah cara kerja dari aplikasi *web*, yaitu[9]:



Gambar 6. Diagram Aplikasi Web

Server web mengenali sebuah halaman *web* dari ekstensi filenya. Halaman *web* statis mempunyai ekstensi file .html atau .htm, sedangkan halaman *web* dinamis berbasis *server* memiliki suatu ekstensi khusus, seperti .asp, .php, .cgi, dan lain-lain. Ekstensi file akan berbeda-beda tergantung pada teknologi *server* aplikasi yang digunakan.

Saat menerima permintaan halaman *web* dinamis berbasis *server*, *server web* membaca kode dinamis yang ada di halaman *web* dan mengirim kode tersebut ke *server aplikasi*. *Server aplikasi* memproses kode tersebut dan menerjemahkannya menjadi halaman statis yang akan dikirim ke *browser* oleh *server web*.

Jika suatu halaman *web* memiliki referensi ke sebuah *database*, *server aplikasi* akan meminta data yang diperlukan dari *server database*. Permintaan data tersebut disebut sebagai *query*. Namun, data dari *server database* tidak bisa dibaca langsung oleh *server*

aplikasi. Oleh karena itu perlu ada suatu *driver database* dalam *server web* yang berfungsi sebagai penghubung antara *server database* dan *server aplikasi*. Setelah *server database* mengetahui data yang diinginkan oleh *server aplikasi*, maka melalui perantaraan *driver database*, *server database* akan mengirimkan data yang dikelompokkan dalam *recordset* ke *server aplikasi*. Selanjutnya *server web* mengirimkan hasil dari *server aplikasi* ke *browser web*.

2.7 Platform Android Mobile

Android merupakan *platform mobile* yang bersifat *open source* yang berjalan diatas Linux kernel sebagai layanan sistem operasinya yang dihasilkan oleh Google, Inc. dan dirilis pada tahun 2007[10]. Sistem *android* merupakan subset perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi *middleware* dan aplikasi inti[17]. Operasi *android* menyediakan manajemen memori, proses, *network*, *driver*, *security*, dan abstraksi.

2.7.1 Fitur-fitur Android

Android menyediakan fitur yang dapat digunakan *developer* dalam pengembangan dari pembangunan aplikasi perangkat lunak. Fitur-fitur tersebut dapat dilihat pada Tabel 2[18].

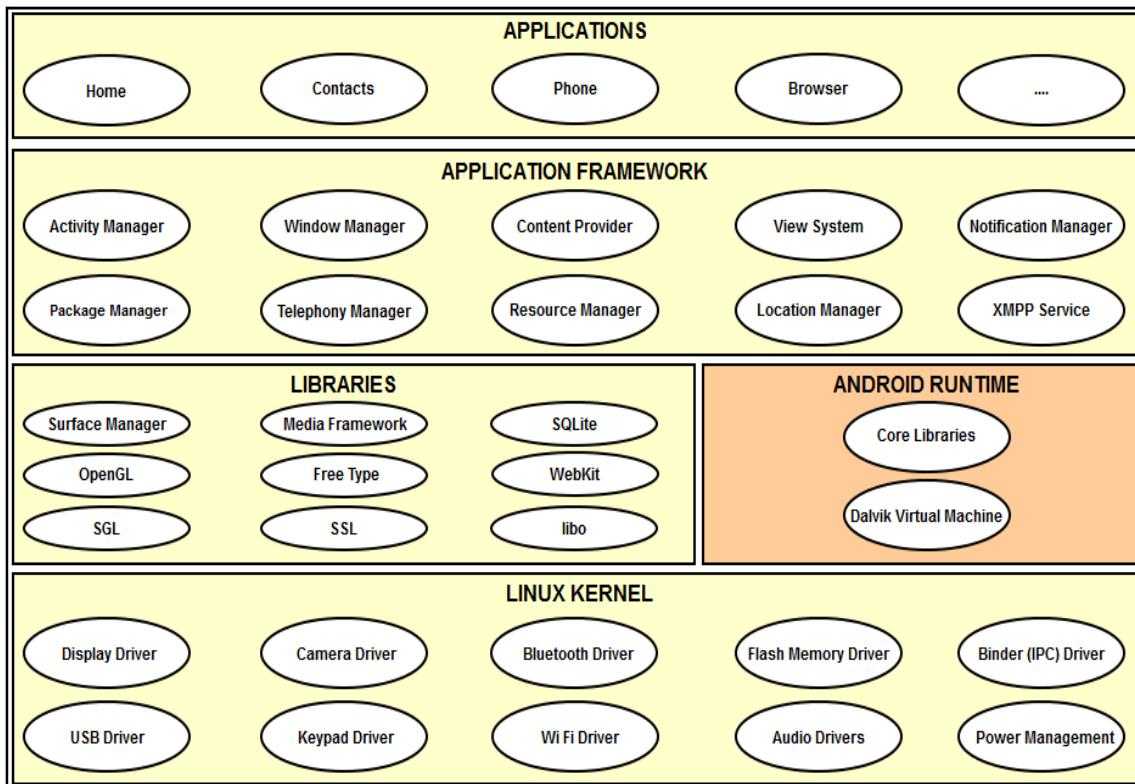
Tabel 2. Fitur-fitur Android

No	Nama Fitur	Keterangan
1.	Framework aplikasi	<i>Framework</i> aplikasi mendukung komponen dan <i>reusable</i> .
2.	Mesin virtual dalvik	<i>Mesin virtual dalvik</i> berfungsi mengoptimalkan perangkat <i>mobile</i> .
3.	Browser terintegrasi	Berdasarkan <i>open source Webkit engine</i> , sistem operasi mengijinkan untuk mengakses <i>web browser</i> yang berfungsi untuk menghubungkan <i>user</i> dengan internet atau dunia maya.
4.	Mengoptimalkan grafis	Grafis yang ada pada <i>android</i> dioptimalkan oleh <i>library</i> grafis 2 dimensi; untuk 3 dimensi berdasarkan spesifikasi <i>OpenGL ES 1.0</i>
5.	SQLite	<i>SQLite</i> digunakan untuk penyimpanan data.
6.	Media support	Media <i>support</i> pada <i>android</i> mendukung audio, video, dan gambar dengan berbagai format (MPEG4, H.264, MP3, ACC, AMR, JPG, PNG, GIF).
7.	GSM telephony	GSM <i>telephony</i> tergantung dari <i>hardware</i> yang digunakan.
8.	Bluetooth, EDGE, 3G, dan Wifi	Bluetooth, EDGE, 3G, dan Wifi tergantung dari <i>hardware</i> yang digunakan.
9.	Camera, GPS, compass, dan	Camera, GPS, compass, dan accelerometer tergantung dari

	accelerometer	<i>hardware</i> yang digunakan.
10.	Lingkungan development	Lingkungan <i>development</i> termasuk perangkat <i>emulator, tools</i> digunakan untuk menemukan error.

2.7.2 Arsitektur Android

Menurut Benjamin Speckmann arsitektur sistem operasi *Android* dapat dilihat pada Gambar 7 berikut[18].



Gambar 7. Arsitektur Android

Penjelasan gambar diatas akan dijelaskan sebagai berikut[18] [19]:

1. Layer Aplikasi

Dari gambar diatas dapat dilihat bahwa lapisan tertinggi dari arsitektur *android* adalah lapisan aplikasi. Lapisan aplikasi adalah lapisan yang paling terlihat pada pengguna saat menjalankan program. Pengguna hanya melihat dan menggunakan aplikasi tanpa mengetahui alur proses yang terjadi pada layer aplikasi ini. Lapisan aplikasi berjalan pada *android runtime* dengan *service* dan kelas yang telah tersedia pada *framework* aplikasi. Lapisan aplikasi *android* memiliki perbedaan dengan sistem operasi lainnya, yaitu semua aplikasi berjalan diatas lapisan aplikasi dengan menggunakan *library* yang sama yaitu API (*Application Programming Interface*).

2. Framework Aplikasi

Framework aplikasi adalah kerangka perangkat lunak yang digunakan dalam menerapkan standar struktur aplikasi untuk sistem operasi tertentu. *Framework* aplikasi juga mengembangkan aplikasi *android* dengan menggunakan kelas-kelas yang dimiliki, menyediakan abstraksi generik untuk mengakses perangkat, dan juga mengatur *user interface* dan sumber daya aplikasi. Adapun hal-hal yang penting dalam framework aplikasi, yaitu:

- a. Activity Manager

Activity manager berfungsi untuk mengontrol berjalannya aplikasi dan menjaga keadaan *backstack* untuk navigasi pengguna.

- b. Content Providers

Content providers berfungsi untuk membuat rangkuman data yang memungkinkan digunakan oleh aplikasi.

- c. Resource Manager

Resources manager berfungsi untuk menyediakan akses sumber daya diluar program, misalnya karakter, file *layout*, dan grafik. Selain itu, *resource manager* juga berfungsi untuk mengelola sumber daya yang ada pada program.

- d. Location Manager

Location manager berfungsi untuk menyediakan informasi secara rinci mengenai lokasi keberadaan perangkat *android*.

- e. Notification Manager

Notification manager berfungsi sebagai penanda peringatan, misalnya pesan masuk, *alarm*, dan lain sebagainya, yang nantinya akan mucul pada status bar.

3. Library

Android menggunakan *library* yang ditulis dengan C/C++. *Library* tersebut akan dipanggil menggunakan *interface* Java. Adapun *library* yang digunakan pada *android*, yaitu:

- a. Media Library

Media library berfungsi untuk merekam serta memutar berbagai macam format video dan audio.

- b. SQLite

SQLite berfungsi untuk mengatur *database* yang akan digunakan pada aplikasi.

c. SSL dan WebKit

SSL dan *WebKit* digunakan untuk *browser* dan keamanan internet.

d. Graphic Library

Graphic library digunakan untuk tampilan 2D dan 3D, misalnya *library* SGL dan openGL.

e. Surface Manager

Surface manager berfungsi untuk mengatur hak akses layer dari berbagai aplikasi.

4. Android Runtime

Runtime merupakan mesin virtual yang menjadikan aplikasi lebih kuat atau tangguh dengan *library* yang ada. *Android runtime* terdiri dari dua komponen. Komponen yang pertama adalah *library* inti, *library* ini menyediakan sebagian besar fungsi yang terdapat pada *library* bahasa pemrograman Java. Komponen yang kedua adalah mesin virtual dalvik yang beroperasi seperti penerjemah antara aplikasi dan sistem operasi.

5. Kernel

Linux kernel akan digunakan *android* untuk manajemen memori, proses, *driver*, manajemen dan jaringan.

2.8 Android Studio vs. Eclipse

Eclipse adalah *Integrated Development Environment* (IDE). Fungsi utama Eclipse adalah untuk mengembangkan aplikasi Java, tetapi dapat juga digunakan untuk pengembangan aplikasi dalam bahasa pemrograman lain melalui penggunaan plugin, termasuk C, C++, Javascript, Perl, Python, Ruby, Erlang, dan lain-lain. Selain itu, Eclipse juga digunakan untuk pengembangan aplikasi *android*. Pengembangan aplikasi *android* membutuhkan Eclipse ADT(*Android Development Tools*). ADT adalah *plugin* Google yang disediakan untuk IDE Eclipse yang dirancang untuk menyediakan lingkungan terintegrasi untuk pembangunan aplikasi *android*[20].

Sedangkan *Android Studio* adalah *Integrated Development Environment* (IDE) yang dirancang khusus untuk mengembangkan aplikasi dengan *platform android*. *Android Studio* adalah pengganti Eclipse ADT sebagai IDE utama Google dalam mengembangkan aplikasi *android*[20].

Tabel 3. Perbedaan Android dan Eclipse

Fitur	Android Studio	Eclipse
-------	----------------	---------

Advanced Android code completion and refactoring	Yes	No
APK signing and keystore management	Yes	Yes
Build system	Gradle	Ant
Build variants and multiple-APK generation	Yes	No
Graphical layout editor	Yes	Yes
Maven-based build dependencies	Yes	No
NDK support	Yes	Yes

Selain perbandingan di atas, *Android Studio* lebih unggul dibandingkan dengan *Eclipse* dari segi grafik. Tetapi *Android Studio* lebih berat dibandingkan *Eclipse* karena membutuhkan spesifikasi komputer yang cukup tinggi. Oleh karena *Android Studio* diperlukan untuk pengembangan *android* saja sedangkan *Eclipse* tidak (yaitu *multi programming*), maka *Android Studio* lebih nyaman digunakan dalam pengembangan aplikasi berbasis *android*.

2.9 Microsoft SQL Server vs. MySQL

Database adalah kumpulan dari item data yang memiliki hubungan satu dengan yang lainnya yang terorganisasi untuk melakukan manipulasi untuk kegunaan tertentu. *Database* diperlukan dikarenakan komponen yang sangat penting dalam sistem informasi karena menyediakan informasi, menentukan kualitas informasi, mengurangi terjadinya *redundancy* data, dan meningkatkan relasi data[21].

MySQL adalah sebuah program *database server* yang memiliki kegunaan untuk mengelola *database* atau memanajemen data. MySQL merupakan turunan dari konsep SQL (*Structured Query Language*) yang merupakan konsep pengoperasian basis data yang dapat dikerjakan dengan mudah. MySQL juga merupakan *software open source* untuk membuat *database* [22].

Microsoft SQL Server adalah sistem manajemen basis data relasional (RDBMS) yang dikembangkan oleh Microsoft. Microsoft SQL Server berfungsi untuk menyimpan data yang dibutuhkan oleh aplikasi lain. SQL Server dapat dijalankan pada komputer yang sama atau komputer dengan jaringan yang berbeda[23].

Berikut akan dijelaskan perbedaan dari Microsof SQL Server dengan MySQL[24].

Platform yang Didukung

Sql Server dikembangkan pada awalnya oleh Microsoft yang berguna untuk sistem operasi Windows secara eksklusif. Baru-baru ini Microsoft mengumumkan untuk

membuat RDMS yang tersedia di Linux, dan Mac OS X (via Docker). Oleh sebab itu, sekarang perusahaan memiliki pilihan untuk menjalankan sistem *database* pada tiga *platform* yang berbeda. Namun pilihan fitur tersebut tidak dimiliki saat menjalankan SQL Server di Linux atau Mac OS X karena SQL Server hanya bekerja di Windows.

Sedangkan MySQL dapat dijalankan di beberapa sistem operasi yaitu Windows, Linux, dan Mac OS X.

Bahasa Pemrograman yang Didukung

MySQL dan SQL Server mendukung beberapa bahasa pemrograman. Kedua RDBMS ini mendukung Java, PHP, C++, Python, Ruby, Visual Basic, Delphi, Go dan R. Namun MySQL mendukung bahasa pemrograman lain seperti Perl, Scheme, Tcl, Haskel dan Eiffel. Dukungan untuk bahasa pemrograman menjadikan MySQL populer di antara berbagai komunitas pengembang.

Mesin Penyimpanan

MySQL mendukung sejumlah mesin penyimpanan. Saat menggunakan MySQL, para *programmer* memiliki banyak pilihan untuk menggunakan mesin penyimpanan *plug-in*. Pada versi sebelumnya RDBM hanya mendukung mesin penyimpanan *non-transaksional*. Oleh sebab itu, *programmer* yang bekerja dengan sistem basis data pada versi lama harus meng-*upgrade* mesin penyimpanannya. Pada saat yang sama, pengembang harus menggunakan mesin penyimpanan tunggal saat bekerja dengan SQL Server sehingga *programmer* harus beralih ke versi terbaru dari RDBMS untuk menggunakan mesin penyimpanan yang lebih baik. Dukungan beberapa *storage engine* membuat MySQL lebih fleksibel.

Filtering

MySQL memungkinkan pengguna untuk memfilter tabel, baris, dan pengguna dalam beberapa cara. Hal tersebut mengharuskan pengguna untuk menyaring tabel, baris, atau pengguna dengan *database* individu. Saat memfilter data, pengembang harus memfilter tabel *database* secara terpisah dengan menjalankan beberapa *query*.

Sedangkan SQL Server memungkinkan pengembang untuk memanfaatkan penyaringan berbaris-baris. Data yang telah difilter di simpan dalam *database* yang terpisah. Oleh karena itu menjadi lebih mudah bagi *developer* untuk memfilter beberapa baris tanpa mempertimbangkan jumlah *database*.

Backup

Saat menggunakan MySQL, pengembang harus mem-*backup* data dengan mengekstrak semua data sebagai pernyataan SQL. Alat yang disediakan oleh RDBMS memblokir *database* saat membuat *backup* data. Fitur ini mengurangi kemungkinan korupsi data saat beralih dari satu versi atau edisi dari MySQL ke yang lain. Namun fitur tersebut membuat restorasi data memakan banyak waktu dikarenakan eksekusi beberapa pernyataan SQL. Berbeda dengan MySQL, SQL Server tidak memblokir *database* saat membuat *backup* data. Fitur ini memungkinkan pengguna untuk membuat *backup* dan memulihkan sejumlah besar data tanpa waktu dan usaha yang besar.

Opsi untuk Menghentikan Eksekusi Query

MySQL tidak mengizinkan pengguna untuk membatalkan *query* ketika sedang berjalan. Pengguna harus membatalkan seluruh proses untuk menghentikan eksekusi *query* SQL. Sedangkan SQL Server dapat membatalkan *query* yang sedang di eksekusi tanpa menghentikan keseluruhan proses. Selain itu, SQL Server juga menggunakan mesin transaksional untuk menjaga agar tetap konsisten.

Keamanan

MySQL memungkinkan pengembang memanipulasi file *database* melalui binari saat menjalankannya. Bahkan juga memungkinkan file *database* diakses dan dimanipulasi oleh proses lain saat *runtime*.

SQL Server tidak mengizinkan proses untuk mengakses atau memanipulasi file *database* atau binernya. Sehingga untuk memanipulasi file, pengguna harus menjalankan fungsi tertentu. Oleh karena itu, para *hacker* tidak dapat memanipulasi atau mengakses data secara langsung.

Edisi

Pengguna dapat memilih dari dua versi yang disediakan MySQL yaitu MySQL Community Server atau MySQL Enterprise Server. MySQL Community Server bersifat *open source* dan gratis, sedangkan MySQL Enterprise Server hadir dengan sejumlah ekstensi.

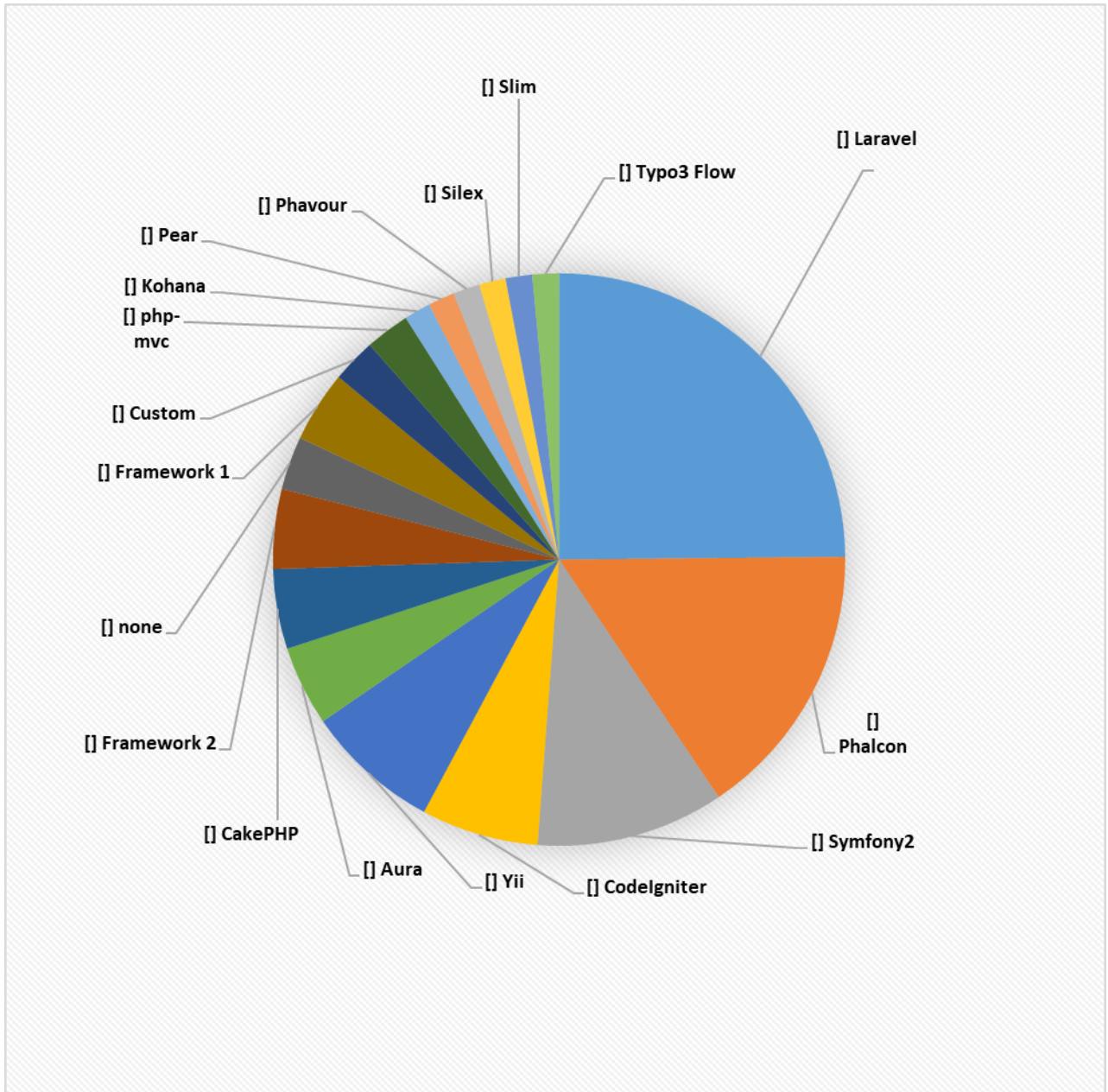
SQL Server tersedia dalam edisi utama dan khusus. Pengguna dapat memilih mana yang akan digunakan dari *enterprise*, *standard*, *web*, *workgroup*, atau *express edition* dari SQL Server. Selain itu pengguna juga dapat menggunakan edisi khusus dari RDBM seperti *azure*, *compact*, *developer*, *embedded*, *evaluation*, *fast track*, and *localDB*.

2.10 Framework Laravel

Framework merupakan kerangka kerja dapat diartikan sebagai kumpulan dari *library* (*class*) yang dapat diturunkan, atau dapat langsung dipakai fungsinya oleh modul-modul atau fungsi yang akan kita kembangkan. Suatu *framework* mendukung perancangan perangkat lunak yang dapat diorganisir dan dikembangkan[1].

Keberadaan *framework* sangat membantu *programmer* dalam pembuatan aplikasi karena dengan adanya *framework*, *programmer* tidak lagi membuat segala sesuatu dari awal melainkan *programmer* hanya perlu memperhatikan pembuatan modul-modul pada aplikasi yang akan dibuat karena *framework* telah menyediakannya[25].

Framework yang akan digunakan pada pengembangan sistem informasi ini adalah *framework laravel*. Laravel adalah *framework* PHP dengan kode terbuka (*open source*) dengan desain MVC (*Model-View-Controller*) yang digunakan untuk membangun aplikasi *website*. *Framework* ini pertama kali dibangun oleh Taylor Otwell. Laravel dibuat untuk mengembangkan aplikasi *web* dengan menggunakan MVC. *Framework laravel* ini bersifat *free open source* dan berjalan di atas PHP (Object-Oriented). *Framework* ini menjadi *framework* yang paling banyak digunakan oleh *programmer* dunia dan menjadi salah satu *framework* terbaik. *Framework laravel* menduduki urutan pertama dengan popularitas sebesar 24,85%. Hal tersebut dapat dilihat pada Gambar 8 berikut[26].



Gambar 8. Perbandingan Framework PHP

Adapun kelebihan dari *framework laravel* yaitu[25]:

1. Laravel memiliki banyak fitur yang tidak dimiliki oleh *framework* lain.
2. Laravel adalah *framework PHP* yang *expressif*, artinya ketika melihat suatu sintaks Laravel, seorang *programmer* akan langsung tahu kegunaan dari sintaks tersebut meskipun belum pernah mempelajarinya apalagi menggunakannya.
3. Laravel bersifat *simple* dikarenakan adanya Eloquent ORM. *Eloquent Object Relation Maping* adalah fitur dari laravel yang memudahkan dalam menjalankan *query database* misalnya *insert*, *update*, *delete*, dan *read* data.
4. Laravel dibuat dengan dokumentasi yang selengkap mungkin. *Code developer*nya dari Laravel selalu menyertakan dokumentasi yang lengkap setiap merilis versi terbarunya.
5. Laravel didukung oleh *Composer*. *Composer* merupakan *Dependency Management PHP* yang membantu untuk mendapatkan *library* yang akan digunakan dan menginstalnya dari internet. Dengan menggunakan *composer*, *library-library* dapat diperoleh dengan mudah dari internet.
6. *Programmer* banyak menggunakan laravel, sehingga banyak tercipta *library* yang mendukung laravel dari penggunaan *programmer*.
7. Laravel memiliki *template engine* tersendiri yaitu *blade*, yang berfungsi memudahkan dalam menampilkan data pada *template HTML*.

Fitur-Fitur Laravel

Pada Tabel 4 akan dijelaskan fitur-fitur yang terdapat pada laravel[25].

Tabel 4. Fitur-fitur Laravel

No	Nama Fitur	Keterangan
1.	Artisan	Perintah-perintah yang diketikkan pada <i>command prompt (command-line interface)</i> yang digunakan untuk melakukan tugas tertentu pada saat membuat aplikasi.
2.	Migration	<i>Migration</i> digunakan dalam pembuatan skema database. Skema database tersebut dapat diperbaharui sehingga tetap menjadi terbaru.
3.	Model	Model digunakan untuk berinteraksi dengan database.
4.	Seeder	<i>Seeder</i> digunakan untuk membuat contoh data pada database.
5.	Routing	<i>Routing</i> digunakan saat mengakses suatu halaman pada aplikasi melalui URL.
6.	Query Builder dan Eloquent ORM	<i>Query builder</i> digunakan untuk menjalankan <i>query</i>

		database dengan lebih mudah dan aman. Dengan <i>query builder</i> juga <i>developer</i> tidak perlu membuat model terlebih dahulu untuk berinteraksi dengan database. <i>Eloquent ORM (Object Relational Mapping)</i> digunakan untuk memetakan basis data ke model objek.
7.	Relationship	<i>Relationship</i> digunakan untuk memudahkan pengelolaan data pada beberapa tabel yang memiliki relasi satu sama lainnya.
8.	View dan Blade Template	<i>View</i> digunakan untuk menampilkan informasi yang telah dibuat kepada pengguna. <i>Blade template</i> digunakan untuk memudahkan <i>developer</i> dalam menampilkan data pada view.
9.	Controller	<i>Controller</i> merupakan jembatan penghubung <i>view</i> dengan model yang berfungsi untuk memproses semua <i>request</i> dan akan ditampilkan ke <i>view</i> .
10.	Authentication	<i>Authentication</i> digunakan untuk membatasi hak akses pengguna melalui <i>login</i> . Dengan adanya fitur ini, maka <i>developer</i> tidak perlu lagi membuat halaman <i>login</i> , melainkan hanya memodifikasi saja.
11.	Middleware	<i>Middleware</i> digunakan untuk memproteksi <i>route</i> dari <i>user</i> yang tidak memiliki hak akses. Jadi <i>user</i> harus memiliki hak akses terlebih dahulu, sehingga dapat mengakses <i>route</i> .

2.11 JavaFX

JavaFX adalah *toolkit graphical user interface* (GUI) dari Java yang memungkinkan pengembang membangun aplikasi *cross-platform* yang beraneka ragam dengan cepat. Dibangun dari dasar, JavaFX mengambil keuntungan dari GPU modern melalui grafis perangkat keras yang dipercepat selama penyediaan antarmuka pemrograman yang dirancang baik yang memungkinkan pengembang menggabungkan kontrol grafis, animasi, dan *user interface*. JavaFX 8 adalah API (*application programming interface*) bahasa Java murni. Tujuan dari JavaFX adalah untuk digunakan di berbagai jenis perangkat, seperti *smartphone*, *TV*, komputer, dan *desktop*[27].

Sebelum JavaFX dibentuk, pengembangan aplikasi sisi klien melibatkan banyak *libraries* dan API yang terpisah untuk mencapai aplikasi dengan fungsional yang tinggi. *Libraries* tersebut mencakup API media, *UI controls*, *web*, 3D, dan 2D. Karena mengintegrasikan API secara bersama-sama bisa menjadi lebih sulit, para *engineer* di Oracle menciptakan kumpulan *libraries* JavaFX baru yang menggabungkan semua kemampuan yang sama dalam satu rumpun[27].

BAB 3 PERENCANAAN DAN ANALISIS

Bab III Perencanaan berisi uraian mengenai inisialisasi proyek, *project plan*, dan analisis yang dilakukan untuk mendapatkan rancangan aplikasi yang akan dikembangkan.

3.1 Inisialisasi Proyek

Rumah Sakit Umum Daerah Doloksanggul merupakan *client* dari pembangunan sistem ini. Sistem yang akan dibangun adalah Sistem Informasi Rumah Sakit yang berbasis *desktop*, *mobile*, dan *web* yang saling terintegrasi dan dimana ketiganya akan dihubungkan oleh *web service*. Pengerjaan sistem informasi rumah sakit ini mencakup tiga modul yaitu modul registrasi, modul rekam medis, dan modul pembayaran.

3.2 Project Plan

Berikut akan dijelaskan bagaimana *developer* membangun sistem informasi rumah sakit:

1. Sistem yang akan dibangun akan menerapkan pendekatan SDLC dengan menggunakan metode MDD.
2. Kelompok yang beranggotakan tiga orang dibagi dalam hal pengerjaan aplikasi *desktop*, *mobile*, dan *web*.
3. Untuk memanajemen waktu maka *developer* membuat target pengerjaan setiap minggunya.
4. Melakukan bimbingan dengan pembimbing mengenai sistem yang akan dibangun.
5. Melakukan kunjungan ke Rumah Sakit Umum Daerah Doloksanggul untuk melakukan *requirement gathering*, validasi, dan UAT (*User Acceptance Testing*).

3.3 Pengumpulan Data

Pada subbab ini akan dijelaskan mengenai pengumpulan data yang dilakukan di Rumah Sakit Umum Daerah Doloksanggul Kabupaten Humbang Hasundutan Provinsi Sumatera Utara yang akan menjadi pengguna sistem. Berikut akan dijelaskan tahapan dalam melakukan pengumpulan data.

3.3.1 Persiapan Survei

Pada bagian ini dijelaskan mengenai persiapan survei yang dilakukan. Pada survei ini dilakukan di Rumah Sakit Umum Daerah Doloksanggul. Teknik yang dipilih dalam melakukan survei yaitu wawancara. Wawancara merupakan kegiatan tanya jawab yang dilakukan oleh pewawancara sebagai penanya dan narasumber sebagai orang yang ditanya. Sebelum melakukan survei adapun persiapan yang dilakukan, yaitu:

1. Melakukan survei internet terlebih dahulu terkait dengan hal yang dipermasalahkan. Hal ini dilakukan untuk memperoleh informasi awal sebagai bahan untuk melaksanakan survei.
2. Membuat *list-list* pertanyaan yang akan diberikan kepada narasumber. Pertanyaan yang akan diajukan harus sesuai dengan topik yang diambil. List pertanyaan dapat dilihat pada Lampiran 2.

3.3.2 Pelaksanaan Survei

Pada bagian ini akan dijelaskan mengenai bagaimana pengamatan melalui aktivitas survei yang dilakukan. Pelaksanaan survei yang dilakukan adalah melakukan pertemuan dengan salah satu staf Rumah Sakit Umum Daerah Doloksanggul yang menjadi penyedia informasi. Setelah melakukan perkenalan, maka *developer* memberikan pertanyaan-pertanyaan kepada *client* terkait dengan sistem yang berjalan saat ini di rumah sakit umum daerah doloksanggul dan *requirement* yang diinginkan pihak rumah sakit dalam pembangunan aplikasi.

Selain itu, *developer* juga mengamati tata letak dari rumah sakit yaitu sebagai berikut. Pada RSUD Doloksanggul terdapat 3 bangunan. Pada bangunan yang pertama yaitu loket merupakan tempat pasien melakukan registrasi. Pada tempat inilah pertama kali pasien berinteraksi dengan pegawai rumah sakit. Jika pasien merupakan pasien BPJS, maka terlebih dahulu mendaftar ke kantor BPJS RSUD yang tersedia pada tempat ini.

Pada bangunan yang kedua merupakan poliklinik dari rumah sakit. Poliklinik yang tersedia dari rumah sakit adalah poliklinik umum, gigi, penyakit dalam, anak, kebidanan/kandungan/obgin, bedah, mata, THT, neurologi/saraf, dan fisioterapi. Di dalam Poliklinik ini juga terdapat kasir, dimana setiap pasien yang ingin melakukan pembayaran harus ke kasir.

Bangunan yang ketiga merupakan rumah sakit. Pada tempat ini tersedia ruangan-ruangan yang menjadi tempat tidur dari pasien rawat inap. Pada tempat ini juga dilakukan pembelian obat dan pembayaran semua tagihan pasien kepada kasir.

3.3.3 Hasil Survei

Adapun hasil yang di dapatkan setelah melakukan survei, yaitu:

1. *Developer* mengetahui bagaimana tata letak rumah sakit umum daerah doloksanggul.
2. *Developer* mengetahui bagaimana sebenarnya alur atau bisnis proses perjalanan pasien dari mulai mendaftar hingga sampai pulang dari rumah sakit.
3. *Developer* mengetahui bagaimana sistem yang berjalan saat ini (*current system*) pada rumah sakit umum daerah doloksanggul.
4. *Developer* mendapatkan data mengenai apa saja *requirement* yang diinginkan oleh rumah sakit umum daerah doloksanggul.

3.4 Analisis Registrasi, Billing System, dan Rekam Medis

Pada subbab ini akan dijelaskan mengenai analisis terhadap registrasi, *billing system*, dan rekam medis yang ada di rumah sakit.

3.4.1 Analisis Registrasi

Modul registrasi pada sistem informasi ini dibatasi dalam cakupan:

1. Pencatatan registrasi pasien lama dan baru, baik anak-anak, dewasa, dan ibu hamil yang memilih perawatan rawat jalan dengan status pasien sebagai pasien umum, jaminan asuransi/perusahaan, miskin, ataupun pasien rujukan.
2. Pencatatan registrasi pasien MCU (*Medical Check Up*) paket, baik MCU pribadi ataupun MCU rekanan.
3. Pencatatan registrasi pasien pindah ruang rawat inap.
4. Pencatatan penerimaan hasil MCU pasien.
5. Pencatatan pengambilan hasil MCU pasien.
6. Pembuatan laporan yang berhubungan dengan registrasi rawat inap, rawat jalan, dan MCU.

Fitur-fitur yang disediakan pada modul registrasi adalah:

1. Reservasi pendaftaran, baik melalui telepon maupun website.
2. Pendaftaran menggunakan barcode reader.
3. Pencatatan *medical record number* secara otomatis.
4. Pencatatan biodata pasien dan data penanggung jawab.
5. Pencatatan data kunjungan pasien, baik untuk pasien baru ataupun pasien lama.
6. Pencatatan kelompok pasien, seperti: Umum, BPJS, ataupun perusahaan.

7. Pencetakan bukti pendaftaran.
8. Pencetakan kartu pasien menggunakan barcode.
9. Pencetakan kartu status.
10. Pencarian data pasien yang sedang menjalani masa perawatan.
11. History kunjungan pasien pada halaman pendaftaran.

Selain dari fitur-fitur yang ada di atas, modul registrasi menghasilkan laporan sebagai berikut:

1. Jumlah kunjungan per unit (Rawat Jalan/Rawat Inap/IGD)
2. Jumlah kunjungan per unit per cara bayar
3. Jumlah kunjungan per unit per wilayah
4. Jumlah kunjungan per unit per cara kunjungan
5. Laporan detail buku register dan summary per unit
6. Laporan detail sensus harian per unit
7. Laporan transaksi yang dibatalkan per unit

3.4.2 Analisis Billing System

Billing system merupakan sistem penagihan berbasis elektronik yang digunakan dalam mempercepat dan mempermudah sistem penagihan yang ada di rumah sakit. Billing system suatu sistem yang sangat penting dalam rumah sakit, khususnya dalam kelancaran dan stabilisasi keuangan dalam sebuah rumah sakit.

Pada menu transaksi dan tindakan *billing system* dilengkapi dengan bagian-bagian yang terdiri dari nomor transaksi, nomor registrasi, NIP, jaminan, nama pasien, alamat, kota, umur, jenis kelamin, tanggal dan jam transaksi, kelas, layanan, tipe tarif, kode tarif, serta dilengkapi dengan *form* keterangan yang berisi mengenai layanan yang telah diberikan kepada pasien beserta total atau jumlah biaya yang harus dibayar pasien. Format transaksi tindakan dilengkapi dengan kolom yang berisi mengenai kode, tindakan medis, tarif, diskon dan subtotal.

Pada transaksi registrasi keluar, format dari *billing system* berisi mengenai nomor registrasi, nomor rekam medik, nomor kwitansi, tanggal/jam keluar, nama pasien, alamat, dan kota. Pada kelengkapan data penagihan berisi mengenai *claim/jaminan utama*, subsidi/diskon, deposit /uang muka, pembayaran tunai, total kekurangan biaya yang diidentifikasi dengan kekurangan biaya pelayanan dan biaya obat-obatan, dan total biaya. Terdapat juga kolom yang berisi mengenai tanggal, keterangan, biaya, jumlah, subtotal, keterangan, dibayar, dan juga kolom kekurangan.

Fitur-fitur pada billing system:

1. Pencatatan jasa tindakan yang dilakukan dokter, perawat, asisten, selama pasien di rawat jalan dengan otomatis sesuai dengan golongan kategori pasien.
2. Pencetakan karcis pelayanan rawat jalan secara otomatis dari *form* registrasi pasien.
3. Master kontrak kerjasama perusahaan atau asuransi (Askes, Jamsostek, Askin) maupun BPJS
4. Breakdown billing pasien perusahaan atau asuransi

Fitur-fitur billing pasien rawat darurat:

1. Membuat pelabelan sesuai status kedaruratan penderita
2. Melakukan pencetakan retribusi layanan pemeriksaan rawat darurat, status pasien, secara otomatis menyesuaikan dengan tarif masing-masing kategori pasien.
3. Melakukan pencatatan jasa tindakan dokter, perawat, asisten selama pasien di UGD/IGD.
4. Melakukan pencetakan karcis pelayanan rawat darurat secara otomatis pada *form* registrasi pasien UGD/IGD.

Fitur-fitur billing pasien rawat jalan:

1. Melakukan pencetakan tagihan layanan pemeriksaan.
2. Status pasien, secara otomatis akan menyesuaikan dengan tarif pada setiap ruang rawat (poli) dengan masing-masing kategori pasien.
3. Melakukan pencetakan laporan pendapatan pada masing-masing departemen.
4. Melakukan pencetakan laporan verifikasi Jasmkesmes, Jamkesda, Askes.
5. Melakukan pencetakan SJP (Surat Jaminan Pelayanan) rawat jalan
6. Breakdown tarif jasa pelayanan dan jasa sarana.
7. List billing pasien rawat inap
8. List billing pasien rawat jalan
9. List billing pasien penunjang
10. List billing BPJS
11. Penerimaan pembayaran biaya perawatan
12. Penerimaan pembayaran pembelian obat tunai
13. Penerimaan pembayaran selisih biaya pasien BPJS
14. Pengalihan tagihan perawatan
15. Entry selisih biaya perawatan pasien BPJS

16. Pemberian diskon biaya perawatan
17. Cetak kwitansi perawatan
18. Cetak tagihan perawatan
19. Cetak penerimaan cash biaya perawatan
20. Cetak pembayaran hutang
21. Informasi penerimaan cash
22. Cetak tagihan ke debitur/BPJS
23. Cetak pengalihan biaya perawatan
24. Cetak potongan biaya perawatan
25. Cetak subsidi/discount biaya perawatan

Fitur-fitur billing pasien rawat inap:

1. Melakukan pencetakan tanda bukti pembayaran jika pasien pulang.
2. Pembagian pendapatan dapat dialokasi sampai dengan sub yang lebih detail, misalnya jasa sarana, jasa pelayanan, dan lain sebagainya.
3. Mendukung multi kategori pasien, sehingga pembiayaan dapat diorganisir pada golongan tunai maupun piutang untuk pasien kerjasama.
4. Mendukung penentuan tarif tindakan berdasarkan kelas perawatan pasien.
5. Mendukung sistem penanganan multi petugas medis, misalnya tindakan operasi yang biasanya dilakukan lebih dari satu petugas.
6. Mendukung sistem pelaporan pendapatan petugas medik secara periodik
7. Pencatatan jasa tindakan dokter, perawat, asisten selama pasien di rawat inap
8. Layanan billing pasien pindah kelas layanan rawat inap
9. Informasi mengenai tempat tidur yang tidak berisi untuk layanan rawat inap
10. Informasi lokasi pasien di rawat untuk layanan rawat inap
11. Informasi billing pasien yang sedang dirawat berupa bill terbayar, terhutang ataupun klaim pasien asuransi untuk layanan rawat inap
12. Informasi billing pasien yang sedang dirawat berupa jasa medis maupun pemakaian obat apotik.
13. Melakukan pencetakan billing pasien pulang untuk layanan rawat inap.
14. Membuat laporan statistik rujukan.

Fitur-fitur yang disediakan pada modul rawat jalan di sistem ini adalah:

1. Pemasukan data jenis pemeriksaan
2. Pemasukan obat dan alkes

3. History kunjungan pasien
4. History rekam medis pasien
5. Rujuk pasien ke rumah sakit lain

Selain fitur-fitur yang disebut di atas, modul rawat jalan menghasilkan laporan sebagai berikut:

1. Daftar pasien rawat jalan
2. Daftar pasien yang dirujuk ke rumah sakit lain
3. Daftar pasien yang dirujuk ke Rawat Inap
4. Daftar pasien yang dirujuk ke poli lain
5. Daftar pasien yang dirujuk ke penunjang medis

Fitur-fitur yang disediakan pada modul rawat inap di sistem ini adalah:

1. Pemanfaatan yang ada di ruangan Nurse Station
2. Pemasukan jenis pemeriksaan
3. Pemasukan obat dan alkes
4. Pemasukan penunjang medis
5. Pemindahan bed
6. Pemindahan ruangan atau kelas perawatan
7. History kunjungan pasien
8. History rekam medis pasien

Selain fitur-fitur yang telah disebut di atas, modul rawat inap menghasilkan laporan sebagai berikut:

1. Daftar pasien rawat inap yang sedang menjalani masa perawatan
2. Daftar pasien yang sudah pulang (*check out*)

4.2.3 Analisis Rekam Medis

a. Pasien

Fungsionalitas sistem pasien adalah:

1. Fungsi *login*
Fungsi ini digunakan untuk dapat masuk ke dalam sistem.
2. Fungsi melihat data rekam medik
Fungsi ini digunakan untuk melihat data tentang rekam medik yang pernah dilakukan oleh pasien.
3. Fungsi *search*

Fungsi ini digunakan untuk mencari data rekam medik dengan menggunakan *keyword* yang diinginkan.

4. Fungsi ubah kata kunci

Fungsi ini digunakan untuk mengubah kata kunci pengguna

b. Dokter

Fungsionalitas sistem dokter adalah:

1. Fungsi mengelola data

Fungsi ini digunakan untuk menambah, mengubah, dan menghapus informasi yang berhubungan dengan data pasien, persetujuan tindakan medik dan surat rawat jalan.

2. Fungsi *login*

Fungsi ini digunakan untuk dapat masuk ke dalam sistem.

c. Bagian Administrasi

Fungsionalitas sistem untuk bagian administrasi adalah:

1. Fungsi mengelola data

Fungsi ini diigunakan untuk menambah, mengubah, dan menghapus data user, informasi yang berkaitan dengan data pasien, persetujuan tindakan medik dan surat rawat jalan pada sistem.

2. Fungsi mencetak laporan

Fungsi ini digunakan untuk mencetak laporan statistik pasien per kategori, surat keterangan kematian, dan surat asuhan keperawatan.

Fitur-fitur yang tersedia pada rekam medik:

1. Pencatatan data sosial pasien, data morbiditas dengan pedoman sistem informasi rumah sakit.
2. Pencarian riwayat kunjungan pasien yang pernah dilakukan pasien pada masing-masing poli rawat jalan.
3. Pencatatan rekam medis pasien rawat jalan sesuai dengan asuhan keperawatan SOAP (Subjective Objective Assesment Planning) untuk mencatat Therapy pasien.
4. Pencatatan no rekam medis pasien rawat jalan secara otomatis.
5. Layanan pelaporan standart Departemen Kesehatan
6. Penyesuaian ICD dan DTD pada tiap-tiap ruang rawat dapat dilakukan sendiri oleh petugas dengan mudah.

Fitur rawat jalan :

1. Informasi pasien
2. List data pasien
3. Isi diagnosa poliklinik
4. Isi tindakan poliklinik
5. Isi dokter poliklinik
6. Check out poliklinik

Fitur rawat inap :

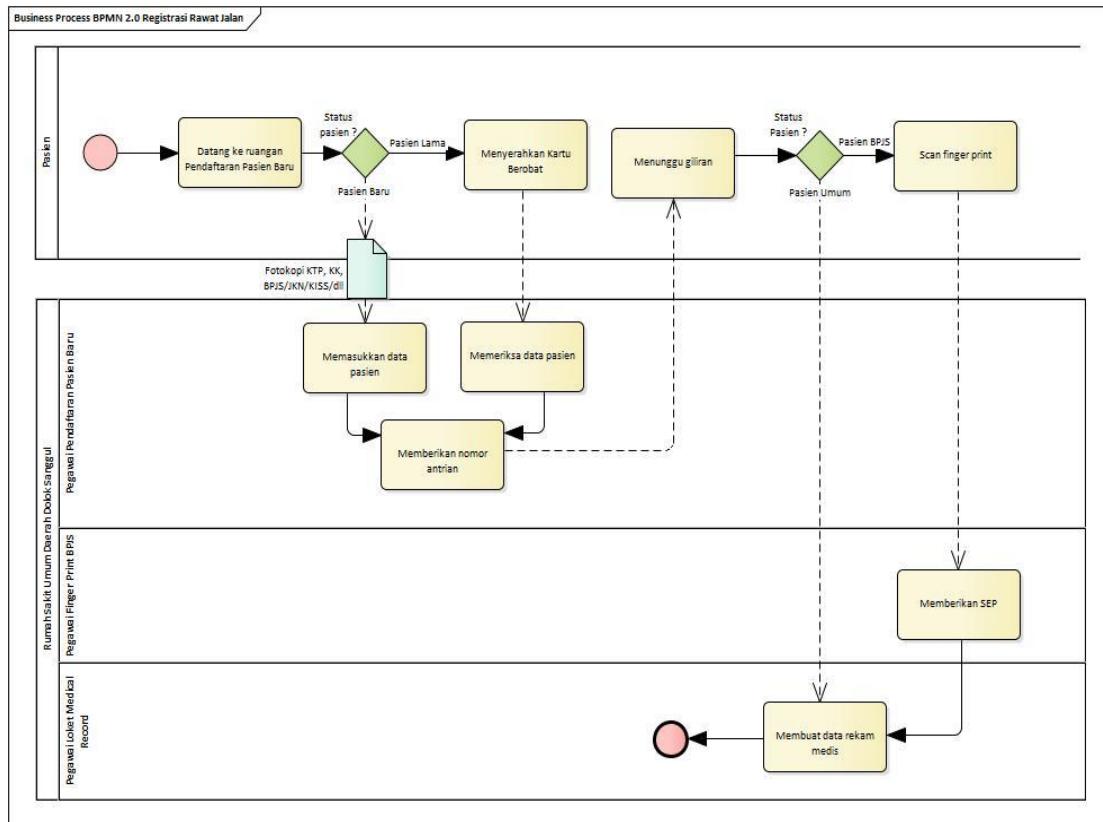
1. Informasi pasien
2. List data pasien
3. Registrasi ruang perawatan
4. Isi biaya kamar/akomodasi otomatis
5. Isi DPJP
6. Isi diagnosa masuk
7. Isi diagnosa kerja
8. Isi diagnosa akhir (primer sekunder)
9. Isi tindakan ruang
10. Pindah kamar
11. Pindah ruang perawatan
12. Informasi biaya rawat inap
13. Informasi biaya penunjang
14. Informasi biaya perawatan
15. Informasi ruang perawatan
16. Informasi kamar terisi dan kamar kosong
17. Informasi detail pasien per kamar
18. Check out pasien
19. Batal check out

3.5 Current System

Pada subbab ini akan diuraikan bagaimana analisis dan penentuan kebutuhan dilaksanakan.

1. Registrasi Rawat Jalan

Berikut merupakan bisnis proses dari registrasi rawat jalan.



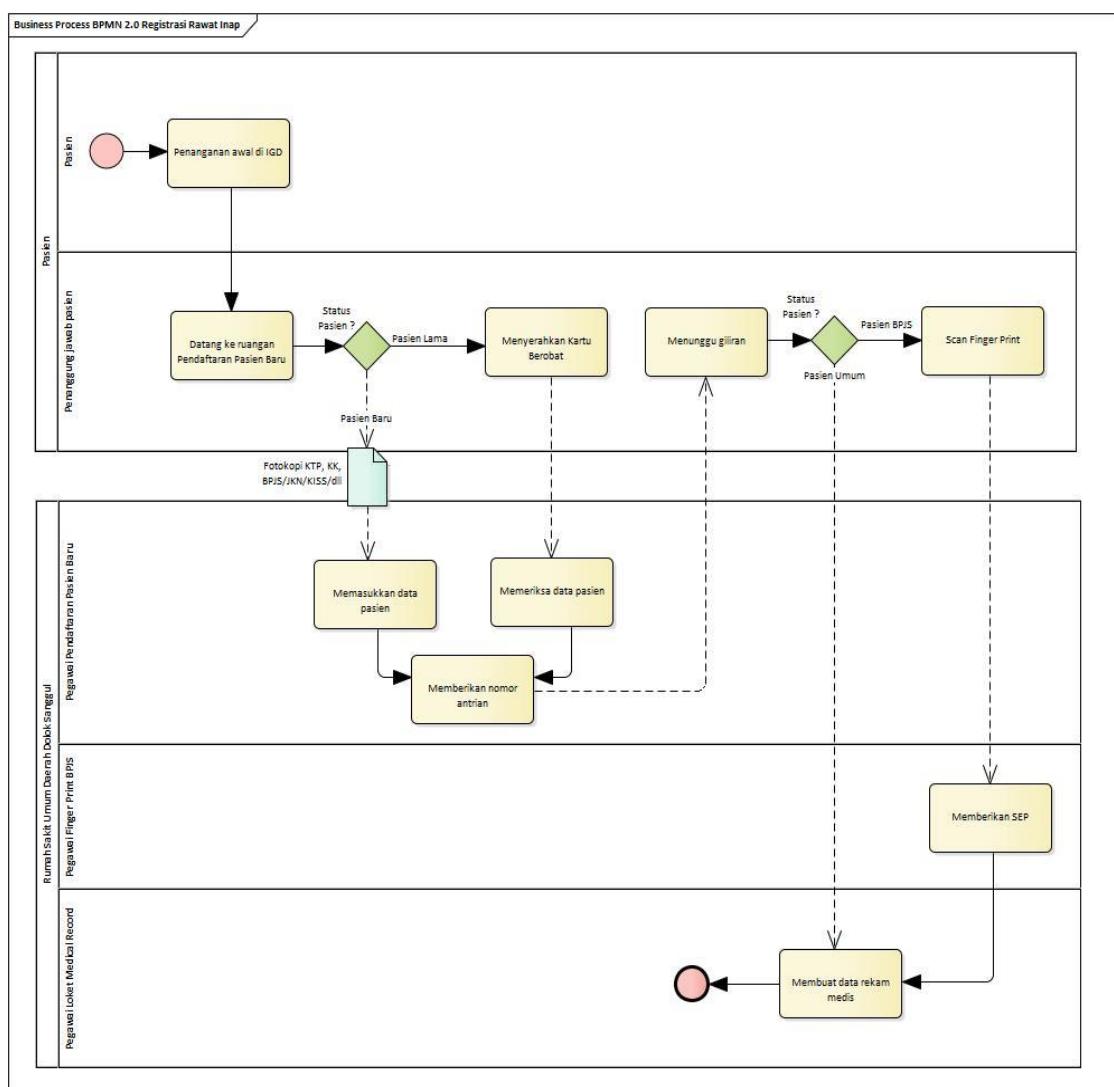
Gambar 9. Current System Registrasi Rawat Jalan

Pada proses registrasi rawat jalan, seorang pasien harus terlebih dahulu mendatangi ruangan Pendaftaran Pasien Baru. Pasien yang merupakan pasien baru (belum pernah tercatat sebagai pasien di rumah sakit) harus membawa fotokopi KTP, KK, BPJS/JKN/KISS/dll yang akan digunakan untuk memasukkan data valid pasien di ruang Pendaftaran Pasien Baru. Namun pasien yang merupakan pasien lama (sudah pernah tercatat sebagai pasien di rumah sakit) hanya perlu menyerahkan Kartu Berobat kepada pegawai untuk dicek keberadaan datanya dalam sistem. Jika pasien tidak membawa Kartu Berobat, maka pasien cukup memberikan identitas umum pasien kepada pegawai untuk digunakan sebagai pencarian dalam menemukan keseluruhan data pasien dalam sistem. Setelah melakukan input atau verifikasi data di ruang Pendaftaran Pasien Baru, pasien akan diberikan nomor antrian dan pasien akan menunggu giliran di ruang tunggu. Pasien akan dipanggil satu per satu sesuai nomor

antrian oleh pegawai. Setelah dari ruang tunggu, pasien akan dipilah sesuai statusnya. Jika pasien merupakan pasien BPJS, maka pertama sekali pasien akan menuju ruang Finger Print untuk memastikan bahwa pasien merupakan pasien BPJS dan pasien akan menerima SEP (Surat Eligibilitas Peserta) untuk menjamin bahwa pasien boleh berobat secara gratis. Setelah itu, pasien BPJS akan menuju Loket Medical Record untuk mengurus data rekam medis pasien. Jika tidak (melainkan pasien umum), pasien akan langsung menuju Loket Medical Record.

2. Proses Registrasi Rawat Inap

Berikut merupakan bisnis proses dari rawat inap.



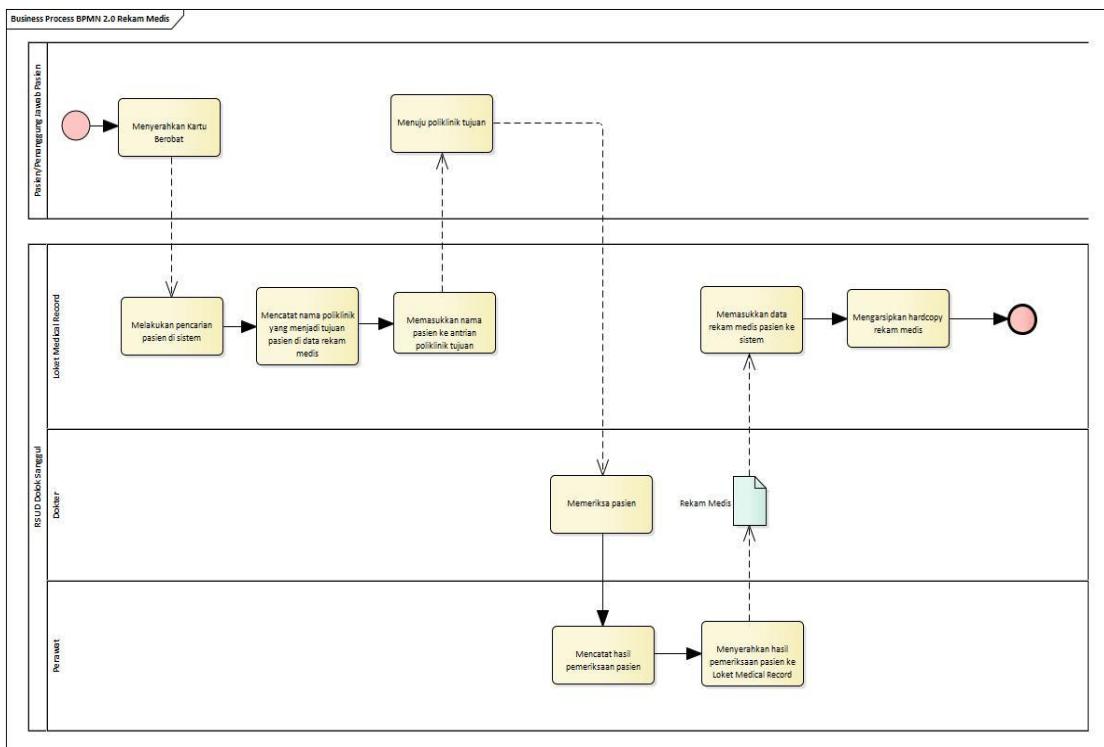
Gambar 10. Current System Registrasi Rawat Inap

Pasien dengan keadaan kesehatan yang darurat atau yang membutuhkan penanganan awal yang cepat akan dimasukkan ke IGD (Instalasi Gawat Darurat). Selama penanganan, pasien dapat menerima pemeriksaan penunjang dari dokter. Selama

pasien ditangani, penanggung jawab pasien akan pergi ke ruangan Pendaftaran Pasien Baru untuk mendaftarkan pasien di rumah sakit. Penanggung jawab pasien akan mengikuti alur seperti yang ada di rawat jalan hingga ke ruangan Loket Medical Record untuk memperoleh data rekam medis pasien.

3. Proses Rekam Medis

Berikut merupakan bisnis proses dari rekam medis.



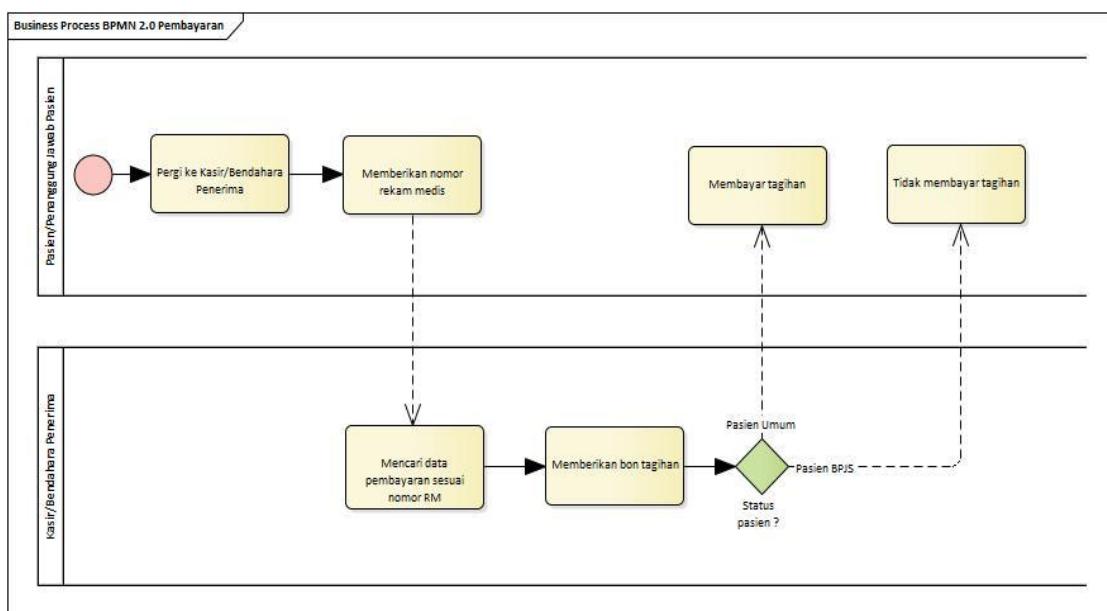
Gambar 11. Current System Proses Rekam Medis

Pasien maupun penanggung jawab pasien mengurus data rekam medis pasien di Loket Medical Record. Pada saat di Loket Medical Record, pasien maupun penanggung jawab pasien hanya perlu memberitahu informasi umum pasien untuk melakukan pencarian di sistem sebab sebelumnya telah dilakukan pengisian data pasien di ruang Pendaftaran Pasien Baru. Bagi pasien rawat jalan, saat di Loket Medical Record akan ditanyakan Poliklinik yang menjadi tujuan pasien dan data tersebut akan segera dicatat oleh petugas di data rekam medis pasien. Bagi pasien rawat inap, data rekam medis dibawa ke ruangan dan akan dilakukan pengisian data seperti keluhan, diagnosa dari dokter, tindakan medis dan jasa pelayanan yang diberikan dokter terhadap pasien, dan lain-lain. Kegiatan pemeriksaan terhadap pasien dilakukan oleh dokter dan perawat. Dokter melakukan pemeriksaan dan perawat mencatat setiap detail hasil pemeriksaan yang dilakukan dokter terhadap pasien di form rekam medis yang sebelumnya

diperoleh dari Loket Medical Record. Setiap lembar form hasil pemeriksaan pasien yang ditulis oleh perawat akan dikumpulkan kembali di Loket Medical Record. Di Loket Medical Record, data dalam form hasil pemeriksaan tersebut akan dimasukkan ke sistem oleh pegawai dan form tersebut diarsipkan dalam sebuah map besar.

4. Proses Billing System

Berikut merupakan bisnis proses dari *billing system*.



Gambar 12. Current System Proses Billing System

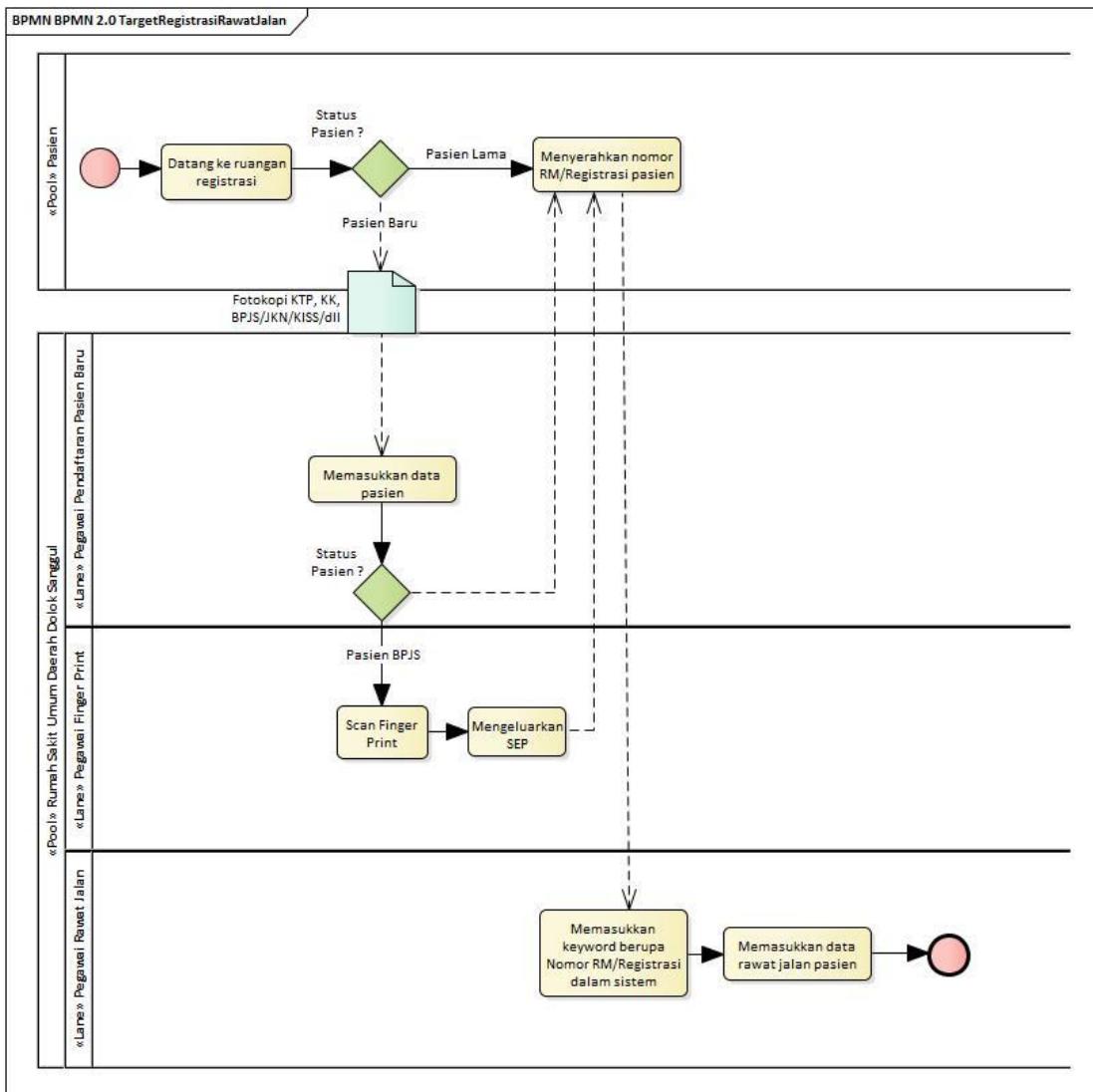
Rekam medis yang berisi hasil pemeriksaan dan tindakan yang dilakukan dokter terhadap pasien akan digunakan sebagai acuan dalam proses pembayaran. Pembayaran yang dilakukan oleh pasien tergantung pada tindakan medis dan jasa pelayanan yang diterima, obat yang dikonsumsi, serta ruangan yang digunakan oleh pasien. Pembayaran yang sah hanya dapat dilakukan di Kasir (Bendahara Penerima) dan tidak dapat dilakukan di tempat lain. Pasien yang merupakan pasien BPJS memiliki tarif pembayaran, namun semuanya akan ditanggung oleh pihak BPJS. Sedangkan pasien umum (non BPJS) akan melakukan pembayaran penuh terhadap total biaya yang ditangguhkan selama ia menjalani proses medis di rumah sakit.

3.6 Target System

Berikut akan dijelaskan target system yang akan dibangun.

1. Registrasi Rawat Jalan

Berikut merupakan bisnis proses dari rawat jalan.



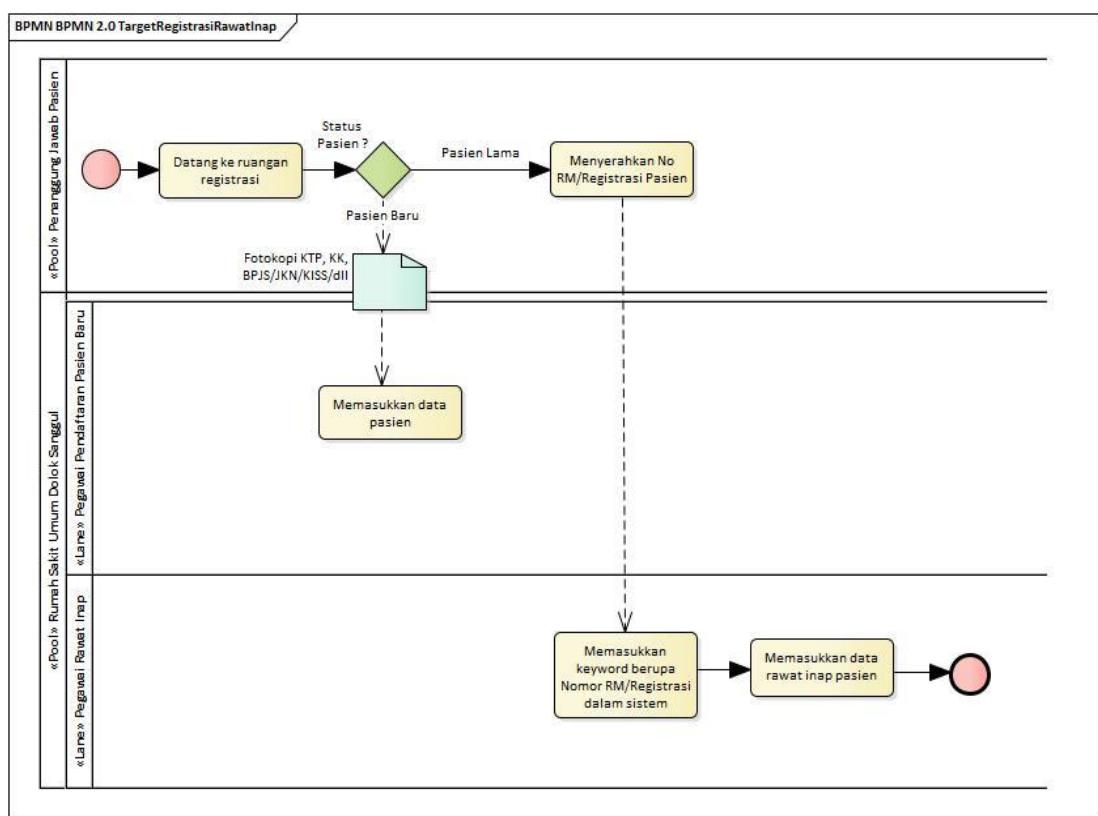
Gambar 13. Target System Registrasi Rawat Jalan

Pegawai registrasi akan melakukan pemasukan data dalam aplikasi desktop. Jika pasien adalah pasien baru (belum pernah melakukan pengobatan di rumah sakit), maka pegawai pendaftaran pasien baru akan memasukkan data pasien secara lengkap yang bersumber dari KTP, KK, BPJS/JKN/KISS/dll terlebih dahulu. Jika pasien adalah pasien BPJS, maka pasien terlebih dahulu harus melakukan *scan finger print* untuk menerima SEP. Namun jika pasien adalah pasien umum, maka tidak perlu melakukan *scan finger print*. Setelah itu, pasien menyerahkan nomor registrasi pasien kepada pegawai rawat jalan untuk dilakukan pencarian sebelum pengisian data rawat

jalan pasien. Jika pasien merupakan pasien lama (sudah pernah melakukan pengobatan di rumah sakit), maka pegawai rawat jalan hanya perlu melakukan pencarian data pasien berdasarkan *keyword* nomor RM pasien dan akan muncul data yang berhubungan dengan *keyword* tersebut setelah pegawai menekan *button* Cari. Selanjutnya, pegawai akan memasukkan data rawat jalan pasien, seperti poliklinik, poli, tanggal masuk, tanggal keluar, nama dokter, dan lain-lain. Pada aplikasi ini, jika pegawai membutuhkan data pasien rawat jalan sesuai tahun tertentu, maka pegawai hanya perlu memasukkan nama tahun dalam kolom pencarian Cetak Pertahun kemudian menekan *button* Cetak Excel dan memilih lokasi penyimpanan *file* tersebut.

2. Registrasi Rawat Inap

Berikut merupakan bisnis proses dari rawat inap.



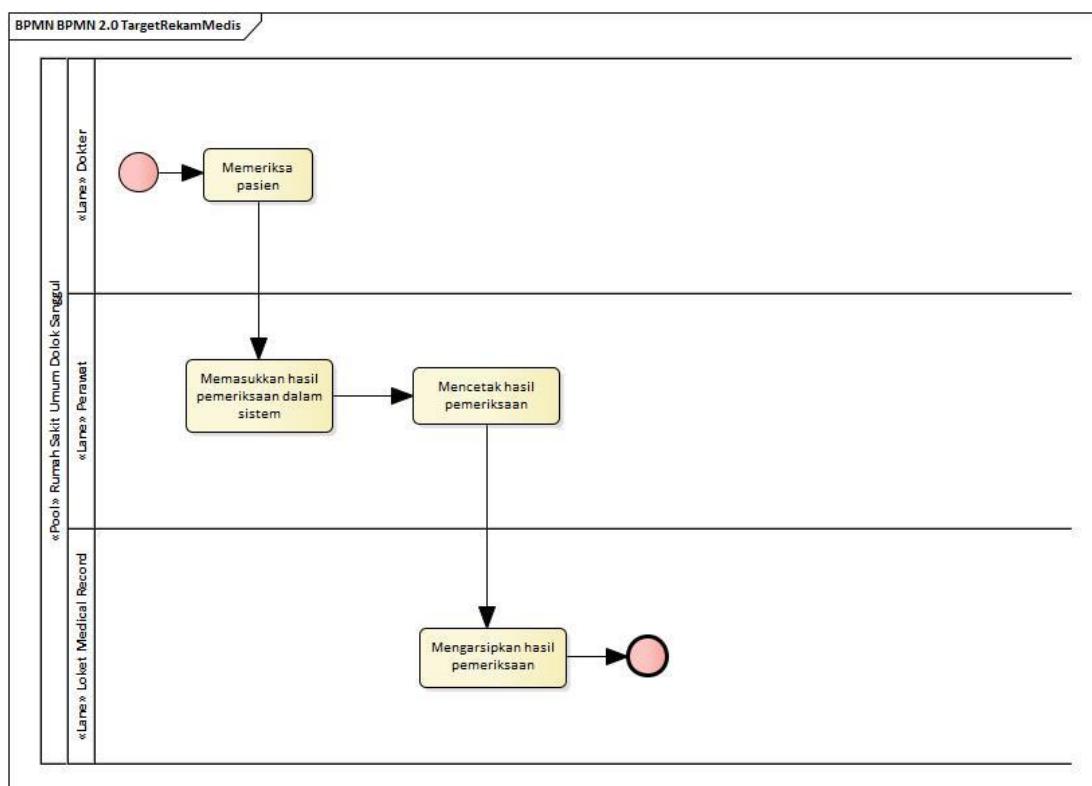
Gambar 14. Target System Registrasi Rawat Inap

Pegawai registrasi akan memasukkan data melalui aplikasi desktop. Jika pasien adalah pasien baru, maka pegawai pendaftaran pasien baru harus memasukkan data pasien secara lengkap yang diwakilkan oleh Penanggung Jawab Pasien. Selanjutnya pasien memberikan nomor registrasi pasien kepada pegawai rawat inap untuk dilakukan pengelolaan data. Namun jika pasien adalah pasien lama, maka pegawai hanya perlu memasukkan nomor RM pasien dan menekan *button* Cari. Setelah menekan *button*

Cari, sistem akan memunculkan data pasien yang telah tersimpan sebelumnya. Kemudian pegawai akan memasukkan data rawat inap pasien seperti asal rujukan, tanggal masuk ke rumah sakit, jam masuk, dan ruangan. Pada aplikasi ini, disediakan juga fitur untuk mencetak data pasien per tahun. Pegawai hanya perlu memasukkan nama tahun di kolom pencarian dan menekan *button* Cetak PDF atau Cetak Excel. Kemudian pegawai akan memilih lokasi penyimpanan *file* yang sesuai.

3. Proses Rekam Medis

Berikut merupakan bisnis proses dari rekam medis.

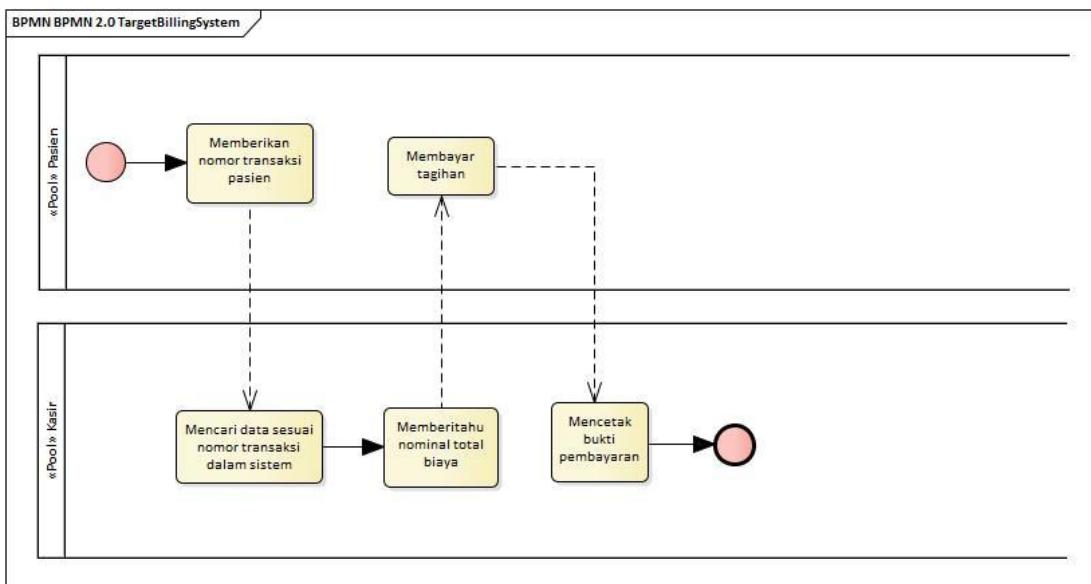


Gambar 15. Target System Proses Rekam Medis

Hasil pemeriksaan yang dilakukan oleh dokter dan perawat dikelola di dalam sistem. Pengolahan data ini dapat dilakukan dalam dua *platform* berbeda, yaitu melalui aplikasi desktop dan aplikasi *mobile*. Data dimasukkan dalam satu *server* sehingga data yang ada dapat dikelola melalui kedua *platform*. Dalam pengolahan data rekam medis ini, data yang muncul tidaklah hanya detail hasil pemeriksaan, melainkan juga total biaya pengobatan pasien yang harus dibayarkan yang kemudian dimasukkan dalam *Billing System*. Jika sudah berhasil mengelola data, hasil pemeriksaan dapat *print* langsung melalui sistem untuk kemudian diarsipkan dalam Loket Medical Record.

4. Proses Billing System

Berikut merupakan bisnis proses dari *billing system*.



Gambar 16. Target System Proses Billing System

Total biaya pengobatan pasien yang diperoleh pada pengelolaan rekam medis akan diproses dalam *Billing System*. Pertama sekali pegawai kasir harus masuk ke dalam sistem. Kemudian pegawai memasukkan nomor transaksi pasien pada kolom Nomor Transaksi. Setelah itu sistem akan menampilkan data pasien beserta total tagihan rumah sakit sesuai nomor transaksi. Pasien dapat membayar penuh tagihan dan dapat mencicil. Ketika pegawai memasukkan nominal yang dibayarkan pasien, sistem akan melakukan kalkulasi biaya untuk jumlah kembalian yang akan diterima pasien. Setelah pengelolaan selesai, pegawai menekan *button Cetak Faktur* untuk mencetak bukti pembayaran yang dilakukan pasien.

3.6.1 User Characteristics

Berikut adalah karakteristik pengguna aplikasi Sistem Informasi Rumah Sakit versi *desktop*, *mobile*, dan *web*.

1. User Group 1

Description of User : Admin

Role : Admin dapat menambah, mengedit, menghapus akun, ruangan, dokter, perawat, dan obat

Prerequisite : Admin harus *login* terlebih dahulu untuk masuk kedalam sistem

Task description : mengelola akun, mengelola ruangan, mengelola dokter

(mengelola jadwal dokter, mengelola tarif dokter, mengelola cuti dokter), mengelola perawat, mengelola tindakan, mengelola tindakan, dan mengelola obat

2. User Group 2

<i>Description of User</i>	:	Dokter
<i>Role</i>	:	Dokter dapat membuat rekam medis pasien yang datang ke rumah sakit
<i>Prerequisite</i>	:	Dokter harus <i>login</i> terlebih dahulu untuk dapat mengakses beberapa fitur yang tersedia
<i>Task description</i>	:	Melihat antrian pasien dan membuat rekam medis untuk pasien, melihat hasil pemeriksaan

3. User Group 3

<i>Description of User</i>	:	Perawat
<i>Role</i>	:	Perawat mengelola data pasien dan dokter
<i>Prerequisite</i>	:	Perawat harus <i>login</i> terlebih dahulu untuk dapat mengakses beberapa fitur yang tersedia
<i>Task description</i>	:	Mengelola data penanggung jawab, melihat data pasien, dan melihat data dokter

4. User Group 4

<i>Description of User</i>	:	Kasir
<i>Role</i>	:	Kasir menampilkan total pembayaran yang harus dibayarkan pasien dan menerima uang tersebut
<i>Prerequisite</i>	:	Kasir harus <i>login</i> terlebih dahulu untuk dapat mengakses beberapa fitur yang tersedia
<i>Task description</i>	:	Melihat daftar pasien yang belum melakukan pembayaran, melihat jumlah pembayaran yang harus dibayarkan pasien (mengelola pembayaran)

5. User Group 5

<i>Description of User</i>	:	Pegawai
<i>Role</i>	:	Pegawai melayani pasien
<i>Prerequisite</i>	:	Pegawai Loket harus <i>login</i> terlebih dahulu untuk dapat mengakses beberapa fitur yang tersedia
<i>Task description</i>	:	Mengelola data pasien, mengelola data rawat jalan,

mengelola data rawat inap

6. User Group 6

Description of User : Direktur

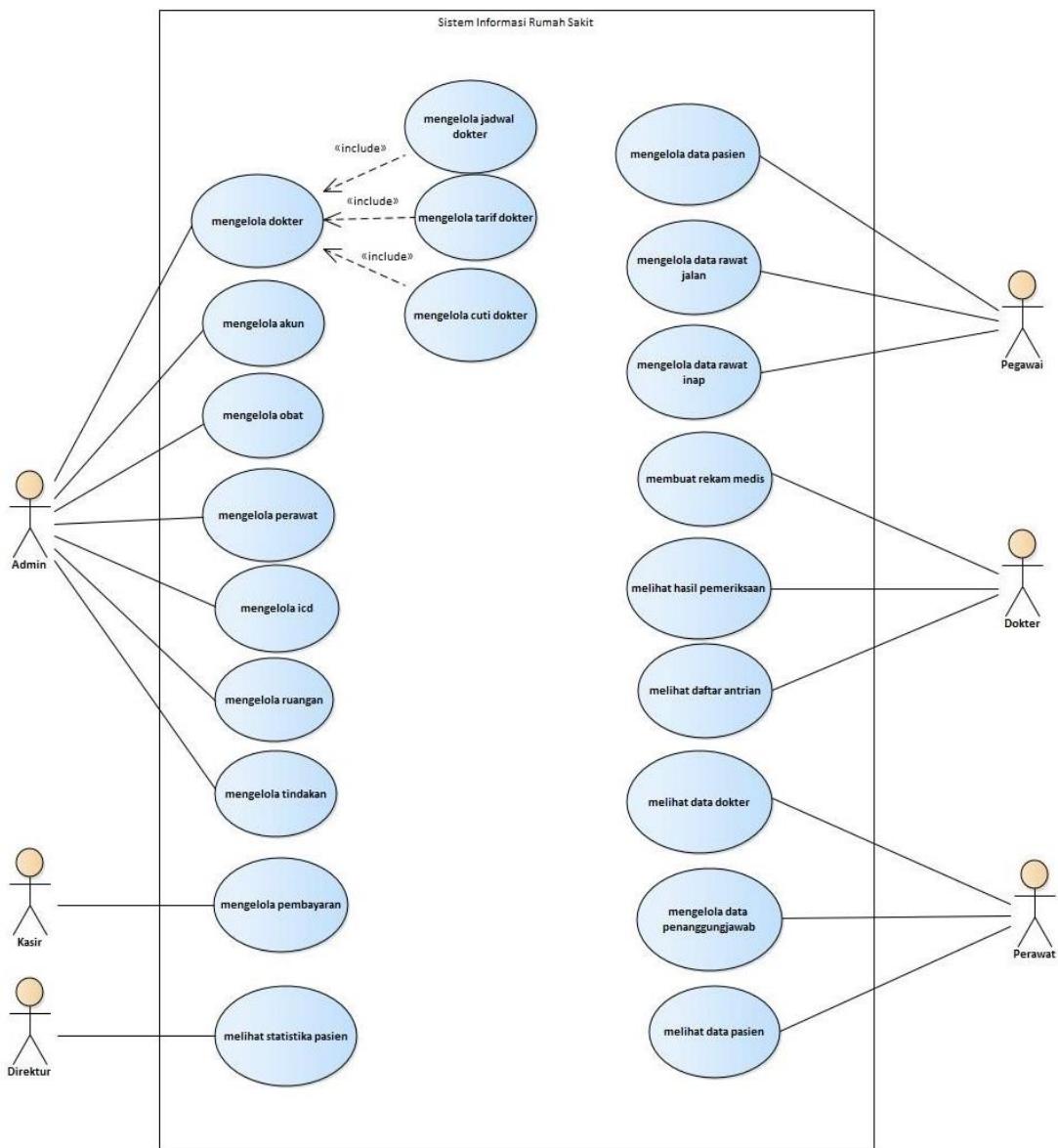
Role : Direktur melihat statistika pasien

Prerequisite : Direktur harus *login* terlebih dahulu untuk dapat masuk ke dalam sistem

Task description : Melihat statistika pasien

3.6.2 Use Case Diagram

Use case diagram menggambarkan kemampuan atau hal apa saja yang dapat dilakukan oleh aktor terhadap suatu sistem. Berikut akan dijelaskan *use case diagram* dari sistem yang akan dibangun.



Gambar 17. Use Case Diagram

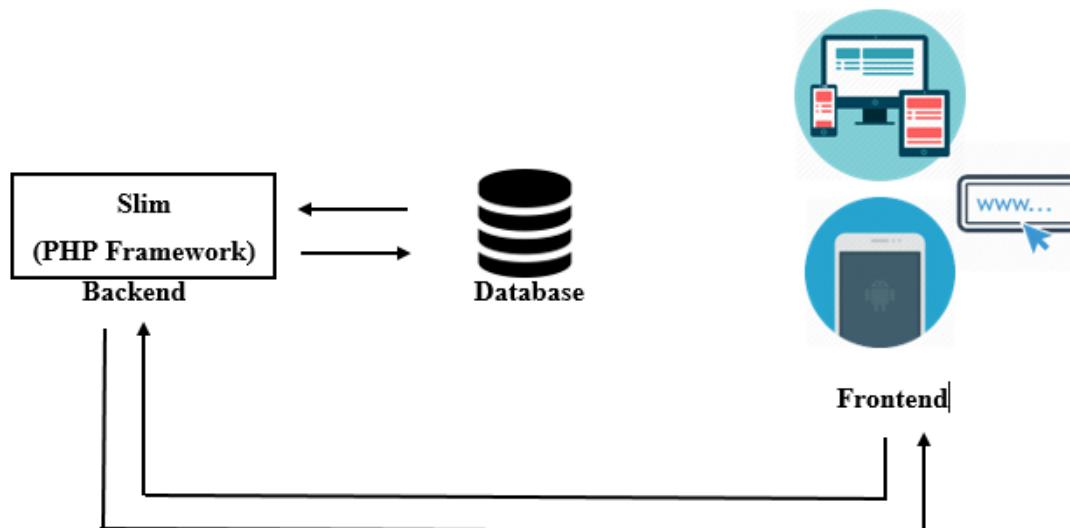
Berdasarkan Gambar 17 *use case* yang dapat dilakukan pada *desktop* adalah mengelola data pasien, mengecek data pasien, mengelola data rawat inap, mengelola data rawat jalan, melihat rekam medis, melihat ruangan, mengelola dokter, mengelola perawat, mengelola obat, mengelola icd, mengelola ruangan, mengelola data tindakan, mengelola data tindakan, mengelola data penanggung jawab, mengelola akun, mengelola pembayaran, melihat list pasien. Pada *mobile use case* yang dapat dilakukan adalah membuat rekam medis, melihat rekam medis, melihat data pasien, melihat data dokter. Sedangkan pada *web use case* yang dapat dilakukan adalah melihat statistika pasien.

BAB 4 DESAIN

Pada bab ini dijelaskan mengenai rancangan diagram integrase, deskripsi aplikasi, desain aplikasi yang meliputi *use case diagram*, *domain model*, *conceptual data*, *physical data*, *class diagram* dan *sequence diagram* yang meliputi kebutuhan aplikasi dan batasan implementasi aplikasi sistem informasi rumah sakit.

4.1 Rancangan Diagram Integrasi

Berdasarkan analisis yang telah dilakukan sebelumnya, maka pada Tugas Akhir ini dirancang diagram integrasi aplikasi yang menjelaskan bagaimana proses aplikasi yang terjadi pada sistem informasi rumah sakit yang akan dirancang. Berikut adalah gambar rancangan diagram integrasi aplikasi yang akan dibangun pada Tugas Akhir ini.



Gambar 18. Rancangan Diagram Integrasi

Keterangan:

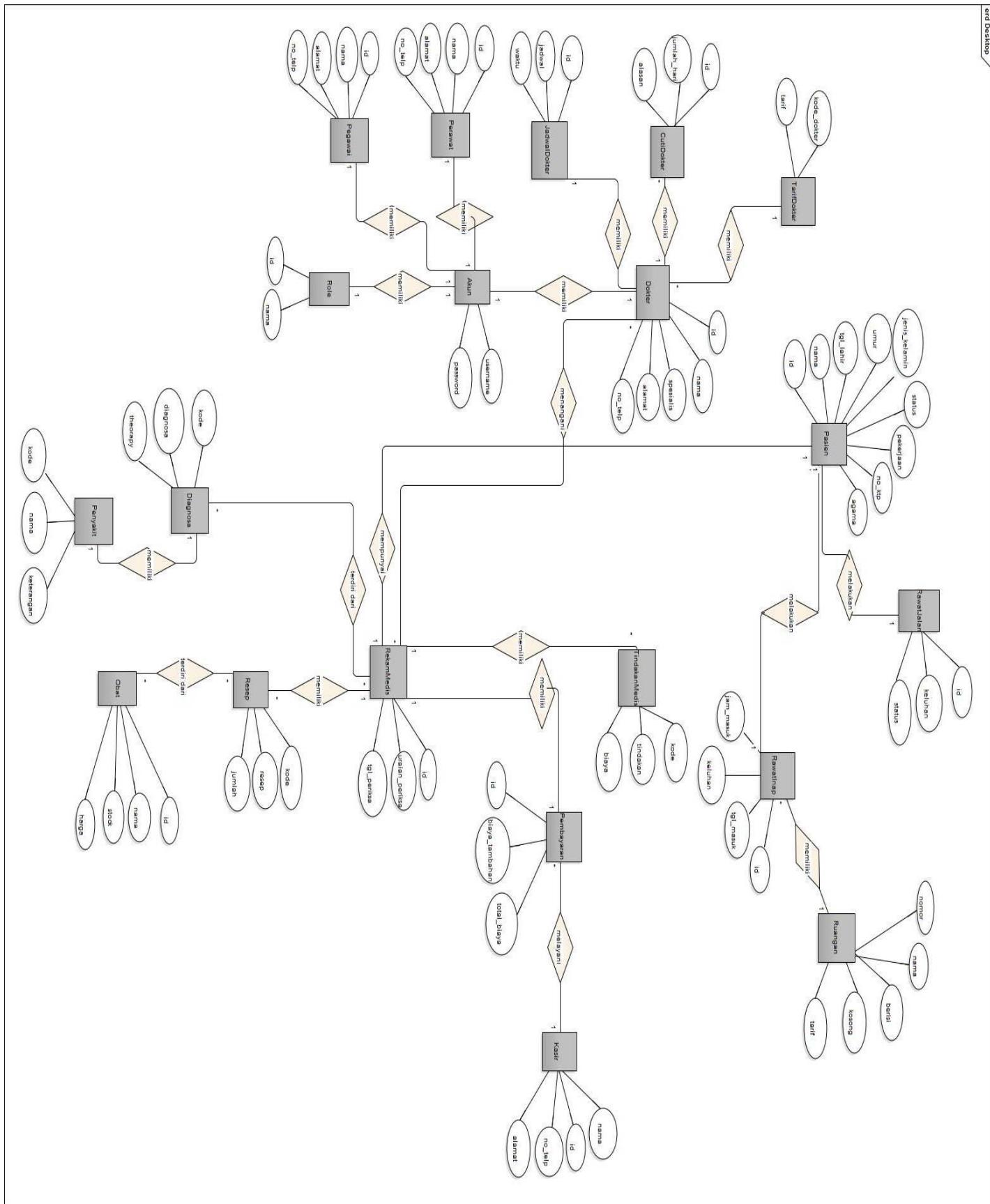
Sisi *frontend* memperoleh data dari database melalui perantaraan *backend* yang adalah PHP. Kemudian database mengembalikan data melalui *backend* untuk ditampilkan di sisi *frontend*.

4.2 Data Requirement

Pada subbab ini dijelaskan kebutuhan data selama pembangunan Sistem Informasi Rumah Sakit dalam bentuk E-R Diagram.

E-R Diagram

E-R diagram yang digunakan dalam pembangunan Sistem Informasi Rumah Sakit dapat dilihat sebagai berikut.



Gambar 19. E-R Diagram Desktop

Keterangan:

Tabel berikut menjelaskan *entity* yang ada pada E-R diagram yang digunakan pada Sistem Informasi Rumah Sakit.

Tabel 5. E-R Diagram Desktop

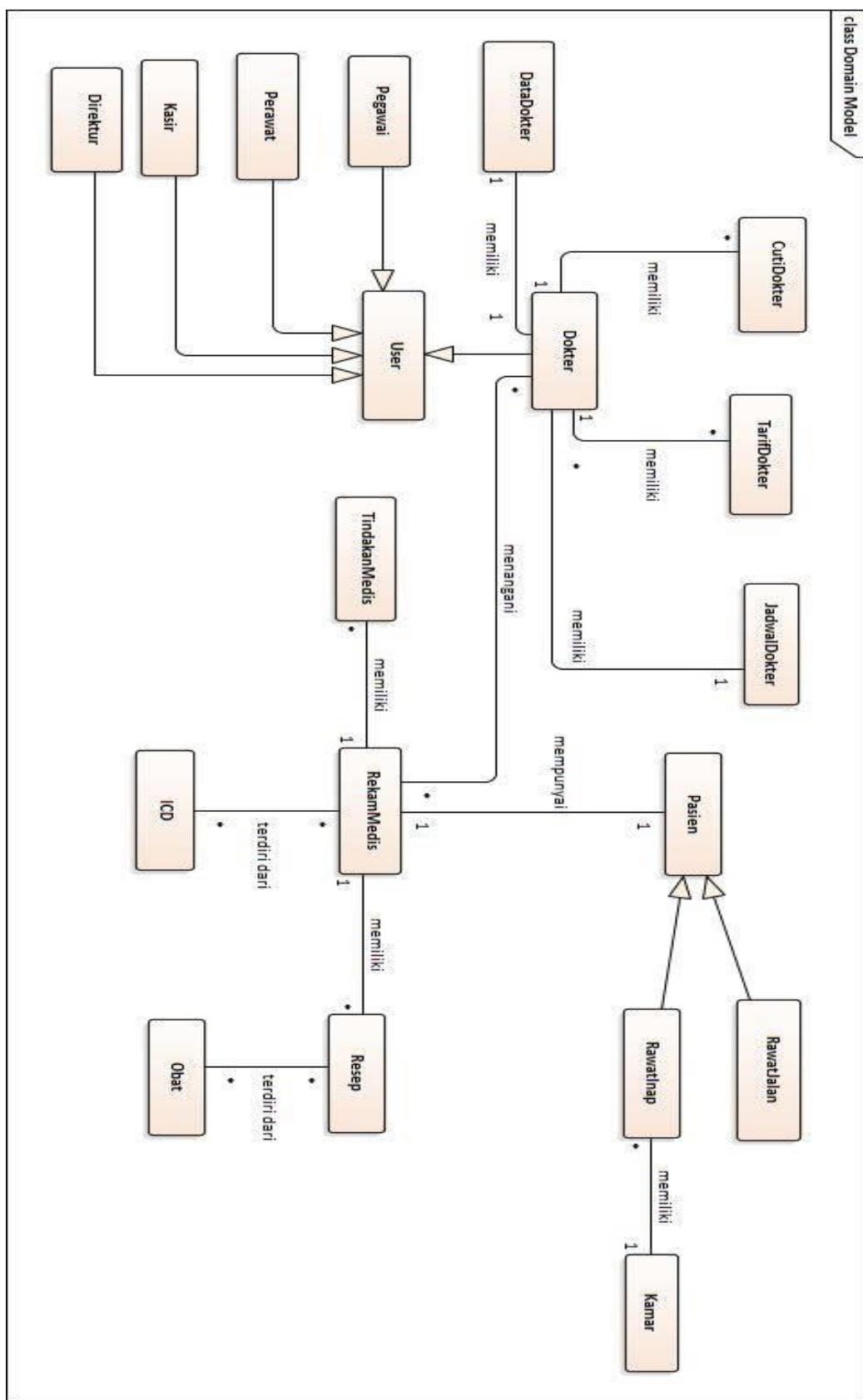
No	Nama Tabel	Keterangan
1.	jadwal_cuti_dokter	Pada tabel ini akan menyimpan data untuk cuti dokter seperti jumlah hari cuti dan alasan melakukan cuti.
2.	biaya_dokter	Data yang disimpan pada tabel ini berupa tarif dari setiap dokter.
3.	jadwal_praktek_dokter	Pada tabel ini berisi mengenai jadwal dari dokter yang terdiri dari data jadwal dan waktu.
4.	data_dokter	Info mengenai dokter akan disimpan pada tabel ini misalnya nama, spesialis, dan info lainnya yang bersangkutan dengan dokter.
5.	rawat_jalan	Pada tabel ini berisi mengenai info dari rawat jalan yang dilakukan pasien.
6.	rawat_inap	Pada tabel ini berisi mengenai info dari rawat inap yang dilakukan pasien.
7.	pasien	Info mengenai pasien akan disimpan pada tabel ini misalnya nama, tanggal lahir, alamat, dan info lainnya yang berhubungan dengan pasien.
8.	ruangan	Data yang disimpan pada tabel ini berupa info ruangan yang ada misalnya nama ruangan, jumlah yang berisi, jumlah yang kosong, dan lain sebagainya.
9.	pegawai	Info mengenai pegawai akan disimpan pada tabel ini misalnya nama pegawai, alamat, dan nomor telepon.
10.	login	Menyimpan data <i>username</i> , <i>password</i> dan <i>typeuser</i> yang akan masuk ke dalam sistem.
11.	perawat	Info mengenai perawat akan disimpan pada tabel ini misalnya nama perawat, alamat, dan nomor telepon.
12.	icd	Tabel ini menyimpan data mengenai semua diagnosa yang diberikan dokter.
13.	tindakan_medis	Semua tindakan medis yang dapat dilakukan pada pasien disimpan pada tabel ini.
14.	rekam_medis	Rekam medis setiap pasien disimpan pada tabel ini.
15.	data_obat	Pada tabel ini berisi mengenai data obat yang tersedia.
16.	penanggung_jawab	Pada tabel ini berisi mengenai data penanggung jawab pasien yang dirawat.
17.	statistika_pasien	Pada tabel ini berisi mengenai statistika pasien yang melakukan perawatan.

4.3 Desain Aplikasi Sistem Informasi Rumah Sakit

pada bab ini dijelaskan diagram yang digunakan sebagai desain dari aplikasi yang dibangun yaitu *domain model*, CDM, PDM, *class diagram*, dan *sequence diagram*.

4.3.1 Domain Model

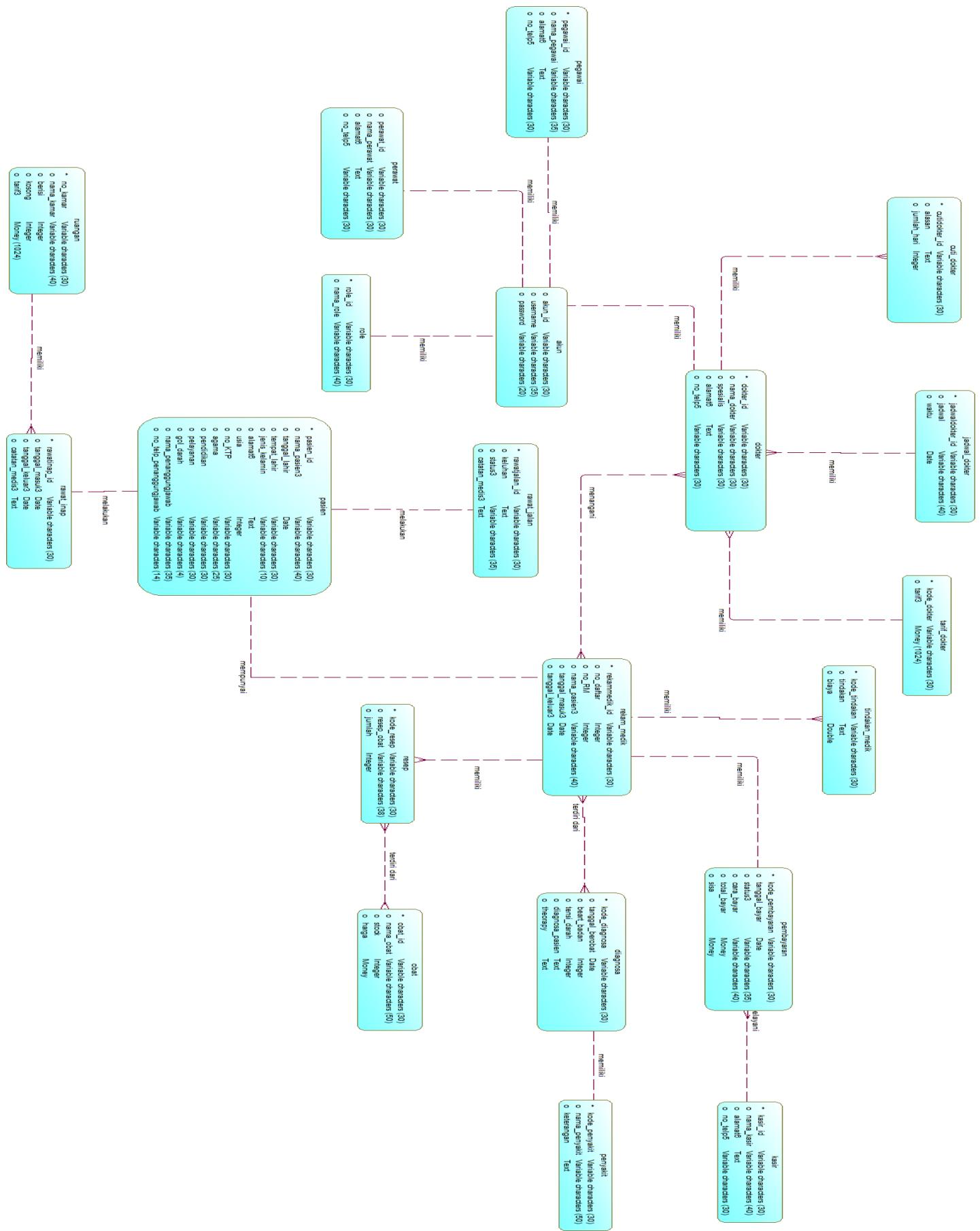
Domain model dari aplikasi ini dapat dilihat pada gambar berikut.



Gambar 20. Domain Model

4.3.2 Conceptual Data Model (CDM)

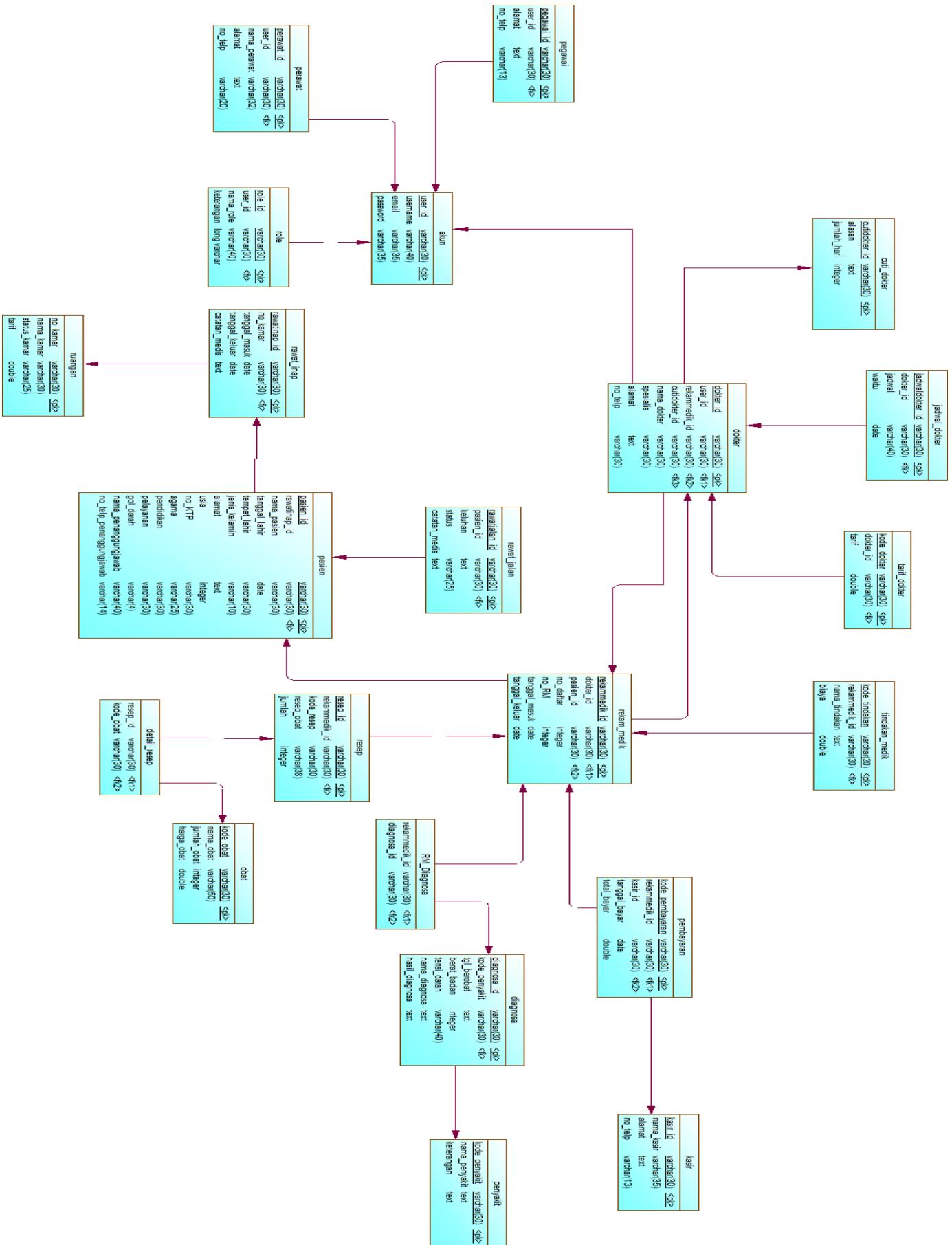
Pada bagian ini digambarkan CDM yang digunakan dalam pembangunan sistem yaitu dapat dilihat sebagai berikut.



Gambar 21. Conceptual Data Model (CDM)

4.3.3 Physical Data Model (PDM)

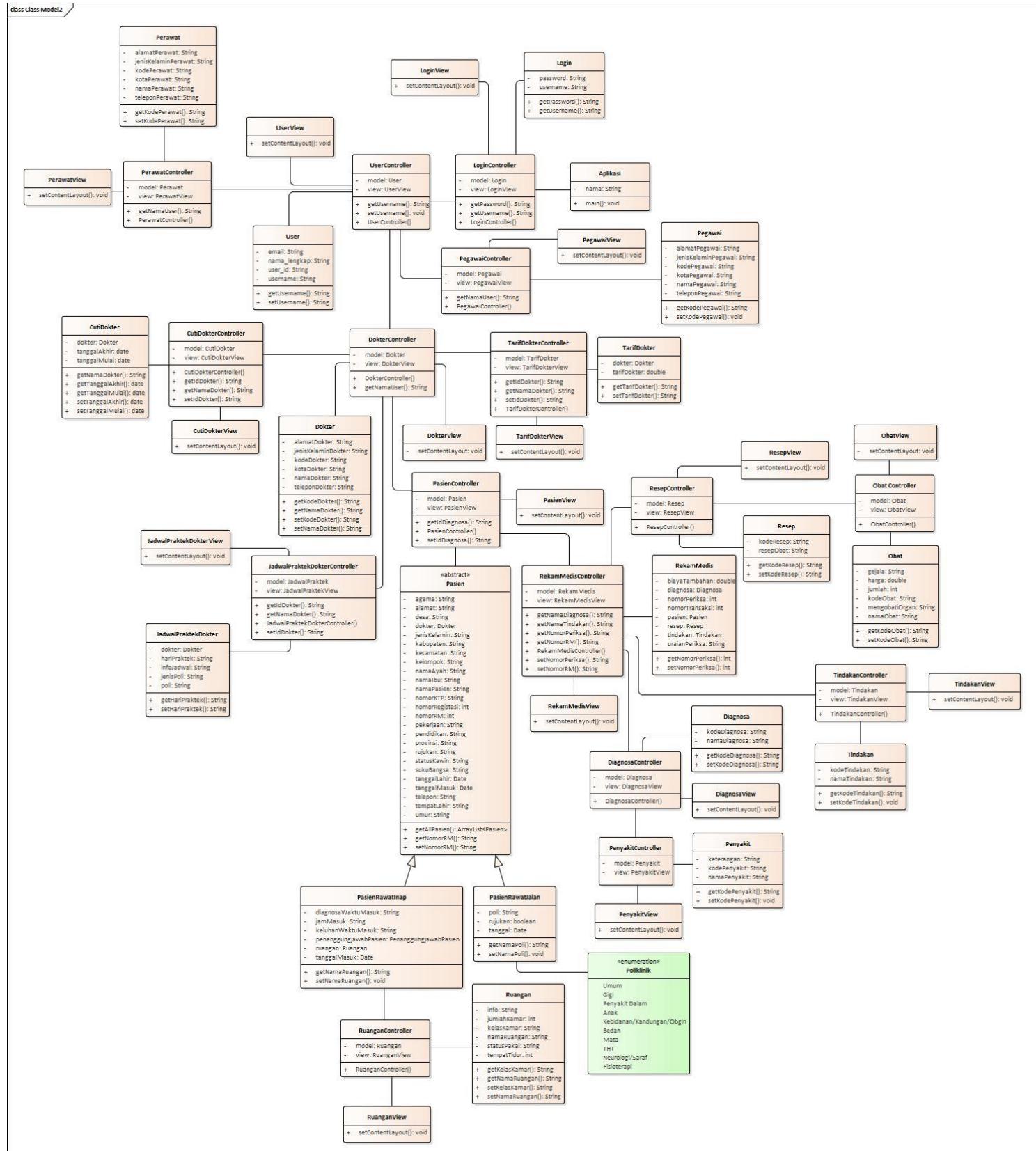
Pada bagian ini digambarkan PDM yang digunakan dalam pembangunan sistem yaitu dapat dilihat sebagai berikut.



Gambar 22. Physical Data Model (PDM)

4.3.4 Class Diagram

Class diagram dari sistem yang akan dibangun dapat dilihat pada gambar berikut.



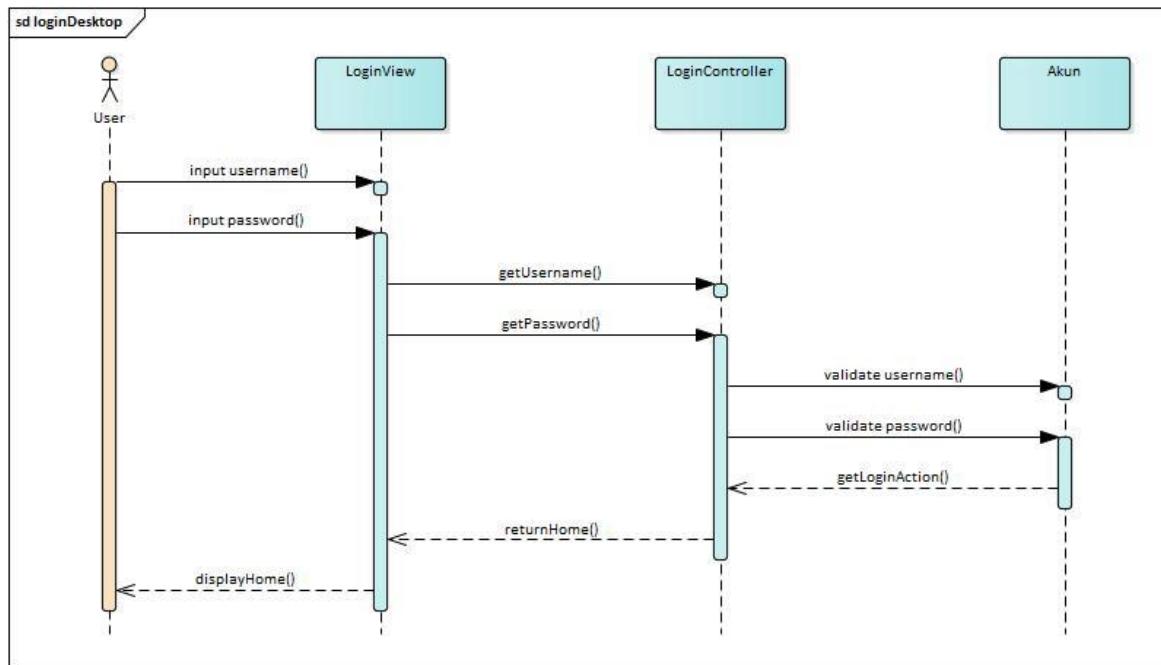
Gambar 23. Class Diagram

4.3.5 Sequence Diagram

Sequence diagram dari Sistem Informasi Rumah Sakit dapat dilihat sebagai berikut.

4.3.5.1 Login

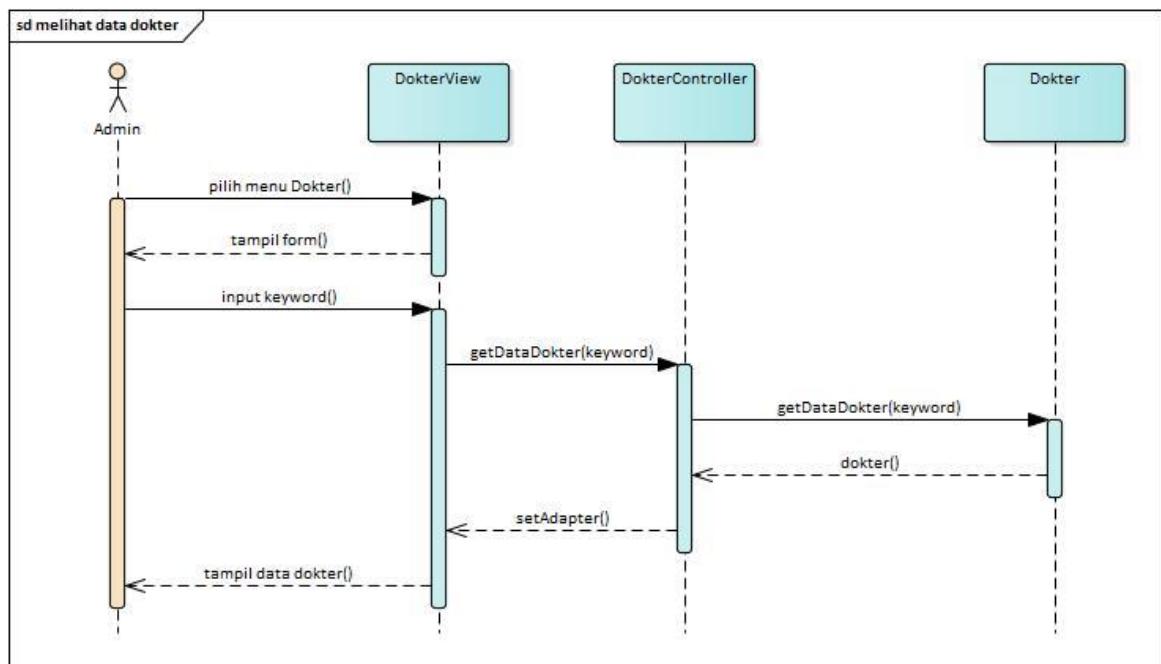
Sequence diagram pada *login* akan dijelaskan pada gambar berikut.



Gambar 24. Sequence Diagram Login

4.3.5.2 Melihat Data Dokter

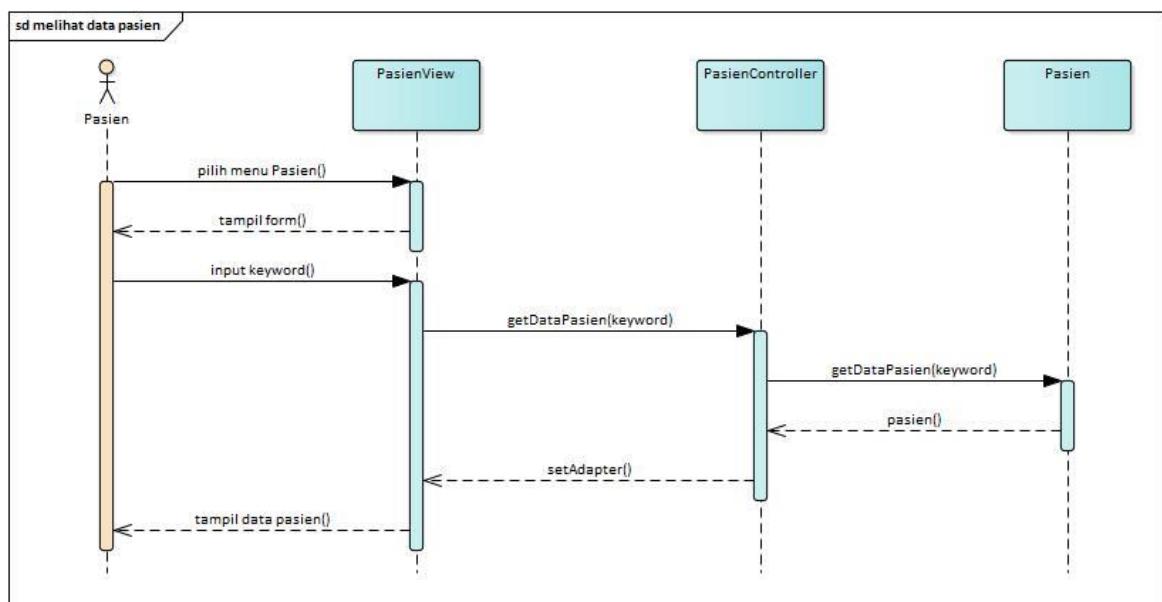
Sequence diagram melihat data dokter akan dijelaskan pada gambar berikut.



Gambar 25. Sequence Diagram Melihat Data Dokter

4.3.5.3 Melihat Data Pasien

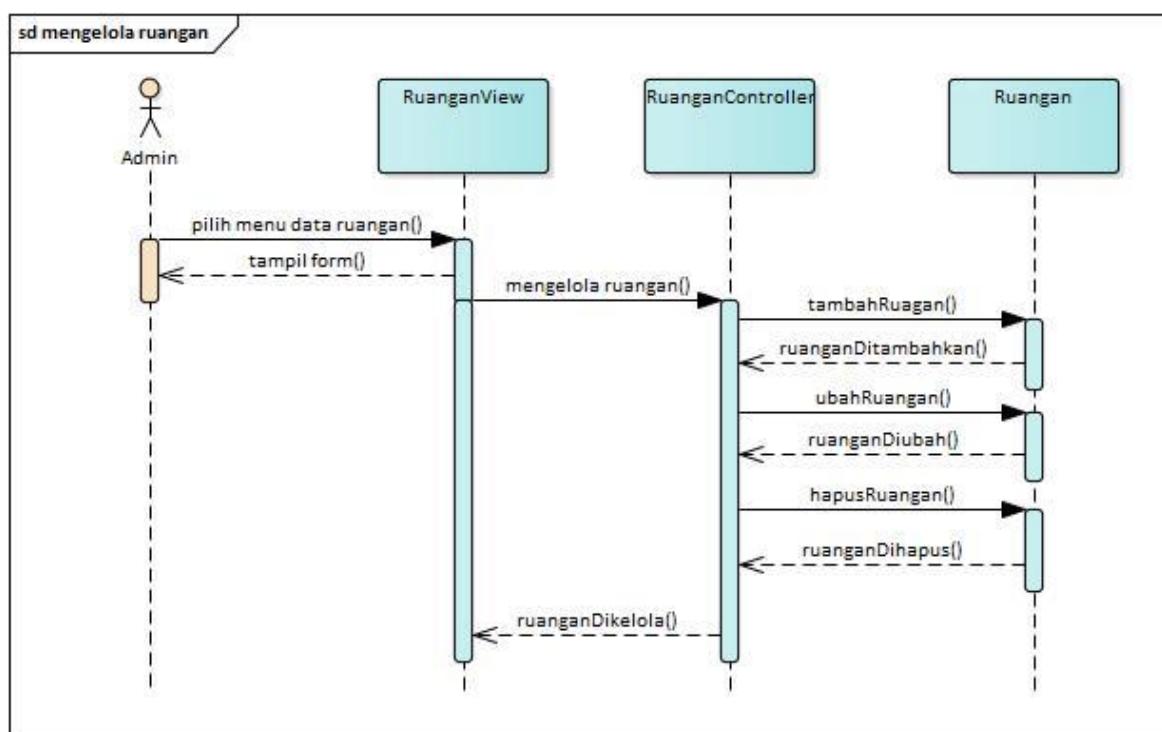
Sequence diagram melihat data pasien akan dijelaskan pada gambar berikut.



Gambar 26. Sequence Diagram Melihat Data Pasien

4.3.5.4 Mengelola Ruangan

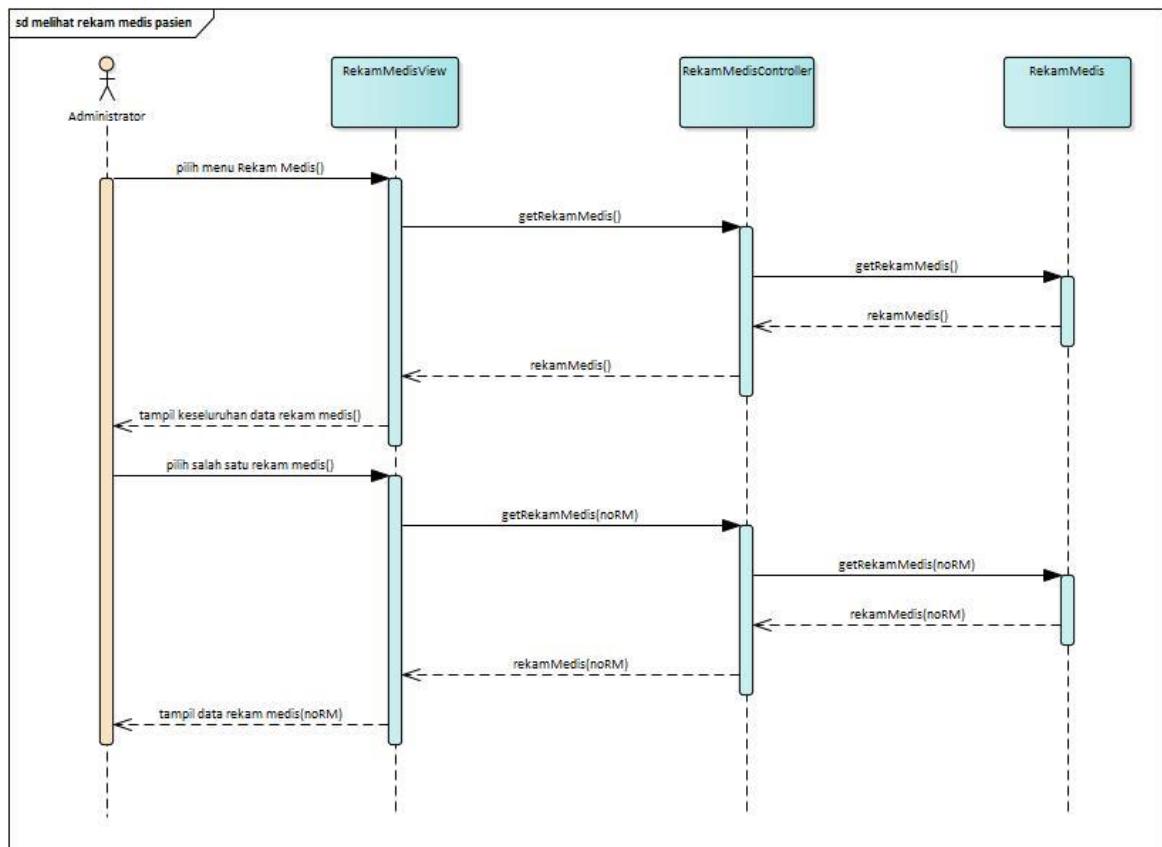
Sequence diagram mengelola ruangan akan dijelaskan pada gambar berikut.



Gambar 27. Sequence Diagram Mengelola Ruangan

4.3.5.5 Melihat Rekam Medis Pasien

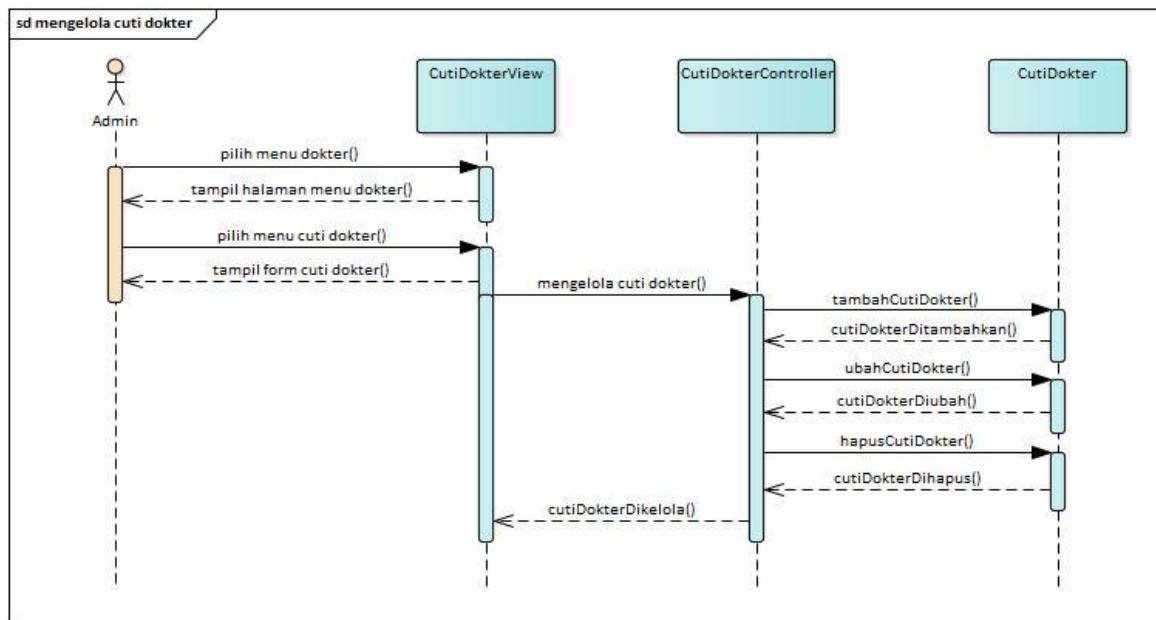
Sequence diagram melihat rekam medis pasien akan dijelaskan pada gambar berikut.



Gambar 28. Sequence Diagram Melihat Rekam Medis Pasien

4.3.5.6 Membuat Data Cuti Dokter

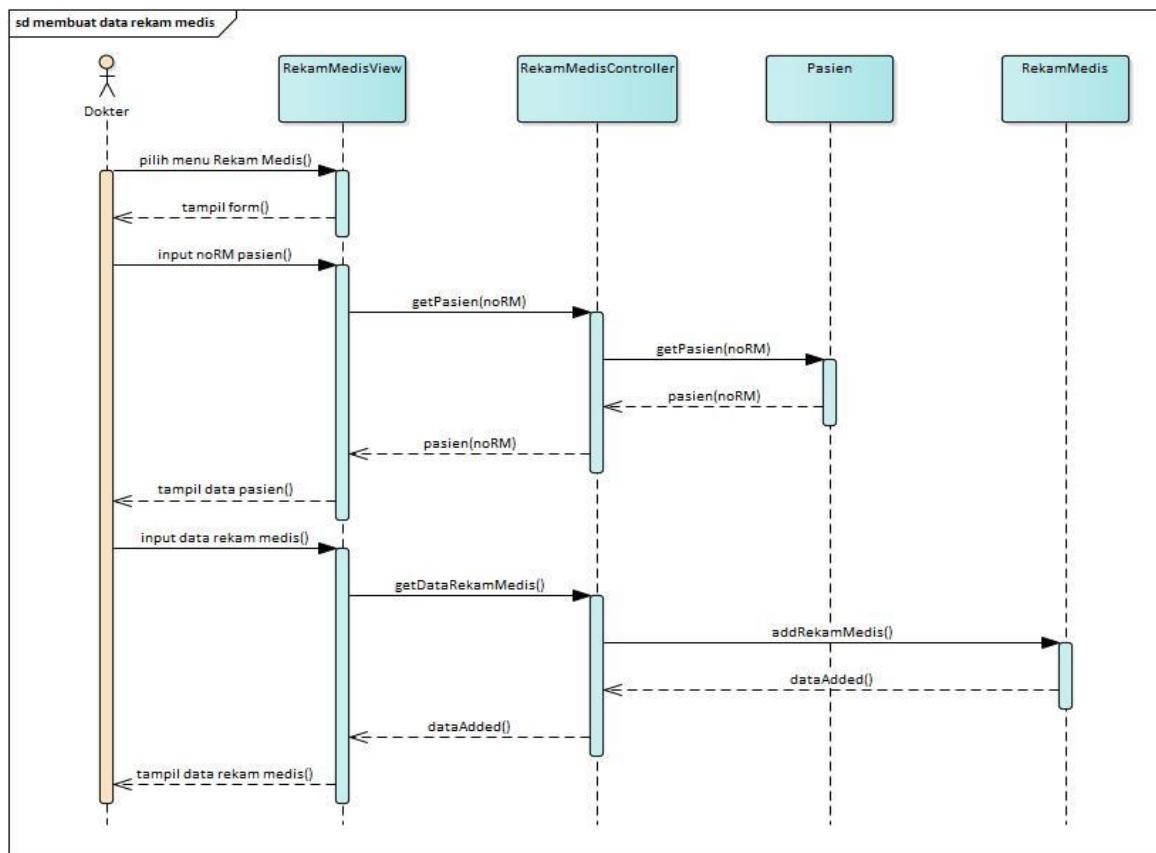
Sequence diagram membuat data cuti dokter akan dijelaskan pada gambar berikut.



Gambar 29. Sequence Diagram Membuat Data Cuti Dokter

4.3.5.7 Membuat Data Rekam Medis

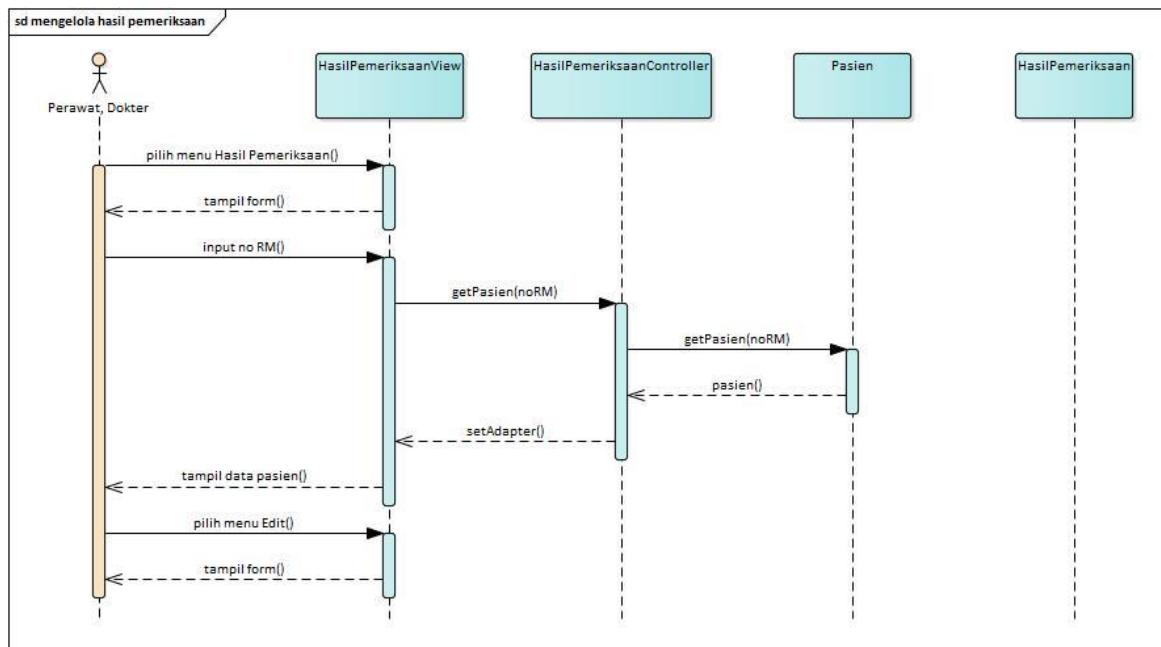
Sequence diagram membuat data rekam medis akan dijelaskan pada gambar berikut.



Gambar 30. Sequence Diagram Membuat Rekam Medis Pasien

4.3.5.8 Mengelola Hasil Pemeriksaan

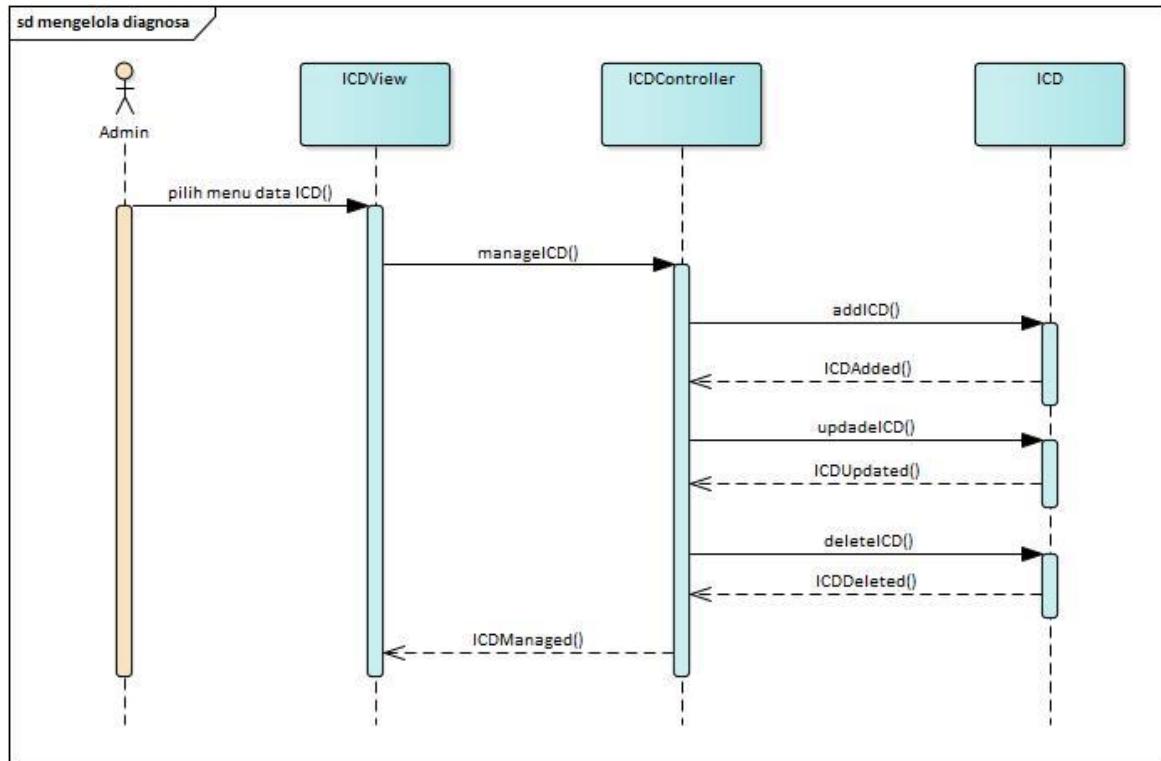
Sequence diagram mengelola hasil pemeriksaan akan dijelaskan pada gambar berikut.



Gambar 31. Sequence Diagram Mengelola Hasil Pemeriksaan

4.3.5.9 Mengelola ICD

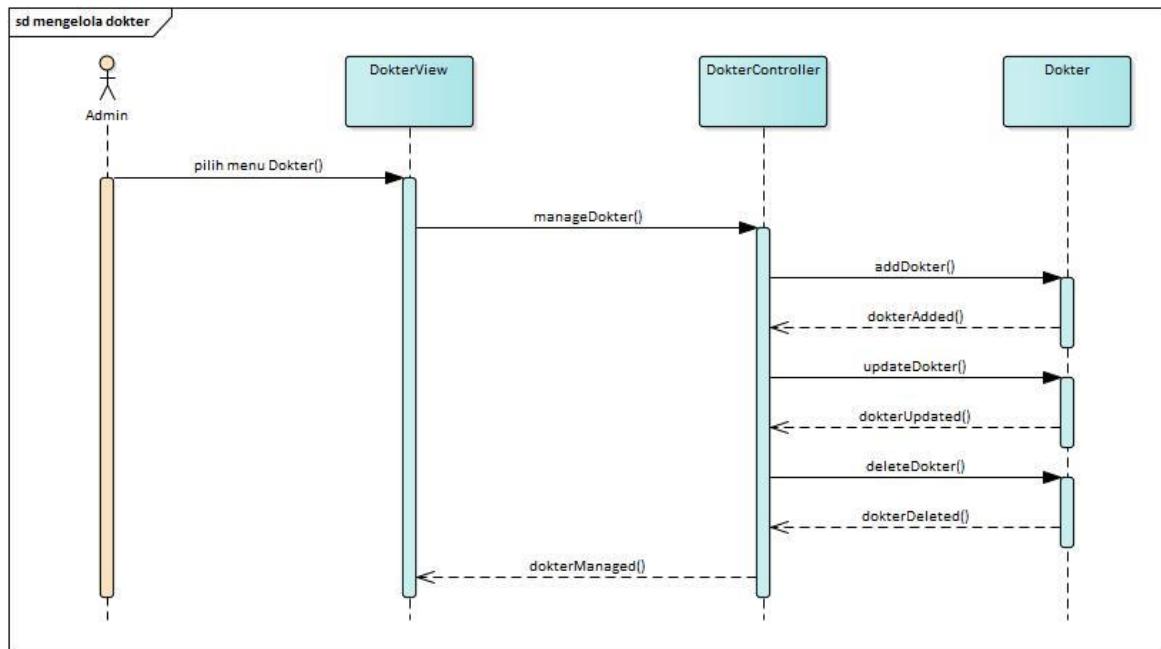
Sequence diagram mengelola ICD akan dijelaskan pada gambar berikut.



Gambar 32. Sequence Diagram Mengelola ICD

4.3.5.10 Mengelola Dokter

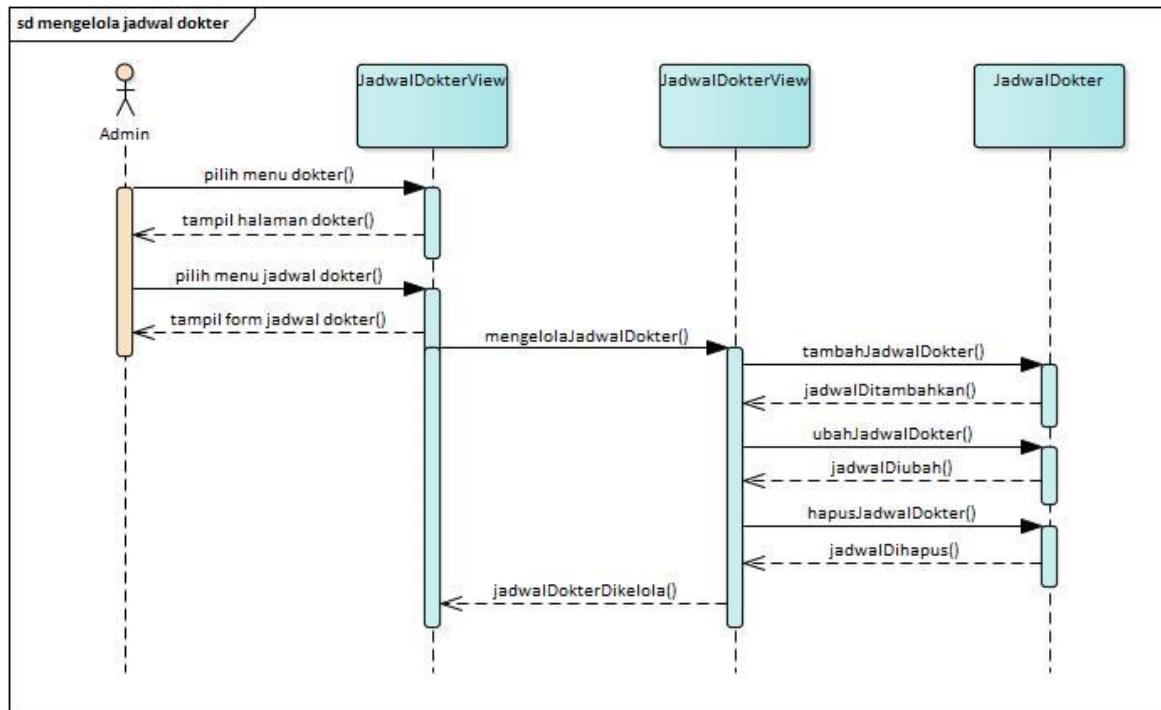
Sequence diagram mengelola dokter akan dijelaskan pada gambar berikut.



Gambar 33. Sequence Diagram Mengelola Dokter

4.3.5.11 Mengelola Jadwal Praktek Dokter

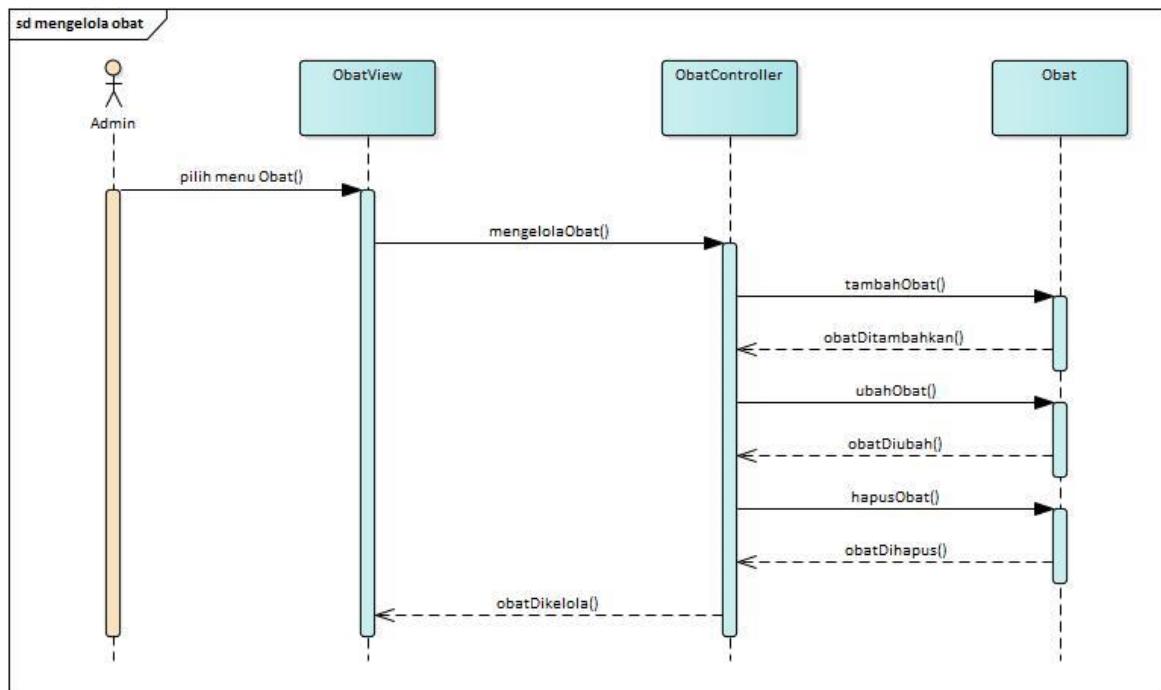
Sequence diagram mengelola jadwal praktek dokter akan dijelaskan pada gambar berikut.



Gambar 34. Sequence Diagram Mengelola Jadwal Praktek Dokter

4.3.5.12 Mengelola Obat

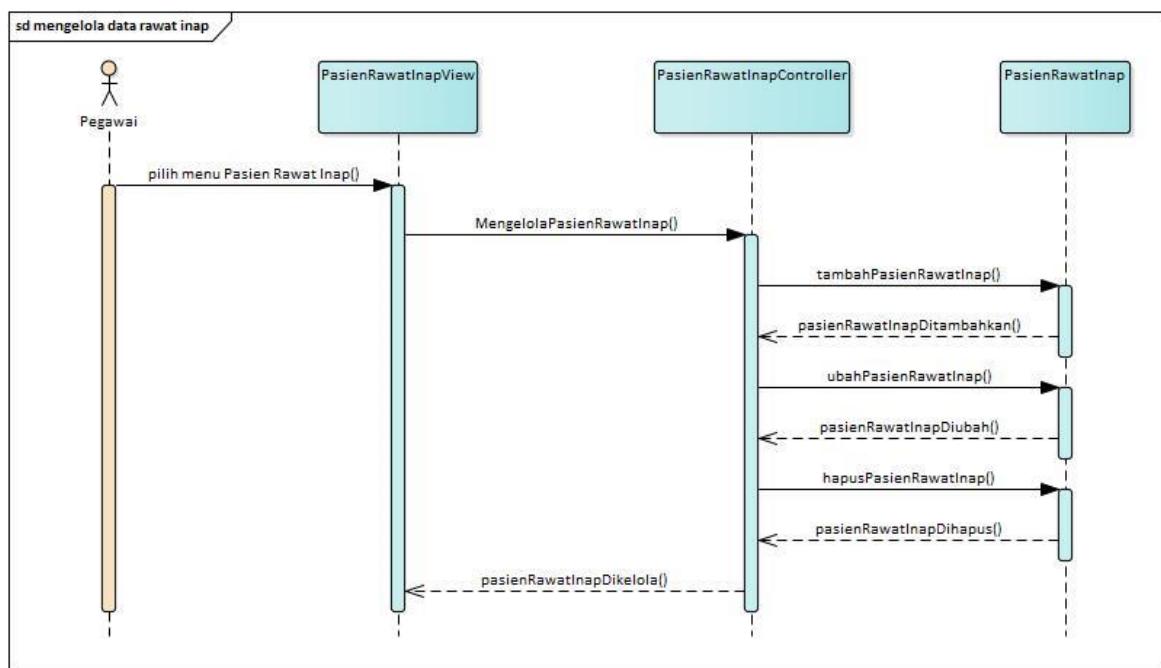
Sequence diagram mengelola obat akan dijelaskan pada gambar berikut.



Gambar 35. Sequence Diagram Mengelola Obat

4.3.5.13 Mengelola Pasien Rawat Inap

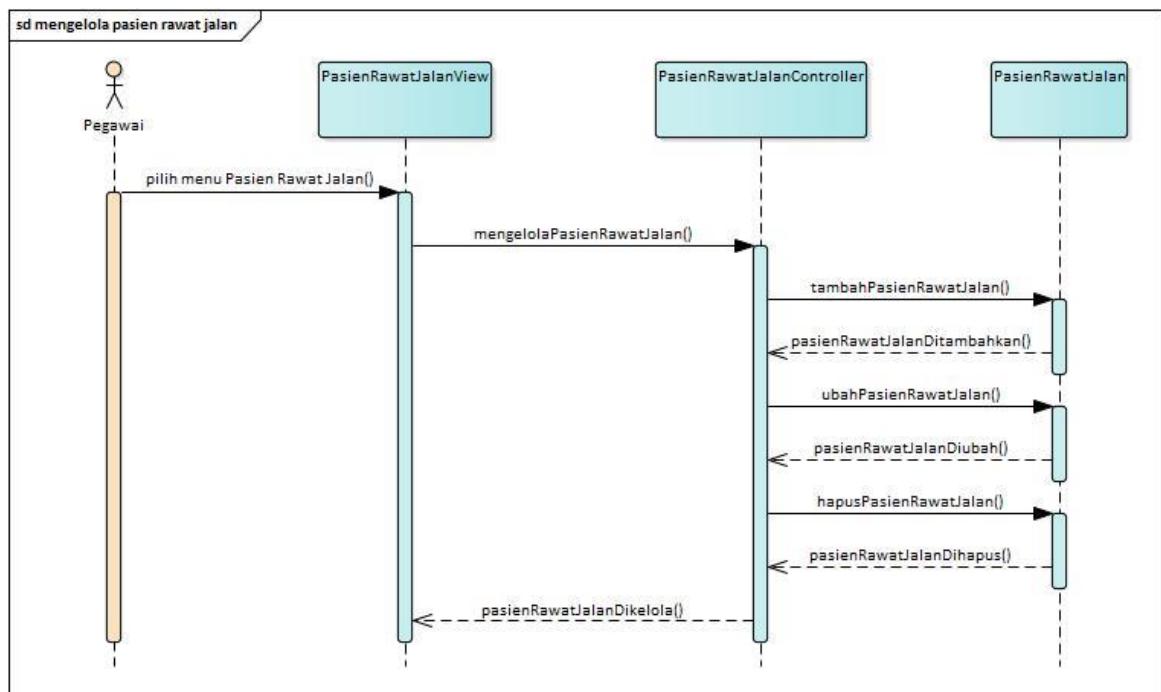
Sequence diagram mengelola pasien rawat inap akan dijelaskan pada gambar berikut.



Gambar 36. Sequence Diagram Mengelola Pasien Rawat Inap

4.3.5.14 Mengelola Pasien Rawat Jalan

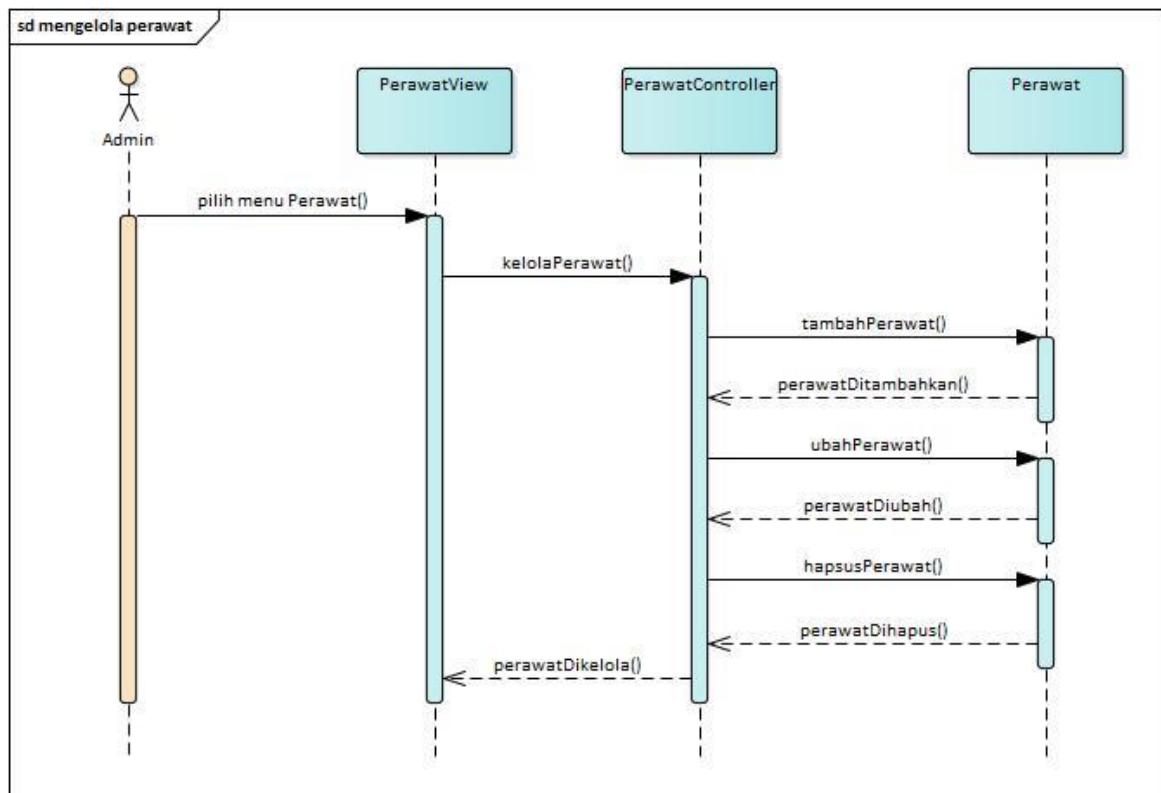
Sequence diagram mengelola pasien rawat jalan akan dijelaskan pada gambar berikut.



Gambar 37. Sequence Diagram Mengelola Pasien Rawat Jalan

4.3.5.15 Mengelola Perawat

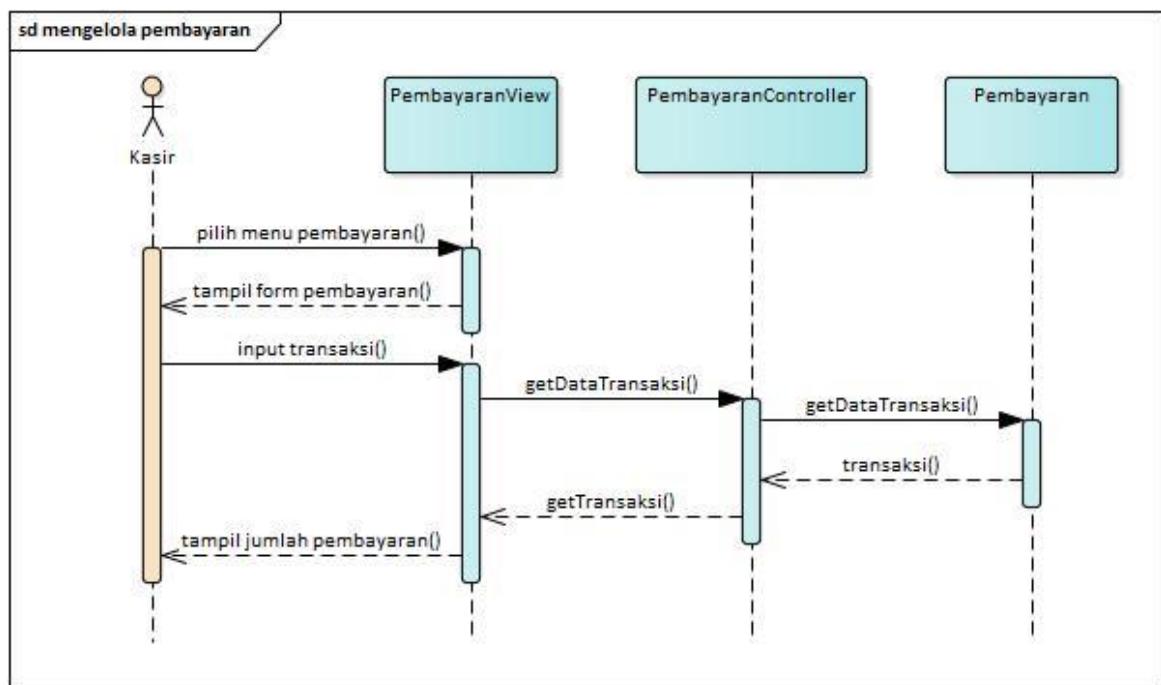
Sequence diagram mengelola perawat akan dijelaskan pada gambar berikut.



Gambar 38. Sequence Diagram Mengelola Perawat

4.3.5.16 Mengelola Pembayaran

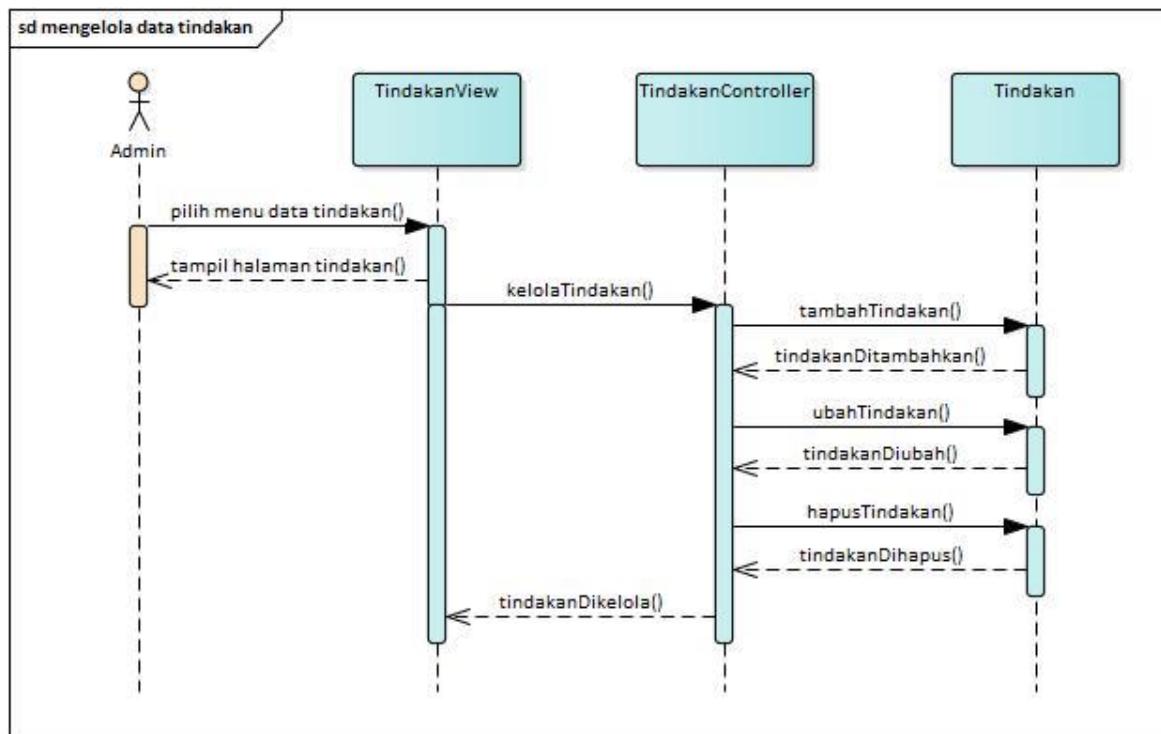
Sequence diagram mengelola pembayaran akan dijelaskan pada gambar berikut.



Gambar 39. Sequence Diagram Mengelola Pembayaran

4.3.5.17 Mengelola Data Tindakan

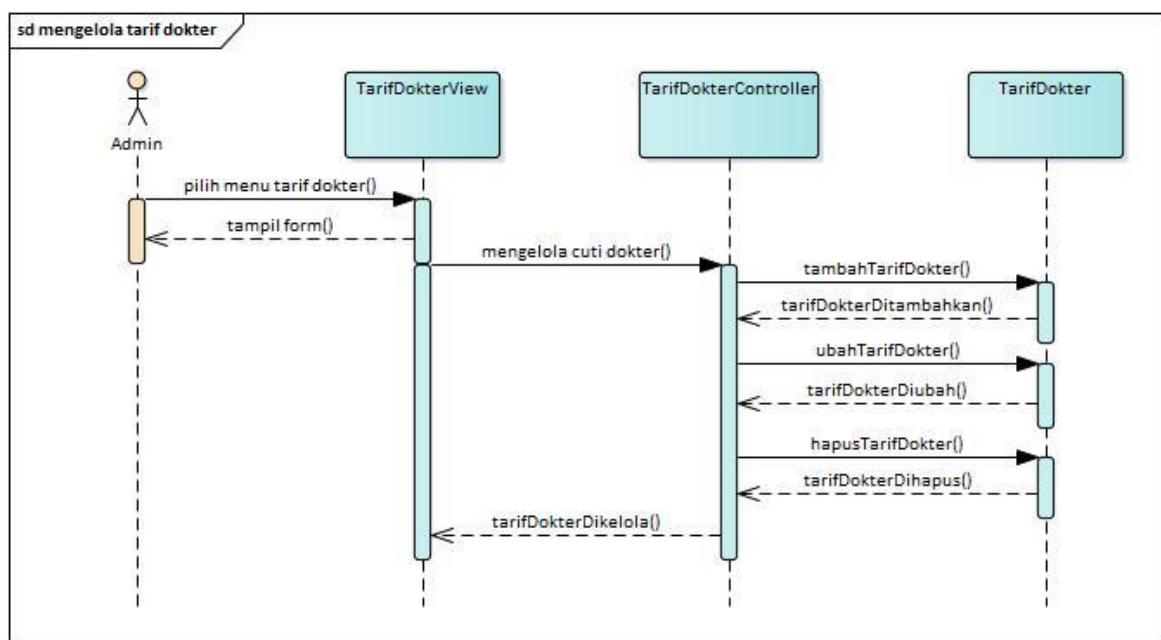
Sequence diagram mengelola data tindakan akan dijelaskan pada gambar berikut.



Gambar 40. Sequence Diagram Mengelola Tindakan

4.3.5.18 Mengelola Tarif Dokter

Sequence diagram mengelola tarif dokter akan dijelaskan pada gambar berikut.



Gambar 41. Sequence Diagram Mengelola Tarif Dokter

4.4 Desain Antarmuka Aplikasi

Pada sub bab ini menjelaskan bagaimana desain antarmuka aplikasi yang akan dibangun pada tiga versi aplikasi, yaitu versi *desktop*, *mobile* dan *web*. Adapun tampilan pada masing-masing versi dapat dilihat pada Lampiran 3.

4.5 Functional Requirement

Functional requirement akan dijelaskan pada Tabel 6 berikut.

Tabel 6. Functional Requirement

No	Main Function	Proses	Keterangan
1	Fungsi autentikasi	Login	Fungsi autentikasi digunakan oleh <i>user</i> untuk masuk ke dalam aplikasi.
2	Fungsi memasukkan data pasien	Memasukkan Data Pasien	Fungsi ini digunakan untuk memasukkan data pasien yang baru pertama sekali melakukan pengobatan di rumah sakit.
3	Fungsi mengecek data pasien	Mengecek Data Pasien	Fungsi ini digunakan untuk mengecek data pasien yang sudah pernah terdaftar sebagai pasien di rumah sakit.
4	Fungsi mengelola pasien poli	Mengelola Pasien Poli	Fungsi ini digunakan untuk mengelola (menambah, mengedit, dan menghapus) data pasien yang merupakan pasien rawat jalan.
5	Fungsi mengelola diagnosa	Mengelola Diagnosa	Fungsi ini digunakan untuk mengelola data diagnosa yang akan dimasukkan ke data pasien.
6	Fungsi mengelola jadwal dokter	Mengelola Jadwal Dokter	Fungsi ini digunakan untuk mengelola jadwal dokter.
7	Fungsi mengelola cuti dokter	Mengelola Cuti Dokter	Fungsi ini digunakan untuk mengelola data cuti dokter.
8	Fungsi mengelola tarif dokter	Mengelola Tarif Dokter	Fungsi ini digunakan untuk mengelola tarif dokter yang akan dimasukkan pada saat pembayaran.
9	Fungsi melihat daftar antrian	Melihat Daftar Antrian	Fungsi ini digunakan untuk melihat daftar antrian pasien yang diperoleh saat pasien mendaftarkan diri untuk berobat di rumah sakit.
10	Fungsi menambahkan hasil pemeriksaan	Memasukkan Hasil Pemeriksaan	Fungsi ini digunakan untuk memasukkan hasil pemeriksaan pasien setelah dokter memeriksa pasien.
11	Fungsi melihat daftar pasien	Melihat Daftar Pasien	Fungsi ini digunakan untuk melihat daftar pasien yang terdaftar sebagai pasien di

			rumah sakit.
12	Fungsi mengelola dokter	Mengelola Dokter	Fungsi ini digunakan untuk mengelola data dokter yang menangani pasien di rumah sakit.
13	Fungsi mengelola perawat	Mengelola Perawat	Fungsi ini digunakan untuk mengelola data perawat yang membantu dokter dalam menangani pasien di rumah sakit.
14	Fungsi mengelola resep	Mengelola Resep	Fungsi ini digunakan untuk mengelola resep yang akan diberikan dokter untuk dikonsumsi oleh pasien.
15	Fungsi mengelola obat	Mengelola Obat	Fungsi ini digunakan untuk mengelola data obat yang akan digunakan untuk pembuatan resep.
16	Fungsi mengelola ruangan	Mengelola Ruangan	Fungsi ini digunakan untuk mengelola data ruangan yang dipakai oleh pasien rawat inap.
17	Fungsi mengelola tindakan medis	Mengelola Tindakan Medis	Fungsi ini digunakan untuk mengelola tindakan medis yang akan diberikan terhadap pasien selama proses perawatan.
18	Fungsi mengelola penyakit	Mengelola Penyakit	Fungsi ini digunakan untuk mengelola penyakit yang akan dimasukkan ke data rekam medis pasien sebagai hasil.
19	Fungsi melihat daftar pasien yang belum bayar	Melihat Daftar Pasien yang Belum Bayar	Fungsi ini digunakan untuk melihat daftar pasien di rumah sakit yang belum melakukan pembayaran di Kasir atau Bendahara Penerima.
20	Fungsi mengelola pasien rawat inap	Mengelola Pasien Rawat Inap	Fungsi ini digunakan untuk mengelola data pasien yang merupakan pasien rawat inap di rumah sakit.

BAB 5 IMPLEMENTASI

Pada bab ini akan dijelaskan deskripsi umum sistem yang meliputi kebutuhan implementasi, batasan implementasi, dan implementasi aplikasi.

5.1 Kebutuhan Implementasi

Pada subbab ini akan dijelaskan mengenai spesifikasi *hardware* dan *software* yang dibutuhkan dalam rancang bangun aplikasi rumah sakit ini. Spesifikasi perangkat keras yang digunakan pada pengimplementasian akan dijelaskan pada tabel berikut ini.

Tabel 7. Spesifikasi Hardware dan Software

No	Hardware	Spesifikasi
1	Processor	Intel ® Core(TM) i5-3230M CPU @2.60GHz (4 CPUs), ~2.6GHz
2	RAM	8GB

No	Software	Spesifikasi
1	<i>Operating System</i>	Windows 10
2	<i>Development Tools</i>	<i>Android Studio, NetBeans IDE 8.2, dan sublime</i>
3	<i>Programming Language</i>	Java, PHP
4	<i>Database Tools</i>	<i>SQLyog</i>
5	<i>Framework</i>	<i>Laravel, Slim</i>
6	<i>Design Tools</i>	<i>Enterprise Architect, Balsamiq Mockups 3, Bizagi</i>

5.2 Batasan Implementasi

Adapun batasan dari pembangun aplikasi Sistem Informasi Rumah Sakit ini adalah sebagai berikut.

1. Rancang bangun sistem informasi rumah sakit ini terdiri dari tiga jenis *user interface*, yaitu versi *desktop*, *mobile*, dan *web*.
2. Aplikasi pada versi *desktop* digunakan oleh admin untuk menambah, mengedit, menghapus akun, ruangan, dokter, perawat, pegawai, diagnose, tindakan, resep, obat dan pegawai menginput data-data pasien dan kasir untuk menampilkan total pembayaran. pada versi *mobile* digunakan oleh dokter untuk membuat rekam medis pasien dan perawat untuk menginput hasil pemeriksaan pasien yang dilakukan oleh dokter. Sedangkan untuk versi *web* digunakan oleh direktur untuk melihat statistika pasien rawat jalan, rawat inap.

5.3 Implementasi Aplikasi

Pada subbab ini dijelaskan mengenai aplikasi yang diimplementasikan dalam penggerjaan Tugas Akhir. Aplikasi sistem informasi rumah sakit yang dibangun merupakan aplikasi yang berbasis *desktop* yang dibangun dengan menggunakan FXML dengan bahasa pemrograman *Java* sebagai proses input data yang akan digunakan pada *mobile*. Pada *mobile* dibangun menggunakan *Android Studio* dengan Bahasa pemrograman *Java*. Pada *web* menggunakan PHP dengan *framework Laravel*. Sedangkan untuk menghubungkan aplikasi sistem informasi ke database menggunakan SQL Server.

BAB 6 HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai bagaimana hasil yang dihasilkan oleh Sistem Informasi Rumah Sakit ini, baik pada versi *desktop*, *mobile*, dan *web*.

6.1 Hasil Aplikasi

Pada sub bab ini menjelaskan bagaimana hasil aplikasi yang telah dibangun pada tiga versi aplikasi, yaitu versi *desktop*, *mobile* dan *web*. Adapun tampilan pada masing-masing versi dapat dilihat pada Lampiran 4.

6.2 Penerapan MDD pada Tugas Akhir

Pada Tugas Akhir pembangunan Sistem Informasi Rumah Sakit menerapkan metode pengembangan MDD, dimana penerapannya dilakukan dengan mengikuti 4 (empat) langkah pengembangan yang dimiliki oleh MDD, yaitu *Inception*, *Elaboration*, *Construction* dan *Transition*. Penjelasan mengenai langkah penerapan MDD telah dijelaskan pada bab 2. Berikut tahapan untuk menerapkan MDD pada sistem informasi yaitu:

1. Tahap *Inception*, pada tahap ini dilakukan kegiatan untuk pencapaian rancangan desain untuk sistem. Pada tahapan ini dilakukan *user requirement* dalam bentuk wawancara kepada narasumber di rumah sakit Doloksanggul. List pertanyaan dapat dilihat pada Lampiran 2.
2. Tahap *Elaboration*, pada tahap ini dilakukan perancangan sistem sesuai dengan rencana yang telah dilakukan pada tahap *inception*. Perancangan yang dilakukan berupa desain *database*, fungsi-fungsi aplikasi, dan desain *requirement* pada aplikasi seperti domain model, class diagram, CDM, PDM, dan lain-lain.
3. Tahap *Construction*, pada tahap ini dilakukan implementasi dari hasil desain ke dalam sebuah *code* program. Pembangunan aplikasi tersebut dilakukan berdasarkan desain yang telah dilakukan pada tahap *elaboration*.
4. Tahap *Transition*, merupakan tahap untuk pengujian sistem. Skenario pengujian yang dilakukan pada sistem dapat dilihat pada Bab 8.2 Skenario Pengujian.

6.3 Pengaruh MDD pada Tugas Akhir

Pada Tugas Akhir ini penggunaan MDD memiliki pengaruh dalam rancang bangun sistem informasi rumah sakit. Adapun pengaruh MDD pada saat pembangunan sistem informasi ini adalah sebagai berikut.

1. MDD lebih cepat

Pada Tugas Akhir ini digunakan MDD yang dapat mengubah desain secara otomatis menjadi sebuah kerangka *code program* yang dapat membantu para *developer* untuk pengembangan yang dilakukan.

2. MDD dapat meningkatkan kualitas

Penggunaan MDD dapat meningkatkan kualitas karena sebuah aplikasi yang dihasilkan dalam sebuah desain model yang baik dapat dengan baik juga diubah ke dalam *code program* seperti yang disebutkan pada point 1.

3. MDD menghasilkan aplikasi yang sensitif terhadap perubahan

Pada proses pengembangan aplikasi dengan menggunakan MDD sering terjadi perubahan baik dari tahap *inception* sampai *construction*. Hal ini dikarenakan jika adanya ketidaksesuaian pada pembangunan aplikasi (implementasi) maka harus dilakukan kembali tahap *inception* dan *elaboration* seperti *requirement gathering* dan desain aplikasi.

BAB 7 PENGUJIAN

Pada bab ini akan dijelaskan mengenai skenario pengujian yang akan dilakukan terhadap aplikasi yang dibangun.

7.1 Tujuan Pengujian

Tujuan dari proses pengujian yang dilakukan pada aplikasi adalah sebagai berikut:

1. Pengujian dilakukan untuk memastikan bahwa aplikasi dapat berjalan dengan baik dan tidak memiliki *error*.
2. Pengujian dilakukan untuk memastikan bahwa aplikasi telah memenuhi kebutuhan *client*.

7.2 Skenario Pengujian

Berikut akan dijelaskan skenario pengujian pada fungsi yang terdapat pada sistem berdasarkan *use case scenario*.

1. Skenario Pengujian Authentication

Pengujian *authentication* dilakukan untuk memastikan bahwa *user* dapat masuk ke dalam sistem. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 8. Pengujian Authentication

Nama Kasus Uji	Authentication pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul					
Tujuan	Menguji apakah <i>user</i> dapat masuk ke dalam sistem					
Deskripsi	Fungsi ini akan menyimpan data <i>user</i>					
Kondisi Awal	<i>User</i> telah terotentikasi dan berada pada tampilan awal Home					
Skenario Uji						
1. <i>User</i> membuka halaman login dan mengisi data yang diperlukan						
2. <i>User</i> masuk ke dalam sistem						
Kriteria Evaluasi Hasil						
<i>User</i> berhasil masuk ke dalam sistem dan data <i>user</i> disimpan di dalam sistem						
Kasus dan Hasil Uji (Data Normal)						
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan			
<i>User</i> memasukkan <i>username</i> dan <i>password</i>	<i>User</i> berhasil masuk ke dalam sistem	Sesuai yang diharapkan	Diterima			
Kasus dan Hasil Uji Coba (Data Tidak Normal)						
<i>User</i> tidak memasukkan <i>username</i> atau <i>password</i> atau keduanya	Tetap di halaman <i>Login</i>	Sesuai yang diharapkan	Diterima			
Catatan						
<i>User</i> berhasil masuk ke dalam sistem sesuai dengan rolenya masing-masing						

2. Skenario Pengujian Mengelola Data Pasien

Pengujian mengelola data pasien dilakukan untuk memastikan bahwa pegawai dapat mengelola data pasien. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 9. Pengujian Mengelola Data Pasien

Nama Kasus Uji	Mengelola data pasien pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul		
Tujuan	Menguji apakah pegawai dapat mengelola data pasien		
Deskripsi	Fungsi ini akan menyimpan dan menampilkan setiap perubahan yang terjadi		
Kondisi Awal	Pegawai telah terotentikasi dan berada pada tampilan awal Home		
Skenario Uji			
1. Pegawai memilih menu mengelola data pasien			
2. Pegawai memilih satu menu : tambah			
3. Pegawai memilih satu menu : edit			
4. Pegawai memilih satu menu : hapus			
5. Pegawai memilih satu menu : batal			
6. Pegawai memilih menu : <i>search</i>			
7. Pegawai memilih menu : cetak exel			
8. Pegawai memilih satu menu : close			
Kriteria Evaluasi Hasil			
Pegawai berhasil menambahkan data pasien pada sistem			
Pegawai berhasil mengubah data pasien pada sistem			
Pegawai berhasil menghapus data pasien pada sistem			
Pegawai berhasil membatalkan mengubah atau menambahkan data pasien pada sistem			
Pegawai berhasil menampilkan data pasien sesuai <i>keyword</i> yang dimasukkan			
Pegawai berhasil mencetak laporan data pasien ke dalam file excel			
Pegawai berhasil menutup atau keluar dari sistem			
Kasus dan Hasil Uji (Data Normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. Pegawai memilih menu tambah	Data pasien akan bertambah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima
2. Pegawai memilih data yang akan diubah dan memilih menu edit	Data pasien akan berubah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima
3. Pegawai memilih data yang akan dihapus dan memilih menu hapus	Data yang dipilih akan terhapus	Sesuai dengan yang diharapkan	Diterima
4. Pegawai memilih menu batal	Data yang dimasukkan akan terhapus dari <i>form</i>	Sesuai dengan yang diharapkan	Diterima
5. Pegawai memilih menu close	Pegawai akan keluar dari sistem	Sesuai dengan yang diharapkan	Diterima
6. Pegawai memilih menu search berdasarkan no rm dan nama pasien	Data akan ditampilkan sesuai no rm atau nama yang diinput pegawai	Sesuai dengan yang diharapkan	Diterima
7. Pegawai memilih menu cetak exel	Semua data pasien akan dicetak ke dalam file excel dan meminta konfirmasi tempat penyimpanannya	Sesuai dengan yang diharapkan	Diterima

Kasus dan Hasil Uji Coba (Data Tidak Normal)			
Pegawai memilih menu simpan namun tidak mengisi <i>form</i>	Sistem akan menampilkan <i>message error</i> dan data tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima
Pegawai memilih menu hapus namun tidak memilih data yang akan dihapus	Sistem akan menampilkan <i>message error</i> dan data tidak akan dihapus	Sesuai dengan yang diharapkan	Diterima
Pegawai memilih menu edit namun tidak memilih data yang akan diedit	Sistem akan menampilkan <i>message error</i> dan data tidak akan diubah	Sesuai dengan yang diharapkan	Diterima
Pegawai membatalkan untuk mencetak data pasien ke file excel	Sistem akan menampilkan <i>message error</i> dan data tidak akan dicetak	Sesuai dengan yang diharapkan	Diterima
Catatan			
Perawat berhasil mengelola data pasien			

3. Skenario Pengujian Melihat Data Pasien

Pengujian melihat data pasien dilakukan untuk memastikan bahwa perawat dapat melihat data pasien. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 10. Pengujian Melihat Data Pasien

Nama Kasus Uji	Melihat data pasien pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul		
Tujuan	Menguji apakah perawat dapat melihat data pasien		
Deskripsi	Fungsi ini akan menampilkan <i>list</i> pasien		
Kondisi Awal	Perawat telah terotentikasi dan berada pada tampilan awal Home		
Skenario Uji			
1. Perawat memilih menu pasien			
2. Perawat melakukan <i>search</i> pasien			
Kriteria Evaluasi Hasil			
Perawat berhasil melihat daftar pasien			
Perawat berhasil melakukan <i>search</i> pasien			
Perawat berhasil melihat detail data pasien			
Kasus dan Hasil Uji (Data Normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. Perawat memilih menu pasien	Sistem akan menampilkan semua <i>list</i> pasien	Sesuai dengan yang diharapkan	Diterima
2. Perawat melakukan pencarian pasien berdasarkan no rm, nama, atau ruangan	List pasien akan ditampilkan sesuai dengan <i>keyword</i> yang dimasukkan perawat	Sesuai dengan yang diharapkan	Diterima
3. Perawat mengklik salah satu data pasien yang ingin dilihat detailnya	Sistem akan menampilkan detail dari data yang dipilih	Sesuai dengan yang diharapkan	Diterima
Kasus dan Hasil Uji Coba (Data Tidak Normal)			
Perawat memilih menu cari namun tidak memasukkan <i>keyword</i>	Sistem akan menampilkan <i>message error</i> dan data pasien	Sesuai dengan yang diharapkan	Diterima

	tidak akan ditampilkan		
Catatan			
Perawat berhasil melihat <i>list</i> pasien			

4. Skenario Pengujian Mengelola Data ICD

Pengujian mengelola data ICD dilakukan untuk memastikan bahwa *admin* dapat mengelola data ICD. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 11. Pengujian Mengelola Data ICD

Nama Kasus Uji	Mengelola data ICD pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul					
Tujuan	Menguji apakah <i>admin</i> dapat menambah, mengubah, menghapus data diagnosa, dan					
Deskripsi	Fungsi ini akan menyimpan dan menampilkan setiap perubahan yang terjadi					
Kondisi Awal	<i>Admin</i> telah terotentikasi dan berada pada tampilan awal Home					
Skenario Uji						
1. <i>Admin</i> memilih menu mengelola diagnosa						
2. <i>Admin</i> memilih satu menu : tambah						
3. <i>Admin</i> memilih satu menu : edit						
4. <i>Admin</i> memilih satu menu : hapus						
5. <i>Admin</i> memilih satu menu : batal						
6. <i>Admin</i> memilih satu menu : reload data						
7. <i>Admin</i> memilih satu menu : close						
8. <i>Admin</i> memilih satu menu : search						
Kriteria Evaluasi Hasil						
<i>Admin</i> berhasil menambahkan data diagnosa pada sistem						
<i>Admin</i> berhasil mengubah data diagnosa pada sistem						
<i>Admin</i> berhasil menghapus data diagnosa pada sistem						
<i>Admin</i> berhasil membatalkan mengubah atau menambahkan data diagnosa pada sistem						
<i>Admin</i> berhasil mereload data pasien pada sistem						
<i>Admin</i> berhasil menutup atau keluar dari sistem						
<i>Admin</i> berhasil menampilkan hasil dari apa yang dicari						
Kasus dan Hasil Uji (Data Normal)						
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan			
1. <i>Admin</i> memilih menu tambah	Data diagnosa akan bertambah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima			
2. <i>Admin</i> memilih data yang akan diubah dan memilih menu edit	Data diagnosa akan berubah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima			
3. <i>Admin</i> memilih data yang akan dihapus dan memilih menu hapus	Data yang dipilih akan terhapus	Sesuai dengan yang diharapkan	Diterima			
4. <i>Admin</i> memilih menu batal	Data yang dimasukkan akan terhapus dari <i>form</i>	Sesuai dengan yang diharapkan	Diterima			
5. <i>Admin</i> memilih menu reload data	Data diagnosa akan di reload	Sesuai dengan yang diharapkan	Diterima			
6. <i>Admin</i> memilih menu close	<i>Admin</i> akan keluar dari sistem	Sesuai dengan yang diharapkan	Diterima			

7. Admin memilih menu search berdasarkan kode ICD dan nama ICD	Data akan ditampilkan sesuai kode atau nama yang diinput <i>admin</i>	Sesuai dengan yang diharapkan	Diterima
Kasus dan Hasil Uji Coba (Data Tidak Normal)			
<i>Admin</i> memilih menu simpan namun tidak mengisi <i>form</i>	Sistem akan menampilkan <i>message error</i> dan data tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima
<i>Admin</i> memilih menu hapus namun tidak memilih data yang akan dihapus	Sistem akan menampilkan <i>message error</i> dan data tidak akan dihapus	Sesuai dengan yang diharapkan	Diterima
<i>Admin</i> memilih menu edit namun tidak memilih data yang akan diedit	Sistem akan menampilkan <i>message error</i> dan data tidak akan diubah	Sesuai dengan yang diharapkan	Diterima
Catatan			
<i>Admin</i> berhasil mengelola data ICD			

5. Skenario Pengujian Mengelola Data Dokter

Pengujian mengelola data dokter dilakukan untuk memastikan bahwa *admin* dapat mengelola data dokter. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 12. Pengujian Mengelola Data Dokter

Nama Kasus Uji	Mengelola data dokter pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul					
Tujuan	Menguji apakah <i>admin</i> dapat mengelola data dokter					
Deskripsi	Fungsi ini akan menyimpan dan menampilkan segala perubahan yang terjadi					
Kondisi Awal	<i>Admin</i> telah terotentifikasi dan berada pada tampilan awal Home					
Skenario Uji						
1. <i>Admin</i> memilih menu mengelola data dokter						
2. <i>Admin</i> memilih satu menu : tambah						
3. <i>Admin</i> memilih satu menu : edit						
4. <i>Admin</i> memilih satu menu : hapus						
5. <i>Admin</i> memilih satu menu : batal						
6. <i>Admin</i> memilih satu menu : close						
Kriteria Evaluasi Hasil						
<i>Admin</i> berhasil menambahkan data dokter pada sistem						
<i>Admin</i> berhasil mengubah data dokter pada sistem						
<i>Admin</i> berhasil menghapus data dokter pada sistem						
<i>Admin</i> berhasil membatalkan mengubah atau menambahkan data dokter pada sistem						
<i>Admin</i> berhasil menutup atau keluar dari sistem						
Kasus dan Hasil Uji (Data Normal)						
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan			
1. <i>Admin</i> memilih menu tambah	Data dokter akan bertambah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima			
2. <i>Admin</i> memilih data yang akan diubah dan memilih menu edit	Data dokter akan berubah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima			
3. <i>Admin</i> memilih data yang akan dihapus dan memilih menu	Data yang dipilih akan terhapus	Sesuai dengan yang diharapkan	Diterima			

hapus			
4. <i>Admin</i> memilih menu batal	Data yang dimasukkan akan terhapus dari <i>form</i>	Sesuai dengan yang diharapkan	Diterima
5. <i>Admin</i> memilih menu close	<i>Admin</i> akan keluar dari sistem	Sesuai dengan yang diharapkan	Diterima
Kasus dan Hasil Uji Coba (Data Tidak Normal)			
<i>Admin</i> memilih menu simpan namun tidak mengisi <i>form</i>	Sistem akan menampilkan <i>message error</i> dan data tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima
<i>Admin</i> memilih menu hapus namun tidak memilih data yang akan dihapus	Sistem akan menampilkan <i>message error</i> dan data tidak akan dihapus	Sesuai dengan yang diharapkan	Diterima
<i>Admin</i> memilih menu edit namun tidak memilih data yang akan diedit	Sistem akan menampilkan <i>message error</i> dan data tidak akan diubah	Sesuai dengan yang diharapkan	Diterima
Catatan			
<i>Admin</i> berhasil mengelola data dokter			

6. Skenario Pengujian Mengelola Tarif Dokter

Pengujian mengelola tarif dokter dilakukan untuk memastikan bahwa *admin* dapat mengelola tarif dokter. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 13. Pengujian Mengelola Tarif Dokter

Nama Kasus Uji	Mengelola tarif dokter pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul		
Tujuan	Menguji apakah <i>admin</i> dapat mengelola tarif dokter		
Deskripsi	Fungsi ini akan menyimpan dan menampilkan segala perubahan yang terjadi		
Kondisi Awal	<i>Admin</i> telah terotentifikasi dan berada pada tampilan awal Home		
Skenario Uji			
1. <i>Admin</i> memilih menu mengelola tarif dokter			
2. <i>Admin</i> memilih satu menu : tambah			
3. <i>Admin</i> memilih satu menu : edit			
4. <i>Admin</i> memilih satu menu : hapus			
5. <i>Admin</i> memilih satu menu : batal			
6. <i>Admin</i> memilih satu menu : close			
Kriteria Evaluasi Hasil			
<i>Admin</i> berhasil menambahkan data tarif dokter pada sistem			
<i>Admin</i> berhasil mengubah data tarif dokter pada sistem			
<i>Admin</i> berhasil menghapus data tarif dokter pada sistem			
<i>Admin</i> berhasil membatalkan mengubah atau menambahkan data tarif dokter pada sistem			
<i>Admin</i> berhasil menutup atau keluar dari sistem			
Kasus dan Hasil Uji (Data Normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. <i>Admin</i> memilih menu tambah	Data tarif dokter akan bertambah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima
2. <i>Admin</i> memilih data yang akan diubah dan memilih menu edit	Data tarif dokter akan berubah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima

3. <i>Admin</i> memilih data yang akan dihapus dan memilih menu hapus	Data yang dipilih akan terhapus	Sesuai dengan yang diharapkan	Diterima
4. <i>Admin</i> memilih menu batal	Data yang dimasukkan akan terhapus dari <i>form</i>	Sesuai dengan yang diharapkan	Diterima
5. <i>Admin</i> memilih menu close	<i>Admin</i> akan keluar dari sistem	Sesuai dengan yang diharapkan	Diterima
Kasus dan Hasil Uji Coba (Data Tidak Normal)			
<i>Admin</i> memilih menu simpan namun tidak mengisi <i>form</i>	Sistem akan menampilkan <i>message error</i> dan data tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima
<i>Admin</i> memilih menu hapus namun tidak memilih data yang akan dihapus	Sistem akan menampilkan <i>message error</i> dan data tidak akan dihapus	Sesuai dengan yang diharapkan	Diterima
<i>Admin</i> memilih menu edit namun tidak memilih data yang akan diedit	Sistem akan menampilkan <i>message error</i> dan data tidak akan diubah	Sesuai dengan yang diharapkan	Diterima
Catatan			
<i>Admin</i> berhasil mengelola data tarif dokter			

7. Skenario Pengujian Mengelola Jadwal Praktek Dokter

Pengujian mengelola jadwal praktek dokter dilakukan untuk memastikan bahwa *admin* dapat mengelola jadwal praktek dokter. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 14. Pengujian Mengelola Jadwal Praktek Dokter

Nama Kasus Uji	Mengelola jadwal praktek dokter pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul		
Tujuan	Menguji apakah <i>admin</i> dapat mengelola jadwal praktek dokter		
Deskripsi	Fungsi ini akan menyimpan dan menampilkan segala perubahan yang terjadi		
Kondisi Awal	<i>Admin</i> telah terotentifikasi dan berada pada tampilan awal Home		
Skenario Uji			
1. <i>Admin</i> memilih menu mengelola jadwal praktek dokter			
2. <i>Admin</i> memilih satu menu : tambah			
3. <i>Admin</i> memilih satu menu : edit			
4. <i>Admin</i> memilih satu menu : hapus			
5. <i>Admin</i> memilih satu menu : batal			
6. <i>Admin</i> memilih satu menu : close			
Kriteria Evaluasi Hasil			
<i>Admin</i> berhasil menambahkan data jadwal praktek dokter pada sistem			
<i>Admin</i> berhasil mengubah data jadwal praktek dokter pada sistem			
<i>Admin</i> berhasil menghapus data jadwal praktek dokter pada sistem			
<i>Admin</i> berhasil membatalkan mengubah atau menambahkan data jadwal praktek dokter pada sistem			
<i>Admin</i> berhasil menutup atau keluar dari sistem			
Kasus dan Hasil Uji (Data Normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. <i>Admin</i> memilih menu tambah	Data jadwal praktek dokter akan bertambah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima
2. <i>Admin</i> memilih data	Data jadwal praktek dokter	Sesuai dengan yang	Diterima

yang akan diubah dan memilih menu edit	akan berubah dan ditampilkan	diharapkan	
3. Admin memilih data yang akan dihapus dan memilih menu hapus	Data yang dipilih akan terhapus	Sesuai dengan yang diharapkan	Diterima
4. Admin memilih menu batal	Data yang dimasukkan akan terhapus dari <i>form</i>	Sesuai dengan yang diharapkan	Diterima
5. Admin memilih menu close	Admin akan keluar dari sistem	Sesuai dengan yang diharapkan	Diterima
Kasus dan Hasil Uji Coba (Data Tidak Normal)			
Admin memilih menu simpan namun tidak mengisi <i>form</i>	Sistem akan menampilkan <i>message error</i> dan data tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima
Admin memilih menu hapus namun tidak memilih data yang akan dihapus	Sistem akan menampilkan <i>message error</i> dan data tidak akan dihapus	Sesuai dengan yang diharapkan	Diterima
Admin memilih menu edit namun tidak memilih data yang akan diedit	Sistem akan menampilkan <i>message error</i> dan data tidak akan diubah	Sesuai dengan yang diharapkan	Diterima
Catatan			
Admin berhasil mengelola data jadwal praktek dokter			

8. Skenario Pengujian Mengelola Cuti Dokter

Pengujian mengelola cuti dokter dilakukan untuk memastikan bahwa *admin* dapat mengelola cuti dokter. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 15. Pengujian Mengelola Cuti Dokter

Nama Kasus Uji	Mengelola cuti dokter pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul		
Tujuan	Menguji apakah <i>admin</i> dapat mengelola cuti dokter		
Deskripsi	Fungsi ini akan menyimpan dan menampilkan segala perubahan yang terjadi		
Kondisi Awal	<i>Admin</i> telah terotentifikasi dan berada pada tampilan awal Home		
Skenario Uji			
1. Admin memilih menu mengelola cuti dokter			
2. Admin memilih satu menu : tambah			
3. Admin memilih satu menu : edit			
4. Admin memilih satu menu : hapus			
5. Admin memilih satu menu : batal			
6. Admin memilih satu menu : close			
Kriteria Evaluasi Hasil			
<i>Admin</i> berhasil menambahkan data cuti dokter pada sistem			
<i>Admin</i> berhasil mengubah data cuti dokter pada sistem			
<i>Admin</i> berhasil menghapus data cuti dokter pada sistem			
<i>Admin</i> berhasil membatalkan mengubah atau menambahkan data cuti dokter pada sistem			
<i>Admin</i> berhasil menutup atau keluar dari sistem			
Kasus dan Hasil Uji (Data Normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. Admin memilih	Data cuti dokter akan	Sesuai dengan yang	Diterima

menu tambah	bertambah dan ditampilkan	diharapkan	
2. <i>Admin</i> memilih data yang akan diubah dan memilih menu edit	Data cuti dokter akan berubah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima
3. <i>Admin</i> memilih data yang akan dihapus dan memilih menu hapus	Data yang dipilih akan terhapus	Sesuai dengan yang diharapkan	Diterima
4. <i>Admin</i> memilih menu batal	Data yang dimasukkan akan terhapus dari <i>form</i>	Sesuai dengan yang diharapkan	Diterima
5. <i>Admin</i> memilih menu close	<i>Admin</i> akan keluar dari sistem	Sesuai dengan yang diharapkan	Diterima
Kasus dan Hasil Uji Coba (Data Tidak Normal)			
<i>Admin</i> memilih menu simpan namun tidak mengisi <i>form</i>	Sistem akan menampilkan <i>message error</i> dan data tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima
<i>Admin</i> memilih menu hapus namun tidak memilih data yang akan dihapus	Sistem akan menampilkan <i>message error</i> dan data tidak akan dihapus	Sesuai dengan yang diharapkan	Diterima
<i>Admin</i> memilih menu edit namun tidak memilih data yang akan diedit	Sistem akan menampilkan <i>message error</i> dan data tidak akan diubah	Sesuai dengan yang diharapkan	Diterima
Catatan			
<i>Admin</i> berhasil mengelola data cuti dokter			

9. Skenario Pengujian Melihat Data Dokter

Pengujian melihat data dokter dilakukan untuk memastikan bahwa perawat dapat melihat data dokter. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 16. Pengujian Melihat Data Dokter

Nama Kasus Uji	Melihat data dokter pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul		
Tujuan	Menguji apakah perawat dapat melihat data dokter		
Deskripsi	Fungsi ini akan menampilkan <i>list</i> dokter		
Kondisi Awal	Perawat telah terotentikasi dan berada pada tampilan awal Home		
Skenario Uji			
1. Perawat memilih menu dokter			
2. Perawat melakukan <i>search</i> dokter			
Kriteria Evaluasi Hasil			
Perawat berhasil melihat daftar dokter			
Perawat berhasil melakukan <i>search</i> dokter			
Perawat berhasil melihat detail data dokter			
Kasus dan Hasil Uji (Data Normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. Perawat memilih menu dokter	Sistem akan menampilkan semua <i>list</i> dokter	Sesuai dengan yang diharapkan	Diterima
2. Perawat melakukan <i>search</i> dokter	List dokter akan	Sesuai dengan yang	Diterima

pencarian dokter berdasarkan kode dokter atau nama dokter	ditampilkan sesuai dengan <i>keyword</i> yang dimasukkan perawat	diharapkan	
3. Perawat mengklik salah satu data dokter yang ingin dilihat detailnya	Sistem akan menampilkan detail dari data dokter yang dipilih	Sesuai dengan yang diharapkan	Diterima
Kasus dan Hasil Uji Coba (Data Tidak Normal)			
Perawat memilih menu cari namun tidak memasukkan <i>keyword</i>	Sistem akan menampilkan <i>message error</i> dan data hasil pemeriksaan tidak akan ditampilkan	Sesuai dengan yang diharapkan	Diterima
Catatan			
Perawat berhasil melihat <i>list</i> dokter			

10. Skenario Pengujian Mengelola Obat

Pengujian mengelola obat dilakukan untuk memastikan bahwa *admin* dapat mengelola obat. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 17. Pengujian Mengelola Obat

Nama Kasus Uji	Mengelola obat pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul					
Tujuan	Menguji apakah <i>admin</i> dapat mengelola data obat					
Deskripsi	Fungsi ini akan menyimpan dan menampilkan segala perubahan yang terjadi					
Kondisi Awal	<i>Admin</i> telah terotentifikasi dan berada pada tampilan awal Home					
Skenario Uji						
1. <i>Admin</i> memilih menu mengelola data perawat						
2. <i>Admin</i> memilih satu menu : tambah						
3. <i>Admin</i> memilih satu menu : edit						
4. <i>Admin</i> memilih satu menu : hapus						
5. <i>Admin</i> memilih satu menu : batal						
6. <i>Admin</i> memilih satu menu : close						
7. <i>Admin</i> memilih menu : <i>search</i>						
Kriteria Evaluasi Hasil						
<i>Admin</i> berhasil menambahkan data obat pada sistem						
<i>Admin</i> berhasil mengubah data obat pada sistem						
<i>Admin</i> berhasil menghapus data obat pada sistem						
<i>Admin</i> berhasil membatalkan mengubah atau menambahkan data obat pada sistem						
<i>Admin</i> berhasil menutup atau keluar dari sistem						
<i>Admin</i> berhasil menampilkan data obat sesuai dengan <i>keyword</i> yang dimasukkan						
Kasus dan Hasil Uji (Data Normal)						
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan			
1. <i>Admin</i> memilih menu tambah	Data obat akan bertambah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima			
2. <i>Admin</i> memilih data yang akan diubah dan memilih menu edit	Data obat akan berubah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima			
3. <i>Admin</i> memilih data yang akan dihapus	Data yang dipilih akan terhapus	Sesuai dengan yang diharapkan	Diterima			

dan memilih menu hapus			
4. Admin memilih menu batal	Data yang dimasukkan akan terhapus dari <i>form</i>	Sesuai dengan yang diharapkan	Diterima
5. Admin memilih menu <i>search</i>	Data akan ditampilkan sesuai dengan <i>keyword</i> yang diinput	Sesuai dengan yang diharapkan	Diterima
6. Admin memilih menu close	Admin akan keluar dari sistem	Sesuai dengan yang diharapkan	Diterima
Kasus dan Hasil Uji Coba (Data Tidak Normal)			
Admin memilih menu simpan namun tidak mengisi <i>form</i>	Sistem akan menampilkan <i>message error</i> dan data tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima
Admin memilih menu hapus namun tidak memilih data yang akan dihapus	Sistem akan menampilkan <i>message error</i> dan data tidak akan dihapus	Sesuai dengan yang diharapkan	Diterima
Admin memilih menu edit namun tidak memilih data yang akan diedit	Sistem akan menampilkan <i>message error</i> dan data tidak akan diubah	Sesuai dengan yang diharapkan	Diterima
Catatan			
Admin berhasil mengelola data obat			

11. Skenario Pengujian Mengelola Data Perawat

Pengujian mengelola data perawat dilakukan untuk memastikan bahwa *admin* dapat mengelola data perawat. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 18. Pengujian Mengelola Data Perawat

Nama Kasus Uji	Mengelola data perawat pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul		
Tujuan	Menguji apakah <i>admin</i> dapat mengelola data perawat		
Deskripsi	Fungsi ini akan menyimpan dan menampilkan segala perubahan yang terjadi		
Kondisi Awal	<i>Admin</i> telah terotentifikasi dan berada pada tampilan awal Home		
Skenario Uji			
9. Admin memilih menu mengelola data perawat			
10. Admin memilih satu menu : tambah			
11. Admin memilih satu menu : edit			
12. Admin memilih satu menu : hapus			
13. Admin memilih satu menu : batal			
14. Admin memilih satu menu : close			
15. Admin memilih menu : search			
Kriteria Evaluasi Hasil			
Admin berhasil menambahkan data perawat pada sistem			
Admin berhasil mengubah data perawat pada sistem			
Admin berhasil menghapus data perawat pada sistem			
Admin berhasil membatalkan mengubah atau menambahkan data perawat pada sistem			
Admin berhasil menutup atau keluar dari sistem			
Admin berhasil menampilkan data perawat sesuai <i>keyword</i> yang dimasukkan			
Kasus dan Hasil Uji (Data Normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. Admin memilih	Data perawat akan	Sesuai dengan yang	Diterima

menu tambah	bertambah dan ditampilkan	diharapkan	
2. <i>Admin</i> memilih data yang akan diubah dan memilih menu edit	Data perawat akan berubah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima
3. <i>Admin</i> memilih data yang akan dihapus dan memilih menu hapus	Data yang dipilih akan terhapus	Sesuai dengan yang diharapkan	Diterima
4. <i>Admin</i> memilih menu batal	Data yang dimasukkan akan terhapus dari <i>form</i>	Sesuai dengan yang diharapkan	Diterima
5. <i>Admin</i> memilih menu close	<i>Admin</i> akan keluar dari sistem	Sesuai dengan yang diharapkan	Diterima
6. <i>Admin</i> memilih menu <i>search</i>	Data akan ditampilkan sesuai dengan <i>keyword</i> yang diinput	Sesuai dengan yang diharapkan	Diterima
Kasus dan Hasil Uji Coba (Data Tidak Normal)			
<i>Admin</i> memilih menu simpan namun tidak mengisi <i>form</i>	Sistem akan menampilkan <i>message error</i> dan data tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima
<i>Admin</i> memilih menu hapus namun tidak memilih data yang akan dihapus	Sistem akan menampilkan <i>message error</i> dan data tidak akan dihapus	Sesuai dengan yang diharapkan	Diterima
<i>Admin</i> memilih menu edit namun tidak memilih data yang akan diedit	Sistem akan menampilkan <i>message error</i> dan data tidak akan diubah	Sesuai dengan yang diharapkan	Diterima
Catatan			
<i>Admin</i> berhasil mengelola data perawat			

12. Skenario Pengujian Mengelola Data Tindakan

Pengujian mengelola data tindakan dilakukan untuk memastikan bahwa *admin* dapat mengelola data tindakan. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 19. Pengujian Mengelola Data Tindakan

Nama Kasus Uji	Mengelola data tindakan pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul
Tujuan	Menguji apakah <i>admin</i> dapat mengelola data tindakan
Deskripsi	Fungsi ini akan menyimpan dan menampilkan segala perubahan yang terjadi
Kondisi Awal	<i>Admin</i> telah terotentifikasi dan berada pada tampilan awal Home
Skenario Uji	
1. <i>Admin</i> memilih menu mengelola data perawat	
2. <i>Admin</i> memilih satu menu : tambah	
3. <i>Admin</i> memilih satu menu : edit	
4. <i>Admin</i> memilih satu menu : hapus	
5. <i>Admin</i> memilih satu menu : batal	
6. <i>Admin</i> memilih satu menu : close	
Kriteria Evaluasi Hasil	
<i>Admin</i> berhasil menambahkan data ruangan pada sistem	
<i>Admin</i> berhasil mengubah data ruangan pada sistem	

<p><i>Admin</i> berhasil menghapus data ruangan pada sistem <i>Admin</i> berhasil membatalkan mengubah atau menambahkan data ruangan pada sistem <i>Admin</i> berhasil menutup atau keluar dari sistem</p>				
Kasus dan Hasil Uji (Data Normal)				
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan	
1. <i>Admin</i> memilih menu tambah	Data ruangan akan bertambah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima	
2. <i>Admin</i> memilih data yang akan diubah dan memilih menu edit	Data ruangan akan berubah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima	
3. <i>Admin</i> memilih data yang akan dihapus dan memilih menu hapus	Data yang dipilih akan terhapus	Sesuai dengan yang diharapkan	Diterima	
4. <i>Admin</i> memilih menu batal	Data yang dimasukkan akan terhapus dari <i>form</i>	Sesuai dengan yang diharapkan	Diterima	
5. <i>Admin</i> memilih menu close	<i>Admin</i> akan keluar dari sistem	Sesuai dengan yang diharapkan	Diterima	
Kasus dan Hasil Uji Coba (Data Tidak Normal)				
<i>Admin</i> memilih menu simpan namun tidak mengisi <i>form</i>	Sistem akan menampilkan <i>message error</i> dan data tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima	
<i>Admin</i> memilih menu hapus namun tidak memilih data yang akan dihapus	Sistem akan menampilkan <i>message error</i> dan data tidak akan dihapus	Sesuai dengan yang diharapkan	Diterima	
<i>Admin</i> memilih menu edit namun tidak memilih data yang akan diedit	Sistem akan menampilkan <i>message error</i> dan data tidak akan diubah	Sesuai dengan yang diharapkan	Diterima	
Catatan				
<i>Admin</i> berhasil mengelola data tindakan				

13. Skenario Pengujian Mengelola Ruangan

Pengujian mengelola ruangan dilakukan untuk memastikan bahwa *admin* dapat mengelola ruangan. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 20. Pengujian Mengelola Ruangan

Nama Kasus Uji	Mengelola ruangan pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul
Tujuan	Menguji apakah <i>admin</i> dapat mengelola data ruangan
Deskripsi	Fungsi ini akan menyimpan dan menampilkan segala perubahan yang terjadi
Kondisi Awal	<i>Admin</i> telah terotentifikasi dan berada pada tampilan awal Home
Skenario Uji	
1. <i>Admin</i> memilih menu mengelola data ruangan 2. <i>Admin</i> memilih satu menu : tambah 3. <i>Admin</i> memilih satu menu : edit 4. <i>Admin</i> memilih satu menu : hapus 5. <i>Admin</i> memilih satu menu : batal 6. <i>Admin</i> memilih menu : search	

<p>7. Admin memilih satu menu : close</p> <p style="text-align: center;">Kriteria Evaluasi Hasil</p> <p><i>Admin berhasil menambahkan data ruangan pada sistem</i> <i>Admin berhasil mengubah data ruangan pada sistem</i> <i>Admin berhasil menghapus data ruangan pada sistem</i> <i>Admin berhasil membatalkan mengubah atau menambahkan data ruangan pada sistem</i> <i>Admin berhasil menampilkan data ruangan sesuai dengan keyword yang dimasukkan</i> <i>Admin berhasil menutup atau keluar dari sistem</i></p>			
Kasus dan Hasil Uji (Data Normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. Admin memilih menu tambah	Data ruangan akan bertambah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima
2. Admin memilih data yang akan diubah dan memilih menu edit	Data ruangan akan berubah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima
3. Admin memilih data yang akan dihapus dan memilih menu hapus	Data yang dipilih akan terhapus	Sesuai dengan yang diharapkan	Diterima
4. Admin memilih menu batal	Data yang dimasukkan akan terhapus dari form	Sesuai dengan yang diharapkan	Diterima
5. Admin memilih menu search	Data akan ditampilkan sesuai dengan keyword yang diinput	Sesuai dengan yang diharapkan	Diterima
6. Admin memilih menu close	Admin akan keluar dari sistem	Sesuai dengan yang diharapkan	Diterima
Kasus dan Hasil Uji Coba (Data Tidak Normal)			
Admin memilih menu simpan namun tidak mengisi form	Sistem akan menampilkan message error dan data tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima
Admin memilih menu hapus namun tidak memilih data yang akan dihapus	Sistem akan menampilkan message error dan data tidak akan dihapus	Sesuai dengan yang diharapkan	Diterima
Admin memilih menu edit namun tidak memilih data yang akan diedit	Sistem akan menampilkan message error dan data tidak akan diubah	Sesuai dengan yang diharapkan	Diterima
Catatan			
Admin berhasil mengelola data ruangan			

14. Skenario Pengujian Mengelola Rawat Jalan

Pengujian mengelola rawat jalan dilakukan untuk memastikan bahwa pegawai dapat mengelola rawat jalan. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 21. Pengujian Mengelola Rawat Jalan

Nama Kasus Uji	Mengelola rawat jalan pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul
Tujuan	Menguji apakah pegawai dapat mengelola rawat jalan
Deskripsi	Fungsi ini akan menyimpan dan menampilkan setiap perubahan yang

	terjadi					
Kondisi Awal	Pegawai telah terotentikasi dan berada pada tampilan awal Home					
Skenario Uji						
1. Pegawai memilih menu mengelola rawat jalan 2. Pegawai memilih satu menu : tambah 3. Pegawai memilih satu menu : edit 4. Pegawai memilih satu menu : hapus 5. Pegawai memilih satu menu : batal 6. Pegawai memilih menu : cetak excel 7. Pegawai memilih satu menu : close						
Kriteria Evaluasi Hasil						
Pegawai berhasil menambahkan data rawat jalan pada sistem Pegawai berhasil mengubah data rawat jalan pada sistem Pegawai berhasil menghapus data rawat jalan pada sistem Pegawai berhasil membatalkan mengubah atau menambahkan data rawat jalan pada sistem Pegawai berhasil mencetak laporan data rawat jalan ke dalam file excel Pegawai berhasil menutup atau keluar dari sistem						
Kasus dan Hasil Uji (Data Normal)						
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan			
1. Pegawai memilih menu tambah	Data rawat jalan akan bertambah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima			
2. Pegawai memilih data yang akan diubah dan memilih menu edit	Data rawat jalan akan berubah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima			
3. Pegawai memilih data yang akan dihapus dan memilih menu hapus	Data yang dipilih akan terhapus	Sesuai dengan yang diharapkan	Diterima			
4. Pegawai memilih menu batal	Data yang dimasukkan akan terhapus dari <i>form</i>	Sesuai dengan yang diharapkan	Diterima			
5. Pegawai memilih menu close	Pegawai akan keluar dari sistem	Sesuai dengan yang diharapkan	Diterima			
6. Pegawai menginput tahun yang akan dicetak dan memilih menu cetak excel	Semua data rawat jalan akan dicetak ke dalam file excel sesuai tahun yang di input dan meminta konfirmasi tempat penyimpanannya	Sesuai dengan yang diharapkan	Diterima			
7. Pegawai memilih menu cetak excel	Semua data rawat jalan akan dicetak ke dalam excel dan meminta konfirmasi tempat penyimpanannya	Sesuai dengan yang diharapkan	Diterima			
Kasus dan Hasil Uji Coba (Data Tidak Normal)						
Pegawai memilih menu simpan namun tidak mengisi <i>form</i>	Sistem akan menampilkan <i>message error</i> dan data tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima			
Pegawai memilih menu hapus namun tidak memilih data yang akan dihapus	Sistem akan menampilkan <i>message error</i> dan data tidak akan dihapus	Sesuai dengan yang diharapkan	Diterima			
Pegawai memilih menu edit namun tidak memilih data	Sistem akan menampilkan <i>message error</i> dan data	Sesuai dengan yang diharapkan	Diterima			

yang akan diedit	tidak akan diubah	Sesuai dengan yang diharapkan	Diterima
Pegawai membatalkan untuk mencetak data pasien ke file excel	Sistem akan menampilkan <i>message error</i> dan data tidak akan dicetak		
Catatan			
Perawat berhasil mengelola data rawat jalan			

15. Skenario Pengujian Mengelola Rawat Inap

Pengujian mengelola rawat inap dilakukan untuk memastikan bahwa pegawai dapat mengelola rawat inap. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 22. Pengujian Mengelola Rawat Inap

Nama Kasus Uji	Mengelola rawat inap pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul					
Tujuan	Menguji apakah pegawai dapat mengelola rawat inap					
Deskripsi	Fungsi ini akan menyimpan dan menampilkan setiap perubahan yang terjadi					
Kondisi Awal	Pegawai telah terotentikasi dan berada pada tampilan awal Home					
Skenario Uji						
1.	Pegawai memilih menu mengelola rawat inap					
2.	Pegawai memilih satu menu : tambah					
3.	Pegawai memilih satu menu : edit					
4.	Pegawai memilih satu menu : hapus					
5.	Pegawai memilih satu menu : batal					
6.	Pegawai memilih menu : cetak exel					
7.	Pegawai memilih satu menu : search					
8.	Pegawai memilih satu menu : pasien keluar					
9.	Pegawai memilih satu menu : close					
Kriteria Evaluasi Hasil						
Pegawai berhasil menambahkan data rawat inap pada sistem						
Pegawai berhasil mengubah data rawat inap pada sistem						
Pegawai berhasil menghapus data rawat inap pada sistem						
Pegawai berhasil membatalkan mengubah atau menambahkan data rawat inap pada sistem						
Pegawai berhasil mencetak laporan data rawat inap ke dalam file excel						
Pegawai berhasil menampilkan data sesuai dengan <i>keyword</i> yang dimasukkan						
Pegawai berhasil mengubah status pasien menjadi tidak pasien rawat inap						
Pegawai berhasil menutup atau keluar dari sistem						
Kasus dan Hasil Uji (Data Normal)						
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan			
1. Pegawai memilih menu tambah	Data rawat inap akan bertambah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima			
2. Pegawai memilih data yang akan diubah dan memilih menu edit	Data rawat inap akan berubah dan ditampilkan	Sesuai dengan yang diharapkan	Diterima			
3. Pegawai memilih data yang akan dihapus dan memilih menu hapus	Data yang dipilih akan terhapus	Sesuai dengan yang diharapkan	Diterima			
4. Pegawai memilih menu batal	Data yang dimasukkan akan terhapus dari <i>form</i>	Sesuai dengan yang diharapkan	Diterima			

5. Pegawai memilih salah satu data pasien yang akan diubah statusnya dan memilih menu pasien keluar	Data pasien yang dipilih akan terhapus dari data rawat inap	Sesuai dengan yang diharapkan	Diterima
6. Pegawai memasukkan no registrasi untuk dilakukan pencarian	Data pasien rawat inap akan ditampilkan sesuai dengan <i>keyword</i> yang dimasukkan	Sesuai dengan yang diharapkan	Diterima
7. Pegawai memilih menu close	Pegawai akan keluar dari sistem	Sesuai dengan yang diharapkan	Diterima
8. Pegawai menginput tahun yang akan dicetak dan memilih menu cetak excel	Semua data rawat jalan akan dicetak ke dalam file excel sesuai tahun yang di input dan meminta konfirmasi tempat penyimpanannya	Sesuai dengan yang diharapkan	Diterima
9. Pegawai memilih menu cetak excel	Semua data rawat jalan akan dicetak ke dalam excel dan meminta konfirmasi tempat penyimpanannya	Sesuai dengan yang diharapkan	Diterima

Kasus dan Hasil Uji Coba (Data Tidak Normal)

Pegawai memilih menu simpan namun tidak mengisi <i>form</i>	Sistem akan menampilkan <i>message error</i> dan data tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima
Pegawai memilih menu hapus namun tidak memilih data yang akan dihapus	Sistem akan menampilkan <i>message error</i> dan data tidak akan dihapus	Sesuai dengan yang diharapkan	Diterima
Pegawai memilih menu edit namun tidak memilih data yang akan diedit	Sistem akan menampilkan <i>message error</i> dan data tidak akan diubah	Sesuai dengan yang diharapkan	Diterima
Pegawai membatalkan untuk mencetak data pasien ke file excel	Sistem akan menampilkan <i>message error</i> dan data tidak akan dicetak	Sesuai dengan yang diharapkan	Diterima

Catatan

Perawat berhasil mengelola data rawat inap

16. Skenario Pengujian Mengelola Rekam Medis

Pengujian mengelola data pasien dilakukan untuk memastikan bahwa dokter dapat mengelola mengelola rekam medis. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 23. Pengujian Mengelola Rekam Medis

Nama Kasus Uji	Mengelola rekam medis pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul					
Tujuan	Menguji apakah dokter dapat mengelola data rekam medis					
Deskripsi	Fungsi ini akan menyimpan dan menampilkan segala perubahan yang terjadi					
Kondisi Awal	Dokter telah terotentikasi dan berada pada tampilan awal Home					
Skenario Uji						
<ol style="list-style-type: none"> 1. Dokter memilih menu mengelola rekam medis 2. Dokter memilih satu menu : cetak 3. Dokter memilih satu menu : tambah 4. Dokter memilih satu menu : close 						
Kriteria Evaluasi Hasil						
Dokter berhasil menambahkan data rekam medis pada sistem Dokter berhasil mencetak rekam medis pasien Dokter berhasil menutup atau keluar dari sistem						
Kasus dan Hasil Uji (Data Normal)						
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan			
1. Dokter memilih salah satu radio button antara rawat inap atau rawat jalan, jika data telah tampil maka dokter akan memilih menu tambah dan mengisi form yang masih kosong 2. Dokter memilih menu cetak 3. Dokter memilih menu close	Data rekam medis akan bertambah dan ditampilkan Data rekam medis akan ditampilkan kemudian dapat dicetak sesuai dengan format yang telah ditentukan Dokter akan keluar dari sistem	Sesuai dengan yang diharapkan Sesuai dengan yang diharapkan Sesuai dengan yang diharapkan	Diterima Diterima Diterima			
Kasus dan Hasil Uji Coba (Data Tidak Normal)						
Dokter memilih menu simpan namun tidak mengisi form	Sistem akan menampilkan <i>message error</i> dan data tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima			
Catatan						
Dokter berhasil mengelola data rekam medis						

17. Skenario Pengujian Mengelola Pembayaran

Pengujian mengelola pembayaran dilakukan untuk memastikan bahwa kasir dapat mengelola pembayaran. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 24. Pengujian Mengelola Pembayaran

Nama Kasus Uji	Mengelola pembayaran pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul					
Tujuan	Menguji apakah kasir dapat mengelola data pembayaran					
Deskripsi	Fungsi ini akan menyimpan dan menampilkan segala perubahan yang terjadi					
Kondisi Awal	Kasir telah terotentikasi dan berada pada tampilan awal Home					
Skenario Uji						
<ol style="list-style-type: none"> 1. Kasir memilih menu pembayaran 2. Kasir memilih menu : bayar 3. Kasir memilih menu : cetak faktur 						
Kriteria Evaluasi Hasil						
Penanggung jawab berhasil melunasi bayaran rumah sakit						
Kasir berhasil mencetak faktur detail pembayaran						
Kasus dan Hasil Uji (Data Normal)						
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan			
1. Kasir memilih menu bayar	Status pembayaran pasien akan berubah dari belum lunas menjadi lunas	Sesuai dengan yang diharapkan	Diterima			
2. Kasir memilih menu cetak faktur	Faktur detail pembayaran akan dicetak	Sesuai dengan yang diharapkan	Diterima			
Kasus dan Hasil Uji Coba (Data Tidak Normal)						
-	-	-	-			
Catatan						
Kasir berhasil mengelola pembayaran pasien						

18. Skenario Pengujian Melihat Daftar Antrian

Pengujian melihat daftar antrian pasien dilakukan untuk memastikan bahwa dokter dapat melihat antrian pasien. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 25. Pengujian Melihat Daftar Antrian

Nama Kasus Uji	Melihat daftar antrian pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul					
Tujuan	Menguji apakah dokter dapat melihat daftar antrian pasien					
Deskripsi	Fungsi ini akan menampilkan daftar antrian pasien yang akan dilayani oleh dokter					
Kondisi Awal	Dokter telah terotentikasi dan berada pada tampilan awal Home					
Skenario Uji						
<ol style="list-style-type: none"> 1. Dokter memilih menu daftar antrian 2. Dokter memilih satu menu : kunci 3. Dokter memilih satu menu : batal kunci 						
Kriteria Evaluasi Hasil						
Dokter berhasil melihat daftar antrian pasien						
Dokter dapat melakukan pencarian sesuai poliklinik dokter						
Kasus dan Hasil Uji (Data Normal)						
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan			
1. Dokter memasukkan poli yang akan	Sistem akan menampilkan daftar antrian sesuai poli yang	Sesuai dengan yang diharapkan	Diterima			

dilihat antriannya	dipilih		
2. Dokter memasukkan poli yang akan dilihat antriannya dan memilih menu kunci	Dokter tidak dapat mengubah poli antrian yang akan dilihat	Sesuai dengan yang diharapkan	Diterima
3. Dokter memilih menu batal kunci	Dokter dapat mengubah poli yang akan dilihat antriannya	Sesuai dengan yang diharapkan	Diterima
Kasus dan Hasil Uji Coba (Data Tidak Normal)			
-	-	-	-
Catatan			
Dokter berhasil melihat daftar antrian pasien sesuai poli yang diinginkan			

19. Skenario Pengujian Melihat Hasil Pemeriksaan

Pengujian melihat hasil pemeriksaan pasien dilakukan untuk memastikan bahwa dokter dapat melihat hasil pemeriksaan pasien. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 26. Pengujian Melihat Hasil Pemeriksaan

Nama Kasus Uji	Melihat hasil pemeriksaan pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul					
Tujuan	Menguji apakah dokter dapat melihat hasil pemeriksaan pasien					
Deskripsi	Fungsi ini akan menampilkan semua hasil pemeriksaan yang pernah dilakukan pasien					
Kondisi Awal	Dokter telah terotentikasi dan berada pada tampilan awal Home					
Skenario Uji						
1. Dokter memilih menu hasil pemeriksaan 2. Dokter melakukan pencarian data hasil pemeriksaan						
Kriteria Evaluasi Hasil						
Dokter berhasil melihat hasil pemeriksaan pasien						
Dokter berhasil melihat detail pasien						
Kasus dan Hasil Uji (Data Normal)						
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan			
1. Dokter memasukkan no rm pasien yang akan dilihat hasil pemeriksannya 2. Dokter mengklik salah satu data hasil pemeriksaan pasien	Sistem akan menampilkan hasil pemeriksaan pasien sesuai dengan <i>keyword</i> yang dimasukkan Detail hasil pemeriksaan pasien akan ditampilkan sesuai dengan yang dipilih	Sesuai dengan yang diharapkan Sesuai dengan yang diharapkan	Diterima Diterima			
Kasus dan Hasil Uji Coba (Data Tidak Normal)						
Dokter memilih menu cari namun tidak memasukkan no rm pasien	Sistem akan menampilkan <i>message error</i> dan data hasil pemeriksaan tidak akan ditampilkan	Sesuai dengan yang diharapkan	Diterima			
Catatan						
Dokter berhasil melihat hasil pemeriksaan pasien						

20. Skenario Pengujian Melihat Statistika Pasien

Pengujian melihat statistika pasien dilakukan untuk memastikan bahwa direktur dapat melihat statistika pasien. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 27. Pengujian Melihat Statistik Pasien

Nama Kasus Uji	Melihat statistika pasien pada Sistem Informasi Rumah Sakit Umum Daerah Doloksanggul					
Tujuan	Menguji apakah direktur dapat melihat statistika pasien					
Deskripsi	Fungsi ini akan menampilkan statistika pasien					
Kondisi Awal	Direktur telah terotentikasi dan berada pada tampilan awal Home					
Skenario Uji						
Direktur masuk ke dalam sistem						
Kriteria Evaluasi Hasil						
Direktur melihat statistika pasien						
Kasus dan Hasil Uji (Data Normal)						
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan			
Direktur masuk ke dalam sistem	Sistem akan menampilkan statistika pasien	Sesuai dengan yang diharapkan	Diterima			
Kasus dan Hasil Uji Coba (Data Tidak Normal)						
-	-	-	-			
Catatan						
Direktur berhasil melihat statistika pasien						

BAB 8 KESIMPULAN DAN SARAN

Pada bab ini akan dijelaskan mengenai kesimpulan dan saran selama penggerjaan Tugas Akhir.

8.1 Kesimpulan

Adapun kesimpulan yang diperoleh selama penggerjaan Tugas Akhir ini, yaitu sebagai berikut:

1. Sistem yang dibangun telah menghasilkan aplikasi berbasis *desktop*, *mobile*, dan *web* yang terintegrasi.
2. Aplikasi yang dibangun dalam tiga *platform* mendapat respon yang positif dari *client* karena dilihat dari hasil UAT. Adapun hasil UAT dapat dilihat pada Lampiran 5.

8.2 Saran

Saran yang dapat diperhatikan untuk pengembangan Tugas Akhir selanjutnya adalah sebagai berikut:

1. Tim pengembang melengkapi modul diluar modul registrasi, rekam medis, dan pembayaran.
2. Mengimplementasikan keamanan sistem pada aplikasi.

DAFTAR PUSTAKA

- [1] Eko Handoyo, A. B, "Aplikasi Sistem Informasi Rumah Sakit Berbasis Web Pada Sub-Sistem Farmasi Menggunakan Framework Prado," *Teknologi Elektro*, vol. 7 No. 1, 2008.
- [2] Guardian Y. Sanjaya, A. R. *Sistem Informasi Rumah Sakit: Kemana arah penggunanya ?* .
- [3] "rumah sakit di Indonesia yang memiliki SIMRS," [Online]. Available: <https://gawaisehat.com/2016/12/01/baru-48-rumah-sakit-di-indonesia-yang-memiliki-simrs-fungsional/>. [Accessed 13 November 2017].
- [4] Saputra, A. B, "Identifikasi Faktor-Faktor Keberhasilan Implementasi Sistem Informasi Manajemen Rumah Sakit. *Jurnal Penelitian Pers dan Komunikasi Pembangunan*," vol. 19 No.3, pp. 135-148, 2016.
- [5] "Penelitian Terapan (Applied Research)," [Online]. Available: <repository.binus.ac.id/content/I0192/I019252523.doc>. [Accessed 18 October 2017].
- [6] Gary B. Shelly, T. J. C. H. J. Rd, *Systems Analysis and Design*. Fourth Edition. United States of America: Course Technology, 2011.
- [7] B. H. W. D. T. Alan Dennis, Systems Analysis and Design with UML Version 2.0 An Object-Oriented Approach. Second Edition. United States of America: John Wiley & Sons, 2005.
- [8] Wahyuningsih, N, "Analisis Lost Patient di Poliklinik Rawat Jalan Rumah Sakit Pertamina Jaya," pp. 41-43, 2009.
- [9] Kadir, A, Pengenalan Sistem Informasi Edisi Revisi, Yogyakarta: ANDI, 2014.
- [10] K. D. T. L. R. Endang Lestari, "Sistem Informasi Rekam Medik Pada Rumah Sakit Bersalin Graha Rap Tanjung Balai Karimun," *Jurnal Sistem Informasi*, vol. 3, NO. 2, pp. 388-389, 2011.
- [11] Ikhsan, "Digital Sense," [Online]. Available: <http://www.digital-sense.net/simrs>. [Accessed 20 October 2017].
- [12] A. C. Sulis Mariyanti, "Burnout Pada Perawat Yang Bertugas Di Ruang Rawat Inap Dan Rawat Jalan Rsab Harapan Kita," *Jurnal Psikologi*, vol. Volume 9 No 2, pp. 48-49, 2011.
- [13] H. H. Pratiwi, "Pengembangan Aplikasi Desktop Sistem Informasi Bmt," p. 18, 2016.
- [14] "lifewire," [Online]. Available: <https://www.lifewire.com/what-is-a-mobile-application-2373354>. [Accessed 28 October 2017].
- [15] Mentari Harmadya, M. H, "Rancang Bangun Aplikasi Try Out Ujian Nasional Sekolah Menengah Pertama (Smp) Berbasis Android," 2015.
- [16] Sampurna, *Menguasai Aplikasi Web Tanpa Pemrograman*, Jakarta: PT Elex Media Komputindo, 2003.
- [17] Noviana, W.(2011). Analisis Sistem Keamanan terhadap Serangan Virus pada Smartphone Berbasis Android dan Symbian OS.
- [18] Speckmann, B, "The Android mobile platform," 2008.
- [19] Heru Supriyono, A. N, "Rancang Bangun Aplikasi Pembelajaran Hadis Untuk Perangkat Mobile Berbasis Android," *Jurnal Informatika*, vol. 8, No. 2, 2014.
- [20] "Perbedaan Android Studio Dengan Eclipse," [Online]. Available:

- <https://www.angon.co.id/news/applications/perbedaan-eclipse-dan-android-studio>. [Accessed 01 November 2017].
- [21] F. Irmansyah, *Pengantar Database*, 2003.
- [22] T. I. Sri Haryanti, "Rancang Bangun Sistem Informasi E-Commerce Untuk Usaha Fashion," *Journal Speed*, vol. 3 No 1, p. 11, 2011.
- [23] "Tutorialspoint,"[Online]. Available: https://www.tutorialspoint.com/ms_sql_server/. [Accessed 28 October 2017].
- [24] "Comparison between MySQL vs MS SQL Server," [Online]. Available: <https://medium.com/@mindfiresolutions.usa/a-comparison-between-mysql-vs-ms-sql-server-58b537e474be>. [Accessed 30 October 2017].
- [25] Abdulloh, R, Membuat Aplikasi Point of Sale dengan Laravel dan Ajax, Jakarta: PT Elex Media Komputindo, 2017.
- [26] TechInputs, 5 January 2017. [Online]. Available: <http://www.techinputs.com/framework/php-framework-comparison/>. [Accessed 28 October 2017].
- [27] Mark Heckler, G. G, *JavaFX 8: Introduction by Example*, Apress, 2014.
- [28] H. T. A. H. B. Ahmad Leo Yudanto, "Rancang Bangun Aplikasi Sistem Informasi Manajemen Laboratorium Biomedik Fakultas Kedokteran Universitas Brawijaya," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, No. 8, p. 629, 2017.
- [29] S. J. Mellor, A. N. Clark and T. Futagami, "Model-Driven Development," IEEE Computer Society, p. 14, 2003.
- [30] B. Selic, in *The Pragmatics of Model-Driven Development*, IEEE Computer Society , p. 19, 2003.
- [31] C. Atkinson and T. Kuhne, "Model-Driven Development: A Metamodeling Foundation," in *Model-driven Development*, IEEE Computer Society, pp. 36-41, 2003.
- [32] B. W. Acep Taryana, "Model-Driven Development : Fase Awal Verifikasi Model Design Rekam Medis Elektronis Menggunakan Perumusan Graf Lengkap," *Model-Driven Development*, vol. 6 Nomor 2, pp. 53 - 64, 2014.
- [33] G. Y. S. A. R. R. B. M. E. N. Evy Hariana, "Penggunaan Sistem Informasi Manajemen Rumah Sakit (SIMRS) di DIY," *Seminar Nasional Sistem Informasi Indonesia*, 2013.
- [34] I. A. M. Yanyan Herdiansyah, "Pembangunan Aplikasi Bantu Dalam Menghafal Al-Qur'an Berbasis Mobile," *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, Vols. Vol. 2, No. 2, 2013.
- [35] D. D. W. J. Fifin Noviyanto, "Sequence diagram akan dijelaskan pada gambar berikut," *Sequence diagram akan dijelaskan pada gambar berikut*, Vols. Vol. 3, No. 2, 2009.

Lampiran 1 – Use Case Scenario

Berikut akan dijelaskan *use case* diagram yang terdapat pada sistem informasi rumah sakit.

Use Case Scenario Login

Use case scenario login dapat dilihat pada tabel berikut.

Tabel 28. Use Case Scenario Login

Use Case ID Number	UC-1	
Use Case Name	Login	
Brief Description	Use case untuk masuk ke dalam sistem	
Actor	User (dokter, pegawai, perawat, direktur)	
Precondition	User telah memiliki akun masing-masing	
Primary Flow of Event	User Action	System Response
	1. User menginput username dan password	
	2. User mengklik tombol login	
		3. Sistem melakukan validasi username dan password
		4. Sistem akan menampilkan halaman utama dengan akun yang sudah terautentikasi
Alternate Flow of Event		
Post Condition	User dapat masuk ke halaman utama	

Use Case Scenario Mengelola Data Pasien

Use case scenario mengelola data pasien dapat dilihat pada tabel berikut.

Tabel 29. Use Case Scenario Mengelola Data Pasien

Use Case ID Number	UC-2	
Use Case Name	Mengelola data pasien	
Brief Description	Use case untuk mengelola data pasien	
Actor	Admin	
Precondition	Admin telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Admin memilih menu data pasien	
		2. Sistem akan menampilkan halaman data pasien
	3. Admin mengklik tombol tambah	
	4. Admin menginput data berupa no registrasi, nama pasien, tempat lahir, tanggal lahir, umur, pekerjaan, dan lain sebagainya	

	5. Admin mengklik tombol simpan	
	6. Data pasien akan disimpan dan akan ditampilkan	
	7. Admin mengklik tombol edit	
	8. Admin menginput data yaitu no registrasi, nama pasien, tempat lahir, tanggal lahir, umur, pekerjaan, dan lain sebagainya	
	9. Admin kembali mengklik tombol edit	
		10. Data pasien akan disimpan dan ditampilkan
	11. Admin memilih salah satu data pasien yang akan dihapus dengan mengkliknya	
	12. Admin mengklik tombol hapus	
		13. Sistem akan menampilkan information dialog
	14. Admin memilih menu hapus	
		15. Data pasien berhasil dihapus
	16. Admin mengklik tombol batal	
		17. Field no registrasi, nama pasien, tempat lahir, tanggal lahir, umur, pekerjaan, dan lain sebagainya akan kosong
	18. Admin mengklik tombol keluar	
		19. Aplikasi akan tertutup
Alternate Flow of Event		
Post Condition	Admin berhasil melakukan penambahan, pengubahan, dan penghapusan data pasien	

Use Case Scenario Melihat Data Pasien

Use case scenario melihat data pasien dapat dilihat pada tabel berikut.

Tabel 30. Use Case Scenario Melihat Data Pasien

Use Case ID Number	UC-3	
Use Case Name	Melihat data pasien	
Brief Description	Use case untuk melihat data pasien	
Actor	Perawat	
Precondition	Perawat telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Perawat memilih menu pasien	

		2. Sistem akan menampilkan seluruh data pasien
	3. Perawat mengklik salah satu data dari pasien	
		4. Sistem akan menampilkan detail dari data pasien
Alternate Flow of Event		
Post Condition	Perawat dapat melihat data pasien	

Use Case Scenario Mengelola Data ICD

Use case scenario mengelola data ICD dapat dilihat pada tabel berikut.

Tabel 31. Use Case Scenario Mengelola Data ICD

Use Case ID Number	UC-4	
Use Case Name	Mengelola data ICD	
Brief Description	Use case untuk mengelola data ICD	
Actor	Admin	
Precondition	Admin telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Admin memilih menu data ICD	
		2. Sistem akan menampilkan halaman data ICD
	3. Admin mengklik tombol tambah	
	4. Admin menginput data berupa kode dan nama ICD	
	5. Admin mengklik tombol simpan	
		6. Data ICD disimpan dan akan ditampilkan
	7. Admin mengklik tombol edit	
	8. Admin kembali menginput kode dan nama ICD	
	9. Admin kembali mengklik tombol edit	
		10. Data ICD akan disimpan dan ditampilkan
	11. Admin mengklik tombol batal	
		12. Field kode dan nama ICD akan kosong
	13. Mengklik tombol reload data	
		14. Sistem akan mereload data
	15. Admin mengklik tombol keluar	
		16. Aplikasi akan tertutup
Alternate Flow of Event		

Post Condition	Admin berhasil dalam melakukan penambahan, pengubahan, dan penghapusan data ICD terkait
----------------	---

Use Case Scenario Mengelola Data Dokter

Use case scenario mengelola data dokter dapat dilihat pada tabel berikut.

Tabel 32. Use Case Scenario Mengelola Data Dokter

Use Case ID Number	UC-5	
Use Case Name	Mengelola data dokter	
Brief Description	Use case untuk mengelola data	
Actor	Admin	
Precondition	Admin telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Admin memilih menu dokter	
		2. Sistem akan menampilkan halaman dokter
	3. Admin memilih menu data dokter	
		4. Sistem akan menampilkan form data dokter
	5. Admin mengklik tombol tambah	
	6. Admin menginput data berupa kode dokter, jenis kelamin, alamat, kota, telepon, dan poli	
	7. Admin mengklik tombol simpan	
		8. Data dokter akan disimpan dan akan ditampilkan
	9. Admin mengklik tombol edit	
	10. Admin menginput data yaitu kode dokter, jenis kelamin, alamat, kota, telepon, dan poli	
	11. Admin kembali mengklik tombol edit	
		12. Data dokter akan disimpan dan ditampilkan
	13. Admin memilih salah satu data dokter yang akan dihapus dengan mengkliknya	
	14. Admin mengklik tombol hapus	
		15. Sistem akan menampilkan information dialog
	16. Admin memilih menu hapus	

		17. Data dokter berhasil dihapus
	18. Admin mengklik tombol batal	
		19. Field kode dokter, jenis kelamin, alamat, kota, telepon, dan poli akan kosong
	20. Admin mengklik tombol keluar	
		21. Aplikasi akan tertutup
Alternate Flow of Event		
Post Condition	Admin berhasil dalam melakukan penambahan, pengubahan, dan penghapusan data dokter	

Use Case Scenario Mengelola Tarif Dokter

Use case scenario mengelola tarif dokter dapat dilihat pada tabel berikut.

Tabel 33. Use Case Scenario Mengelola Tarif Dokter

Use Case ID Number	UC-6	
Use Case Name	Mengelola tarif dokter	
Brief Description	Use case untuk mengelola tarif dokter	
Actor	Admin	
Precondition	Admin telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Admin memilih menu dokter	
		2. Sistem akan menampilkan halaman dokter
	3. Admin memilih menu tarif dokter	
		4. Sistem akan menampilkan form data tarif dokter
	5. Admin mengklik tombol tambah	
	6. Admin menginput data berupa kode, nama, dan tarif dokter	
	7. Admin mengklik tombol simpan	
		8. Data tarif dokter akan disimpan dan akan ditampilkan
	9. Admin mengklik tombol edit	
	10. Admin menginput data yaitu kode, nama, dan tarif dokter	
	11. Admin kembali mengklik tombol edit	
		12. Data tarif dokter akan disimpan dan ditampilkan
	13. Admin memilih salah satu data tarif dokter yang akan dihapus	

	dengan mengkliknya	
14. Admin mengklik tombol hapus		
	15. Sistem akan menampilkan information dialog	
16. Admin memilih menu hapus		
	17. Data tarif dokter berhasil dihapus	
18. Admin mengklik tombol batal		
	19. Field kode, nama, dan tarif dokter akan kosong	
20. Admin mengklik tombol keluar		
	21. Aplikasi akan tertutup	
Alternate Flow of Event		
Post Condition	Admin berhasil mengelola tarif dokter (menambah, mengedit, dan menghapus data)	

Use Case Scenario Mengelola Jadwal Praktek Dokter

Use case scenario mengelola jadwal praktek dokter dapat dilihat pada tabel berikut.

Tabel 34. Use Case Scenario Mengelola Jadwal Praktek Dokter

Use Case ID Number	UC-7	
Use Case Name	Mengelola jadwal praktek dokter	
Brief Description	Use case untuk mengelola jadwal praktek dokter	
Actor	Admin	
Precondition	Admin telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Admin memilih menu dokter	
		2. Sistem akan menampilkan halaman dokter
	3. Admin memilih menu jadwal praktek dokter	
		4. Sistem akan menampilkan form data jadwal praktek dokter
	5. Admin mengklik tombol tambah	
	6. Admin menginput data berupa kode, nama, hari praktek, jenis poli, dan info jadwal	
	7. Admin mengklik tombol simpan	
		8. Data jadwal praktek dokter akan disimpan dan akan ditampilkan
	9. Admin mengklik	

	tombol edit	
	10. Admin menginput data yaitu kode, nama, hari praktek, jenis poli, dan info jadwal	
	11. Admin kembali mengklik tombol edit	
		12. Data jadwal praktek dokter akan disimpan dan ditampilkan
	13. Admin memilih salah satu jadwal praktek dokter yang akan dihapus dengan mengkliknya	
	14. Admin mengklik tombol hapus	
		15. Sistem akan menampilkan information dialog
	16. Admin memilih menu hapus	
		17. Data jadwal praktek dokter berhasil dihapus
	18. Admin mengklik tombol batal	
		19. Field kode, nama, hari praktek, jenis poli, dan info jadwal akan kosong
	20. Admin mengklik tombol keluar	
		21. Aplikasi akan tertutup
Alternate Flow of Event		
Post Condition	Admin berhasil mengelola data jadwal praktek dari setiap dokter	

Use Case Scenario Mengelola Cuti Dokter

Use case scenario cuti dokter dapat dilihat pada tabel berikut.

Tabel 35. Use Case Scenario Mengelola Cuti Dokter

Use Case ID Number	UC-8	
Use Case Name	Mengelola cuti dokter	
Brief Description	Use case untuk mengelola cuti dokter	
Actor	Admin	
Precondition	Admin telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Admin memilih menu dokter	
		2. Sistem akan menampilkan halaman dokter
	3. Admin memilih menu cuti dokter	
		4. Sistem akan menampilkan form data cuti dokter
	5. Admin mengklik tombol tambah	
	6. Admin menginput data berupa kode dokter, nama dokter, tanggal mulai, dan sampai tanggal	
	7. Admin mengklik tombol simpan	
		8. Data cuti dokter akan disimpan dan akan ditampilkan
	9. Admin mengklik tombol edit	
	10. Admin menginput data yaitu kode dokter, nama dokter, tanggal mulai, dan sampai tanggal	
	11. Admin kembali mengklik tombol edit	
		12. Data cuti dokter akan disimpan dan ditampilkan
	13. Admin memilih salah satu data cuti dokter yang akan dihapus dengan mengkliknya	
	14. Admin mengklik tombol hapus	
		15. Sistem akan menampilkan information dialog
	16. Admin memilih menu hapus	
		17. Data cuti dokter berhasil dihapus

	18. Admin mengklik tombol batal	
		19. Field kode dokter, nama dokter, tanggal mulai, dan sampai tanggal akan kosong
	20. Admin mengklik tombol keluar	
		21. Aplikasi akan tertutup
Alternate Flow of Event		
Post Condition	Admin berhasil dalam mengelola data cuti dokter	

Use Case Scenario Melihat Data Dokter

Use case scenario melihat data dokter dapat dilihat pada tabel berikut.

Tabel 36. Use Case Scenario Melihat Data Dokter

Use Case ID Number	UC-9	
Use Case Name	Melihat data dokter	
Brief Description	Use case untuk melihat data dokter	
Actor	Perawat	
Precondition	Perawat telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Perawat memilih menu dokter	
		2. Sistem akan menampilkan data dokter
	3. Perawat mengklik salah satu data dari dokter	
		4. Sistem akan menampilkan detail dari data dokter
Alternate Flow of Event		
Post Condition	Perawat dapat melihat data dokter	

Use Case Scenario Mengelola Obat

Use case scenario mengelola obat dapat dilihat pada tabel berikut.

Tabel 37. Use Case Scenario Mengelola Data Obat

Use Case ID Number	UC-10	
Use Case Name	Mengelola Obat	
Brief Description	Use case untuk mengelola obat	
Actor	Admin	
Precondition	Admin telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Admin memilih menu data obat	
		2. Sistem akan menampilkan halaman data obat
	3. Admin mengklik tombol tambah	
	4. Admin menginput data berupa kode resep, reserp obat, jumlah, dan harga	

	5. Admin mengklik tombol simpan	
	6. Data obat disimpan dan akan ditampilkan	
	7. Admin mengklik tombol edit	
	8. Admin kembali menginput kode resep, reserp obat, jumlah, dan harga	
	9. Admin kembali mengklik tombol edit	
		10. Data obat akan disimpan dan ditampilkan
	11. Admin memilih salah satu data obat yang akan dihapus dengan mengkliknya	
	12. Admin mengklik tombol hapus	
		13. Sistem menampilkan menampilkan information dialog
	14. Admin memilih menu hapus	
		15. Data obat berhasil dihapus
	16. Admin mengklik tombol batal	
		17. Field kode resep, reserp obat, jumlah, dan harga akan kosong
	18. Admin mengklik tombol keluar	
		19. Aplikasi akan tertutup
Alternate Flow of Event		
Post Condition	Admin berhasil melakukan penambahan, pengubahan, dan penghapusan data obat terkait	

Use Case Scenario Mengelola Data Perawat

Use case scenario mengelola data perawat dapat dilihat pada tabel berikut.

Tabel 38. Use Case Scenario Mengelola Data Perawat

Use Case ID Number	UC-11	
Use Case Name	Mengelola data perawat	
Brief Description	Use case untuk mengelola data perawat	
Actor	Admin	
Precondition	Admin telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Admin memilih menu data perawat	
		2. Sistem akan menampilkan halaman data perawat
	3. Admin mengklik tombol tambah	
	4. Admin menginput data berupa kode perawat, nama, jenis kelamin, alamat, kota, dan nomor telepon	
	5. Admin mengklik tombol simpan	
		6. Data perawat disimpan dan akan ditampilkan
	7. Admin mengklik tombol edit	
	8. Admin kembali menginput kode perawat, nama, jenis kelamin, alamat, kota, dan nomor telepon	
	9. Admin kembali mengklik tombol edit	
		10. Data perawat akan disimpan dan ditampilkan
	11. Admin memilih salah satu data perawat yang akan dihapus dengan mengkliknya	
	12. Admin mengklik tombol hapus	
		13. Sistem akan menampilkan information dialog
	14. Admin memilih menu hapus	
		15. Data perawat berhasil dihapus
	16. Admin mengklik tombol batal	
		17. Field kode perawat, nama, jenis kelamin, alamat, kota, dan

		nomor telepon akan kosong
	18. Admin mengklik tombol keluar	
		19. Aplikasi akan tertutup
Alternate Flow of Event		
Post Condition	Admin berhasil dalam melakukan penambahan, pengubahan, dan penghapusan terkait data perawat	

Use Case Scenario Mengelola Data Tindakan

Use case scenario mengelola data tindakan dapat dilihat pada tabel berikut.

Tabel 39. Use Case Scenario Mengelola Data Tindakan

Use Case ID Number	UC-12	
Use Case Name	Mengelola Data Tindakan	
Brief Description	Use case untuk mengelola data tindakan	
Actor	Admin	
Precondition	Admin telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Admin memilih menu data tindakan	
		2. Sistem akan menampilkan halaman tindakan
	3. Admin mengklik tombol tambah	
	4. Admin menginput data berupa kode, nama, dan harga masing-masing tindakan	
	5. Admin mengklik tombol simpan	
		6. Data tindakan disimpan dan akan ditampilkan
	7. Admin mengklik tombol edit	
	8. Admin kembali menginput kode, nama, dan harga tindakan	
	9. Admin kembali mengklik tombol edit	
		10. Data tindakan akan disimpan dan ditampilkan
	11. Admin memilih salah satu tindakan yang akan dihapus dengan mengkliknya	
	12. Admin mengklik tombol hapus	
		13. Sistem menampilkan information dialog
	14. Admin memilih menu hapus	
		15. Data tindakan

		berhasil dihapus
	16. Admin mengklik tombol batal	
		17. Field kode, nama, dan harga tindakan akan kosong
	18. Admin mengklik tombol keluar	
	19. Aplikasi akan tertutup	
Alternate Flow of Event		
Post Condition	Admin berhasil dalam melakukan penambahan, pengeditan, dan penghapusan terkaitan data tindakan yang dilakukan	

Use Case Scenario Mengelola Ruangan

Use case scenario mengelola ruangan dapat dilihat pada tabel berikut.

Tabel 40. Use Case Scenario Mengelola Ruangan

Use Case ID Number	UC-13	
Use Case Name	Mengelola Ruangan	
Brief Description	Use case untuk mengelola ruangan	
Actor	Admin	
Precondition	Admin telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Admin memilih menu data ruangan	
		2. Sistem menampilkan halaman mengelola ruangan
	3. Admin mengklik tombol tambah	
	4. Admin menginput data berupa kode kamar, jenis kamar, dan lantai keberadaan kamar	
	5. Admin mengklik tombol simpan	
		6. Data ruangan disimpan dan akan ditampilkan
	7. Admin mengklik tombol edit	
	8. Admin kembali menginput kode kamar, jenis kamar, dan lantai keberadaan kamar	
	9. Admin kembali mengklik tombol edit	
		10. Data ruangan akan disimpan dan ditampilkan
	11. Admin memilih salah satu data ruangan yang akan dihapus dengan mengkliknya	
	12. Admin mengklik tombol hapus	

	13. Sistem menampilkan information dialog
	14. Admin memilih menu hapus
	15. Data ruangan berhasil dihapus
	16. Admin mengklik tombol batal
	17. Field kode kamar, jenis kamar, dan lantai keberadaan kamar akan kosong
	18. Admin mengklik tombol keluar
	19. Aplikasi akan tertutup
Alternate Flow of Event	
Post Condition	Admin berhasil dalam penambahan, pengubahan, dan penghapusan terkait pengelolaan ruangan

Use Case Scenario Mengelola Rawat Jalan

Use case scenario mengelola rawat jalan dapat dilihat pada tabel berikut.

Tabel 41. Use Case Scenario Mengelola Rawat Jalan

Use Case ID Number	UC-14	
Use Case Name	Mengelola rawat jalan	
Brief Description	Use case untuk mengelola rawat jalan	
Actor	Pegawai	
Precondition	Pegawai telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Pegawai memilih menu rawat jalan	
		2. Sistem akan menampilkan halaman rawat jalan
	3. Pegawai mengklik tombol tambah	
	4. Pegawai menginput data berupa no registrasi, no RM, nama pasien, poliklinik, tanggal masuk, tanggal keluar, nama dokter, dan sebagainya	
	5. Pegawai mengklik tombol simpan	
		6. Data rawat jalan disimpan dan akan ditampilkan
	7. Pegawai mengklik tombol edit	
	8. Pegawai kembali menginput data berupa no registrasi, no RM, nama pasien, poliklinik, tanggal masuk, tanggal keluar, nama dokter,	

	dan sebagainya	
9. Pegawai kembali mengklik tombol edit		
	10. Data rawat jalan akan disimpan dan ditampilkan	
11. Pegawai memilih salah satu data rawat jalan yang akan dihapus dengan mengkliknya		
12. Pegawai mengklik tombol hapus		
	13. Sistem akan menampilkan information dialog	
14. Pegawai memilih menu hapus		
	15. Data rawat jalan berhasil dihapus	
16. Pegawai mengklik tombol batal		
	17. Field berupa no registrasi, no RM, nama pasien, poliklinik, tanggal masuk, tanggal keluar, nama dokter, dan sebagainya akan kosong	
18. Pegawai mengklik tombol keluar		
	19. Aplikasi akan tertutup	
Alternate Flow of Event		
Post Condition	Admin berhasil dalam penambahan, pengubahan, dan penghapusan pengelolaan rawat jalan pasien	

Use Case Scenario Mengelola Rawat Inap

Use case scenario mengelola rawat inap dapat dilihat pada tabel berikut.

Tabel 42. Use Case Scenario Mengelola Rawat Inap

Use Case ID Number	UC-15	
Use Case Name	Mengelola rawat inap	
Brief Description	Use case untuk mengelola rawat inap	
Actor	Pegawai	
Precondition	Pegawai telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Pegawai memilih menu rawat inap	
		2. Sistem akan menampilkan halaman rawat inap
	3. Pegawai mengklik tombol tambah	
	4. Pegawai menginput data berupa no registrasi, no RM, nama pasien, nama dokter, nama perawat, asal rujukan, dan sebagainya	
	5. Pegawai mengklik tombol simpan	
		6. Data rawat inap disimpan dan akan ditampilkan
	7. Pegawai mengklik tombol edit	
	8. Pegawai kembali menginput data berupa no registrasi, no RM, nama pasien, nama dokter, nama perawat, asal rujukan, dan sebagainya	
	9. Pegawai kembali mengklik tombol edit	
		10. Data rawat inap akan disimpan dan kemudian ditampilkan
	11. Pegawai memilih salah satu data rawat inap yang akan dihapus dengan mengkliknya	
	12. Pegawai mengklik tombol hapus	
		13. Sistem akan menampilkan information dialog
	14. Pegawai memilih menu hapus	
		15. Data rawat inap

		berhasil dihapus
	16. Pegawai mengklik tombol batal	
		17. Field berupa no registrasi, no RM, nama pasien, nama dokter, nama perawat, asal rujukan, dan sebagainya akan kosong
	18. Pegawai mengklik tombol keluar	
		19. Aplikasi akan tertutup
Alternate Flow of Event		
Post Condition	Pegawai berhasil dalam melakukan penambahan, pengubahan, dan penghapusan pada pengelolaan rawat inap terkait	

Use Case Scenario Mengelola Rekam Medis

Use case scenario mengelola rekam medis dapat dilihat pada tabel berikut.

Tabel 43. Use Case Scenario Mengelola Rekam Medis

Use Case ID Number	UC-16	
Use Case Name	Mengelola rekam medis	
Brief Description	Use case untuk mengelola rekam medis	
Actor	Dokter	
Precondition	Dokter telah melakukan pemeriksaan pasien dan telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Dokter memilih menu rekam medis	
		2. Sistem akan menampilkan halaman rekam medis
	3. Dokter mengklik tombol tambah untuk menambahkan data rekam medis	
	4. Dokter akan memilih rekam medis untuk rawat jalan atau rawat inap dan mengisi form berupa no RM, no registrasi, no transaksi, nama pasien, tanggal periksa nama dokter, uraian periksa, tindakan, pilihan obat, dan diagnose yang diberikan	
	5. Dokter mengklik tombol simpan	
		6. Data akan tersimpan dan akan ditampilkan
Alternate Flow of Event		
Post Condition	Admin berhasil dalam mengelola rekam medis pasien	

Use Case Scenario Mengelola Hasil Pemeriksaan

Use case scenario mengelola hasil pemeriksaan dapat dilihat pada tabel berikut.

Tabel 44. Use Case Scenario Mengelola Hasil Pemeriksaan

Use Case ID Number	UC-17	
Use Case Name	Mengelola hasil pemeriksaan	
Brief Description	Use case untuk mengelola hasil pemeriksaan	
Actor	Dokter	
Precondition	Dokter telah berhasil masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Dokter dan perawat memilih menu hasil pemeriksaan	
		2. Sistem akan menampilkan halaman untuk menu hasil pemeriksaan
	3. Dokter dan perawat memilih menu hasil pemeriksaan	
		4. Sistem akan menampilkan seluruh hasil pemeriksaan dari pasien
	5. Dokter dan perawat mengklik tambah untuk menambahkan data hasil pemeriksaan	
		6. Sistem akan menampilkan <i>form</i> untuk menginput hasil pemeriksaan pasien
	7. Dokter menginput hasil pemeriksaan dan mengklik tombol save	
		8. Sistem akan menyimpan data hasil pemeriksaan dan menampilkannya
Alternate Flow of Event		
Post Condition	Dokter dapat melihat dan menambah hasil pemeriksaan pasien	

Use Case Scenario Mengelola Pembayaran

Use case scenario mengelola pembayaran dapat dilihat pada tabel berikut.

Tabel 45. Use Case Scenario Mengelola Pembayaran

Use Case ID Number	UC-19	
Use Case Name	Mengelola pembayaran	
Brief Description	Use case untuk mengelola pembayaran	
Actor	Kasir	
Precondition	Kasir telah masuk ke dalam sistem	
Primary Flow of Event	User Action	System Response
	1. Kasir memilih menu pembayaran	
		2. Sistem menampilkan form pembayaran
	3. Kasir menginput no transaksi, no registrasi, no RM, dan nama pasien	
		4. Sistem akan menampilkan data pasien dan jumlah yang harus dibayarkan
	5. Kasir menginput jumlah uang yang diberikan pasien	
		6. Sistem menampilkan kembalian dari uang yang dibayarkan
	7. Kasir menekan tombol bayar	
Alternate Flow of Event		
Post Condition	Kasir dapat melakukan pembayaran terhadap pasien	

Lampiran 2 – List Pertanyaan

Pewawancara : **Bagaimana alur rawat jalan di rumah sakit ini, Pak ?**

Narasumber : Pertama sekali, pasien yang ingin berobat pergi ke Loket Registrasi (ambil menunjuk ruangan yang merupakan Loket Registrasi). Untuk pasien baru atau yang belum pernah tercatat untuk menjalani pengobatan di rumah sakit ini akan diminta fotokopi berkas seperti KTP, KK, BPJS/JKN/KISS/dll oleh pegawai. Berkas-berkas ini dibutuhkan untuk pengisian data valid pasien ke sistem. Namun, jika pasien tersebut merupakan pasien lama atau pasien yang sudah pernah menjalani pengobatan di rumah sakit ini dan telah tercatat sebagai pasien, pegawai hanya meminta pasien untuk menyerahkan Kartu Berobat yang digunakan untuk mengecek keberadaan data pasien dalam sistem. Jika pasien tidak membawa Kartu Berobat, maka pasien cukup memberikan identitas umum pasien seperti nama atau nomor rekam medis kepada pegawai untuk digunakan sebagai kunci pencarian data pasien dalam sistem. Setelah melakukan input dan verifikasi data di Loket Registrasi, pasien akan diberikan nomor antrian dan pasien diarahkan untuk menunggu di ruang tunggu. Pasien akan dipanggil satu per satu oleh pegawai sesuai nomor antrian yang didapat sebelumnya. Jika pasien merupakan pasien BPJS, maka pasien akan menuju ruang *Finger Print* untuk memastikan bahwa pasien benar-benar merupakan pasien BPJS yang aktif dan nantinya akan diberikan SEP (Surat Eligibilitas Peserta) yang digunakan untuk menjamin bahwa pasien layak untuk menerima pengobatan gratis di rumah sakit. Pasien yang tidak memiliki SEP tidak berhak untuk menerima pengobatan gratis dan akan membayar biaya pengobatan sendiri. Setelah itu, pasien akan menuju Loket Medical Record untuk mengurus data rekam medis pasien. Untuk pasien umum (bukan merupakan pasien BPJS), akan langsung menuju loket ini (Loket Medical Record). Setelah itu, pasien akan pergi ke poli yang menjadi tujuannya. Jika pasien ingin mengurus surat, maka pasien pergi ke Poli Umum. Jika pasien bermasalah dengan gigi, maka pasien pergi ke Poli Gigi, demikian seterusnya. Di poli, dokter dan perawat akan mengecek *hardcopy* rekam medis pasien. Kemudian setelah pemeriksaan, dokter ataupun perawat

akan mengisi *hardcopy* rekam medis seperti data diagnosa, obat, dan lain-lain. Pengisian data rekam medis akan dimasukkan ke dalam sistem dan sekaligus diisi pada *hardcopy* rekam medis. Ketika pemasukan data dalam sistem, sistem juga telah mengeluarkan biaya total yang harus dibayarkan pasien di akhir. Semua transaksi yang ada akan tersimpan di server rumah sakit.

Pewawancara : Bagaimana jika alur rawat inap, Pak ?

Narasumber : Jika rawat inap biasanya untuk pasien yang kritis. Pasien yang masuk IGD akan didata oleh pegawai yang ada disana. Jika pasien merupakan pasien lama (sudah pernah menjalani pengobatan di rumah sakit), maka pasien telah memiliki rekam medis. Namun jika belum, maka pegawai akan terlebih dahulu membuat rekam medis pasien. Setelah menjalani pengobatan di IGD, maka pasien akan dipindahkan ke ruangan. Sesampainya di ruangan, perawat ruangan juga akan mengelola data terkait pasien yang menggunakan ruangan tersebut.

Pewawancara : Apakah kasir di rumah sakit ini terletak di setiap lantai, Pak ?

Narasumber : Tidak, kasir di rumah ini hanya ada satu. Semua transaksi di rumah sakit harus dibayarkan disana. Jika tidak, maka itu tidak dianggap sebagai transaksi yang sah.

Pewawancara : Apakah di rumah sakit ini ada *Nurse Station*-nya, Pak ?

Narasumber : Tidak ada *Nurse Station*. Tapi di setiap ruangan rumah sakit sudah ada komputer yang dijalankan sistem rumah sakit berbasis web.

Pewawancara : Bagaimana cara penanganan antrian pasien di rumah sakit, Pak ?

Narasumber : Pasien yang datang setiap hari tidaklah menuju satu poli saja. Semua pasien akan terpecah sesuai poli tujuannya masing-masing. Semua pasien akan menunggu giliran di poli sesuai antrian yang sudah ada. Dokter akan menangani pasien sesuai waktu yang mampu ditangani oleh dokter.

Pewawancara : Pak, sesuai dengan yang Bapak sebutkan tadi bahwa pasien BPJS harus melakukan *Finger Print*. Apakah berarti hal ini bahwa sistem itu sudah terhubung secara nasional

Narasumber : Ya, sistem BPJS itu sudah terhubung dengan data BPJS Indonesia. Sebab jika pasien tidak melakukan *Finger Print* namun berobat, biaya pengobatannya tidak akan dibayar oleh BPJS nantinya. Oleh karena itu, sistem harus selalu *online* selama 24 jam dan tidak boleh terputus.

Pewawancara : **Apakah benar jika pasien BPJS tidak akan dipungut biaya pengobatan sedikit pun, Pak ?**

Narasumber : Ya, benar sekali. Biaya pengobatan akan menjadi nol rupiah. Demikian pula dengan biaya obat si pasien. Hal ini berlaku juga walaupun si pasien menderita penyakit yang berat.

Pewawancara : **Apakah penggunaan BPJS di rumah sakit berarti pengurusan suratnya dilakukan langsung ke pusat, Pak ?**

Narasumber : Pengurusan BPJS dilakukan di kantor BPJS-nya sendiri, di luar rumah sakit. Namun pelayanannya dilakukan di rumah sakit. Jadi jika seseorang yang ingin menjadi anggota BPJS, mengurus berkasnya harus di kantor BPJS.

Pewawancara : **Dalam pelaksanaan Tugas Akhir kami ini, kami akan membangun sistem informasi rumah sakit dalam tiga *platform* yaitu web, android, dan desktop. Namun ketiganya akan terintegrasi satu sama lain. Tujuan kami untuk membuatnya dalam android sendiri adalah untuk mempermudah pengaksesan data tanpa harus ke komputer. Jika ada sistem yang seperti itu, apakah kira-kira rumah sakit ini membutuhkannya, Pak ?**

Narasumber : Jika berbicara mengenai kebutuhan, rumah sakit ini pasti membutuhkannya sebab proses pengolahan datanya juga akan lebih cepat. Mengingat juga bahwa teknologi yang semakin berkembang, aplikasi seperti itu sudah banyak digunakan.

Pewawancara : **Pak, apakah benar jika pasien yang datang berkunjung harus membawa data-data seperti Kartu Keluarga dan KTP ? Apakah betul jika semuanya harus dipenuhi ?**

Narasumber : Ya, itu benar. Namun itu harus dipenuhi oleh pasien baru, pasien yang belum pernah melakukan pengobatan ke rumah sakit sebelumnya.

Pewawancara : **Jadi bagaimana jika pasien lama, Pak ?**

Narasumber : Pasien lama tidak perlu menyerahkan data seperti pasien baru. Sebab, data pasien yang lama masih ada tersimpan di sistem dan tidak perlu penambahan data yang sama kembali.

Pewawancara : **Apakah semua dokter harus *stand by* di rumah sakit setiap hari, Pak ?**

Narasumber : Tentu tidak. Dokter di rumah sakit memiliki jadwalnya masing-masing dari rumah sakit. Jadi tidak semua dokter *stand by* di rumah sakit selama 24 jam.

Pewawancara : **Pak, apakah yang terjadi jika ruangan di rumah sakit ini penuh ?**

Apakah harus dilakukan rujukan terhadap pasien ?

Narasumber : Hal itu tergantung pada pasien dan keluarganya. Jika ada persetujuan dari pihak pasien, maka akan dilakukan rujukan. Jika tidak maka pasien harus menunggu di IGD sampai ada kamar kosong.

Pewawancara : **Jika terjadi peristiwa habis stok obat, apakah pasien akan dirujuk ke tempat lain ?**

Narasumber : Itu adalah kebijakan direktur. Sebenarnya sangat jarang terjadi kehabisan stok obat di rumah sakit ini. Sebab semua stok obat dikontrol dengan baik untuk mencegah kehabisan stok.

Pewawancara : **Pak, di rumah sakit ini, jika pasien pertama kali datang ke rumah sakit apakah datanya langsung masuk ke sistem atau masih di tulis terlebih dahulu di kertas ?**

Narasumber : Data yang ada langsung masuk ke sistem tanpa di kertas terlebih dahulu.

Pewawancara : **Bagaimana penyimpanan database dalam sistem yang ada sekarang, Pak ?**

Narasumber : Database akan dihapus dalam waktu sekali lima tahun.

Pewawancara : **Bagaimana cara pengisian hasil pemeriksaan dari pasien, Pak ?**

Narasumber : Pengisian itu dilakukan di masing-masing ruangan oleh dokter maupun perawat. Awalnya dokter maupun pasien memasukkan data dalam kertas, kemudian akan dipindahkan sistem sehingga terlihat keseluruhan data lengkapnya. Dan pada pukul 3 hingga 4 sore, *hardcopy* pemeriksaan pasien akan kembali ke Loket Medical Record.

Pewawancara : **Apakah dokter memiliki biaya tambahan untuk dirinya sendiri, Pak ?**

Narasumber : Bukan biaya tambahan, tapi itu merupakan tarif jasa dokter atas tindakan yang diberikan dokter kepada pasien.

Pewawancara : **Berapa kalikah pasien dikunjungi oleh perawat ataupun dokter setiap harinya, Pak ?**

Narasumber : Jika dokter, waktunya adalah dua kali sehari yaitu pagi dan sore. Namun, pe`rawat tidak ada jam kunjungan karena perawat memang harus stand by di ruangan pasien.

Pewawancara : **Apakah disini menyediakan kartu untuk pasien, Pak ?**

Narasumber : Ya ada, namanya adalah Kartu Berobat.

Lampiran 3 – Desain Antarmuka Aplikasi

Lampiran 3 berikut akan menjelaskan rancangan antarmuka aplikasi yang akan dibangun.

Desain Aplikasi Desktop

Pada bagian ini ditampilkan desain antarmuka yang akan diimplementasikan pada aplikasi *desktop* dan digunakan oleh *user*.

Gambar 42. Desain Aplikasi Desktop – Home Login untuk Pegawai Registrasi

Gambar 44. Desain Aplikasi Desktop – PopUp Simpan Data Pasien

Gambar 43. Desain Aplikasi Desktop – Form Pasien

Gambar 45. Desain Aplikasi Desktop - PopUp Data Pasien Berhasil Disimpan

DATA PASIEN

No Registrasi	0000000	No KTP	0000000000000000	Provinsi	Sumatera Utara
No RM	1111111	Nama Ayah	Boy	Kabupaten	Toba Samosir
Nama Pasien	Dita	Nama Ibu	Rini	Kecamatan	Loguboti
Tempat Lahir	Bolige	Kelompok	Dewasa	Desa	Situluoma
Tanggal Lahir	01/01/1998	Suku Bangsa	Indonesia	Alamat	Jalan Sisingamangaraja
Umur	21	Telepon	081267548390		
Pekerjaan	Mahasiswa	Agama	Kristen Protestan		
Jenis Kelamin	Perempuan				
Status Kawin	Belum Menikah				

Cari Pasien
 No RM Nama Pasien
 search

Tambah **Edit** **Simpan** **Hapus** **Batal** **Keluar**

Gambar 46. Desain Aplikasi Desktop – Edit Data Pasien

DATA PASIEN

No Registrasi	0000000	No KTP	0000000000000000	Provinsi	Sumatera Utara
No RM	1111111	Nama Ayah	Boy	Kabupaten	Toba Samosir
Nama Pasien	Dita	Nama Ibu	Rini	Kecamatan	Loguboti
Tempat Lahir	Bolige	Kelompok	Dewasa	Desa	Situluoma
Tanggal Lahir	01/01/1998	Suku Bangsa	Indonesia	Alamat	Jalan Sisingamangaraja
Umur	21	Telepon	081267548390		
Pekerjaan	Mahasiswa	Agama	Kristen Protestan		
Jenis Kelamin	Perempuan				
Status Kawin	Belum Menikah				

Hapus
Data pasien tidak berhasil dihapus

Cari Pasien
 No RM Nama Pasien
 search

Tambah **Edit** **Simpan** **Hapus** **Batal** **Keluar**

Gambar 48. Desain Aplikasi Desktop – PopUp Gagal Hapus Data Pasien

DATA PASIEN

No Registrasi	0000000	No KTP	0000000000000000	Provinsi	Sumatera Utara
No RM	1111111	Nama Ayah	Boy	Kabupaten	Toba Samosir
Nama Pasien	Dita	Nama Ibu	Rini	Kecamatan	Loguboti
Tempat Lahir	Bolige	Kelompok	Dewasa	Desa	Situluoma
Tanggal Lahir	01/01/1998	Suku Bangsa	Indonesia	Alamat	Jalan Sisingamangaraja
Umur	21	Telepon	081267548390		
Pekerjaan	Mahasiswa	Agama	Kristen Protestan		
Jenis Kelamin	Perempuan				
Status Kawin	Belum Menikah				

Hapus
Apakah Anda yakin ingin menghapus data pasien ini?
 Tidak Ya
 search

Tambah **Edit** **Simpan** **Hapus** **Batal** **Keluar**

Gambar 47. Desain Aplikasi Desktop - PopUp Hapus Data Pasien

DATA PASIEN

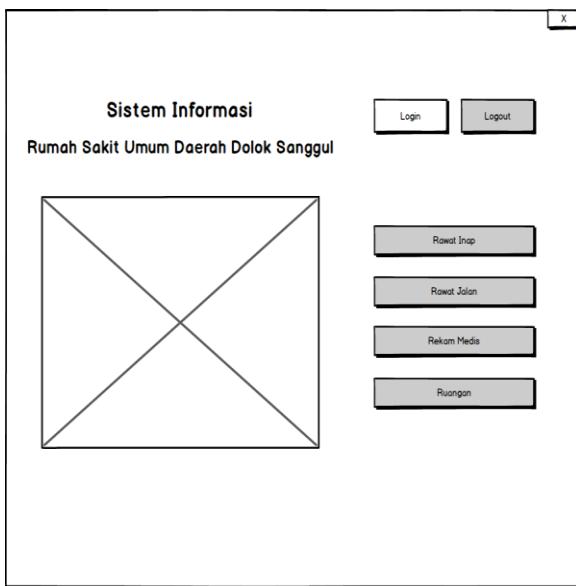
No Registrasi	0000000	No KTP	0000000000000000	Provinsi	Sumatera Utara
No RM	1111111	Nama Ayah	Boy	Kabupaten	Toba Samosir
Nama Pasien	Dita	Nama Ibu	Rini	Kecamatan	Loguboti
Tempat Lahir	Bolige	Kelompok	Dewasa	Desa	Situluoma
Tanggal Lahir	01/01/1998	Suku Bangsa	Indonesia	Alamat	Jalan Sisingamangaraja
Umur	21	Telepon	081267548390		
Pekerjaan	Mahasiswa	Agama	Kristen Protestan		
Jenis Kelamin	Perempuan				
Status Kawin	Belum Menikah				

Hapus
Data pasien berhasil dihapus

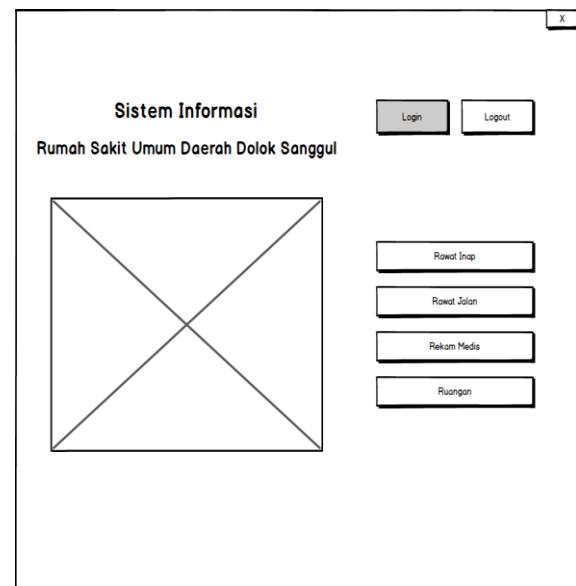
Cari Pasien
 No RM Nama Pasien
 search

Tambah **Edit** **Simpan** **Hapus** **Batal** **Keluar**

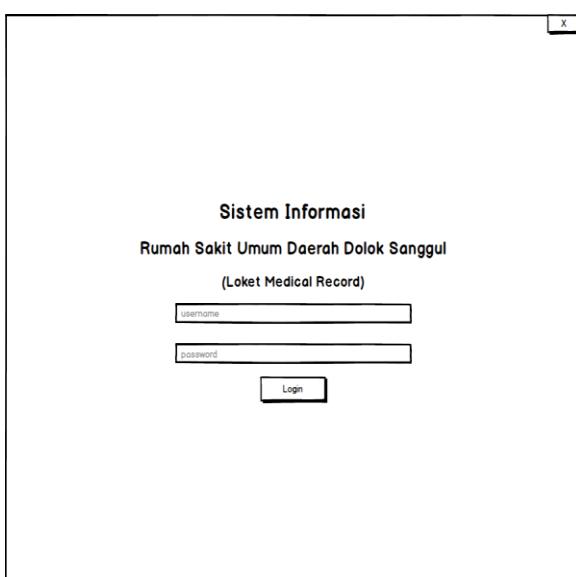
Gambar 49. Desain Aplikasi Desktop - PopUp Berhasil Hapus Data Pasien



Gambar 50. Desain Aplikasi Desktop - Home sebelum login untuk Pegawai Rekam Medis



Gambar 52. Desain Aplikasi Desktop - Home Setelah Login untuk Pegawai Rekam Medis



Gambar 51. Desain Aplikasi Desktop - Login untuk Pegawai Rekam Medis

The screenshot shows a detailed form titled 'RAWAT INAP'. The form includes various input fields and dropdown menus for patient information such as 'No Registrasi', 'Nama Pasien', 'Nama Dokter', 'Asal Rujukan', 'Tanggal Masuk RS', 'Tanggal Keluar RS', 'Jam Masuk', 'Pilih Ruangan', 'Keluhan Waktu Masuk', 'Diagnosa Waktu Masuk', and 'Data Penanggung Jawab' (Name, Profession, Address, Telephone, Gender). It also includes sections for 'Hubungan dengan Posien' (Relationship to Patient) with options like 'Ayah', 'O Suami', 'O Ibu', 'O Istri', 'O Anak', and 'O Pengguru'. At the bottom, there are buttons for 'Tambah', 'Edit', 'Batal', 'Simpan', 'Hapus', and 'Keluar', along with a preview table for registered patients.

Gambar 53. Desain Aplikasi Desktop - Form Rawat Inap

Gambar 54. Desain Aplikasi Desktop - Form Rawat Jalan

X

Sistem Informasi

Rumah Sakit Umum Daerah Dolok Sanggul

Poli : (Nama Poli)

Login

Logout

Antrian Pasien

Rekam Media

Gambar 56. Desain Aplikasi Desktop - Home Sebelum Login untuk Perawat dan Dokter

Gambar 55. Desain Aplikasi Desktop - Data Rekam Medis

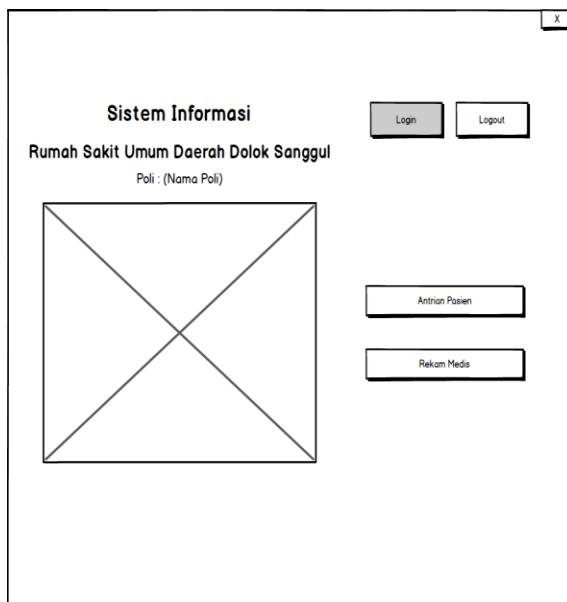
X

Sistem Informasi

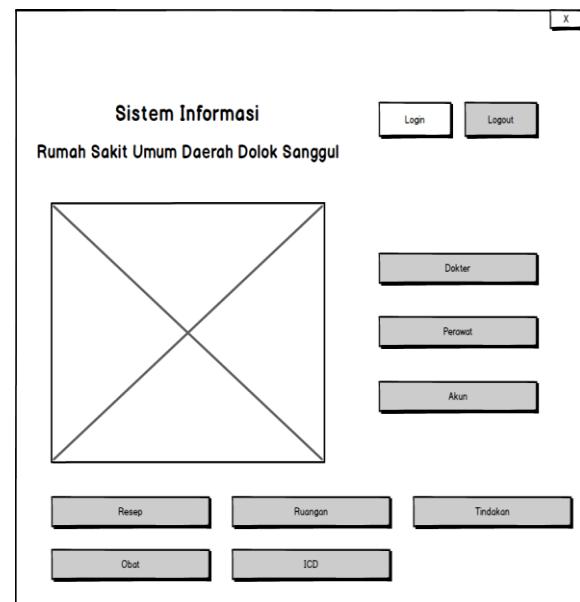
Rumah Sakit Umum Daerah Dolok Sanggul

(Poliklinik)

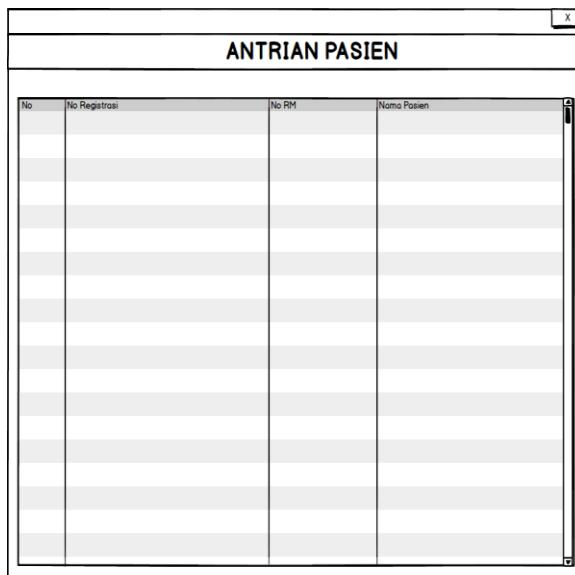
Gambar 57. Desain Aplikasi Desktop - Login untuk Perawat dan Dokter



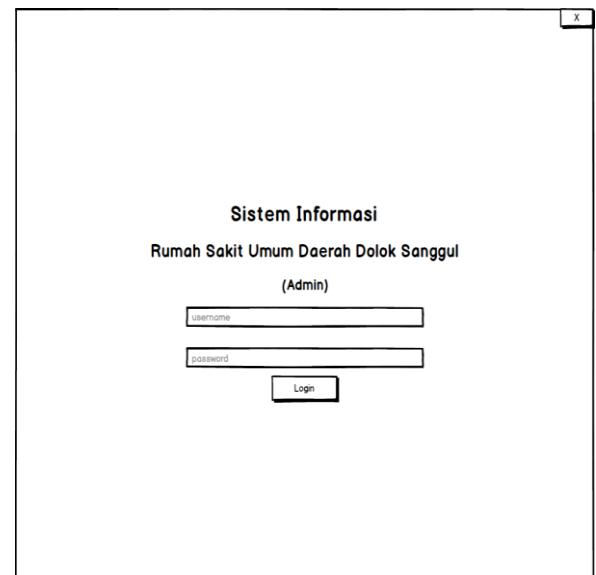
Gambar 58. Desain Aplikasi Desktop - Home Setelah Login untuk Perawat dan Dokter



Gambar 60. Desain Aplikasi Desktop - Home Sebelum Login untuk Admin



Gambar 59. Desain Aplikasi Desktop - Antrian Pasien untuk Rawat Jalan



Gambar 61. Desain Aplikasi Desktop - Login untuk Admin

The screenshot shows the login page of the RSUD Dolok Sanggul Information System. The title "Sistem Informasi Rumah Sakit Umum Daerah Dolok Sanggul" is at the top. A large square graphic with a diagonal cross is on the left. On the right are buttons for "Login" and "Logout". Below the login area are buttons for "Dokter", "Perawat", and "Akun". At the bottom are buttons for "Resep", "Ruangan", "Tindakan", "Obat", and "ICD".

Gambar 62. Desain Aplikasi Desktop - Home Setelah Login untuk Admin

Gambar 64. Desain Aplikasi Desktop - Form ICD

Gambar 63. Desain Aplikasi Desktop - Form Obat

RUANGAN																																							
Nama Ruangan	<input type="text"/>																																						
No Kamar	<input type="text"/>																																						
Kelas	<input type="text"/>																																						
Bed	<input type="text"/>																																						
<input type="button" value="Tambah"/>		<input type="button" value="Edit"/>	<input type="button" value="Batal"/>																																				
<input type="button" value="Simpan"/>		<input type="button" value="Hapus"/>	<input type="button" value="Keluar"/>																																				
<table border="1"> <thead> <tr> <th>Nama Ruangan</th> <th>No Kamar</th> <th>Kelas</th> <th>Bed</th> </tr> </thead> <tbody> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </tbody> </table>				Nama Ruangan	No Kamar	Kelas	Bed																																
Nama Ruangan	No Kamar	Kelas	Bed																																				

Gambar 65. Desain Aplikasi Desktop - Form Ruangan

PERAWAT					
Kode Perawat:	<input type="text"/>				
Nama:	<input type="text"/>				
Jenis Kelamin:	<input type="text"/> <input checked="" type="radio"/>				
Alamat:	<input type="text"/>				
Kota:	<input type="text"/>				
Telepon:	<input type="text"/>				
<input type="button" value="Tambah"/>		<input type="button" value="Edit"/>	<input type="button" value="Batal"/>		
<input type="button" value="Simpan"/>		<input type="button" value="Hapus"/>	<input type="button" value="Keluar"/>		
Kode Perawat:	Nama:	Jenis Kelamin:	Alamat:	Kota:	Telepon:

Gambar 66. Desain Aplikasi Desktop - Form Perawat

Gambar 68. Desain Aplikasi Desktop - Form Data Dokter

Gambar 67. Desain Aplikasi Desktop - Form Akun

Gambar 69. Desain Aplikasi Desktop - Form Tarif Dokter

JADWAL PRAKTEK DOKTER

Nama Dokter:

Hari Praktek:

- Senin
- Rabu
- Jumat
- Minggu
- Selasa
- Kamis
- Sabtu
- Pagi
- Siang
- Sore

Info Jadwal:
(Note: Jadwal Praktek dimulai pukul 10.25 WIB)

Tambah Edit Batal
Simpan Hapus Keluar

Senin	Selasa	Rabu	Kamis	Jumat	Sabtu	Minggu

Gambar 70. Desain Aplikasi Desktop - Form Jadwal Praktek Dokter

TINDAKAN

Kode Tindakan:

Nama Tindakan:

Tambah Edit Batal
Simpan Hapus Keluar

Kode Tindakan	Nama Tindakan

Gambar 72. Desain Aplikasi Desktop - Form Tindakan

CUTI DOKTER

Nama Dokter:

Tanggal Mulai: /

Sampai Tanggal: /

Tambah Edit Batal
Simpan Hapus Keluar

Nama Dokter	Tanggal Mulai	Sampai Tanggal

Gambar 71. Desain Aplikasi Desktop - Form Cuti Dokter

Sistem Informasi
Rumah Sakit Umum Daerah Dolok Sanggul

Login Logout

Gambar 73. Desain Aplikasi Desktop - Home Login untuk Kasir

X

Gambar 74. Desain Aplikasi Desktop - Login Kasir

PEMBAYARAN							
No Transaksi			No RM				
No Periksa			Nama Pasien				
Tanggal Periksa	Tindakan	Obat	Biaya Tambahan	Tarif			
TOTAL XXXXXXXX							
Jumlah Dibayar	Rp			Kembalian	Rp		

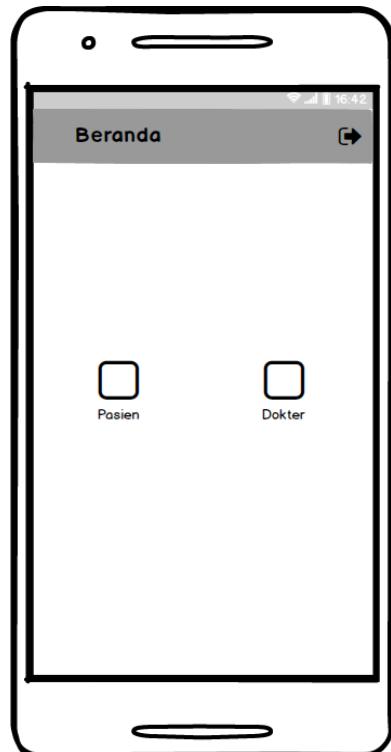
Gambar 75. Desain Aplikasi Desktop - Form Pembayaran

Desain Aplikasi Mobile

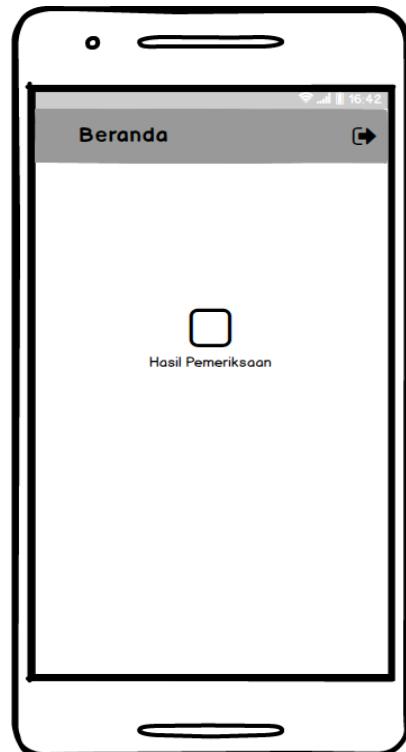
Pada bagian ini ditampilkan desain antarmuka yang akan diimplementasikan pada aplikasi *mobile* dan digunakan oleh dokter dan perawat.



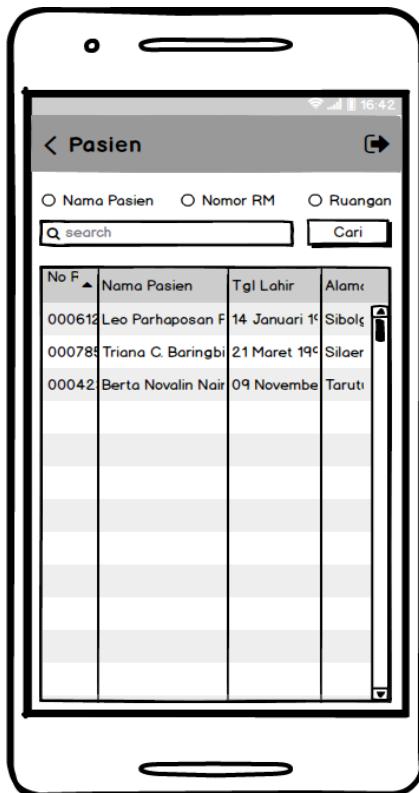
Gambar 76. Desain Aplikasi Mobile - Home



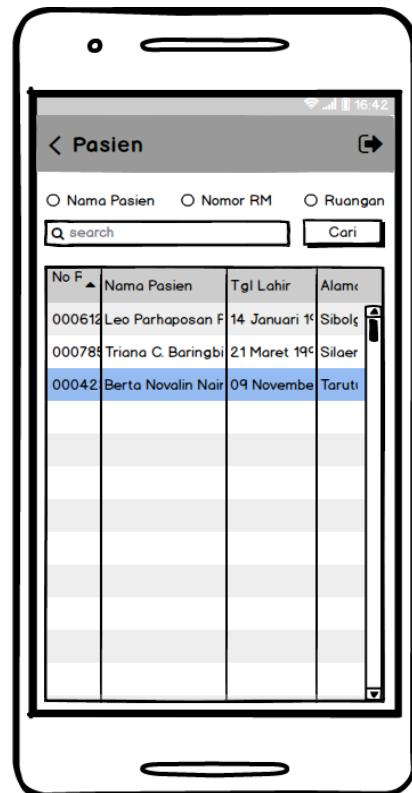
Gambar 77. Desain Aplikasi Mobile – Beranda



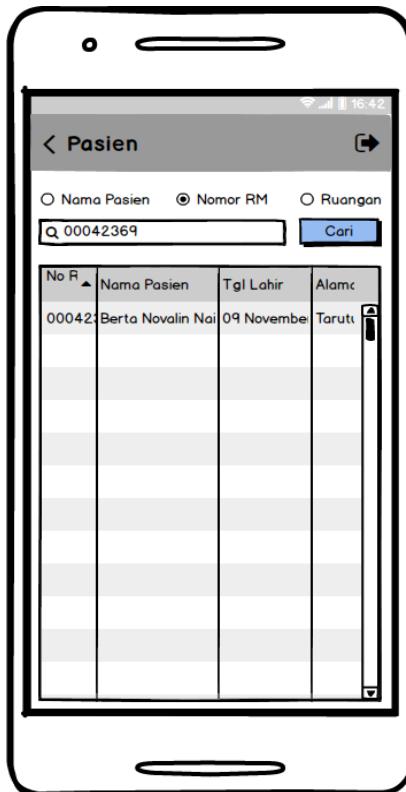
Gambar 78. Desain Aplikasi Mobile - Beranda
Hasil Pemeriksaan



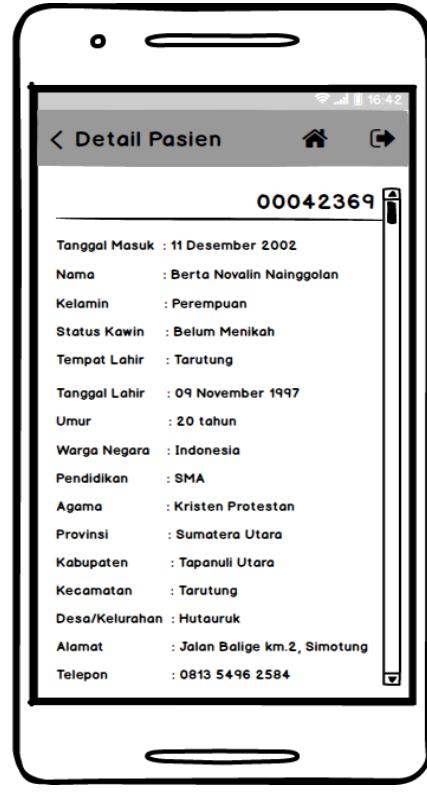
Gambar 79. Desain Aplikasi Mobile – Pasien



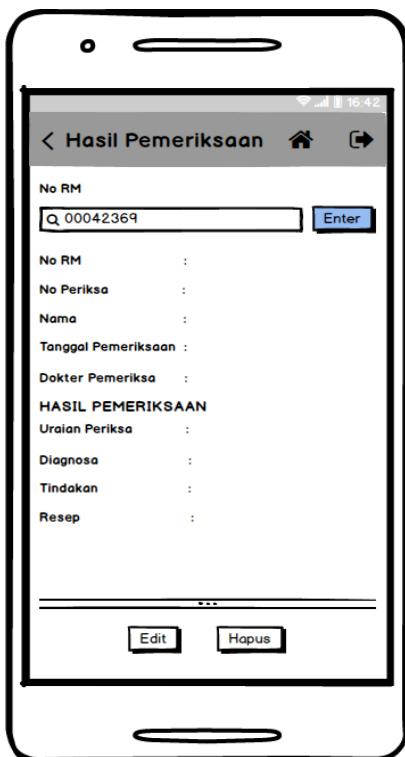
Gambar 81. Desain Aplikasi Mobile – Lihat Detail Pasien



Gambar 80. Desain Aplikasi Mobile - Search Pasien



Gambar 82. Desain Aplikasi Mobile - Detail Pasien



Gambar 83. Desain Aplikasi Mobile - Pemeriksaan



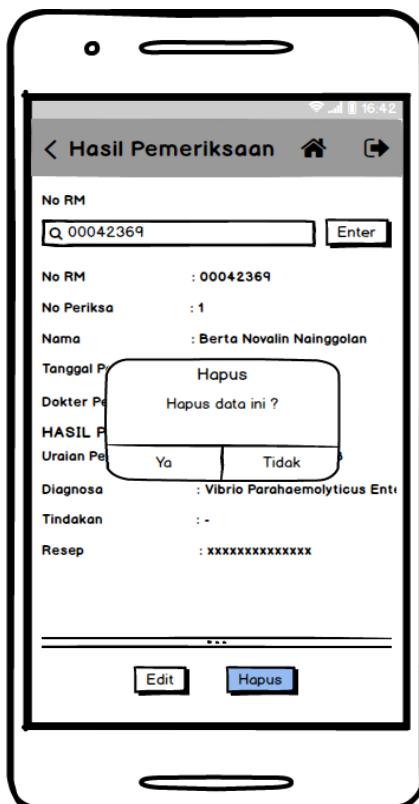
Gambar 85. Desain Aplikasi Mobile - Edit Data Pemeriksaan



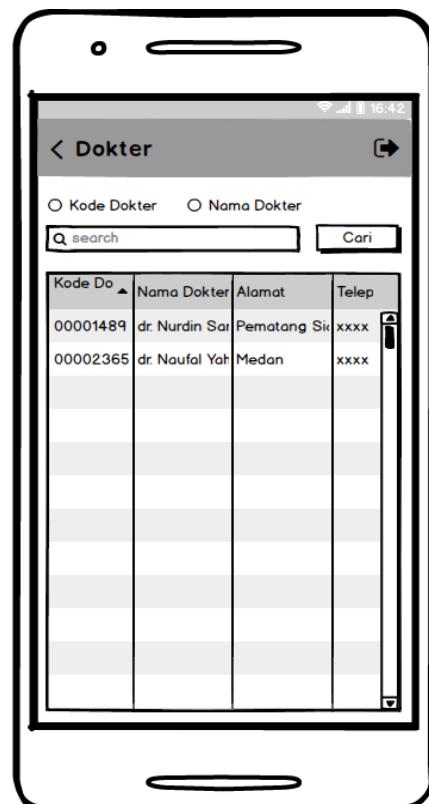
Gambar 84. Desain Aplikasi Mobile - Hasil Pemeriksaan



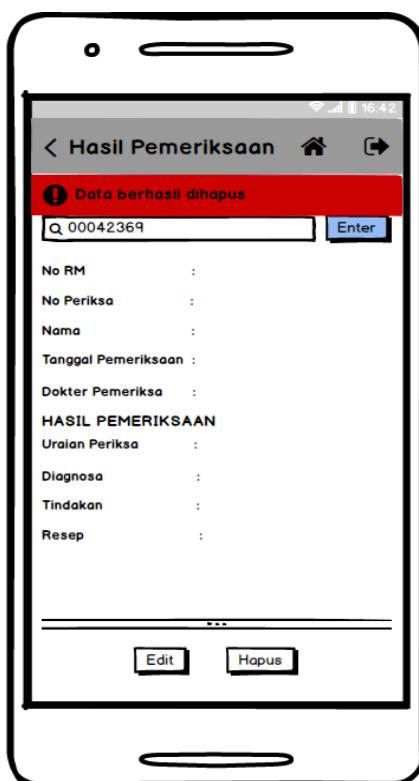
Gambar 86. Desain Aplikasi Mobile - Notifikasi Data Pemeriksaan Berhasil Diubah



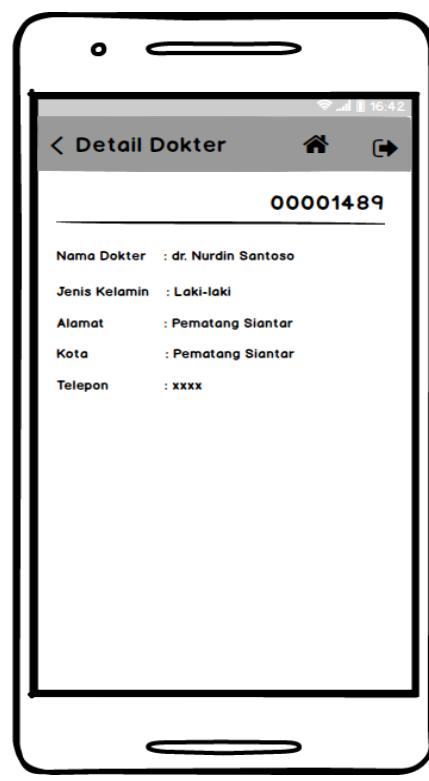
Gambar 87. Desain Aplikasi Mobile - Hapus Hasil Pemeriksaan



Gambar 89. Desain Aplikasi Mobile - Search Dokter

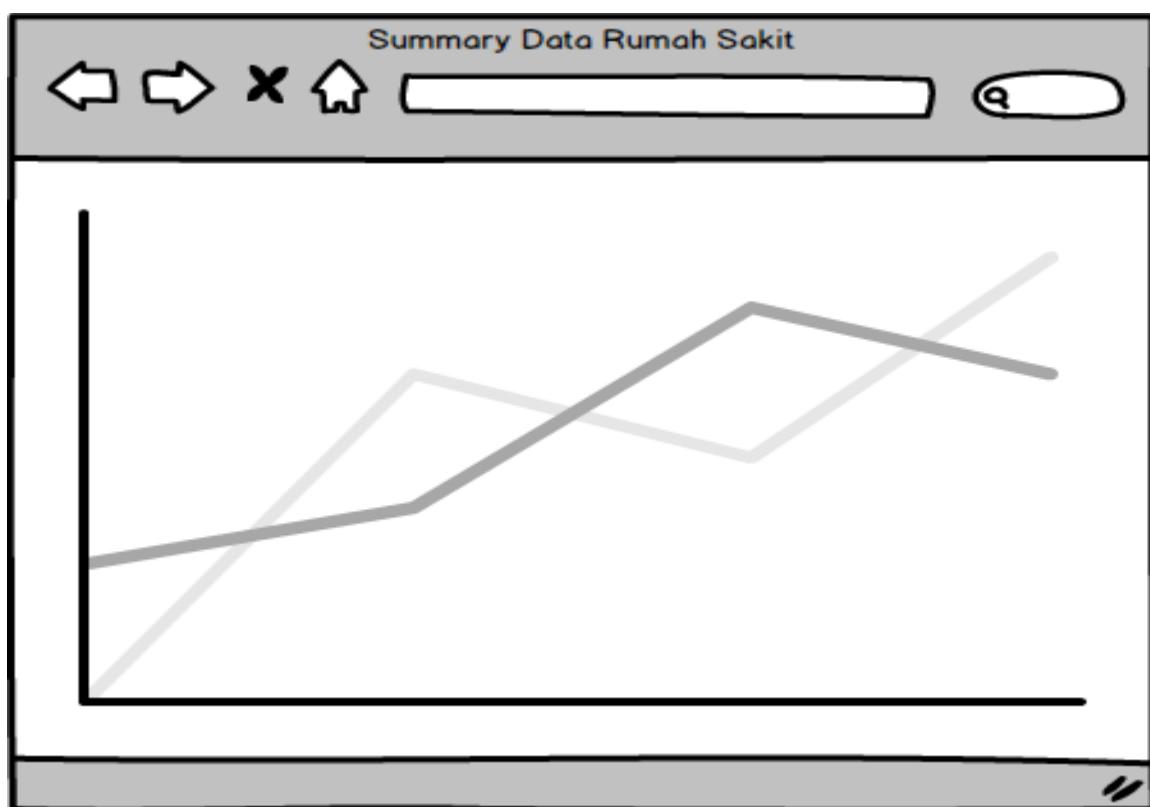


Gambar 88. Desain Aplikasi Mobile - Notifikasi Hasil Pemeriksaan Berhasil Dihapus



Gambar 90. Desain Aplikasi Mobile - Detail Dokter

Desain Aplikasi Web



Lampiran 4 – Hasil Aplikasi

Pada Lampiran 4 ini menampilkan bagaimana hasil aplikasi baik dari versi *desktop*, *mobile* dan *web* dalam bentuk UI aplikasi. Pada aplikasi juga telah dilakukan pengujian terhadap masing-masing fungsi baik dari versi *mobile* maupun *web*. Adapun hasil UI yang dihasilkan adalah seperti berikut.

Antarmuka Desktop

Pada bagian ini akan diuraikan antarmuka Rancang Bangun Sistem Informasi Rumah Sakit versi *desktop*.

1. Tampilan Mengelola Rawat Inap

Tampilan dari mengelola rawat inap seperti menambah, mengedit, menyimpan, menghapus, dan membatalkan dapat dilihat pada gambar berikut.

No Registrasi	No RM	Nama Pasien	Nama Dokter	Nama Perawat	Asal Rujukan	Tanggal Masuk RS	Tanggal Keluar RS	Jam Masuk	Ruangan	Keluhan wkt. Masuk	Diagnosa	Nama Pen.	Pekerjaan	Juliana Sia...	PNS
00957	00750	Immanuel ...	Bernard Si...	Pembina Sl...	RSUD Med...	2018-05-03	2018-05-19	09:48:55	414 II	Sakit perut...	-	Juliana Sia...	PNS		

Gambar 91. Tampilan Rawat Inap pada Desktop

2. Tampilan Mengelola Rawat Jalan

Tampilan dari mengelola rawat jalan seperti menambah, mengedit, menyimpan, menghapus, dan membatalkan dapat dilihat pada gambar berikut.

No Registrasi	No RM	Nama Pasien	Poli	Tanggal Masuk	Tanggal Keluar	Nama dokter	Alamat walli	Pekerjaan walli	Telepon walli	Jenis kelamin	Hubungan dengan pasien	
00513	00552	Sariaman S...	THT	Sore	2018-05-03	2018-05-11	Bambang s...	Pembina Sl...	Wiraswasta	Balige	081299481...	L
											Ayah	

Gambar 92. Tampilan Rawat Jalan pada Desktop

3. Tampilan Mengelola Rekam Medis

Tampilan dari mengelola rekam medis seperti cetak, tambah dan simpan dapat dilihat pada gambar berikut.

No RM	No Registrasi	No Transaksi	Nama Pasien	Tanggal Periksa	Nama Dokter
00750	00357	00458	Immanuel Sit...	2018-05-03	Indra Manik
					Sakit maag
					EGD
					Antimot
					KEMATROVE

Gambar 93. Tampilan Rekam Medis pada Desktop

4. Tampilan Mengelola Ruangan

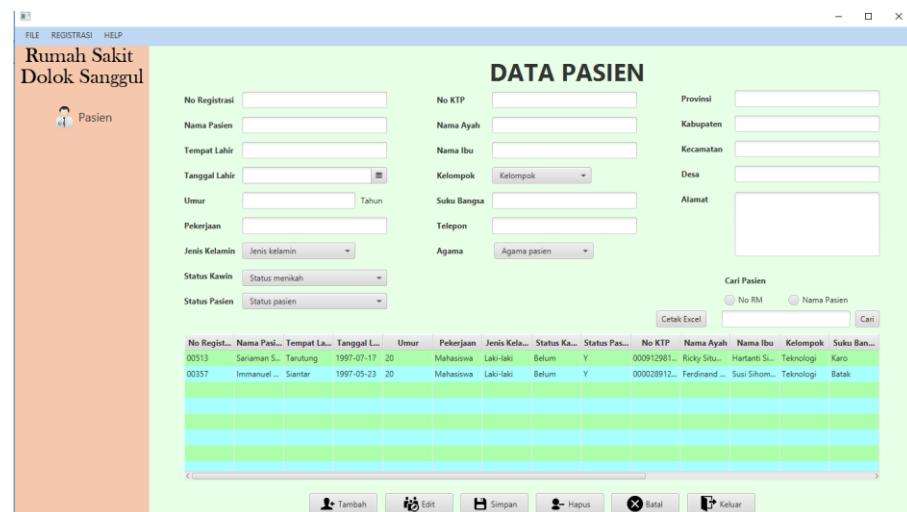
Tampilan mengelola ruangan seperti menambah, mengedit, menyimpan, menghapus, dan membatalkan dapat dilihat pada gambar berikut.



Gambar 94. Tampilan Mengelola Ruangan pada Desktop

5. Tampilan Mengelola Pasien

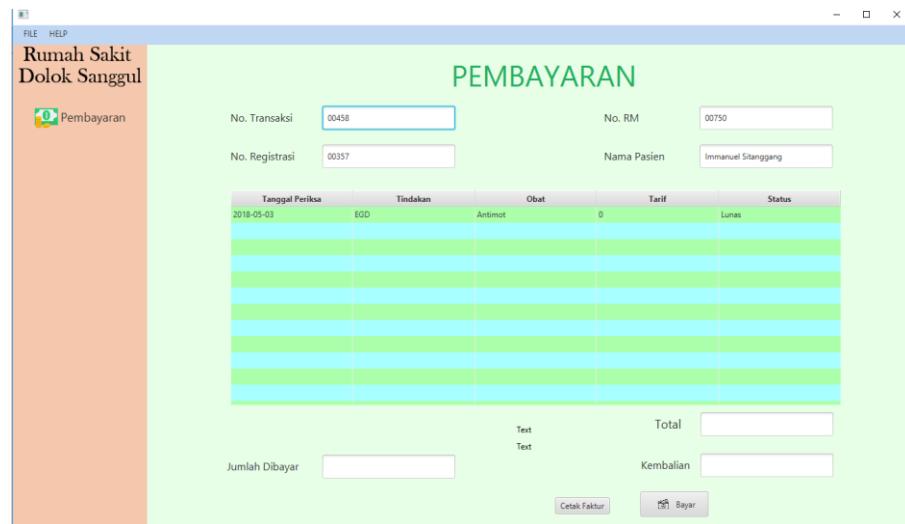
Tampilan mengelola pasien seperti menambah, mengedit, menyimpan, menghapus, dan membatalkan dapat dilihat pada gambar berikut.



Gambar 95. Tampilan Mengelola Pasien pada Desktop

6. Tampilan Mengelola Pembayaran

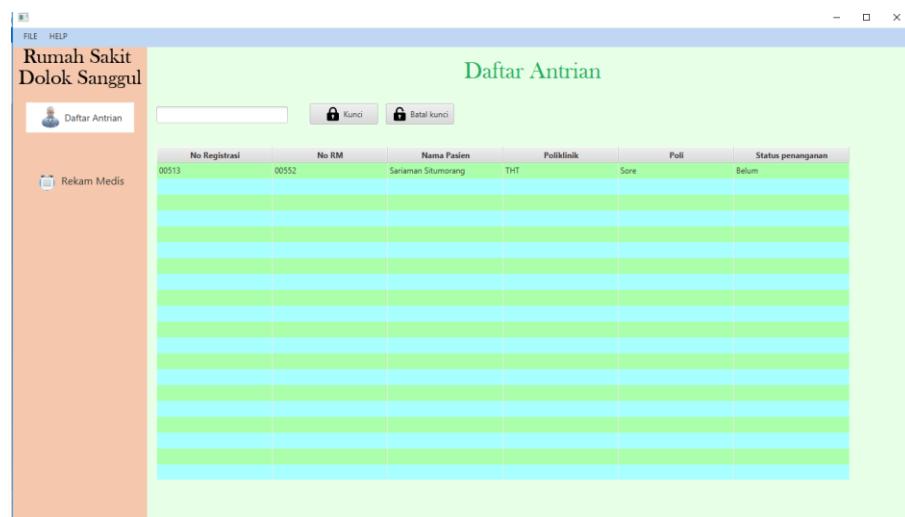
Tampilan mengelola pembayaran dapat dilihat pada gambar berikut.



Gambar 96. Tampilan Mengelola Pembayaran pada Desktop

7. Tampilan Melihat Daftar Antrian

Tampilan melihat daftar antrian berdasarkan poli dapat dilihat pada gambar berikut.



Gambar 97. Tampilan Melihat Daftar Antrian pada Desktop

8. Tampilan Mengelola Dokter

Berikut merupakan tampilan dari mengelola dokter.

The screenshot shows the 'DOKTER' management screen. On the left, there is a sidebar with icons for managing doctors, nurses, drugs, rooms, actions, and diagnoses. The main area has fields for doctor code, name, gender, address, city, phone number, and specialty. Below the fields is a table header with columns for doctor code, name, address, city, phone number, and specialty. A message 'No content in table' is shown. At the bottom are buttons for Add (+), Edit, Save (Simpan), Delete (Hapus), Cancel (Batal), and Exit (Keluar).

Gambar 98. Tampilan Mengelola Dokter

9. Tampilan Mengelola Tarif Dokter

Berikut merupakan tampilan mengelola tarif dokter.

The screenshot shows the 'TARIF DOKTER' management screen. It has fields for doctor code, name, and tariff. Below the fields is a table header with columns for doctor code, name, and tariff. A message 'No content in table' is shown. At the bottom are buttons for Add (+), Edit, Save (Simpan), Delete (Hapus), Cancel (Batal), and Exit (Keluar).

Gambar 99. Tampilan Mengelola Tarif Dokter

10. Tampilan Mengelola Jadwal Praktek Dokter

Berikut merupakan tampilan dari mengelola jadwal praktek dokter.

Gambar 100. Tampilan Mengelola Jadwal Praktek Dokter

11. Tampilan Mengelola Cuti Dokter

Berikut merupakan tampilan dari mengelola cuti dokter.

Gambar 101. Tampilan Mengelola Cuti Dokter

12. Tampilan Mengelola Perawat

Berikut merupakan tampilan mengelola perawat.

Kode...	Nama perawat	Jenis Kelamin	Alamat	Kota	Telepon
0584	Feby	Perempuan	Silaen	Porsea	081200728016
0192	Pembina Siahaan	Laki-Laki	Medan	Medan	029898392423

Gambar 102. Tampilan Mengelola Perawat

13. Tampilan Mengelola Data Obat

Berikut merupakan tampilan dari mengelola data obat.

Kode obat	Nama obat	Jumlah	Harga
O-0004	Antimot	10	100000
O-0008	Antimo	10	30000

Gambar 103. Tampilan Mengelola Data Obat

14. Tampilan Mengelola Tindakan

Berikut merupakan tampilan dari mengelola tindakan.

FILE HELP

Rumah Sakit Dolok Sanggul



Mengelola dokter



Mengelola perawat



Mengelola obat



Mengelola ruangan



Mengelola tindakan



Mengelola diagnosa

TINDAKAN

Kode Tindakan	<input type="text"/>	Nama Tindakan	<input type="text"/>
Harga Tindakan	<input type="text"/>		
Kode tindakan	Nama tindakan	Harga tindakan	
T-0001	Suntik KB	25000	
T-0002	Replantasi Gigi	50000	
T-0003	EGD	80000	

 Tambah
 Edit
 Simpan
 Hapus
 Batal
 Keluar

Gambar 104. Tampilan Mengelola Tindakan

15. Tampilan Mengelola ICD

Berikut merupakan tampilan dari mengelola ICD.

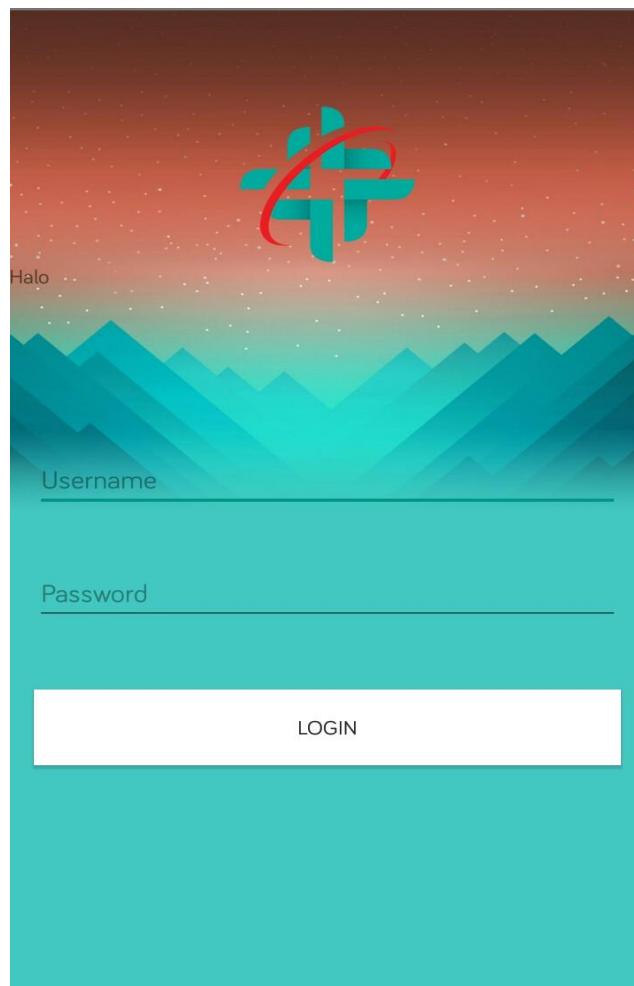
Gambar 105. Tampilan Mengelola ICD

Antarmuka Mobile

Pada bagian ini akan diuraikan antarmuka Rancang Bangun Sistem Informasi Rumah Sakit versi *mobile*.

1. Tampilan *Login*

Tampilan *login* pada *mobile* dapat dilihat pada gambar berikut.



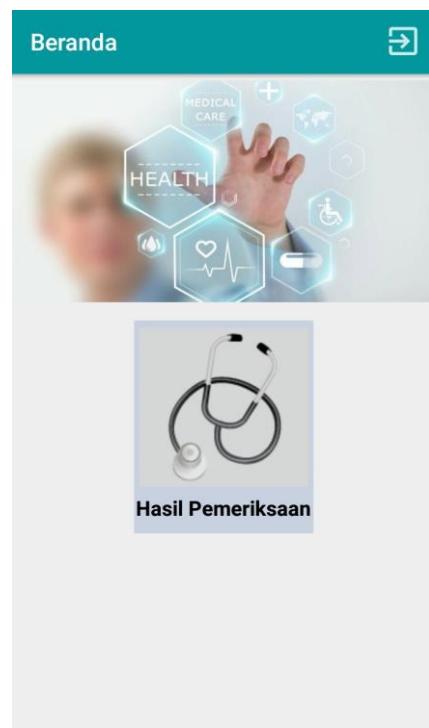
Gambar 106. Tampilan Login

2. Tampilan Halaman Utama

Tampilan halaman utama perawat dan dokter dapat dilihat pada gambar berikut.



Gambar 107. Tampilan Halaman Utama Perawat



Gambar 108. Tampilan Halaman Utama Dokter

3. Tampilan Hasil Pemeriksaan

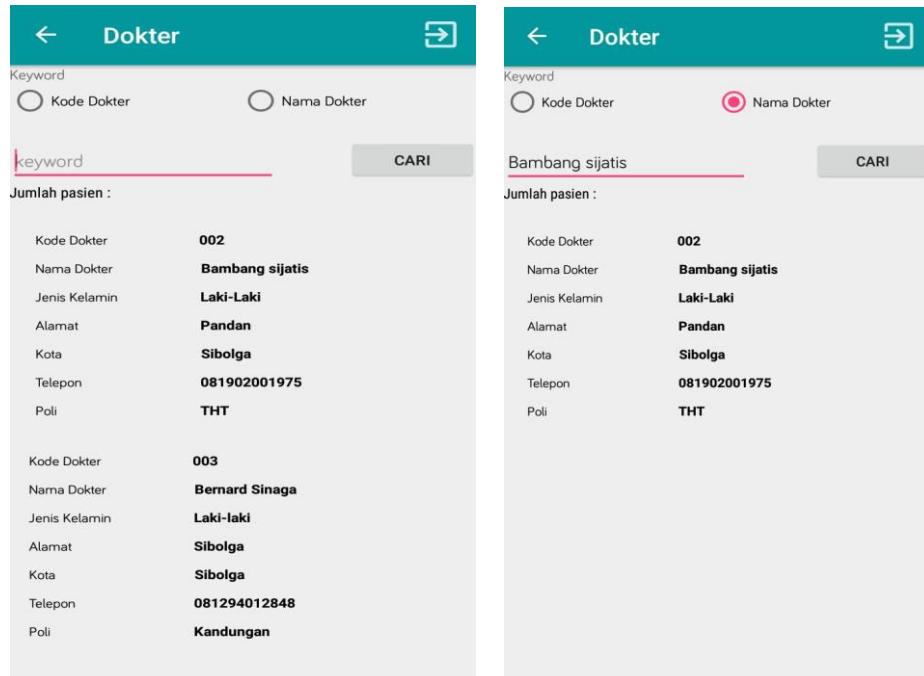
Tampilan hasil pemeriksaan dapat dilihat pada gambar berikut. Pada halaman ini dapat dilakukan pencarian hasil pemeriksaan pasien yaitu berupa nama pasien, nomor RM, dan ruangan. Setelah mengklik salah satu dari data pasien, maka akan muncul detail dari pasien tersebut.

Hasil Pemeriksaan	
No RM	<input type="text" value="321"/>
	<input type="button" value="CARI"/>
No RM	321
No Registrasi	00354
No Transaksi	00581
Nama Pasien	Leo Pakpahan
Tanggal Periksa	2018-04-01
Nama Dokter	Bambang Sijatis
No RM	321
No Registrasi	00355
No Transaksi	00351
Nama Pasien	Leo Pakpahan
Tanggal Periksa	2018-04-02
Nama Dokter	Sry Aritonang

Gambar 109. Tampilan Hasil Pemeriksaan

4. Tampilan Mengelola Dokter

Tampilan mengelola dokter dapat dilihat pada gambar berikut. Pada halaman ini dapat dilakukan pencarian dokter berdasarkan kode dan nama dokter. Setelah salah satu dari data dokter di klik maka akan ditampilkan detail dari dokter tersebut.



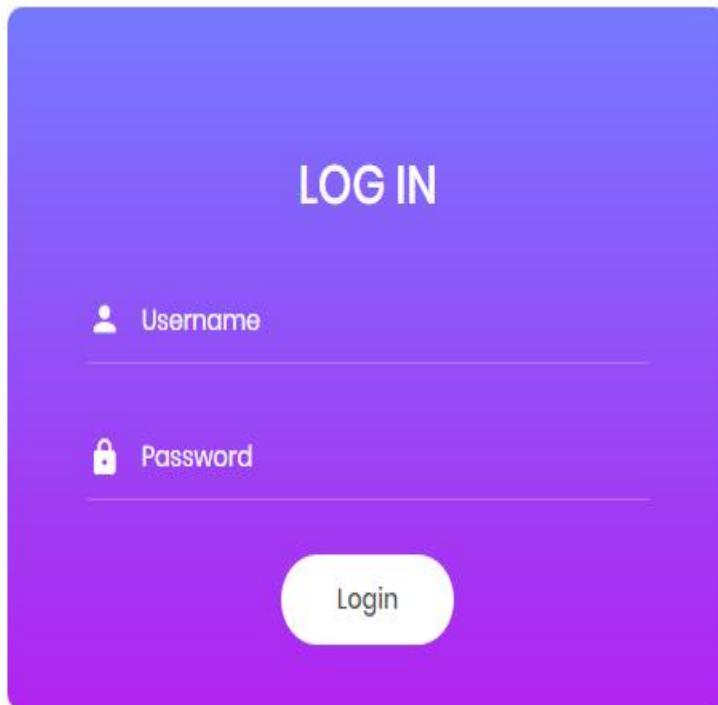
Gambar 110. Tampilan Mengelola Dokter

Antarmuka Web

Pada bagian ini merupakan tampilan Rancang Bangun Sistem Informasi Rumah Sakit versi *web*.

1. Tampilan *Login*

Tampilan *login* pada *web* dapat dilihat pada gambar berikut.



Gambar 111. Tampilan Login pada Web

2. Tampilan dashboard direktur

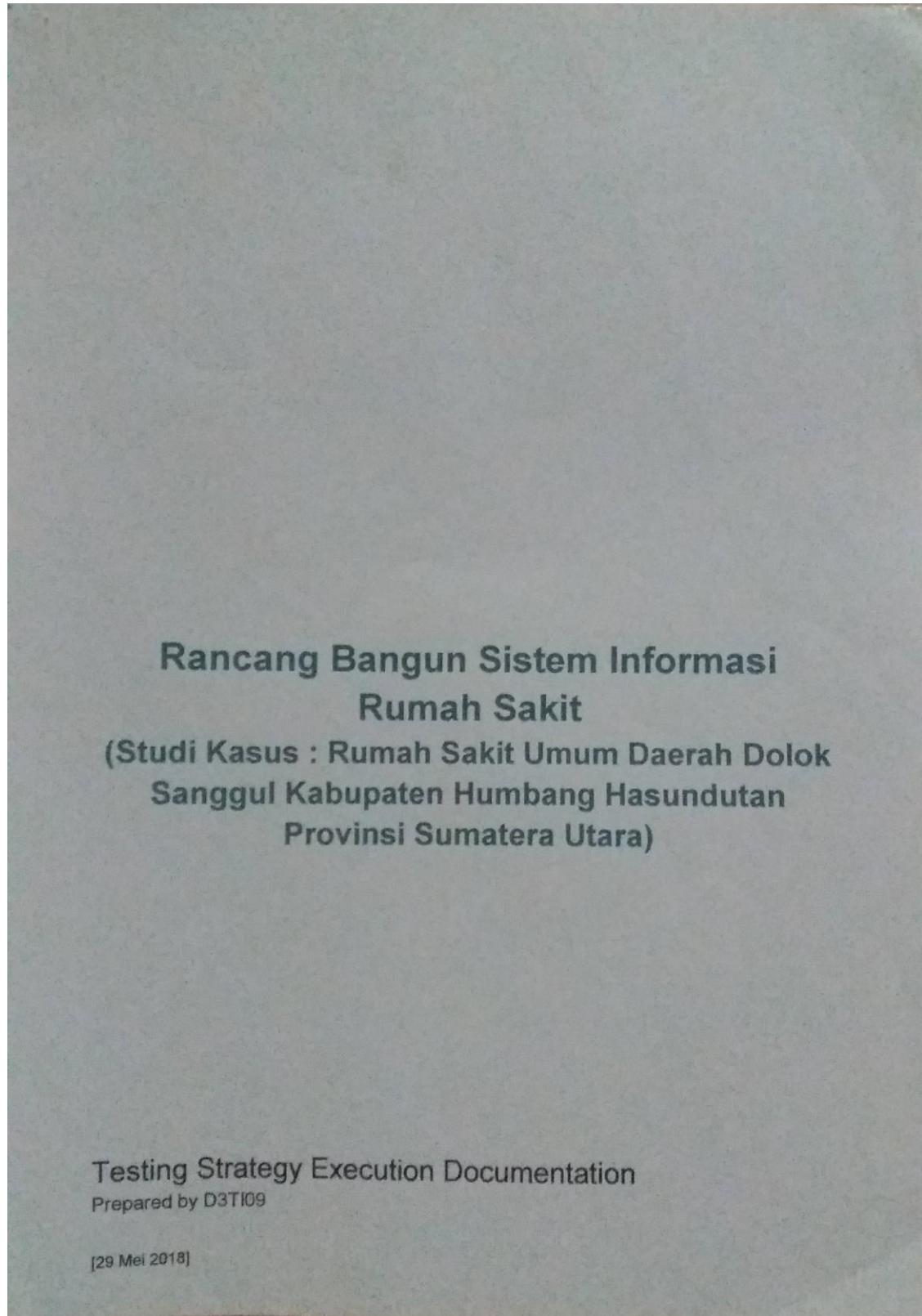
Tampilan dashboard direktur dapat dilihat pada gambar berikut.

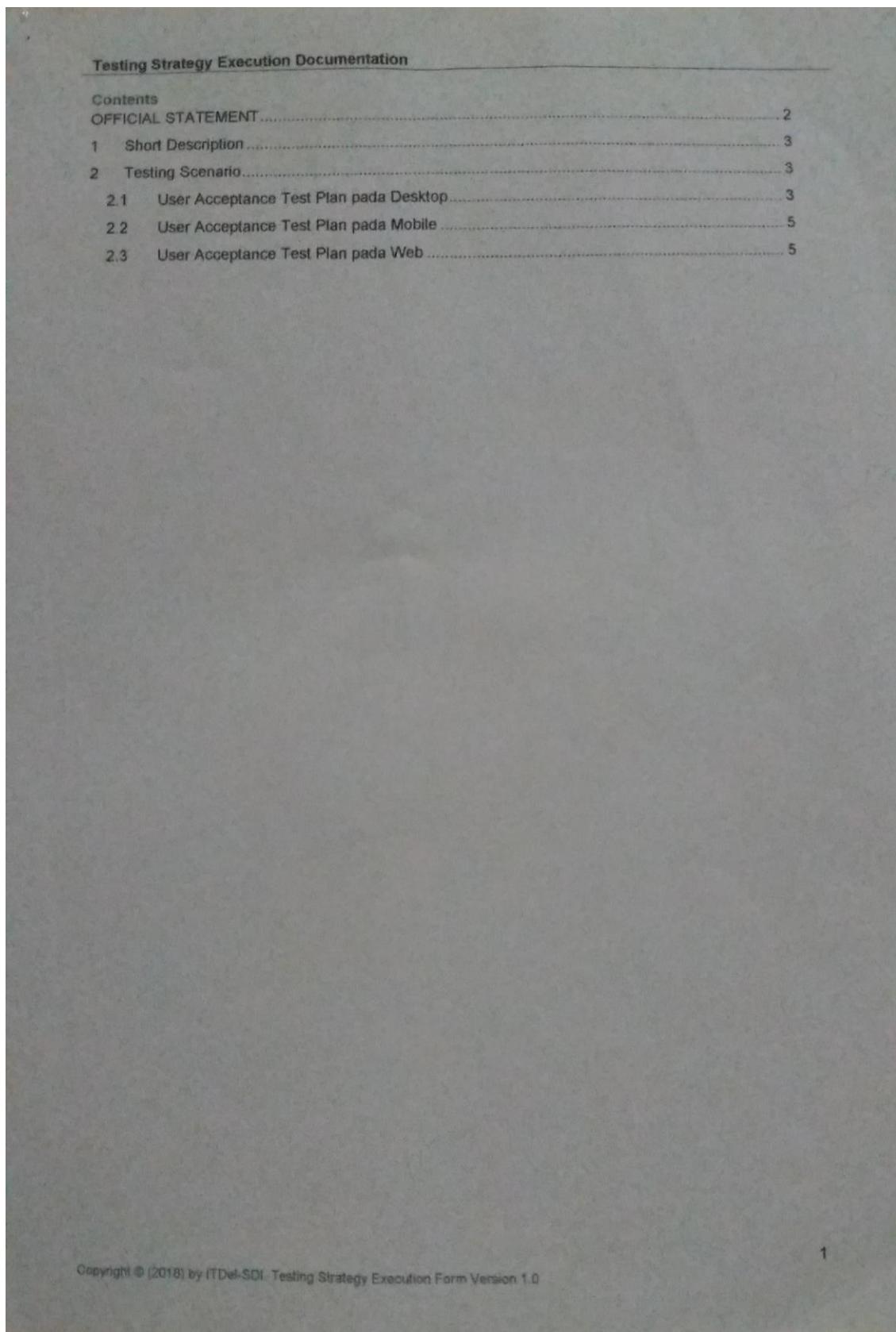


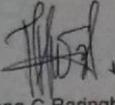
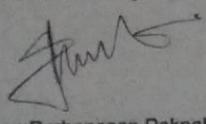
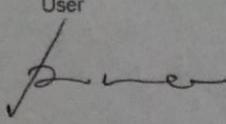
Gambar 112. Tampilan dashboard direktur

Lampiran 5 – Hasil UAT

Pada Lampiran 5 ini menampilkan hasil UAT yang telah dilakukan.





Testing Strategy Execution Documentation			
OFFICIAL STATEMENT			
PROJECT INFORMATION			
Client Name	Robert Daniel Purba	Project Name	Rancang Bangun Sistem Informasi RSUD Dolok Sanggul
Project Phase	User Acceptance Testing	Date	29 Mei 2018
DESCRIPTION			
On 29 May 2018 at Dolok Sanggul, has been conducted User Acceptance Testing for Rancang Bangun Sistem Informasi RSUD Dolok Sanggul.			
The activity has been accomplished by D3TI09, as the party of Del Institute of Technology and witnesses by Robert Daniel Purba as the party of RSUD Dolok Sanggul for Rancang Bangun Sistem Informasi RSUD Dolok Sanggul.			
<p>Note (if any)</p> <p>* Kembangkan dengan model lebih lengkap / Kompleks.</p>			
<p>User Rate</p> <p>95</p>			
This document was made to be used as appropriate.			
ACKNOWLEDGE			
Dolok Sanggul, 29 Mei 2018			
Tim	 (Triana C Baringbing)		
 (Leo Parhaposan Pakpahan)		 (Robert Daniel Purba)	
 (Berta Novalin Nainggolan)			

Testing Strategy Execution Documentation

1 Short Description

Aplikasi yang dibangun adalah Sistem Informasi Rumah Sakit untuk Rumah Sakit Umum Daerah Dolok Sanggul. Sistem informasi ini mencakup tiga modul, yaitu modul registrasi, rekam medis, dan pembayaran. Aplikasi dibangun dalam tiga platform, yaitu desktop, mobile, dan web, dimana ketiganya saling terintegrasi.

2 Testing Scenario

2.1 User Acceptance Test Plan pada Desktop

No.	Description	Expected Result	Criteria
TC-01	Login	User masuk ke sistem	✓
TC-02	Menambah data pasien	Data pasien bertambah	✓
TC-03	Mengedit data pasien	Data pasien dapat diubah	✓
TC-04	Melihat data pasien	Tampil data pasien	✓
TC-05	Menambah data dokter	Data dokter bertambah	✓
TC-06	Mengedit data dokter	Data dokter dapat diubah	✓
TC-07	Melihat data dokter	Tampil data dokter	✓
TC-08	Menambah data ruangan	Data ruangan bertambah	✓
TC-09	Mengedit data ruangan	Data ruangan dapat diubah	✓
TC-10	Melihat data ruangan	Tampil data ruangan	✓
TC-11	Menghapus data ruangan	Data ruangan dapat dihapus	✓
TC-12	Menambah data perawat	Data perawat bertambah	✓
TC-13	Mengedit data perawat	Data perawat dapat diubah	✓
TC-14	Melihat data perawat	Tampil data perawat	✓
TC-15	Menambah data obat	Data obat bertambah	✓
TC-16	Mengedit data obat	Data obat dapat diubah	✓
TC-17	Melihat data obat	Tampil data obat	✓
TC-18	Menghapus data obat	Data obat dapat dihapus	✓
TC-19	Menambah data akun	Data akun bertambah	✓
TC-20	Mengedit data akun	Data akun dapat diubah	✓
TC-21	Melihat data akun	Tampil data akun	✓
TC-22	Menghapus data akun	Data akun dapat dihapus	✓
TC-23	Menambah data diagnosa	Data diagnosa bertambah	✓
TC-24	Mengedit data diagnosa	Data diagnosa dapat diubah	✓
TC-25	Melihat data diagnosa	Tampil data diagnosa	✓

3

Copyright © (2018) by ITDel-SDI. Testing Strategy Execution Form Version 1.0

Testing Strategy Execution Documentation

No.	Description	Expected Result	Criteria
TC-26	Menghapus data diagnosa	Data diagnosa dapat dihapus	✓
TC-27	Menambah data tindakan	Data tindakan bertambah	✓
TC-28	Mengedit data tindakan	Data tindakan dapat diubah	✓
TC-29	Melihat data tindakan	Tampil data tindakan	✓
TC-30	Menghapus data tindakan	Data tindakan dapat dihapus	✓
TC-31	Mengelola data jadwal dokter	Data jadwal dokter dapat dikelola	✓
TC-32	Mengelola data tarif dokter	Data tarif dokter dapat dikelola	✓
TC-33	Mengelola jadwal praktek dokter	Data jadwal praktek dokter dapat dikelola	✓
TC-34	Mengelola data rawat jalan	Data rawat jalan dapat dikelola	✓
TC-35	Mengelola data rawat inap	Data rawat inap dapat dikelola	✓
TC-36	Melihat rekam medis pasien	Tampil data rekam medis pasien	✓
TC-37	Melihat antrian pasien	Tampil antrian pasien	✓
TC-38	Membuat rekam medis	Rekam medis pasien bertambah	✓
TC-39	Mengelola pembayaran	Pembayaran berhasil dikelola	✓
TC-40	Mencetak form pembayaran	Form pembayaran berhasil dicetak	✓
TC-41	Mencetak form rekam medis	Form rekam medis berhasil dicetak	✓
TC-42	Pencarian	Berhasil melakukan pencarian sesuai keyword yang dimasukkan	✓
TC-43	Pencetakan data pasien	Data pasien berhasil dicetak dalam bentuk laporan	✓
TC-44	Pencetakan rawat inap per tahun	Rawat inap per tahun berhasil dicetak	✓
TC-45	Pencetakan rawat jalan per tahun	Rawat jalan per tahun berhasil dicetak	✓

Testing Strategy Execution Documentation
2.2 User Acceptance Test Plan pada Mobile

No.	Description	Expected Result	Criteria
TC-01	Login	User masuk ke sistem	✓
TC-02	Menampilkan data pasien	Tampil data pasien	✓
TC-03	Menampilkan data dokter	Tampil data dokter	✓
TC-04	Pencarian	Menampilkan data sesuai keyword pencarian	✓
TC-05	Menampilkan data rekam medis	Tampil data rekam medis	✓
TC-06	Menambah data rekam medis	Data rekam medis bertambah	✓

2.3 User Acceptance Test Plan pada Web

No.	Description	Expected Result	Criteria
TC-01	Login	User masuk ke sistem	✓
TC-02	Melihat statistika pasien	Tampil data statistika pasien	✓