

**PEMBANGUNAN SISTEM INFORMASI REKAM MEDIS
RAMDANI SKINCARE & SPA MALANG DENGAN METODE
PROTOTYPING BERBASIS WEB**

SKRIPSI

Untuk memenuhi Sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Cindy Cunday Cicimby

NIM:145150207111130



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG**

2021



PENGESAHAN

PEMBANGUNAN SISTEM INFORMASI REKAM MEDIS RAMDANI SKINCARE & SPA MALANG
DENGAN METODE *PROTOTYPING* BERBASIS WEB

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Cindy Cunday Cicimby

NIM: 145150207111130

Skrripsi ini telah diuji dan dinyatakan lulus pada
29 Juli 2021

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing 2


Denny Sagita Rusdianto, S.Kom., M.Kom.

NIP: 19851124 201504 1 001


Randy Cahya Wihandika, S.ST., M.Kom.

NIK: 201405 880206 1 001

Mengetahui

Ketua Jurusan Teknik Informatika




Achmad Basuki, S.T., M.MG., Ph.D.

NIP: 19741118 200312 1 002

PERNYATAAN ORISINALITAS

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

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

Malang, 29 Juli 2021



Cindy Cunday Cicimby
NIM: 145150207111130

KATA PENGANTAR

Segala puji atas kehadiran Tuhan YME atas izin dan rahmat-Nya sehingga laporan skripsi yang berjudul “Pembangunan Sistem Informasi Rekam Medis Ramdani *Skincare & Spa* Malang dengan Metode *Prototyping* Berbasis Web” ini dapat terselesaikan. Perlu disadari bahwa laporan skripsi ini tidak akan berjalan dengan semestinya tanpa ada dukungan dari beberapa pihak. Oleh karena itu, akan disampaikan beberapa rasa hormat dan terima kasih kepada:

1. Kedua orang tua, adik-adik, serta keluarga besar atas doa dan dukungan yang diberikan selama proses studi hingga saat ini.
2. Bapak Denny Sagita Rusdianto, S.Kom., M.Kom sebagai dosen pembimbing I dan Bapak Randy Cahya Wihandika, S.ST., M.Kom selaku dosen pembimbing II yang telah mendampingi dan mengarahkan penulis dengan sabar dalam menyelesaikan skripsi ini dan juga telah memberikan masukan, saran dan ilmu kepada penulis.
3. Bapak Achmad Basuki, S.T., M.MG., Ph.D., selaku Ketua Jurusan Teknik Informatika dan Bapak Adhitya Bhawiyuga S.Kom., M.Sc., selaku Kepala Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Brawijaya.
4. Muhammad Jaka Dzulfikar yang telah membantu memberikan dukungan tiada henti dan mendukung proses pengerjaan skripsi ini.
5. Segenap civitas akademika Fakultas Ilmu Komputer yang membantu kelancaran penyusunan skripsi penulis.

Dengan segala kerendahan hati, penulis menyadari bahwa penyusunan skripsi ini jauh dari kata sempurna. Sehingga penulis mengharapkan kritik dan saran yang membangun untuk memperbaiki laporan skripsi ini dan semoga laporan skripsi ini dapat bermanfaat serta dapat memberikan informasi bagi kita semua.

Malang, 29 Juli 2021

Penulis

Cindy Cunday Cicimby

cindycmby@student.ub.ac.id

ABSTRAK

Cindy Cunday Cicimby, Pembangunan Sistem Informasi Rekam Medis Ramdani Skincare & Spa Malang dengan Metode Prototyping Berbasis Web

Pembimbing: Denny Sagita Rusdianto, S.Kom., M.Kom. dan Randy Cahya Wihandika, S.ST., M.Kom.

Ramdani Skincare & Spa ialah salah satu klinik kecantikan yang berada di Kota Malang, pengelolaan rekam medis di klinik kecantikan ini dipegang oleh Admin dan Dokter yang meliputi input data pendaftaran pasien, pendataan riwayat penyakit pasien hingga mengirimkan laporan kepada direktur utama, dan juga Direktur yang mengelola Admin dan Dokter. Permasalahan ditemukan pada proses pengelolaan rekam medis di Ramdani Skincare & Spa yang dilakukan secara manual menggunakan *hardcopy*, permasalahan yang menyebabkan pengarsipan data diri pasien tidak efektif dan tidak adanya keberlanjutan dari riwayat kunjungan pasien. Masalah lain yang ditimbulkan yaitu proses pelayanan dari mulai pendaftaran hingga pelaporan klinik sangat lambat, hal tersebut memicu ketidakpuasan pelanggan. Dari penjabaran permasalahan yang diperoleh, maka dibuatlah sebuah aplikasi *web base* agar pengelolaan rekam medis berjalan dengan cepat, tepat, dan akurat. Sistem rekam medis berbasis web adalah sebuah sistem yang memudahkan pelayanan, pencatatan, dan pengarsipan data diri pasien di sebuah layanan kesehatan guna mengurangi kesalahan yang sering terjadi. Pengembangan sistem di Ramdani Skincare & Spa Malang menggunakan metode *prototyping* yang meliputi tahapan analisis kebutuhan, implementasi, serta pengujian. Tahap analisis kebutuhan diperoleh empat puluh empat kebutuhan fungsional dari satu kali iterasi. PHP merupakan bahasa pemrograman yang digunakan saat tahap implementasi pada kerangka kerja *CodeIgniter*. Tahap pengujian meliputi pengujian fungsional serta non fungsional. Pengujian fungsional meliputi uji unit yang berhasil melakukan jalur *testing* berdasarkan tiga kasus uji dan uji *validation* menunjukkan 100% valid dari tujuh puluh tiga kasus uji. Pengujian non fungsional melalui *compability* menunjukkan hasil sistem dapat diterima oleh pengguna dan dapat berjalan baik pada *web browser*.

Kata kunci: Ramdani Skincare & Spa, Rekam Medis, Pengembangan sistem, Prototipe, Website

ABSTRACT

Cindy Cunday Cicimby, *Development of Ramdani Skincare & Spa Malang Medical Record Information System with a Web-based Prototyping Method*

Advisors: Denny Sagita Rusdianto, S.Kom., M.Kom. dan Randy Cahya Wihandika, S.ST., M.Kom.

Ramdani Skincare & Spa is one of the beauty clinics in Malang City, the management of medical records at this beauty clinic is held by the Admin and the Doctor which includes inputting patient registration data, collecting patient history data, and sending reports to the main director, and also the Director who manages Admins dan Doctors. Problems were found in the medical record management process at Ramdani Skincare & Spa which was done manually using hardcopy, causing ineffective patient data archiving and the absence of continuity of patient visit history. Another problem that arises is that the service process from registration to clinical reporting is very slow, this triggers customer dissatisfaction. The medical record information system is a system that facilitates the service, recording, and archiving of patient personal data in a health service in order to reduce errors that often occur. From the problems, a web-based application was made so that the management of medical records runs quickly, precisely and accurately. System development at Ramdani Skincare & Spa Malang using the prototyping method. System development includes the stages of needs analysis, implementation, and testing. The requirements analysis stage gets forty-four functional requirements obtained from one iteration. Implementation stage uses PHP programming language CodeIgniter framework. Testing stages include functional and non-functional testing. Functional testing includes unit testing that successfully performs a test path based on three test cases and validation testing produces one hundred percent valid from seventy-three cases. Non-functional testing using compatibility results can be accepted by users and can run well on the browser.

Keywords: *Ramdani Skincare & Spa, Medical Record, System Development, Prototyping, Web*

DAFTAR ISI

PEMBANGUNAN SISTEM INFORMASI REKAM MEDIS RAMDANI SKINCARE & SPA MALANG DENGAN METODE <i>PROTOTYPING</i> BERBASIS <i>WEB</i>	i
PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS.....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN.....	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Manfaat.....	2
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN.....	4
2.1 Kajian Pustaka.....	4
2.2 Perbedaan Penelitian Terdahulu dan Sekarang.....	5
2.3 <i>Ramdani Skincare & Spa</i>	7
2.4 Sistem Rekam Medis.....	7
2.5 Proses Bisnis <i>Ramdani Skincare & Spa</i>	8
2.6 <i>PHP</i>	12
2.7 <i>Database MySQL</i>	12
2.7.1 <i>Database</i>	12
2.7.2 <i>MySQL</i>	12
2.8 <i>Framework CodeIgniter</i>	12
2.8.1 <i>CodeIgniter</i>	13



2.9 Prototype	13
2.10 Unified Modelling Language (UML).....	14
2.10.1 Use Case Diagram	14
2.10.2 Sequence Diagram	15
2.10.3 Class Diagram	15
2.11 Pengujian Perangkat Lunak.....	16
2.11.1 Desain Test Case	16
BAB 3 METODOLOGI PENELITIAN	19
3.1 Studi Pustaka	19
3.2 Analisis Kebutuhan Sistem.....	20
3.3 Perancangan Sistem	20
3.4 Pembangunan Prototipe.....	21
3.5 Evaluasi Prototipe	21
3.6 Implementasi Sistem	21
3.7 Pengujian Sistem	21
3.8 Penarikan Kesimpulan dan Saran	22
BAB 4 ANALISIS KEBUTUHAN	23
4.1 Gambaran Umum Sistem.....	23
4.2 Elisitasi Kebutuhan	23
4.3 Identifikasi Aktor	23
4.4 Daftar Kebutuhan	24
4.4.1 Kebutuhan Fungsional	24
4.4.2 Kebutuhan Non Fungsional.....	30
4.5 Use Case.....	31
4.5.1 Diagram Use Case	31
4.5.2 Skenario Use Case	31
BAB 5 PERANCANGAN DAN IMPLEMENTASI	47
5.1 Perancangan Sistem	47
5.1.1 Perancangan Sequence Diagram	47
5.1.2 Perancangan Class Diagram.....	49
5.1.3 Perancangan ERD (Entity Relationship Diagram).....	50
5.1.4 Perancangan Data	51

5.1.5 Perancangan Algoritme.....	54
5.1.6 Perancangan Antarmuka.....	55
5.2 Implementasi Sistem	59
5.2.1 Spesifikasi Sistem	59
5.3 Implementasi Algoritme	60
5.3.1 Implementasi Algoritme Menambah Data Rekam Medis.....	60
5.3.2 Implementasi Algoritme Menambah Pasien.....	61
5.3.3 Implementasi Algoritme Menghapus Admin	61
5.4 Implementasi Antarmuka	62
5.4.1 Implementasi Antarmuka Menambah Data Rekam Medis	62
5.4.2 Implementasi Antarmuka Menambah Pasien.....	63
5.4.3 Implementasi Antarmuka Menghapus Admin	64
BAB 6 PENGUJIAN	65
6.1 Pengujian Unit	65
6.1.1 Pengujian Unit Menambah Data Rekam Medis	65
6.1.2 Pengujian Unit Menambah Pasien.....	67
6.1.3 Pengujian Unit Menghapus Admin	68
6.2 Pengujian Integrasi	70
6.2.1 Pengujian Integrasi Method Add_Rekam_Medis() dengan Method Add_Rekam_Medis(\$id_pasien, \$id_doketer, \$rekam_medis, \$tanggal)	71
6.3 Pengujian Validasi.....	73
6.4 Pengujian <i>Compatibility</i>	100
BAB 7 PENUTUP	101
7.1 Kesimpulan	101
7.2 Saran	101
DAFTAR REFERENSI.....	102
LAMPIRAN.....	103
A.1 FORMULIR PENDAFTARAN	103
A.2 DATA REKAM MEDIS PASIEN 1	104
A.3 DATA REKAM MEDIS PASIEN 2	105
A.4 DATA REKAM MEDIS PASIEN 3	106

DAFTAR TABEL

Tabel 2.1 Tabel Perbedaan Penelitian Terdahulu dan Sekarang	5
Tabel 2.2 Tabel Simbol pada Diagram Proses Bisnis Ramdani Skincare	8
Tabel 2.3 Tabel Nilai Kompleksitas	17
Tabel 4.1 Spesifikasi Kebutuhan Fungsional Admin Iterasi Pertama	24
Tabel 4.2 Spesifikasi Kebutuhan Fungsional Dokter Iterasi Pertama	25
Tabel 4.3 Spesifikasi Kebutuhan Fungsional Direktur Iterasi Pertama	26
Tabel 4.4 Spesifikasi Kebutuhan Fungsional Admin Iterasi Kedua	27
Tabel 4.5 Spesifikasi Kebutuhan Fungsional Dokter Iterasi Kedua	28
Tabel 4.6 Spesifikasi Kebutuhan Fungsional Direktur Iterasi Kedua	29
Tabel 4.7 Spesifikasi Kebutuhan Non Fungsional Sistem	30
Tabel 4.8 Skenario <i>Use Case Login</i>	32
Tabel 4.9 Skenario <i>Use Case Logout</i>	32
Tabel 4.10 Skenario <i>Use Case Mengubah Profil</i>	33
Tabel 4.11 Skenario <i>Use Case Mengirim Laporan</i>	33
Tabel 4.12 Skenario <i>Use Case Menambah Data Pasien</i>	34
Tabel 4.13 Skenario <i>Use Case Mengubah Data Pasien</i>	34
Tabel 4.14 Skenario <i>Use Case Mencari Data Pasien</i>	35
Tabel 4.15 Skenario <i>Use Case Menghapus Data Pasien</i>	35
Tabel 4.16 Skenario <i>Use Case Melihat Data Pasien</i>	36
Tabel 4.17 Skenario <i>Use Case Melihat Profil</i>	36
Tabel 4.18 Skenario <i>Use Case Menambah Data Rekam Medis</i>	36
Tabel 4.19 Skenario <i>Use Case Mencari Data Rekam Medis</i>	37
Tabel 4.20 Skenario <i>Use Case Mengubah Data Rekam Medis</i>	38
Tabel 4.21 Skenario <i>Use Case Menghapus Data Rekam Medis</i>	38
Tabel 4.22 Skenario <i>Use Case Melihat Data Rekam Medis</i>	39
Tabel 4.23 Skenario <i>Use Case Melihat Masa Berlaku SIP</i>	39
Tabel 4.24 Skenario <i>Use Case Mengunduh Laporan Admin</i>	39
Tabel 4.25 Skenario <i>Use Case Mengunduh Laporan Dokter</i>	40
Tabel 4.26 Skenario <i>Use Case Menghapus Laporan Admin</i>	40
Tabel 4.27 Skenario <i>Use Case Menghapus Laporan Dokter</i>	41



Tabel 4.28 Skenario <i>Use Case</i> Melihat Laporan <i>Admin</i>	41
Tabel 4.29 Skenario <i>Use Case</i> Melihat Laporan Dokter.....	41
Tabel 4.30 Skenario <i>Use Case</i> Menambah <i>Admin</i>	42
Tabel 4.31 Skenario <i>Use Case</i> Menambah Dokter.....	42
Tabel 4.32 Skenario <i>Use Case</i> Menghapus <i>Admin</i>	43
Tabel 4.33 Skenario <i>Use Case</i> Menghapus Dokter.....	43
Tabel 4.34 Skenario <i>Use Case</i> Melihat Masa Berlaku SIP Dokter	43
Tabel 4.35 Skenario <i>Use Case</i> Mengubah <i>Admin</i>	44
Tabel 4.36 Skenario <i>Use Case</i> Mengubah Dokter.....	44
Tabel 4.37 Skenario <i>Use Case</i> Melihat Daftar <i>Admin</i>	45
Tabel 4.38 Skenario <i>Use Case</i> Melihat Daftar Dokter.....	45
Tabel 4.39 Skenario <i>Use Case</i> Mencari <i>Admin</i>	45
Tabel 4.40 Skenario <i>Use Case</i> Mencari Dokter.....	46
Tabel 5.1 Tabel <i>Admin</i>	52
Tabel 5.2 Tabel Pasien.....	52
Tabel 5.3 Tabel Rekam Medis.....	53
Tabel 5.4 Tabel Direktur.....	53
Tabel 5.5 Tabel Dokter.....	53
Tabel 5.6 Tabel Laporan.....	53
Tabel 5.7 Tabel Algoritme Menambah Data Rekam Medis	54
Tabel 5.8 Tabel Algoritme Menambah Pasien.....	54
Tabel 5.9 Tabel Algoritme Menghapus <i>Admin</i>	55
Tabel 5.10 Tabel Penjelasan Perancangan Antarmuka Halaman Menambah Data Rekam Medis	56
Tabel 5.11 Tabel Penjelasan Perancangan Antarmuka Halaman Menambah Pasien.....	57
Tabel 5.12 Tabel Penjelasan Perancangan Antarmuka Halaman <i>Delete Admin</i> ...	58
Tabel 5.13 Tabel Spesifikasi Perangkat Keras.....	59
Tabel 5.14 Tabel Spesifikasi Perangkat Lunak.....	59
Tabel 6.1 Pembentukan Node Algoritme Fungsi <i>Add_Rekam_Medis()</i>	65
Tabel 6.2 Pengujian Unit Fungsi <i>Add_Rekam_Medis()</i>	66
Tabel 6.3 Pembentukan Node Algoritme fungsi <i>Add_Pasien()</i>	67

Tabel 6.4 Pengujian Unit Fungsi Add_Pasien()	68
Tabel 6.5 Pembentukan Node Algoritme fungsi Delete_Admin().....	69
Tabel 6.6 Pengujian Unit Fungsi Delete_Admin()	70
Tabel 6.7 Pembentukan Node Algoritme Fungsi Add_Rekam_Medis() Kelas Dokter_Controller.....	71
Tabel 6.8 Pembentukan Node Algoritme Fungsi Add_Rekam_Medis() Kelas Model_Rekam_Medis.....	71
Tabel 6.9 Pengujian Unit Fungsi Add_Rekam_Medis()	72
Tabel 6.10 Kasus Uji Login Admin	73
Tabel 6.11 Kasus Uji Login Admin	73
Tabel 6.12 Kasus Uji Login Admin	74
Tabel 6.13 Kasus Uji Login Admin	74
Tabel 6.14 Kasus Uji Login Dokter.....	74
Tabel 6.15 Kasus Uji Login Dokter.....	75
Tabel 6.16 Kasus Uji Login Dokter.....	75
Tabel 6.17 Kasus Uji Login Dokter.....	75
Tabel 6.18 Kasus Uji Login Direktur	76
Tabel 6.19 Kasus Uji Login Direktur	76
Tabel 6.20 Kasus Uji Login Direktur	76
Tabel 6.21 Kasus Uji Login Direktur	77
Tabel 6.22 Kasus Uji Mencari Data Rekam Medis.....	77
Tabel 6.23 Kasus Uji Mencari Data Rekam Medis.....	77
Tabel 6.24 Kasus Uji Menambah Data Rekam Medis	78
Tabel 6.25 Kasus Uji Menambah Data Rekam Medis	78
Tabel 6.26 Kasus Uji Menambah Data Rekam Medis	78
Tabel 6.27 Kasus Uji Logout Admin.....	79
Tabel 6.28 Kasus Uji Logout Admin.....	79
Tabel 6.29 Kasus Uji Logout Dokter.....	80
Tabel 6.30 Kasus Uji Logout Dokter.....	80
Tabel 6.31 Kasus Uji Logout Direktur.....	80
Tabel 6.32 Kasus Uji Logout Direktur.....	81
Tabel 6.33 Kasus Uji Mengubah Profil Admin.....	81

Tabel 6.34 Kasus Uji Mengubah Profil Dokter	82
Tabel 6.35 Kasus Uji Mengubah Profil Direktur	82
Tabel 6.36 Kasus Uji Mengirim Laporan <i>Admin</i>	82
Tabel 6.37 Kasus Uji Mengirim Laporan <i>Admin</i>	83
Tabel 6.38 Kasus Uji Mengirim Laporan Dokter	83
Tabel 6.39 Kasus Uji Mengirim Laporan Dokter	84
Tabel 6.40 Kasus Uji Menambah Data Pasien	84
Tabel 6.41 Kasus Uji Menambah Data Pasien	85
Tabel 6.42 Kasus Uji Mengubah Data Pasien	85
Tabel 6.43 Kasus Uji Mencari Data Pasien pada <i>Admin</i>	85
Tabel 6.44 Kasus Uji Mencari Data Pasien pada <i>Admin</i>	86
Tabel 6.45 Kasus Uji Mencari Data Pasien pada Dokter	86
Tabel 6.46 Kasus Uji Mencari Data Pasien pada Dokter	86
Tabel 6.47 Kasus Uji Menghapus Data Pasien	87
Tabel 6.48 Kasus Uji Menghapus Data Pasien	87
Tabel 6.49 Kasus Uji Melihat Data Pasien pada <i>Admin</i>	88
Tabel 6.50 Kasus Uji Melihat Data Pasien pada Dokter	88
Tabel 6.51 Kasus Uji Melihat Profil pada <i>Admin</i>	88
Tabel 6.52 Kasus Uji Melihat Profil pada Dokter	89
Tabel 6.53 Kasus Uji Melihat Profil pada Direktur	89
Tabel 6.54 Kasus Uji Mengubah Data Rekam Medis	89
Tabel 6.55 Kasus Uji Menghapus Data Rekam Medis	90
Tabel 6.56 Kasus Uji Menghapus Data Rekam Medis	90
Tabel 6.57 Kasus Uji Melihat Data Rekam Medis	91
Tabel 6.58 Kasus Uji Melihat Masa Berlaku SIP	91
Tabel 6.59 Kasus Uji Mengunduh Laporan <i>Admin</i>	91
Tabel 6.60 Kasus Uji Mengunduh Laporan Dokter	92
Tabel 6.61 Kasus Uji Menghapus Laporan <i>Admin</i>	92
Tabel 6.62 Kasus Uji Menghapus Laporan <i>Admin</i>	92
Tabel 6.63 Kasus Uji Menghapus Laporan Dokter	93
Tabel 6.64 Kasus Uji Menghapus Laporan Dokter	93
Tabel 6.65 Kasus Uji Melihat Laporan Dokter	93



Tabel 6.66 Kasus Uji Melihat Laporan <i>Admin</i>	94
Tabel 6.67 Kasus Uji Menambah <i>Admin</i>	94
Tabel 6.68 Kasus Uji Menambah <i>Admin</i>	94
Tabel 6.69 Kasus Uji Menambah Dokter	95
Tabel 6.70 Kasus Uji Menambah Dokter	95
Tabel 6.71 Kasus Uji Menghapus Dokter	95
Tabel 6.72 Kasus Uji Menghapus Dokter	96
Tabel 6.73 Kasus Uji Menghapus <i>Admin</i>	96
Tabel 6.74 Kasus Uji Menghapus <i>Admin</i>	97
Tabel 6.75 Kasus Uji Melihat Masa Berlaku SIP Dokter	97
Tabel 6.76 Kasus Uji Mengubah <i>Admin</i>	97
Tabel 6.77 Kasus Uji Mengubah Dokter	98
Tabel 6.78 Kasus Uji Melihat Daftar Dokter	98
Tabel 6.79 Kasus Uji Melihat Daftar <i>Admin</i>	98
Tabel 6.80 Kasus Uji Mencari <i>Admin</i>	99
Tabel 6.81 Kasus Uji Mencari <i>Admin</i>	99
Tabel 6.82 Kasus Uji Mencari Dokter	99
Tabel 6.83 Kasus Uji Mencari Dokter	100

DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi Ramdani Skincare & Spa.....	7
Gambar 2.2 Proses Bisnis as-is Ramdani Skincare & Spa.....	10
Gambar 2.3 Proses Bisnis to-be Ramdani Skincare & Spa.....	11
Gambar 2.4 Model Alur <i>Prototype</i>	13
Gambar 2.5 <i>Use Case Diagram</i>	15
Gambar 2.6 <i>Sequence Diagram</i>	15
Gambar 2.7 <i>Class Diagram</i>	16
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	19
Gambar 4.1 Diagram <i>Use Case Admin, Dokter, dan Direktur</i>	31
Gambar 5.1 <i>Sequence Diagram</i> Menambah Data Rekam Medis.....	47
Gambar 5.2 <i>Sequence Diagram</i> Menambah Pasien.....	48
Gambar 5.3 <i>Sequence Diagram</i> Menghapus <i>Admin</i>	48
Gambar 5.4 Perancangan <i>Class Diagram</i> pada <i>Controller</i>	49
Gambar 5.5 Perancangan <i>Class Diagram</i> pada <i>Model</i>	50
Gambar 5.6 Perancangan <i>Class Diagram</i> pada <i>View</i>	50
Gambar 5.7 Perancangan <i>Entity Relationship Diagram</i>	51
Gambar 5.8 Perancangan <i>Physical Data Model</i>	52
Gambar 5.9 Perancangan Antarmuka Halaman Menambah Data Rekam Medis.....	56
Gambar 5.10 Perancangan Antarmuka Halaman Menambah Pasien.....	57
Gambar 5.11 Perancangan Antarmuka Halaman <i>Delete Admin</i>	58
Gambar 5.12 Implementasi Antarmuka Menambah Data Rekam Medis.....	62
Gambar 5.13 Implementasi Antarmuka Menambah Pasien.....	63
Gambar 5.14 Implementasi Antarmuka Menambah Pasien (Lanjutan).....	63
Gambar 5.15 Implementasi Antarmuka Menghapus <i>Admin</i>	64
Gambar 6.1 <i>Flow Graph</i> <i>Add_Rekam_Medis()</i>	66
Gambar 6.2 <i>Flow Graph</i> <i>Add_Pasien()</i>	68
Gambar 6.3 <i>Flow Graph</i> <i>Delete_Admin()</i>	69
Gambar 6.4 Method <i>Add_Rekam_Medis()</i> dari kelas <i>Controller</i> dan <i>Model</i>	70
Gambar 6.5 <i>Flow Graph</i> <i>Add_Rekam_Medis()</i>	72
Gambar 6.6 Hasil Pengujian <i>Compatibility</i> Aplikasi.....	100

DAFTAR LAMPIRAN

Lampiran 1 Formulir Pendaftaran.....	103
Lampiran 2 Data Rekam Medis Pasien 1.....	104
Lampiran 3 Data Rekam Medis Pasien 2.....	105
Lampiran 4 Data Rekam Medis Pasien 3.....	106



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Kesigapan petugas medis serta ketepatan dalam melayani pasien di klinik atau rumah sakit sangat dibutuhkan. Namun, pencarian data pasien yang banyak secara manual tentu memakan waktu cukup lama bagi paramedis untuk melakukan tindakan, pasien sering kali memiliki nomor rekam medis ganda karena proses pendaftaran hanya tersedia secara manual (*hardcopy*) sehingga sering kali hilang dan tidak ada rekam medisnya, manajemen klinik/rumah sakit yang tidak rapi, serta lamanya antrian pelayanan dapat menjadi masalah pada klinik atau rumah sakit yang masih menggunakan metode manual untuk merekam riwayat medis pasien. Dengan adanya permasalahan tersebut, sangat dibutuhkan sebuah sistem rekam medis yang dapat memudahkan dan mempercepat pekerjaan paramedis, serta dapat pula meningkatkan eminensi pelayanan.

Kualitas pelayanan termasuk totalitas dari karakteristik barang dan jasa yang menampilkan kemampuannya dalam memuaskan kebutuhan pelanggan, baik yang nampak jelas maupun yang tidak terlihat. Pemberian pelayanan yang memiliki kualitas tinggi terhadap pelanggan adalah hal wajib yang harus dilaksanakan oleh pelaku di bidang jasa jika perusahaan hendak mencapai keberhasilan (Kotler, 2000).

Berdasarkan penelitian yang telah dilakukan sebelumnya oleh Susanto (2011) membahas tentang pembangunan sistem informasi rekam medis berbasis web dengan studi kasus di RSUD Pacitan. Permasalahan yang dihadapi yaitu, ketika pasien berobat di rumah sakit yang berbeda. Kemudian data rekam medis milik pasien yang ada hanya ada di rumah sakit/klinik lama tempat pasien berobat, sementara belum ada riwayat rekam medis pasien di tempat baru pasien berobat. Pencatatan rekam medis secara manual masih sulit untuk dipindahkan antar rumah sakit. Sementara pasien yang berobat tentunya harus mendapatkan perlakuan berbeda-beda secara individu sesuai dengan riwayat penyakitnya. Penelitian ini turut membuktikan lebih efisiennya sistem kerja di RSUD Pacitan dengan adanya sistem tersebut.

Berdasarkan hasil wawancara dengan pihak Ramdani *Skincare & Spa*, diperoleh fakta bahwa Ramdani *Skincare & Spa* telah menerapkan antrian pelayanan serta pencatatan dan pengarsipan data diri pasien. Namun permasalahan yang terjadi saat ini adalah Ramdani *Skincare & Spa* masih melakukan itu semua secara manual (*hardcopy*), yang menyebabkan pengarsipan data diri pasien menjadi tidak efektif karena data pasien yang banyak yang harus segera diolah, serta data yang berkelanjutan dari riwayat kunjungan pasien. Dan juga menyebabkan proses pendaftaran, pemeriksaan, pembayaran, dan pelaporan di klinik sangat lambat karena belum adanya sistem yang mengatur dengan baik.

Maka solusi yang dibutuhkan untuk memudahkan karyawan Ramdani *Skincare & Spa* dalam mengatasi masalah yang ada yaitu dengan membuat sistem rekam medis yang menggunakan metode *prototyping*. Penulis memilih untuk

menggunakan metode *prototyping* karena metode ini memiliki jalan keluar untuk permasalahan yang dihadapi. Metode ini juga turut mengundang pengguna dan *stakeholder* dalam proses pembangunan dan pengembangan sistem, sehingga metode ini dapat mempersingkat waktu dan biaya, serta mengurangi risiko terjadinya. Pengembang dan pengguna juga ikut terlibat proses dalam pengembangannya dengan metode ini, maka bisa terjadi proses interaksi yang baik antara pengguna dan pengembang, sehingga tidak terjadi penafsiran yang salah.

1.2 Rumusan Masalah

Berdasarkan kerangka masalah yang telah diuraikan pada sub bab 1.1, maka rumusan masalahnya dapat dipecah menjadi beberapa poin sebagai berikut:

1. Bagaimana menganalisis kebutuhan pada sistem rekam medis Ramdani *Skincare & Spa*?
2. Bagaimana merancang dan mengimplementasikan sistem rekam medis Ramdani *Skincare & Spa*?
3. Bagaimana hasil pengujian dan analisis sistem guna mengetahui tingkat kualitas kinerja sistem rekam medis Ramdani *Skincare & Spa*?

1.3 Tujuan

Tujuan dari penelitian ini yang didasarkan atas rumusan masalah yang telah diuraikan dijelaskan pada poin-poin berikut:

1. Mengetahui mengenai analisis kebutuhan pada sistem rekam medis Ramdani *Skincare & Spa*.
2. Merancang dan mengimplementasikan sistem rekam medis Ramdani *Skincare & Spa*.
3. Mengetahui hasil pengujian dan analisis sistem guna mengetahui tingkat kualitas kinerja sistem rekam medis Ramdani *Skincare & Spa*.

1.4 Manfaat

Manfaat penelitian ini bagi petugas medis Ramdani *Skincare & Spa* adalah dapat menyelesaikan permasalahan rekam data medis pasien yang tidak tercatat rapi menjadi rapi dan teratur sehingga lebih terorganisir, dapat memudahkan pencarian data pasien, dapat menghindari nomor rekam medis ganda, menghindari kesalahan tindakan medis, mempermudah proses pembuatan laporan, serta mempercepat proses pelayanan pasien di Ramdani *Skincare & Spa* oleh paramedis. Lalu, manfaat penelitian ini bagi pasien adalah pasien akan mendapatkan pelayanan lebih cepat dan teratur, serta pasien dapat terhindar dari kesalahan penanganan medis.

1.5 Batasan Masalah

Ditentukannya batasan masalah tujuannya adalah agar penelitian mempunyai fokus yang pasti dan jelas. Berdasarkan rumusan masalah yang telah diuraikan, maka didapatkan batasan masalah sebagai berikut:

1. Penelitian dilakukan di Ramdani *Skinicare & Spa* Malang.
2. Pengisian rekam medis pasien hanya bisa dilakukan oleh petugas kesehatan yang melayani di Ramdani *Skinicare & Spa* Malang.

1.6 Sistematika Pembahasan

Penataan penulisan mengenai penelitian ini secara garis besar dipaparkan sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini berisi mengenai latar belakang masalahnya, rumusan masalahnya agar lebih terfokus, tujuan penelitian dilakukan, manfaat dilakukannya penelitian ini, batasan masalah yang ditetapkan dalam penelitian, serta penataan pembahasannya.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan mengenai berbagai teori yang diimplementasikan dalam pengerjaan penelitian ini. Teori-teori tersebut didapatkan dari berbagai sumber seperti literatur, penelitian sebelumnya, jurnal, dan internet.

BAB 3 METODOLOGI PENELITIAN

Bab ini membahas mengenai metodologi yang digunakan yang meliputi studi pustaka penelitian, analisis kebutuhan yang dibutuhkan sistem, perancangan yang akan dilakukan, pembangunan prototipe, evaluasi prototipe, implementasi, pengujian, penarikan kesimpulan dan saran.

BAB 4 ANALISIS DAN PERANCANGAN

Bab ini berisi pembahasan mengenai penganalisisan kebutuhan dari sistem yang akan dibangun.

BAB 5 IMPLEMENTASI

Bab ini berisi kode-kode untuk membuat sistem dalam penelitian ini yang terdiri dari tahap perancangan dan implementasi algoritme.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai hasil pengujiannya, pembahasannya, serta hasil analisis dari algoritme yang dimanfaatkan dalam penelitian ini.

BAB 7 PENUTUP

Bab ini membahas mengenai kesimpulan akhir dari penelitian ini dan juga saran untuk peneliti selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini mengulas mengenai kepastakaan peneliti-peneliti sebelumnya dan teori-teori yang mendukung penelitian ini.

2.1 Kajian Pustaka

Kajian pustaka berikut berdasarkan dengan tiga penelitian terdahulu yang menjadi referensi penelitian ini.

Sebuah analisis oleh Susanto (2011) membahas tentang pembangunan sistem informasi rekam medis yang diimplementasikan di RSUD Pacitan dengan basis web. Permasalahan yang dihadapi yaitu, saat pasien berpindah rumah sakit. Pasien telah memiliki rekam medis di rumah sakit sebelumnya, namun saat pasien berpindah ke rumah sakit lain, pencatatan rekam medis harus dimulai dari awal lagi karena pencatatan manual dengan kertas sangat sulit untuk dipindahkan antar rumah sakit. Serta hasil dari penelitian ini menunjukkan lebih efisiennya sistem kerja di RSUD Pacitan dengan adanya sistem tersebut.

Penelitian berikutnya yang dilakukan oleh Cahyanti (2012) membahas tentang bagaimana membangun sebuah sistem informasi manajemen di Puskesmas Pakis Baru Nawangan guna memudahkan proses pendaftaran pasien, serta pencarian, dan pelaporan. Permasalahan yang dihadapi yaitu, dalam proses pendaftaran pasien di Puskesmas Pakis Baru Nawangan sering kali masih terjadi masalah dalam pengidentifikasian pasien lama dengan pasien baru yang dikarenakan data pasien yang tidak rapi atau tidak ada di dalam sistem. Hasil dari penelitian ini menyatakan bahwa proses perekaman data medis menjadi lebih mudah, *real time*, dan cepat.

Gavidi (2015) membahas mengenai bagaimana mengembangkan sebuah sistem rekam medis untuk pasien rawat jalan di Fiji dalam penelitiannya yang berjudul *Development of an Android Application for an Electronic Medical Record System in an Outpatient Environment for Healthcare in Fiji*. Permasalahan yang dihadapi adalah departemen rawat jalan di rumah sakit negara berkembang biasanya kekurangan staf dan tidak cukup untuk menangani sejumlah besar pasien yang datang. Maka, dibuatlah sistem berbasis Android yang dapat memudahkan praktisi medis dan pasien dalam mengatur janji temu, serta sistem juga dapat mengirimkan pesan pengingat secara *real time* ketika waktu yang ditentukan telah dekat.

Berdasarkan penelitian yang telah diuraikan tersebut, maka digunakanlah sistem rekam medis sebagai media penyimpanan informasi serta rekam medis di Ramdani *Skincare & Spa* karena kemampuannya yang dapat menyimpan informasi yang dibutuhkan secara *real time*, cepat, mudah, dan akurat.

2.2 Perbedaan Penelitian Terdahulu dan Sekarang

Penelitian ini mengambil referensi dari beberapa penelitian terdahulu yang relevan. Namun, dalam penelitian ini ditemukan adanya beberapa perbedaan dengan penelitian terdahulu. Berikut pada Tabel 2.1 dijelaskan secara terperinci perbedaan yang ada antara penelitian terdahulu dan sekarang.

Tabel 2.1 Tabel Perbedaan Penelitian Terdahulu dan Sekarang

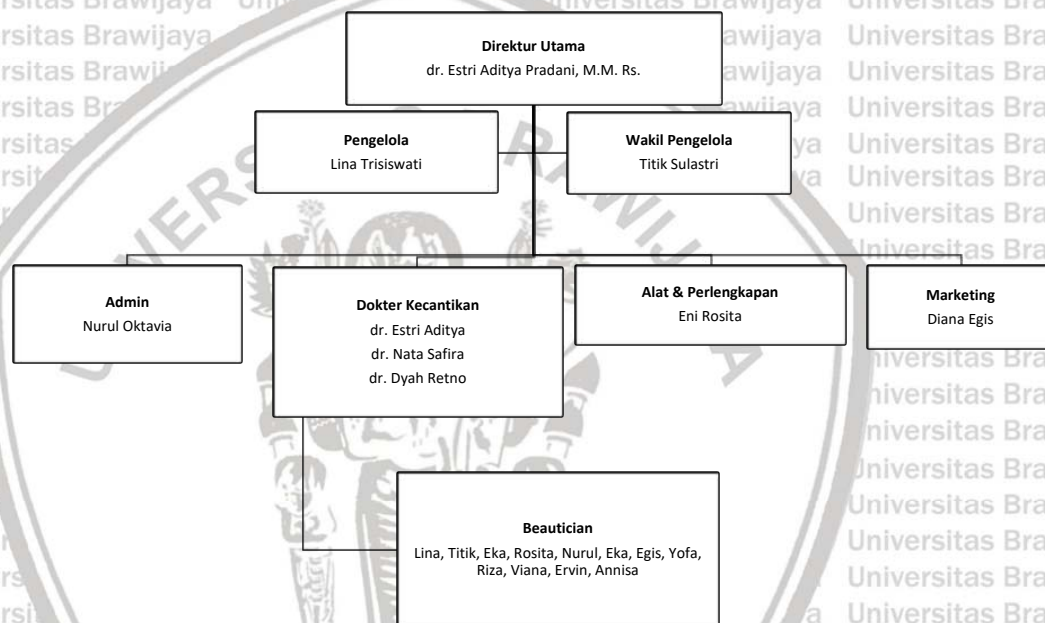
No.	Nama Peneliti Terdahulu	Judul Penelitian	Perbedaan
1.	Gunawan Susanto	Sistem Informasi Rekam Medis Pada Rumah Sakit Umum Daerah (RSUD) Pacitan Berbasis Web Base	<p>Fungsi:</p> <ul style="list-style-type: none"> - Penelitian terdahulu menerapkan fungsi klasifikasi penyakit berdasarkan data epidemiologi dan data statistik. - Penelitian sekarang tidak menerapkan fungsi tersebut. <p>Periode penelitian:</p> <ul style="list-style-type: none"> - Penelitian terdahulu dilaksanakan pada tahun 2011. - Penelitian sekarang dilaksanakan pada tahun 2019. <p>Objek penelitian:</p> <ul style="list-style-type: none"> - Penelitian terdahulu objeknya adalah RSUD Pacitan. - Penelitian sekarang objeknya adalah klinik Ramdani <i>SkinCare & Spa</i> Malang.
2.	Ana Cahyanti Nur	Pembangunan Sistem Informasi Manajemen Puskesmas Pakis Baru Nawangan	<p>Sistem yang dibangun:</p> <ul style="list-style-type: none"> - Penelitian terdahulu membangun sistem informasi manajemennya.

			<p>- Penelitian sekarang membangun sistem rekam medis.</p> <p>Periode penelitian:</p> <p>- Penelitian terdahulu dilaksanakan pada tahun 2012.</p> <p>- Penelitian sekarang dilaksanakan pada tahun 2019.</p> <p>Objek penelitian:</p> <p>- Penelitian terdahulu objeknya adalah Puskesmas Pakis Baru Nawangan.</p> <p>Penelitian sekarang objeknya adalah klinik Ramdani <i>Skincare & Spa</i> Malang.</p>
3.	Bulou Gavidi	<i>Development of an Android Application for an Electronic Medical Record System in an Outpatient Environment for Healthcare in Fiji</i>	<p>Platform:</p> <p>- Penelitian terdahulu berbasis Android.</p> <p>- Penelitian sekarang berbasis web.</p> <p>Periode penelitian:</p> <p>- Penelitian terdahulu dilaksanakan pada tahun 2015.</p> <p>- Penelitian sekarang dilaksanakan pada tahun 2019.</p> <p>Objek penelitian:</p> <p>- Penelitian terdahulu objeknya adalah pusat kesehatan di Fiji.</p> <p>Penelitian sekarang objeknya adalah klinik Ramdani <i>Skincare & Spa</i> Malang.</p>

2.3 Ramdani Skincare & Spa

Ramdani Skincare & Spa merupakan sebuah klinik kecantikan yang ditangani oleh dokter-dokter terpercaya untuk pria dan wanita yang telah ada sejak tahun 2014 dan berlokasi di Jl. Raya Jatikerto No. 78 Bedali, Jatikerto, Kromengan, Malang, Jawa Timur 65165.

Ramdani Skincare & Spa dipimpin oleh pemiliknya yang juga menjabat sebagai direktur Ramdani Skincare & Spa. Direktur utama bertanggung jawab penuh atas semua hal yang berkaitan dengan Ramdani Skincare & Spa. Admin bertugas menginput data diri pasien pada saat proses pendaftaran pasien, serta para dokter bertugas untuk menginput riwayat penyakit pasien dan tindakan yang dilakukan lalu kemudian mengirimkan laporan kepada direktur utama. Struktur organisasi Ramdani Skincare & Spa dijelaskan pada Gambar 2.1.



Gambar 2.1 Struktur Organisasi Ramdani Skincare & Spa

Sumber: Ramdani Skincare & Spa (2021)

2.4 Sistem Rekam Medis

Menurut IDI (2005), rekaman yang tercatat dalam bentuk tulisan maupun gambaran pelayanan yang diberikan oleh pemberi pelayanan medis kepada seorang pasien ialah pengertian dari rekam medis.

Sistem rekam medis merupakan sistem yang berfungsi memudahkan proses penyimpanan data pasien suatu rumah sakit atau klinik, serta bertujuan untuk mengurangi kesalahan yang biasa terjadi dengan proses memasukkan data secara manual.

Dalam UU RI No.29 Tahun 2009 tentang Praktik Kedokteran pada bagian penjelasan pasal 46 ayat 1 dinyatakan bahwa yang dimaksud "rekam medis adalah

berkas berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan, dan pelayanan lain yang telah diberikan kepada pasien”.



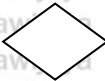


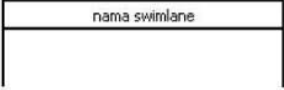
Dalam Peraturan Menteri Kesehatan Republik Indonesia No. 269/MenKes/Per/III/ 2008 menyatakan bahwa yang dimaksud dengan rekam medis adalah “berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain kepada pasien.”

Pengertian layanan rekam medis dalam Pedoman Penyelenggaraan dan Prosedur Rekam Medis Rumah Sakit (2006) adalah “Keterangan baik yang tertulis maupun yang terekam tentang identitas, anamneses, pemeriksaan fisik, laboratorium, diagnosa serta segala pelayanan dan tindakan medis yang diberikan kepada pasien, dan pengobatan baik yang dirawat inap, rawat jalan, maupun pelayanan gawat darurat”. Rekam medis jika dikaji lebih dalam maka akan mempunyai makna yang lebih kompleks yang bukan hanya berisi catatan kondisi dan keadaan pasien, tetapi di dalam rekam medis akan tercermin dan terekam segala informasi menyangkut pasien yang menjadi dasar dalam menentukan tindakan lebih lanjut dalam upaya pelayanan maupun tindakan medis lain yang diberikan kepada seorang pasien yang datang ke rumah sakit.

2.5 Proses Bisnis Ramdani Skincare & Spa

Proses bisnis merupakan alur proses berjalannya bisnis dalam suatu perusahaan. Proses bisnis pada penelitian ini dibuat sebagai tolak ukur perbedaan proses yang terjadi saat sebelum ada sistem dan setelah dibangun sistem rekam medis. Proses bisnis berikut digambarkan dengan *activity diagram*. Berikut pada table 2.2 adalah penjelasan dari berbagai simbol yang digunakan beserta deskripsinya.

Tabel 2.2 Tabel Simbol pada Diagram Proses Bisnis Ramdani Skincare

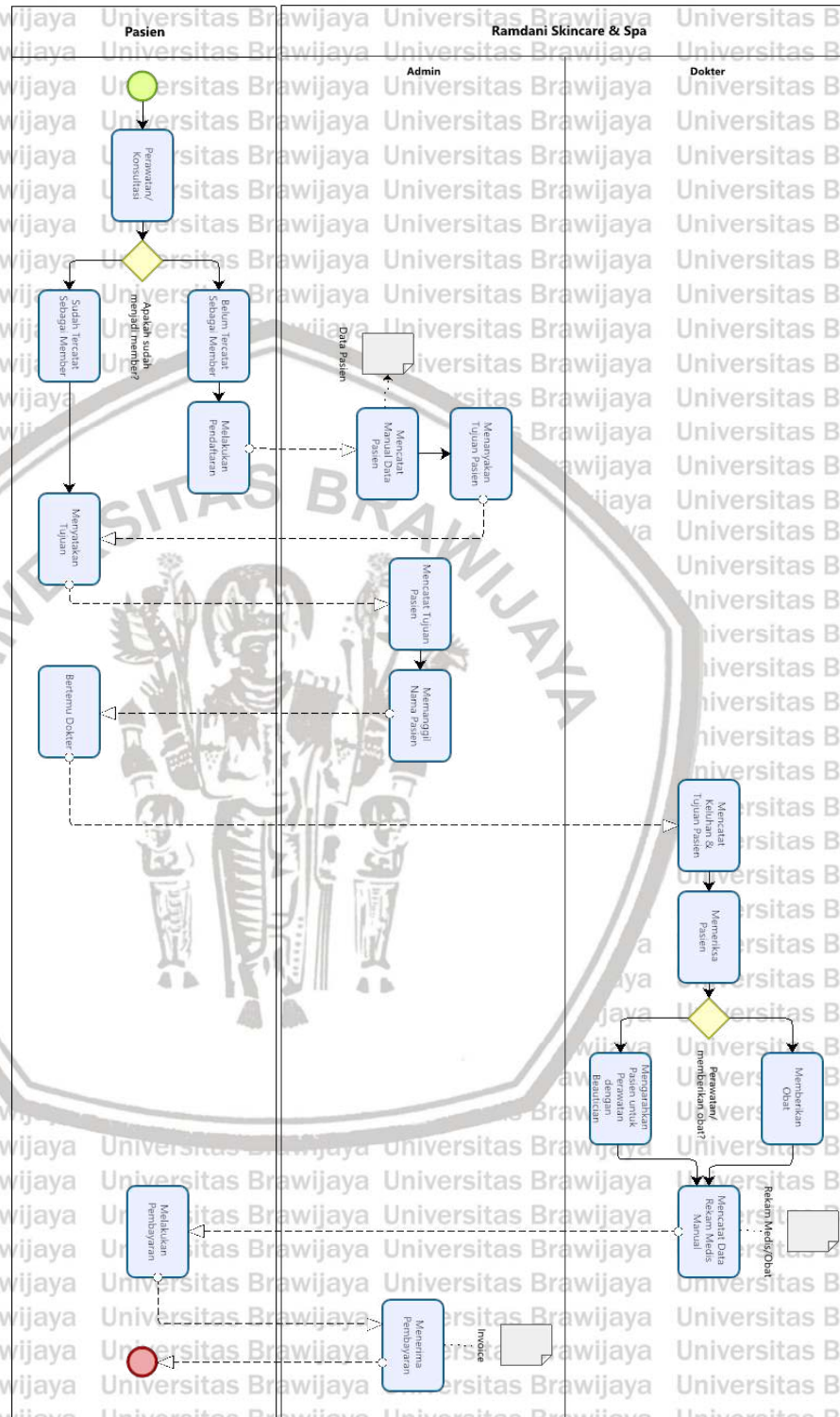
Simbol	Nama Simbol	Deskripsi
	Status awal	Kondisi mula kegiatan di sistem
	Aktivitas	Kegiatan yang sistem sedang lakukan
	Percabangan/ <i>decision</i>	Percabangan saat dihadapkan dengan opsi aktivitas yang lebih dari satu
	Penggabungan	Pengintegrasian lebih dari satu aktivitas
	Status akhir	Kondisi aktivitas sistem terakhir
	<i>Swimlane</i>	Pemisah antar organisasi/aktor yang memiliki tanggung jawab dengan aktivitas yang terjadi

Simbol	Nama Simbol	Deskripsi
	<i>Fork</i>	Aktivitas yang dilakukan secara paralel
	<i>Join</i>	Aktivitas yang digabungkan menjadi satu

Sumber: Satzinger, Jackson, & Burd (2012)



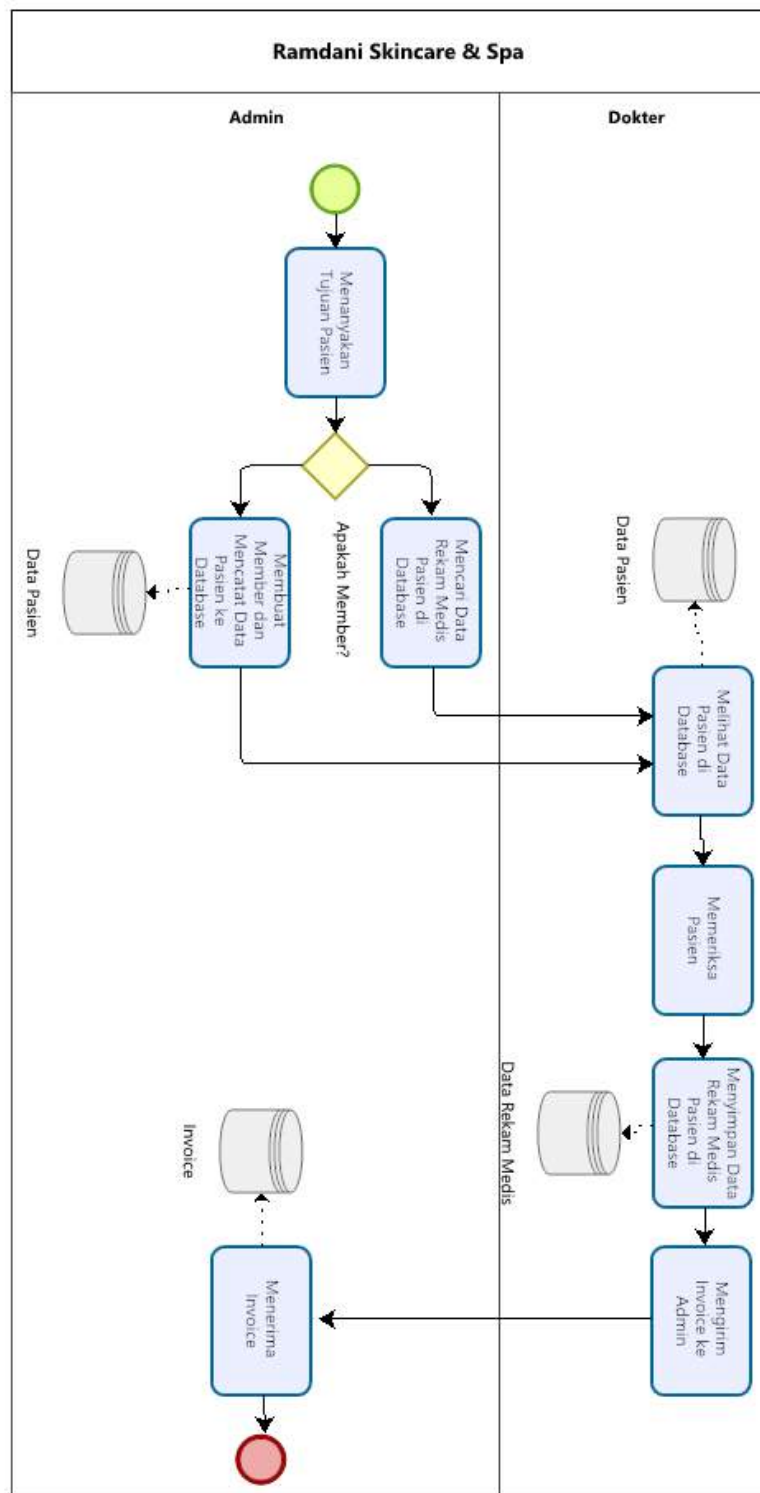
Berikut pada Gambar 2.2 adalah proses bisnis Ramdani Skincare & Spa sebelum adanya sistem.



Gambar 2.2 Proses Bisnis as-is Ramdani Skincare & Spa

Sumber: Ramdani Skincare & Spa (2021)

Berikut pada Gambar 2.3 merupakan proses bisnis Ramdani Skincare & Spa setelah ada sistem.



Gambar 2.3 Proses Bisnis to-be Ramdani Skincare & Spa

Sumber: Ramdani Skincare & Spa (2021)

2.6 PHP

PHP (PHP Hypertext Preprocessor) adalah bahasa pemrograman *scripting* berlisensi *Open Source*. *PHP* ialah bahasa pemrograman dengan kode-kode untuk mengolah data dan kemudian mengirimkan kembali ke *browser* menjadi kode *HTML*.

PHP tidak mengamati taraf kecocokan dari *user* yang mengaksesnya, *PHP* mengamati melalui kapasitas dan kecocokan *server* yang menjalankannya, itu dikarenakan *PHP* dijalankan di sisi *server* (Nugroho, 2005).

2.7 Database MySQL

2.7.1 Database

Database atau basis data, menurut Fathansyah (2001) digambarkan sebagai kelompok data yang saling terhubung sedemikian rupa untuk digunakan kembali dengan mudah dan cepat. Ada pula yang disebut dengan *DBMS (Database Management System)*, yang merupakan program komputer untuk memasukkan, mengubah, menghapus, memanipulasi, dan juga mendapatkan data dan informasi secara efisien.

2.7.2 MySQL

MySQL ialah program *database server* yang dapat mengirim data dan menerima data dengan kecepatan yang cukup cepat. *MySQL* merupakan piranti lunak yang dapat memudahkan pengaturan *database* dalam pembuatan *field*, *record*, dan *table*. *MySQL* juga dapat menghapus serta mengubah *database* yang sudah ada. *SQL* dapat dikatakan sebagai bahasa penyambung dengan *database* (Nugroho, 2005).

Menurut Imansyah (2003), fungsi-fungsi yang digunakan pada bahasa *SQL* antara lain:

1. *Select* berfungsi untuk memperlihatkan data sesuai kriteria yang kita inginkan.
2. *Create* berfungsi untuk membuat tabel yang belum ada.
3. *Insert* berfungsi untuk memasukan atau menambah baris pada tabel.
4. *Update* berfungsi untuk mengubah isi data tabel.
5. *Delete* berfungsi untuk menghilangkan baris/*record* data di tabel.
6. *Drop* berfungsi untuk menghilangkan tabel serta *database*.

2.8 Framework CodeIgniter

Sebuah perangkat lunak untuk memudahkan *programmer* dalam pembuatan aplikasi atau *web* dengan isi yang terdiri dari bermacam fungsi, ada juga *plugin*, dan juga konsep yang akan membentuk suatu sistem, itulah pengertian dari *Framework*. Dengan *framework*, sebuah aplikasi akan runtut dan rapi.

Siena (2009) *framework* ialah *library* yang terorganisasikan pada desain arsitektur demi memberikan kemudahan, ketepatan, memberikan kecepatan, dan juga konsistensi dalam pengembangan suatu aplikasi.

2.8.1 CodeIgniter

CodeIgniter ialah salah satu *framework PHP* yang dirancang dengan konsepsi *MVC (Model, View, Controller)*. *CodeIgniter* berbasis *MVC* dan disalin ke *PHP* yang kemudian didesain guna menaikkan eminensi suatu piranti lunak dengan cara memangkas ongkos pengembangan serta pemeliharannya, dan juga menghasilkan struktur pemrograman yang rapi (baik dari sisi kode dan struktur *file PHP*-nya), serta memberikan standar *coding* yang memudahkan kita maupun orang lain untuk mempelajari kembali sistem yang dibangun.

Menurut Blanco & Upton (2009) *CodeIgniter* ialah sumber terbuka *PHP framework* yang mudah diambil alih, dinaikkan menjadi lebih baik untuk para *PHP programmers* yang membutuhkan alat sederhana dan juga baik guna membangun aplikasi web dengan fitur penuh. *CodeIgniter* merupakan *MVC framework* yang dirancang untuk mempermudah seluruh penggunaannya.

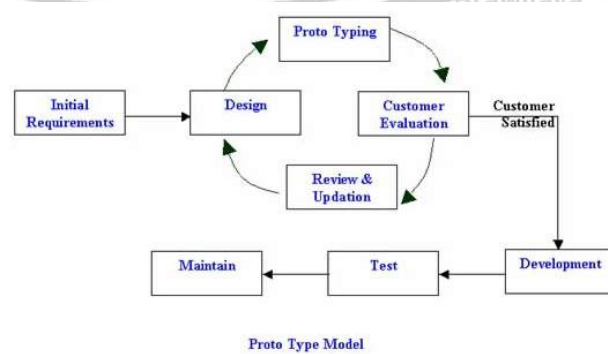
2.9 Prototype

Pada pembangunan sistem rekam medis ini, peneliti memakai metode *System Development Life Cycle*. Alasan digunakannya metode ini yaitu dapat mewujudkan reaksi sistem yang lebih baik dikarenakan sistem dianalisis dan didesain secara keseluruhan sebelum dibuat, sehingga peluang terjadi kesalahan sangat kecil, serta sesuai untuk digunakan pada pembangunan sistem berskala kecil seperti ini.

SDLC atau *System Development Life Cycle* adalah metode yang bisa dipakai guna mengembangkan suatu sistem yang juga dapat ditafsirkan sebagai kegiatan dalam membangun sebuah sistem dengan melalui beberapa tahapan. Terdapat beberapa model *SDLC* yang biasa digunakan, seperti *waterfall*, *fountain*, *spiral*, *rapid*, *prototype*, *incremental* (Isnardi, 2016).

Prototype merupakan metode pembangunan sistem yang sangat kuat secara terstruktur dan bisa terus disempurnakan. Dengan penggunaan metode *prototype*, sistem ini dapat terus disempurnakan dan dibangun kembali. Teknik ini membuat pengembang mendapatkan kemungkinan penambahan fitur yang tidak didapat selama fase *requirement* dan perancangan.

Berikut pada Gambar 2.4 adalah gambar model alur *prototype*:



Gambar 2.4 Model Alur *Prototype*

Sumber: Sommerville (2011)

1. *Initial Requirements*

Tahapan dimana *stakeholder* dan pengembang bertemu demi membicarakan *software* yang ingin dibangun dan mengidentifikasi kebutuhan sistem.

2. *Design*

Menggambarkan desain piranti lunak yang akan diterapkan pada sesuatu yang dapat dipahami oleh pengguna dengan mudah.

3. *Prototyping*

Pembuatan prototipe sebagai bentuk gambaran sistem yang hendak diimplementasikan.

4. *Customer Evaluation*

Pelanggan meninjau jika sistem yang diinginkan telah setara dengan yang diharapkan ataukah belum.

5. *Review & Updation*

Tahap ini merupakan tahap lanjutan dari *Customer Evaluation*. Jika sistem sudah sesuai dengan keinginan pengguna, proses akan berlanjut ke tahap implementasi. Jika belum, proses iterasi yang dilakukan ulang mulai dari desain sampai evaluasi *customer* tetap akan berlanjut hingga sesuai dengan keinginan pengguna.

6. *Development*

Pada tahap ini, *prototype* yang telah disepakati akan diterjemahkan ke dalam bahasa pemrograman yang sesuai.

7. *Test*

Ketika sistem telah menjadi perangkat lunak siap pakai, maka akan dilakukan pengujian terhadap sistem apakah bisa berjalan dengan baik ataukah tidak.

8. *Maintain*

Pemeliharaan sistem.

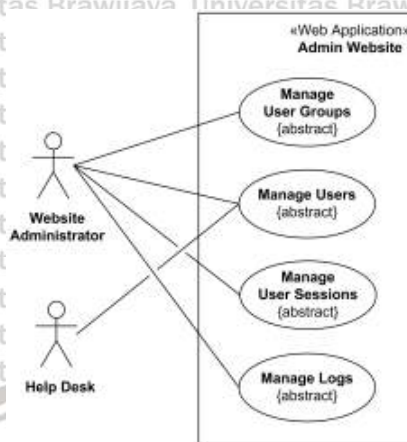
2.10 *Unified Modelling Language (UML)*

Unified Modelling Language (UML) menurut Fowler (2005) ialah anggota disposisi sampel yang diayomi oleh meta model tunggal, yang bertugas mengakomodasi penggambaran serta rancangan *software* dengan pemrograman berorientasi objek (OO)(khususnya).

2.10.1 *Use Case Diagram*

Use Case merupakan metode perekaman persyaratan praktis sebuah sistem. *Use Case* mendefinisikan korelasi antar pengguna-pengguna sistem dengan sistem yang dibuat. *Use Case Diagram* menunjukkan setiap *use case* yang harus digunakan aktor serta menunjukkan setiap *use case* yang dapat memasuki *use*

case yang lain dan juga hubungan antara user atau aktor dengan use case (Fowler, 2005). Berikut pada Gambar 2.5 ialah contoh use case diagram:

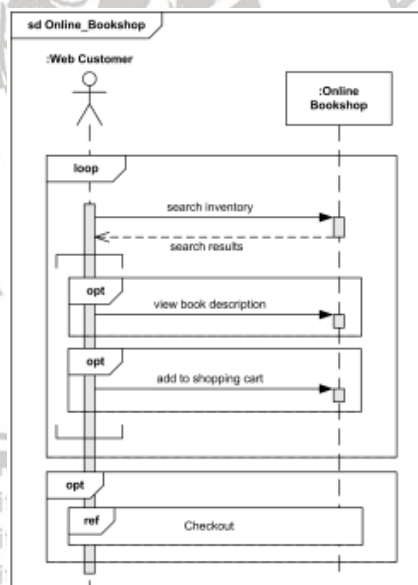


Gambar 2.5 Use Case Diagram

Sumber: UML Diagrams (2009)

2.10.2 Sequence Diagram

Bagan dua dimensi dengan obyeknya digambarkan dalam format melintang serta lifeline yang digambarkan dalam format berdiri/tegak, disebut dengan sequence diagram (Munawar, 2005). Berikut pada Gambar 2.6 adalah desain sequence diagram:



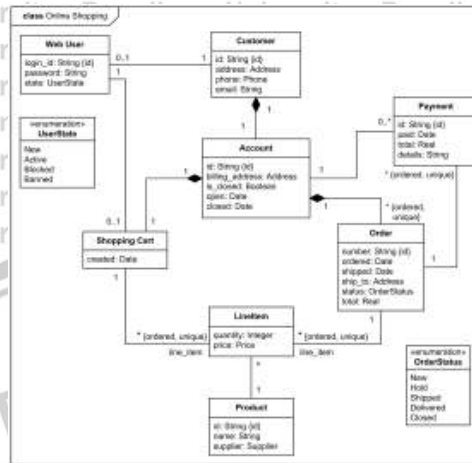
Gambar 2.6 Sequence Diagram

Sumber: UML Diagrams (2009)

2.10.3 Class Diagram

Salah satu diagram utama UML yang berfungsi mendefinisikan class pada sebuah sistem, ialah class diagram.

Class diagram menurut Munawar (2005) adalah kumpulan beberapa objek serupa. Sebuah objek dengan keadaan singkat serta tingkah laku. Serta *state* sebuah objek yang diartikan sebagai kondisi objek tersebut yang diakui dalam *attribute*. Sementara perilaku suatu objek menggambarkan bagaimana sebuah objek beraksi serta memberikan reaksi. Berikut Gambar 2.7 menggambarkan desain *class diagram*:



Gambar 2.7 Class Diagram

Sumber: UML Diagrams (2009)

2.11 Pengujian Perangkat Lunak

Pengujian perangkat lunak ialah aksi yang dilaksanakan guna memperoleh keterangan mengenai kualitas produk atau layanan yang sedang dicoba (Kaner, 2006). Terdapat tingkatan-tingkatan dalam pengujian perangkat lunak, yaitu:

1. *Unit testing*, merupakan pengujian yang dilakukan pada unit terkecil suatu kode seperti fungsi, kelas, atau modul.
2. *Integration testing*, merupakan pengujian yang dilakukan pada interaksi fungsi-fungsi yang ada pada sistem sehingga fungsi-fungsi tersebut dapat terintegrasi dengan benar.
3. *System testing*, yaitu bentuk pengujian keseluruhan terhadap sistem.

2.11.1 Desain Test Case

Metode *test case* yang dapat digunakan ialah yang memberikan kemungkinan cukup tinggi dalam menemukan kesalahan. Dalam penelitian ini, penulis menggunakan *Test Case White Box*, *Black Box*, dan *Compatibility Testing*.

2.11.1.1 Pengujian White Box (fungsionalitas)

Dilakukannya pengujian kode program yang berdasarkan pada pembuktian terhadap uraian pemodelan. Program yang benar seutuhnya akan didapatkan dengan *white box testing* karena pengujian dilaksanakan sesuai dengan bagaimana suatu *software* mengeluarkan *output* dari *input* yang dimasukkan.

Pengujian *white-box* dengan *Basis Path Testing* merupakan satu di antara beberapa teknik pengujian *white-box*. Teknik ini dipakai guna mengukur kompleksitas logika dari suatu logika dan juga sebagai pedoman untuk memperoleh konsistensi jalur aplikasi (Pare, 2013).

Berikut ialah kasus uji yang akan diperoleh dengan metode *white-box testing*:

1. Mengetahui bahwa seluruh jalur independen sudah pernah terlewati setidaknya sebanyak satu kali.
2. Mengaplikasikan seluruh keputusan valid benar dan salah.
3. Mengaplikasikan keseluruhan *loop* dalam batas fungsional.
4. Membuat struktur data privat guna membuktikan keabsahannya.

Pengujian *white-box* dikerjakan dengan teknik *basis path*. Kode program digunakan dalam kiat ini untuk mencari setiap jalur yang bisa dilalui serta untuk mendefinisikan kesalahan-kesalahan yang ada pada kode program. Metode *basis path* memberikan hasil berupa sejumlah jalur yang dilewati suatu operasi dari awal hingga mendapatkan hasil yang diinginkan. Hasil tersebut kemudian digunakan untuk menjabarkan *test case*. Lalu ada *cyclomatic complexity* yang digunakan sebagai matrik pengukur kekompleksitan program dengan jumlah jalur mandiri.

Berikut ialah cara menghitung nilai kekompleksitan:

- a. $V(G) = E - N + 2$
- b. $V(G) = P + 1$
- c. $V(G) = R$

Keterangan

E = *Edges*

N = *Nodes*

P = *Predicate Node*

R = *Region*

Tabel 2.3 menampilkan tabel nilai kekompleksitan yang didasarkan dari jumlah jalur independen.

Tabel 2.3 Tabel Nilai Kompleksitas

Jumlah Kompleksitas	Arti
1-10	Runtut serta ditulis dengan benar Kemampuan untuk diuji tinggi Ongkos murah serta mudah
10-20	Kode program saling berhubungan Kemampuan untuk diuji sedang Ongkos sedang dan agak susah

20-40	Kode program sangat saling terhubung Kemampuan untuk diuji rendah Mahal dan sulit
>40	Ongkos serta usaha sangat tinggi Tidak memiliki kemampuan untuk diuji

2.11.1.2 Pengujian Black Box (fungsionalitas)

Pengujian *black-box* ialah pengujian yang menguji fungsionalitas sistem tanpa mengamati struktur internal sistem tersebut atau tanpa mengamati kode sumber. Pengujian ini dapat diterapkan pada tingkatan unit, integrasi, dan sistem, namun seringkali dijumpai pada tingkatan yang tinggi (Ammann, 2016).

Pengujian *black-box* yang akan dilaksanakan yaitu dengan menggunakan teknik yang bernama *State Transitions Testing*. Dengan teknik ini, *input* dan *output* yang dihasilkan diharapkan harus sesuai dengan keinginan pengguna dengan cara pengguna memasukkan *input* tertentu dan akan mengeluarkan *output* tertentu yang kemudian akan dicocokkan dengan sistem, lalu kemudian diketahui apakah *input* dan *output* yang dihasilkan sesuai dengan sistem atau belum. Pengujian *black box* dilakukan untuk mendapati kesalahan seperti fungsi yang tidak berjalan dengan tepat, antarmuka tidak sesuai yang diharapkan, dan struktur data yang salah.

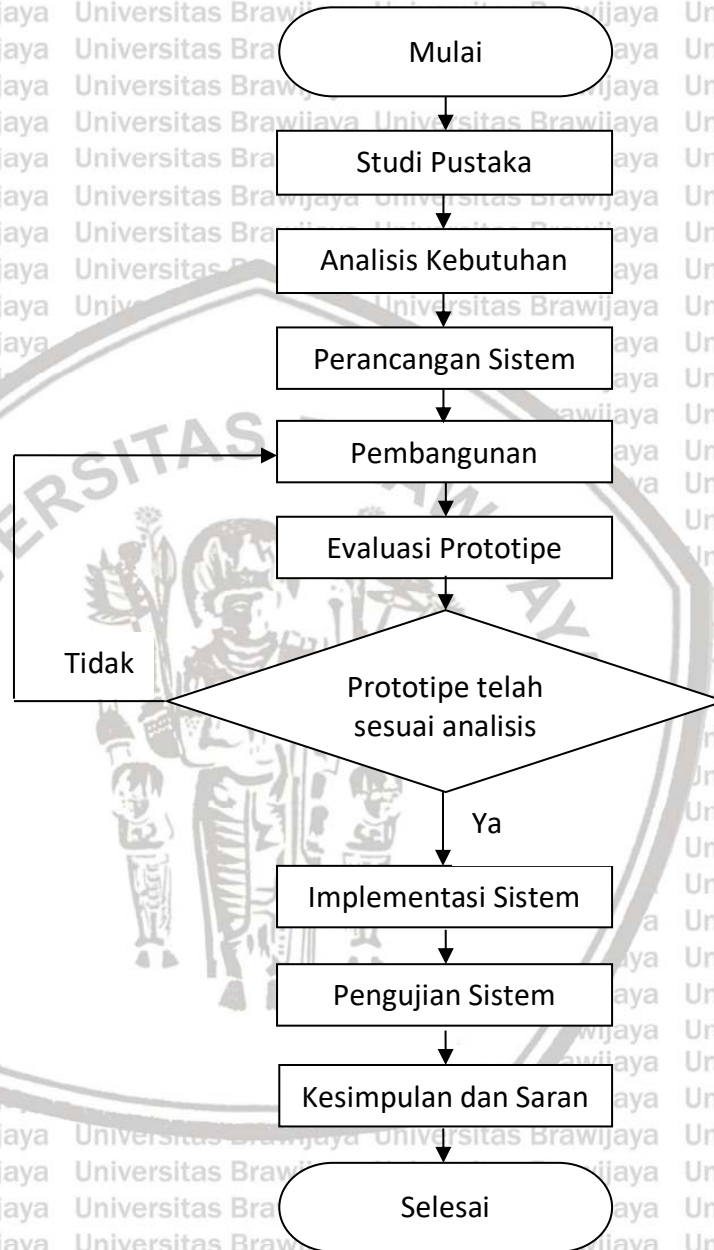
2.11.1.3 Pengujian Compatibility (non fungsional)

Pengujian *Compatibility* ialah pengujian yang dilaksanakan demi mengetahui bagaimana jalannya sistem yang dibuat, apakah berjalan baik dan benar saat dioperasikan di sistem operasi yang berbeda, *browser* yang berbeda, berbagai ukuran layar, dan berbagai koneksi *internet* yang berbeda (Hass, 2014).

Pada sistem informasi manajemen dan rekam medis Ramdani *Skincare & Spa* ini akan dicoba melakukan pengujian *compatibility* pada beberapa *browser*, seperti Google Chrome dan Mozilla Firefox, dan juga pada perangkat komputer yang berbeda menggunakan aplikasi *SortSite* dan akan didapatkan hasil pengujian valid atau tidak.

BAB 3 METODOLOGI PENELITIAN

Bab ini mendefinisikan alur pola penelitian yang dikerjakan oleh penulis. Alur pola penelitian tersebut dapat terlihat di Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1 Studi Pustaka

Dalam tahapan ini, penulis akan mempelajari berbagai teori yang dipakai sebagai referensi dasar penelitian dalam menunjang penulisan penelitian. Teori-

teori tersebut dapat diperoleh melalui jurnal, buku, *internet*, dan lain-lain. Teori yang digunakan untuk mendukung penelitian ini, yaitu:

1. *PHP*
2. *MySQL database*
3. Metode *Prototyping*
4. *Framework CodeIgniter*
5. Sistem Rekam Medis.
6. Ramdani *Skincare & Spa* Malang
7. *SDLC*
8. Pengujian Perangkat Lunak

3.2 Analisis Kebutuhan Sistem

Tingkatan untuk menjabarkan hal-hal yang diperlukan sistem sebelum mewujudkan menjadi kode program disebut dengan analisis kebutuhan. Analisis kebutuhan ini dapat diperoleh dengan mengadakan wawancara dan diskusi dengan pihak yang terkait sehingga sistem ini dapat terjabarkan secara terperinci. Berikut ialah tahapan analisis kebutuhan sistem rekam medis Ramdani *Skincare & Spa*:

1. Proses Elisitasi Kebutuhan
Dalam proses elisitasi kebutuhan, wawancara terhadap direktur dan petugas medis Ramdani *Skincare & Spa* akan dilakukan demi mendapati permasalahan yang ada dan juga keperluan yang diperlukan oleh sistem yang hendak bangun.
2. Proses Spesifikasi Kebutuhan
Dalam kiat berikut ini dilaksanakan kiat penjabaran kebutuhan yang telah ada menjadi lebih rinci dan distingtif. Lalu akan dibuat *use case diagram* dan *use case scenario* sebagai pemodelan kebutuhannya.
3. Proses Validasi Kebutuhan
Pada tahap ini, pengecekan kembali dilakukan agar semua kebutuhan yang diharapkan pengguna ada pada sistem yang akan diwujudkan nanti.
4. Proses Manajemen Kebutuhan
Diciptakannya proses membuat kode kebutuhan dan mengatur prioritas kebutuhan dalam proses manajemen kebutuhan.

3.3 Perancangan Sistem

Perancangan sistem memiliki kegunaan sebagai refleksi dari sistem yang aktual. Penulis menggunakan *OOD (Object Oriented Design)* dengan bentuk *UML (Unified Modelling Language)*. Lalu, digunakan diagram kelas sebagai refleksi hubungan antar *database*. Penulis juga menggunakan *use case diagram* sebagai penjelasan korelasi antara pengguna dengan sistem yang lain maupun dengan aktor lain. Untuk aliran data, penulis mengimplementasikannya dengan *activity diagram*. Penulis juga menggunakan *sequence diagram* sebagai gambaran pertukaran pesan antar obyek ketika melakukan tugas.

3.4 Pembangunan Prototipe

Pada pembangunan prototipe ini terjadi proses pembangunan prototipe (contoh gambaran sistem yang akan dibangun) yang berpusat pada penyampaian kepada pelanggan, seperti memanifestasikan format input dan output. Proses ini akan mengalami iterasi atau perulangan sampai prototipe sesuai dengan keinginan pelanggan.

3.5 Evaluasi Prototipe

Pada tahap ini, prototipe yang sudah jadi segera dievaluasi. Jika telah sesuai ketentuan dan keinginan pengguna, maka segera lanjut ke proses implementasi sistem. Namun jika belum sempurna, akan dilakukan pemodelan/pembangunan ulang prototipe sampai sempurna lalu bisa dilanjutkan ke tahap membuat kode pada implementasi sistem.

3.6 Implementasi Sistem

Pada tahap ini, prototipe harus sudah sesuai dengan keinginan pelanggan sehingga bisa segera dilakukan proses pengkodean. Berikut merupakan *software* yang digunakan untuk mempraktikkan sistem ini melalui tahap pemrograman adalah:

1. XAMPP.
2. PHPMyAdmin untuk pemrograman *database*.
3. Sublime sebagai *editor*.
4. PHP sebagai Bahasa pemrograman di sisi server.
5. HTML digunakan sebagai struktur website.
6. CSS untuk desain *user interface*.
7. Browser (Google Chrome dan Safari) untuk menampilkan dan menguji sistem rekam medis.

3.7 Pengujian Sistem

Penelitian ini dilaksanakan dengan pengujian *White Box* dan *Black Box*, serta *compatibility testing*. Pengujian yang dilaksanakan demi mencocokkan fungsionalitas sistem dan non fungsionalitas sistem yang dibuat ialah sebagai berikut:

1. Pengujian Unit

Metode *basis path* yang dapat memastikan kesalahan-kesalahan yang ada pada kode program dan didesain untuk melakukan semua jalur yang ada digunakan sebagai teknik pengujian unit pada penelitian ini. *Basis path* memanifestasikan jumlah jalan yang dilewati suatu fungsional dari awal mula sampai hasil akhir yang diinginkan. Hasil ini kemudian akan dipakai guna menentukan *test case*. Adapun matrik yang dipakai untuk menghitung kekompleksitasan program dengan jumlah jalur mandiri ialah *cyclomatic complexity*.

2. Pengujian Validasi

Pengujian validasi akan dilaksanakan dengan teknik *black box testing* bernama *State Transitions Testing*. Pengujian ini bekerja dengan cara harus menyesuaikan masukan dan keluaran yang didapatkan dengan yang diharapkan pengguna. Setiap *test case* mempunyai syarat untuk dikatakan berhasil, yaitu status mulai, masukan, *output* sesuai yang diharapkan, dan status akhir yang juga sesuai dengan yang diharapkan.

3. Pengujian Integrasi

Pengujian integrasi adalah pengujian yang dilakukan pada interaksi fungsi-fungsi yang ada pada sistem sehingga fungsi-fungsi tersebut dapat terintegrasi dengan benar.

4. Pengujian *Compatibility*

Pengujian *compatibility* dijalankan demi mendapati apakah sistem yang dibuat sanggup dijalankan pada *browser* yang berbeda. Sistem akan diuji dengan cara dijalankan pada berbagai macam *browser* dan akan dilihat bagaimana jalannya sistem, apakah baik ataukah tidak.

3.8 Penarikan Kesimpulan dan Saran

Sesudah melalui tahapan penganalisisan kebutuhan, perancangan, pengimplementasian, penajalan, dan penganalisisan, maka tiba saatnya untuk menarik kesimpulan. Kesimpulan dapat dipetik dari hasil pengujian dan penganalisisan sistem. Kemudian akan dilakukan penulisan saran sebagai tahap akhir dari penelitian ini. Saran tersebut berguna dalam menjadikan kesalahan-kesalahan yang terjadi menjadi lebih baik dan dapat menjadi acuan untuk penyempurnaan penulisan serta pengembangan penelitian selanjutnya.



BAB 4 ANALISIS KEBUTUHAN

Hasil yang pasti didapatkan oleh perangkat lunak yang akan dibuat dari proses analisis kebutuhan ini ialah spesifikasi kebutuhan. Ada dua ragam analisis dari kebutuhan sistem, yaitu penganalisisan kebutuhan yang bersifat fungsional serta penganalisisan kebutuhan yang bersifat non fungsional. Hal ini bertujuan agar dapat mempermudah perancangan sistem selanjutnya.

4.1 Gambaran Umum Sistem

Skema rekam medis yang diciptakan dengan tujuan agar pencatatan rekam medis di Ramdani *Skincare & Spa* dapat dilakukan dengan mudah dalam melakukan pencatatan rekam medis serta proses administrasi. Sistem ini terdiri dari empat aktor yang memiliki akses berbeda-beda di dalam sistem.

Sistem rekam medis ini menggunakan konsep waktu *real time*, sehingga informasi-informasi di dalamnya bersifat sesuai dengan waktu yang sebenarnya dan selalu diperbarui. Sistem ini hanya bisa diakses oleh pegawai Ramdani *Skincare & Spa*.

4.2 Elisitasi Kebutuhan

Untuk mengetahui permasalahan yang terjadi pada Ramdani *Skincare & Spa*, maka dibutuhkan proses elisitasi kebutuhan. Tahapan ini dilakukan dengan melakukan wawancara kepada direktur utama dengan tujuan untuk mengetahui permasalahan lebih mendalam.

Berdasarkan hasil wawancara dengan pihak Ramdani *Skincare & Spa*, diperoleh fakta bahwa Ramdani *Skincare & Spa* telah menerapkan antrian pelayanan serta pencatatan dan pengarsipan data diri pasien. Namun permasalahan yang terjadi saat ini adalah Ramdani *Skincare & Spa* masih melakukan itu semua secara manual (*hardcopy*), yang menyebabkan pengarsipan data diri pasien menjadi tidak efektif karena banyaknya jumlah data-data pasien yang dikerjakan dan data berkelanjutan dari Riwayat pasien. Permasalahan ini juga menyebabkan proses pendaftaran, pemeriksaan, pembayaran, dan pelaporan di klinik sangat lambat karena belum adanya sistem yang mengatur dengan baik.

Maka solusi yang dibutuhkan untuk memudahkan karyawan Ramdani *Skincare & Spa* dalam mengatasi masalah yang ada yaitu dengan membuat sistem rekam medis yang menggunakan metode *prototyping*.

4.3 Identifikasi Aktor

Pengidentifikasian aktor dilakukan untuk mendapatkan siapa saja yang turut menggunakan sistem. Ada beberapa aktor yang berhubungan dengan sistem informasi Ramdani *Skincare & Spa* ini. Sistem ini memiliki tiga aktor, yaitu:

1. Admin

Admin melakukan proses pencarian pasien, melihat daftar pasien, menambah pasien, memilih pasien, mengubah data pasien, serta menghapus pasien, dan juga mengirim laporan kepada direktur.

2. Dokter

Dokter dapat melakukan proses melihat biodata dan jejak rekam medis pasien, memasukkan daftar obat dan tindakan yang diberikan kepada pasien, serta mengirim laporan kepada direktur.

3. Direktur

Direktur dapat melihat dan mengelola segala kegiatan yang berjalan di dalam sistem. Direktur juga dapat mengunduh laporan harian yang dikirimkan oleh admin dan dokter, serta dapat mengelola data *admin* dan dokter.

4.4 Daftar Kebutuhan

4.4.1 Kebutuhan Fungsional

Indeks yang isinya terdiri dari berbagai macam proses yang harus disiapkan oleh sistem yang mencakup cara sistem bereaksi terhadap masukan tertentu serta perilaku sistem pada situasi tertentu merupakan pengertian dari kebutuhan fungsional. Jenis perangkat lunak yang digunakan dan siapa yang akan menjadi pengguna sistem memegang peranan penting bagi kebutuhan fungsional. Daftar kebutuhan dispesifikasikan menjadi dua yaitu, spesifikasi kebutuhan fungsional admin dan spesifikasi kebutuhan fungsional member. Pada kebutuhan fungsional akan diberikan kode SRS-RSS-F-X yang mengandung pengertian dari SRS (*System Requirement Spesification*), RSS (*Ramdani Skincare & Spa*), F (*fungsionalitas*), dan X (*nomor definisi kebutuhan utama*). Sementara pada kebutuhan non fungsional akan diberikan kode SRS-RSS-NF-X, dimana NF adalah singkatan dari non fungsional. Berikut akan dijelaskan spesifikasi kebutuhan fungsional admin di Tabel 4.1.

Tabel 4.1 Spesifikasi Kebutuhan Fungsional Admin Iterasi Pertama

Kode Fungsi	Nama Fungsi	Deskripsi
SRS-RSS-F-1	Login sebagai <i>admin</i>	<i>Admin</i> bisa masuk ke dalam sistem dan juga hanya admin yang berhak mengelola sistem
SRS-RSS-F-2	Logout sebagai <i>admin</i>	<i>Admin</i> bisa keluar dari sistem
SRS-RSS-F-3	Menambah Data Pasien	<i>Admin</i> dapat menambah data diri pasien lama/baru pada sistem

Kode Fungsi	Nama Fungsi	Deskripsi
SRS-RSS-F-4	Mengubah Data Pasien	<i>Admin</i> dapat mengubah data pasien pada sistem
SRS-RSS-F-5	Mencari Data Pasien	<i>Admin</i> dapat mencari data pasien pada sistem
SRS-RSS-F-6	Menghapus Data Pasien	<i>Admin</i> dapat menghapus data pasien pada sistem jika sudah tidak menjadi pasien di <i>Ramdani Skincare & Spa</i> lagi
SRS-RSS-F-7	Melihat Profil	<i>Admin</i> dapat melihat profil pada sistem
SRS-RSS-F-8	Mengubah Profil	<i>Admin</i> dapat mengubah profil pada sistem
SRS-RSS-F-9	Melihat Data Pasien	<i>Admin</i> dapat melihat data pasien pada sistem

Berikut akan dijelaskan detail kebutuhan fungsional dokter di Tabel 4.2.

Tabel 4.2 Spesifikasi Kebutuhan Fungsional Dokter Iterasi Pertama

Kode Fungsi	Nama Fungsi	Deskripsi
SRS-RSS-F-10	Login sebagai dokter	Dokter dapat masuk ke dalam sistem
SRS-RSS-F-11	Logout sebagai dokter	Dokter dapat keluar dari sistem
SRS-RSS-F-12	Menambah Data Rekam Medis Pasien	Dokter bisa menambahkan berkas riwayat perawatan pasien dengan menginputkan obat dan tindakan yang dilakukan untuk pasien
SRS-RSS-F-13	Mencari Data Pasien	Dokter bisa mencari data diri pasien pada sistem
SRS-RSS-F-14	Menghapus Data Rekam Medis Pasien	Dokter bisa menghapus berkas pasien pada sistem jika sudah tidak menjadi pasien di <i>Ramdani Skincare & Spa</i> lagi
SRS-RSS-F-15	Mengubah Data Rekam Medis Pasien	Dokter bisa mengubah berkas riwayat perawatan pasien pada sistem

Kode Fungsi	Nama Fungsi	Deskripsi
SRS-RSS-F-16	Melihat Data Pasien	Dokter dapat melihat data pasien pada sistem
SRS-RSS-F-17	Melihat Profil	Dokter dapat melihat profil pada sistem
SRS-RSS-F-18	Mengubah Profil	Dokter dapat mengubah profil pada sistem
SRS-RSS-F-19	Melihat Masa Berlaku SIP	Dokter dapat melihat masa berlaku SIP miliknya pada sistem
SRS-RSS-F-20	Melihat Data Rekam Medis Pasien	Dokter dapat melihat data rekam medis pasien pada sistem
SRS-RSS-F-21	Mencari Data Rekam Medis Pasien	Dokter dapat mencari data riwayat Kesehatan pasien pada sistem

Berikut akan dijelaskan spesifikasi kebutuhan fungsional Direktur di Tabel 4.3.

Tabel 4.3 Spesifikasi Kebutuhan Fungsional Direktur Iterasi Pertama

Kode Fungsi	Nama Fungsi	Deskripsi
SRS-RSS-F-22	Login sebagai direktur	Direktur dapat masuk ke dalam sistem
SRS-RSS-F-23	Logout sebagai direktur	Direktur dapat keluar dari sistem
SRS-RSS-F-24	Menambah Dokter	Direktur dapat menambah dokter pada sistem
SRS-RSS-F-25	Menghapus Dokter	Direktur dapat menghapus dokter pada sistem
SRS-RSS-F-26	Mengubah Dokter	Direktur dapat mengubah dokter pada sistem
SRS-RSS-F-27	Melihat Daftar Dokter	Direktur dapat menyaksikan daftar dokter pada sistem
SRS-RSS-F-28	Mencari Dokter	Direktur dapat mencari dokter pada sistem
SRS-RSS-F-29	Melihat Masa Berlaku SIP Dokter	Direktur dapat melihat masa berlaku SIP seluruh dokter pada sistem

Kode Fungsi	Nama Fungsi	Deskripsi
SRS-RSS-F-30	Menambah Admin	Direktur dapat menambah admin pada sistem
SRS-RSS-F-31	Menghapus Admin	Direktur dapat menghapus admin pada sistem
SRS-RSS-F-32	Mengubah <i>Admin</i>	Direktur dapat mengubah <i>admin</i> pada sistem
SRS-RSS-F-33	Melihat Daftar <i>Admin</i>	Direktur dapat melihat daftar <i>admin</i> pada sistem
SRS-RSS-F-34	Mencari <i>Admin</i>	Direktur dapat mencari <i>admin</i> pada sistem
SRS-RSS-F-35	Mengedit Profil	Direktur dapat mengubah profil pada sistem
SRS-RSS-F-36	Melihat Profil	Direktur dapat melihat profil pada sistem

Terdapat sedikit perubahan dari rancangan spesifikasi kebutuhan fungsional iterasi pertama pada *Admin* karena kurang sesuai dengan keinginan *user* yaitu dengan menambahkan fungsi mengirim laporan kepada direktur, maka berikut ini adalah spesifikasi kebutuhan fungsional *Admin* iterasi kedua agar sesuai dengan keinginan *user*. Spesifikasi kebutuhan fungsional *admin* iterasi kedua ditunjukkan pada Tabel 4.4.

Tabel 4.4 Spesifikasi Kebutuhan Fungsional *Admin* Iterasi Kedua

Kode Fungsi	Nama Fungsi	Deskripsi
SRS-RSS-F-1	Login sebagai <i>admin</i>	<i>Admin</i> bisa masuk ke sistem dan hanya admin yang berhak mengelola sistem
SRS-RSS-F-2	Logout sebagai <i>admin</i>	<i>Admin</i> bisa keluar sistem
SRS-RSS-F-3	Menambah Data Pasien	<i>Admin</i> bisa menambah data diri pasien lama/baru pada sistem
SRS-RSS-F-4	Mengubah Data Pasien	<i>Admin</i> dapat mengubah data pasien pada sistem
SRS-RSS-F-5	Mencari Data Pasien	<i>Admin</i> dapat mencari data pasien pada sistem
SRS-RSS-F-6	Menghapus Data Pasien	<i>Admin</i> dapat menghapus data pasien pada sistem jika sudah tidak menjadi pasien di <i>Ramdani Skincare & Spa</i> lagi

Kode Fungsi	Nama Fungsi	Deskripsi
SRS-RSS-F-7	Melihat Profil	<i>Admin</i> dapat melihat profil pada sistem
SRS-RSS-F-8	Mengubah Profil	<i>Admin</i> dapat mengubah profil pada sistem
SRS-RSS-F-9	Melihat Data Pasien	<i>Admin</i> dapat melihat data pasien pada sistem
SRS-RSS-F-10	Mengirim Laporan	<i>Admin</i> dapat mengirim laporan data pasien harian kepada direktur

Terdapat sedikit perubahan dari rancangan spesifikasi kebutuhan fungsional iterasi pertama pada Dokter karena kurang sesuai dengan keinginan *user*, yaitu dengan menambahkan fungsi mengirim laporan kepada direktur, maka berikut ini adalah spesifikasi kebutuhan fungsional Dokter iterasi kedua agar sesuai dengan keinginan *user*. Spesifikasi kebutuhan fungsional dokter iterasi kedua ditunjukkan pada Tabel 4.5.

Tabel 4.5 Spesifikasi Kebutuhan Fungsional Dokter Iterasi Kedua

Kode Fungsi	Nama Fungsi	Deskripsi
SRS-RSS-F-11	Login sebagai dokter	Dokter bisa masuk ke sistem
SRS-RSS-F-12	Logout sebagai dokter	Dokter bisa keluar dari sistem
SRS-RSS-F-13	Menambah Data Rekam Medis Pasien	Dokter dapat menambahkan data rekam medis pasien dengan menginputkan obat serta pemberian perawatan yang diberikan kepada pasien
SRS-RSS-F-14	Mencari Data Pasien	Dokter bisa mencari data diri pasien pada sistem
SRS-RSS-F-15	Menghapus Data Rekam Medis Pasien	Dokter bisa menghapus data pasien pada sistem jika sudah tidak menjadi pasien di <i>Ramdani Skincare & Spa</i> lagi
SRS-RSS-F-16	Mengubah Data Rekam Medis Pasien	Dokter dapat mengubah data rekam medis pasien pada sistem
SRS-RSS-F-17	Melihat Data Pasien	Dokter dapat melihat data pasien pada sistem
SRS-RSS-F-18	Melihat Profil	Dokter dapat melihat profil pada sistem

Kode Fungsi	Nama Fungsi	Deskripsi
SRS-RSS-F-19	Mengubah Profil	Dokter dapat mengubah profil pada sistem
SRS-RSS-F-20	Melihat Masa Berlaku SIP	Dokter dapat melihat masa berlaku SIP miliknya pada sistem
SRS-RSS-F-21	Melihat Data Rekam Medis Pasien	Dokter bisa menyaksikan data rekam medis pasien di sistem
SRS-RSS-F-22	Mencari Data Rekam Medis Pasien	Dokter bisa mencari data rekam medis pasien di sistem
SRS-RSS-F-23	Mengirim Laporan	Dokter dapat mengirim laporan data pasien harian kepada direktur

Berdasarkan perubahan yang terjadi pada rancangan spesifikasi kebutuhan fungsional iterasi pertama *Admin* dan *Dokter*, maka pada aktor *Direktur* pun ikut berubah dengan ditambahkannya fungsi melihat laporan *admin*, melihat laporan dokter, menghapus laporan *admin*, menghapus laporan dokter, mengunduh laporan *admin*, dan mengunduh laporan dokter. Berikut ini adalah spesifikasi kebutuhan fungsional iterasi kedua *Direktur* yang ditunjukkan di Tabel 4.6.

Tabel 4.6 Spesifikasi Kebutuhan Fungsional *Direktur* Iterasi Kedua

Kode Fungsi	Nama Fungsi	Deskripsi
SRS-RSS-F-24	Login sebagai direktur	Direktur bisa masuk ke sistem
SRS-RSS-F-25	Logout sebagai direktur	Direktur bisa keluar sistem
SRS-RSS-F-26	Menambah Dokter	Direktur dapat menambah dokter pada sistem
SRS-RSS-F-27	Menghapus Dokter	Direktur dapat menghapus dokter pada sistem
SRS-RSS-F-28	Mengubah Dokter	Direktur dapat mengubah dokter pada sistem
SRS-RSS-F-29	Melihat Daftar Dokter	Direktur dapat menyaksikan daftar dokter pada sistem
SRS-RSS-F-30	Mencari Dokter	Direktur dapat mencari dokter pada sistem
SRS-RSS-F-31	Melihat Masa Berlaku SIP Dokter	Direktur dapat melihat masa berlaku SIP seluruh dokter pada sistem

Kode Fungsi	Nama Fungsi	Deskripsi
SRS-RSS-F-32	Menambah Admin	Direktur dapat menambah admin pada sistem
SRS-RSS-F-33	Menghapus Admin	Direktur dapat menghapus admin pada sistem
SRS-RSS-F-34	Mengubah <i>Admin</i>	Direktur dapat mengubah <i>admin</i> pada sistem
SRS-RSS-F-35	Melihat Daftar <i>Admin</i>	Direktur dapat melihat daftar <i>admin</i> pada sistem
SRS-RSS-F-36	Mencari <i>Admin</i>	Direktur dapat mencari <i>admin</i> pada sistem
SRS-RSS-F-37	Mengubah Profil	Direktur dapat mengubah profil pada sistem
SRS-RSS-F-38	Melihat Profil	Direktur dapat melihat profil pada sistem
SRS-RSS-F-39	Melihat Laporan Dokter	Direktur dapat melihat laporan dokter pada sistem
SRS-RSS-F-40	Menghapus Laporan Dokter	Direktur dapat menghapus laporan dokter pada sistem
SRS-RSS-F-41	Mengunduh Laporan Dokter	Direktur dapat mengunduh laporan dokter pada sistem
SRS-RSS-F-42	Melihat Laporan <i>Admin</i>	Direktur dapat melihat laporan <i>admin</i> pada sistem
SRS-RSS-F-43	Menghapus Laporan <i>Admin</i>	Direktur dapat menghapus laporan <i>admin</i> pada sistem
SRS-RSS-F-44	Mengunduh Laporan <i>Admin</i>	Direktur dapat mengunduh laporan <i>admin</i> pada sistem

4.4.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan segala jenis keperluan yang mendukung kinerja sistem agar bisa berfungsi dengan benar dan sesuai kebutuhan *stakeholder*. Berikut akan dijelaskan kebutuhan non fungsional sistem pada Tabel 4.7.

Tabel 4.7 Spesifikasi Kebutuhan Non Fungsional Sistem

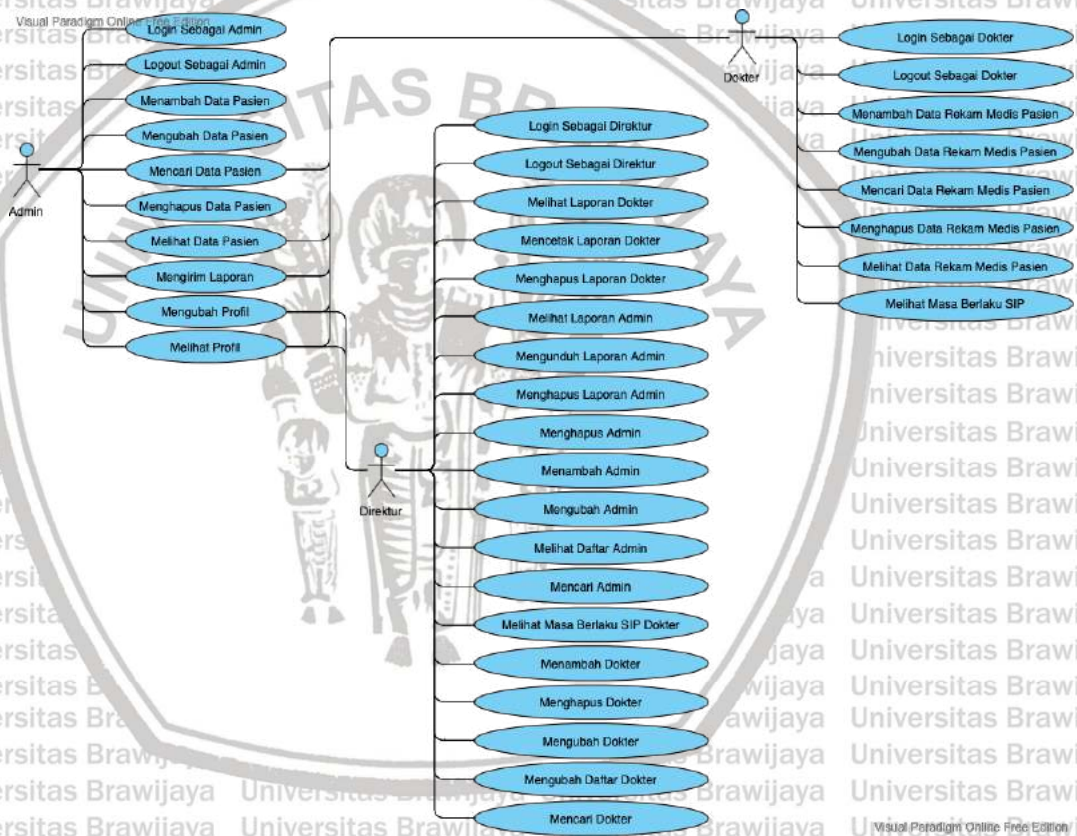
Kode Fungsi	Nama Fungsi	Deskripsi
SRS-RSS-NF-1	<i>Compatibility</i>	Sistem bisa dicapai dari semua <i>browser</i> , layaknya Google Chrome,

Kode Fungsi	Nama Fungsi	Deskripsi
		Mozilla Firefox, dll. Sistem juga dapat diakses dalam waktu yang singkat

4.5 Use Case

4.5.1 Diagram Use Case

Use Case Diagram ialah sebuah penggambaran fungsional yang dimiliki sistem yang menjelaskan tentang pertukaran pesan yang terjadi antarsistem dan aktornya. Pada sistem ini, akan dijelaskan *Use Case Diagram* mengenai hubungan fungsionalitas setiap fitur yang ada pada sistem beserta *Use Case Scenario*. Berikut akan ditunjukkan diagram *use case* para aktor pada Gambar 4.1.



Gambar 4.1 Diagram Use Case Admin, Dokter, dan Direktur

4.5.2 Skenario Use Case

Skenario *use case* ialah deskripsi rinci setiap kasus uji yang dijelaskan pada diagram *use case*. Terdapat beberapa bagian pada skenario *use case*, antara lain nama penggunaan kasus, kode kebutuhan kasus, *user* yang berhubungan, tujuan dari adanya *use case*, adanya *pre condition*, adanya *main flow*, adanya pula *alternative flow*, dan *post condition*. *Pre condition* yaitu syarat yang harus terpenuhi oleh sistem sebelum dapat mengoperasikan *main flow* pada *use case*.

Main flow ialah runtutan pembuatan *use case* sampai menghasilkan hal yang diharapkan terjadi di *post condition*. Sementara *Alternative flow* merupakan skenario pengerjaan jikalau dalam pengerjaannya *main flow* mengalami skenario yang tidak didefinisikan pada *main flow*, lalu *post condition* ialah kondisi terakhir yang diperlihatkan oleh sistem sesudah *user* mengoperasikan *main flow*. Skenario *use case* akan dijabarkan pada Tabel 4.8 hingga Tabel 4.40.

Tabel 4.8 Skenario Use Case Login

Nama Use Case	Login
Aktor	Admin, Dokter, Direktur
Tujuan	Spesifikasi <i>use case</i> ini dipakai guna mendefinisikan kegiatan aktor melakukan login
Pre Condition	Sistem memperlihatkan beranda <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor menginputkan username dan password pada halaman login 2. Aktor memilih login sebagai siapa (admin/dokter/direktur) 3. Aktor memencet tombol login 4. Sistem melakukan proses login untuk masuk ke sistem
Alternative Flow	<ol style="list-style-type: none"> 1. Jika nama pengguna atau kata sandi tidak benar, sistem akan memperlihatkan notifikasi kesalahan pada nama pengguna atau kata sandi 2. Jika aktor login bukan sebagai diri sendiri/bukan sebagai hak aksesnya, sistem akan memperlihatkan peringatan kesalahan pada login sebagai orang lain
Post Condition	Sistem memperlihatkan halaman <i>dashboard</i> masing-masing aktor. Aktor dapat mengakses halaman <i>website</i> sesuai hak akses.

Tabel 4.9 Skenario Use Case Logout

Nama Use Case	Logout
Aktor	Admin, Dokter, Direktur
Tujuan	Spesifikasi <i>use case</i> ini dipakai guna menjelaskan kegiatan aktor melakukan <i>logout</i>
Pre Condition	Sistem memperlihatkan halaman utama
Main Flow	<ol style="list-style-type: none"> 1. Aktor memencet tombol <i>logout</i> di pojok kanan atas halaman utama 2. Sistem memperlihatkan sebuah <i>dialog box</i> memastikan apakah aktor yakin untuk keluar atau tidak 3. Aktor mengklik tombol <i>logout</i> untuk keluar

<i>Alternative Flow</i>	Aktor memencet tombol <i>cancel</i> untuk membatalkan keluar dari sistem, maka aktor masih berada di dalam sistem (sistem masih menampilkan halaman <i>dashboard</i>)
<i>Post Condition</i>	Sistem memperlihatkan halaman <i>login</i>

Tabel 4.10 Skenario Use Case Mengubah Profil

Nama Use Case	Mengubah Profil
Aktor	<i>Admin</i> , Dokter, Direktur
Tujuan	Spesifikasi use case ini dipakai guna menjelaskan kegiatan aktor melakukan ubah profil
<i>Pre Condition</i>	Sistem memperlihatkan halaman utama
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor memencet tombol <i>Profile</i> di sebelah kanan pojok atas halaman utama 2. Sistem memperlihatkan halaman profil yang berisi username, <i>password</i>, dan foto 3. Aktor melakukan perubahan, lalu memencet tombol <i>update profile</i>
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem memperlihatkan pemberitahuan bahwa perubahan berhasil disimpan

Tabel 4.11 Skenario Use Case Mengirim Laporan

Nama Use Case	Mengirim Laporan
Aktor	<i>Admin</i> , Dokter
Tujuan	Spesifikasi use case ini dipakai guna menjelaskan kegiatan aktor dalam mengirim laporan kepada direktur
<i>Pre Condition</i>	Sistem memperlihatkan beranda utama
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Aktor memencet menu Make Report 2. Aktor mengisi data yang terdiri dari Report Subject, Report Message, dan mengunggah file laporan 3. Aktor memencet tombol save report
<i>Alternative Flow</i>	Jika aktor tidak mengisi kolom <i>Report Subject</i> atau <i>Report Message</i> , dan juga tidak mengunggah file, maka sistem akan memperlihatkan peringatan untuk mengisi kolom <i>Report Subject</i> , <i>Report Message</i> , serta mengunggah file

<i>Post Condition</i>	Sistem memperlihatkan pemberitahuan bahwa proses pengunggahan laporan berhasil dan sistem kembali ke halaman laporan
-----------------------	--

Tabel 4.12 Skenario Use Case Menambah Data Pasien

Nama Use Case	Menambah Data Pasien
Aktor	<i>Admin</i>
Tujuan	Spesifikasi <i>use case</i> ini dipakai guna menjelaskan kegiatan aktor dalam menambah data pasien
<i>Pre Condition</i>	Aktor berhasil masuk dan sistem memperlihatkan halaman <i>Patient Registration</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengisikan data diri pasien baru yang berisi ID pasien (berdasarkan nomor KTP), nama, nomor telepon, tempat dan tanggal lahir, alamat, golongan darah, tinggi dan berat badan, jenis kelamin, jenis kulit (berminyak/kering/kombinasi), alergi, rekam medis, dan tanggal pendaftaran/konsultasi 2. Aktor memencet tombol <i>Patient Registration</i>
<i>Alternative Flow</i>	Jika aktor tidak mengisi seluruh kolom yang ada pada halaman <i>Patient Registration</i> , maka sistem akan memperlihatkan himbuan bahwa ada kolom yang masih kosong
<i>Post Condition</i>	Sistem memperlihatkan pesan bahwa pasien berhasil didaftarkan

Tabel 4.13 Skenario Use Case Mengubah Data Pasien

Nama Use Case	Mengubah Data Pasien
Aktor	<i>Admin</i>
Tujuan	Spesifikasi <i>use case</i> ini dipakai guna menjelaskan aktivitas aktor dalam mengubah data pasien
<i>Pre Condition</i>	Aktor berhasil masuk dan sistem memperlihatkan halaman <i>Edit Patient</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor meletakkan kursor di kolom pencarian di sebelah kiri atas halaman, lalu mengetikkan nomor ID pasien, dan kemudian memencet tombol enter pada <i>keyboard</i> 2. Sistem memperlihatkan data pasien yang dicari 3. Aktor memencet tombol berwarna kuning bertuliskan <i>edit</i>

	<ol style="list-style-type: none"> 4. Sistem memperlihatkan halaman data diri pasien yang akan diubah 5. Aktor melakukan perubahan, lalu memencet tombol <i>update patient data</i>
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem memperlihatkan pemberitahuan bahwa perubahan telah berhasil disimpan

Tabel 4.14 Skenario Use Case Mencari Data Pasien

Nama Use Case	Mencari Data Pasien
Aktor	Admin, Dokter
Tujuan	Spesifikasi <i>use case</i> ini dipakai guna menjelaskan kegiatan aktor dalam mencari data pasien
<i>Pre Condition</i>	Aktor berhasil masuk dan sistem menampilkan halaman <i>View Patient List</i>
<i>Main Flow</i>	Aktor meletakkan kursor di kolom pencarian di sebelah kiri atas halaman, lalu mengetikkan nomor ID pasien, dan kemudian menekan tombol <i>enter</i> pada <i>keyboard</i>
<i>Alternative Flow</i>	Jika aktor mengetik nomor ID yang tidak terdaftar sebagai pasien lalu memencet tombol <i>enter</i> , maka sistem akan memperlihatkan halaman kosong (tidak ditemukan data)
<i>Post Condition</i>	Sistem memperlihatkan data pasien yang dicari

Tabel 4.15 Skenario Use Case Menghapus Data Pasien

Nama Use Case	Menghapus Data Pasien
Aktor	Admin
Tujuan	Spesifikasi <i>use case</i> ini dipakai guna menjelaskan kegiatan aktor dalam menghapus data pasien
<i>Pre Condition</i>	Aktor berhasil masuk dan sistem memperlihatkan halaman <i>View Patient List</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor meletakkan kursor pada kolom pencarian di sebelah kiri atas halaman, lalu aktor mengetik nomor ID pasien yang akan dihapus 2. Aktor memencet tombol <i>enter</i> pada <i>keyboard</i> 3. Sistem memperlihatkan hasil pencarian 4. Aktor memencet tombol berwarna merah bertuliskan <i>delete</i>

	5. Sistem memperlihatkan peringatan yakin untuk menghapus atau tidak
	6. Aktor memencet tombol <i>delete</i> (yakin untuk menghapus)
<i>Alternative Flow</i>	Jika aktor memencet tombol <i>cancel</i> , maka sistem batal menghapus data yang dipilih dan kembali ke halaman <i>View Patient List</i>
<i>Post Condition</i>	Sistem kembali ke halaman <i>View Patient List</i>

Tabel 4.16 Skenario Use Case Melihat Data Pasien

Nama Use Case	Menghapus Data Pasien
Aktor	<i>Admin</i> , Dokter
Tujuan	Spesifikasi use case ini dipakai guna menjelaskan kegiatan aktor dalam melihat data pasien
<i>Pre Condition</i>	Aktor berhasil masuk dan sistem memperlihatkan beranda utama
<i>Main Flow</i>	Aktor memencet menu <i>View Patient List</i>
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem memperlihatkan beranda <i>View Patient List</i>

Tabel 4.17 Skenario Use Case Melihat Profil

Nama Use Case	Menghapus Data Pasien
Aktor	<i>Admin</i> , Dokter, Direktur
Tujuan	Spesifikasi use case ini dipakai guna menjelaskan kegiatan aktor melihat profil
<i>Pre Condition</i>	Sistem memperlihatkan halaman utama
<i>Main Flow</i>	Aktor memencet tombol <i>Profile</i> di sebelah kanan pojok atas beranda utama
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem memperlihatkan beranda profil

Tabel 4.18 Skenario Use Case Menambah Data Rekam Medis

Nama Use Case	Menambah Data Rekam Medis
Aktor	Dokter

Tujuan	Spesifikasi <i>use case</i> ini dipakai dokter guna menambah data rekam medis pasien
<i>Pre Condition</i>	Aktor berhasil masuk dan sistem memperlihatkan halaman <i>Add Medical History</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memasukkan nomor ID pasien yang sudah terdaftar di kolom <i>Patient ID</i> dan mengisi data obat serta tindakan (jika ada) yang diberikan kepada pasien di kolom <i>Medical Report</i>, serta mengisi tanggal konsultasi/pemeriksaan pada hari tersebut di kolom <i>Date</i> 2. Aktor memencet tombol <i>save report</i>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor tidak mengisi kolom <i>Patient ID</i> atau salah mengisi nomor ID pasien, maka sistem akan memperlihatkan pesan bahwa ID pasien tidak terdaftar 2. Jika aktor mengisi ID pasien yang terdaftar dengan benar namun tidak mengisi kolom <i>Medical Report</i> dan <i>Date</i>, maka sistem akan memperlihatkan himbauan untuk memenuhi kolom yang masih kosong
<i>Post Condition</i>	Sistem memperlihatkan pesan bahwa data rekam medis pasien berhasil tersimpan di <i>database</i>

Tabel 4.19 Skenario *Use Case* Mencari Data Rekam Medis

Nama <i>Use Case</i>	Mencari Data Rekam Medis
Aktor	Dokter
Tujuan	Spesifikasi <i>use case</i> ini dipakai dokter guna mencari data rekam medis pasien
<i>Pre Condition</i>	Aktor berhasil masuk dan sistem memperlihatkan halaman <i>View Medical History</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor meletakkan kursor pada kolom pencarian di sebelah kiri atas halaman, lalu aktor mengetik nomor ID pasien untuk melihat data rekam medis pasien 2. Aktor memencet tombol <i>enter</i> pada <i>keyboard</i>
<i>Alternative Flow</i>	Jika aktor mengetik nomor ID yang tidak terdaftar sebagai pasien lalu memencet tombol <i>enter</i> , maka sistem akan memperlihatkan beranda kosong (tidak ditemukan <i>file</i>)
<i>Post Condition</i>	Sistem memperlihatkan data pasien yang dicari

Tabel 4.20 Skenario *Use Case* Mengubah Data Rekam Medis

Nama <i>Use Case</i>	Mengubah Data Rekam Medis
Aktor	Dokter
Tujuan	Spesifikasi <i>use case</i> ini dipakai dokter guna mengubah data rekam medis pasien
<i>Pre Condition</i>	Aktor berhasil masuk dan sistem memperlihatkan halaman <i>View Medical History</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor meletakkan kursor pada kolom pencarian di sebelah kiri atas halaman, lalu aktor mengetik nomor ID pasien untuk melihat data rekam medis pasien 2. Aktor memencet tombol <i>enter</i> pada <i>keyboard</i> 3. Sistem memperlihatkan hasil pencarian 4. Aktor memencet tombol berwarna kuning bertuliskan <i>Edit Medical History</i> 5. Sistem memperlihatkan halaman <i>Edit Medical History</i> 6. Aktor mengisi perubahan pada data rekam medis pasien, lalu menekan tombol <i>update medical history</i>
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem memperlihatkan notifikasi bahwa data pasien telah terubah serta tersimpan di <i>database</i>

Tabel 4.21 Skenario *Use Case* Menghapus Data Rekam Medis

Nama <i>Use Case</i>	Menghapus Data Rekam Medis
Aktor	Dokter
Tujuan	Spesifikasi <i>use case</i> ini dipakai dokter guna menghapus data rekam medis pasien
<i>Pre Condition</i>	Aktor berhasil masuk dan sistem memperlihatkan halaman <i>View Medical History</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor meletakkan kursor pada kolom pencarian di sebelah kiri atas halaman, lalu aktor mengetik nomor ID pasien untuk melihat data rekam medis pasien 2. Aktor memencet tombol <i>enter</i> pada <i>keyboard</i> 3. Sistem menampilkan hasil pencarian 4. Aktor memencet tombol berwarna merah bertuliskan <i>delete</i> 5. Sistem memperlihatkan peringatan yakin untuk menghapus atau tidak 6. Aktor memencet tombol <i>delete</i> (yakin untuk menghapus)

<i>Alternative Flow</i>	Jika aktor memencet tombol <i>cancel</i> , maka sistem batal menghapus data yang dipilih dan kembali ke halaman <i>View Medical History</i>
<i>Post Condition</i>	Sistem kembali ke halaman <i>View Medical History</i>

Tabel 4.22 Skenario Use Case Melihat Data Rekam Medis

Nama Use Case	Melihat Data Rekam Medis
Aktor	Dokter
Tujuan	Spesifikasi <i>use case</i> ini dipakai dokter guna melihat data rekam medis pasien
<i>Pre Condition</i>	Sistem memperlihatkan halaman utama
<i>Main Flow</i>	Aktor memencet menu <i>View Medical History</i>
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem memperlihatkan halaman <i>View Medical History</i> yang berisi data-data rekam medis pasien

Tabel 4.23 Skenario Use Case Melihat Masa Berlaku SIP

Nama Use Case	Melihat Masa Berlaku SIP
Aktor	Dokter
Tujuan	Spesifikasi <i>use case</i> ini dipakai dokter guna melihat masa berlaku SIP miliknya
<i>Pre Condition</i>	Sistem memperlihatkan beranda utama
<i>Main Flow</i>	Aktor memencet menu <i>View Notification</i>
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem memperlihatkan halaman <i>Notification</i> yang berisi informasi/peringatan mengenai masa berlaku SIP milik aktor

Tabel 4.24 Skenario Use Case Mengunduh Laporan Admin

Nama Use Case	Mengunduh Laporan Admin
Aktor	Direktur
Tujuan	Spesifikasi <i>use case</i> ini dipakai aktor guna mengunduh laporan <i>admin</i> ke <i>device</i> milik aktor

<i>Pre Condition</i>	Aktor berhasil masuk dan sistem memperlihatkan halaman <i>Admin's Report List</i>
<i>Main flow</i>	Aktor memencet tombol unduh berkas
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Berkas berhasil diunduh dan tersimpan di <i>device</i> milik aktor

Tabel 4.25 Skenario Use Case Mengunduh Laporan Dokter

Nama Use Case	Mengunduh Laporan Dokter
Aktor	Direktur
Tujuan	Spesifikasi <i>use case</i> ini dipakai aktor guna mengunduh laporan dokter ke <i>device</i> milik aktor
<i>Pre Condition</i>	Aktor berhasil masuk dan sistem memperlihatkan halaman <i>Doctor's Report List</i>
<i>Main flow</i>	Aktor memencet tombol unduh berkas
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Berkas berhasil diunduh dan tersimpan di <i>device</i> milik aktor

Tabel 4.26 Skenario Use Case Menghapus Laporan Admin

Nama Use Case	Menghapus Laporan Admin
Aktor	Direktur
Tujuan	Spesifikasi <i>use case</i> ini dipakai direktur guna menghapus laporan yang diterima dari Admin
<i>Pre Condition</i>	Aktor berhasil masuk dan sistem memperlihatkan halaman <i>Admin's Report List</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memencet tombol berwarna merah bertuliskan <i>delete</i> 2. Aktor memencet tombol <i>delete</i> (yakin untuk menghapus)
<i>Alternative Flow</i>	Jika aktor memencet tombol <i>cancel</i> , maka laporan batal dihapus dan sistem akan memperlihatkan kembali halaman <i>View Admin's Report List</i>
<i>Post Condition</i>	Sistem kembali ke halaman <i>View Admin's Report List</i> (laporan berhasil dihapus)

Tabel 4.27 Skenario Use Case Menghapus Laporan Dokter

Nama Use Case	Menghapus Laporan Dokter
Aktor	Direktur
Tujuan	Spesifikasi <i>use case</i> ini dipakai direktur guna menghapus laporan yang diterima dari dokter
Pre Condition	Aktor berhasil masuk dan sistem menampilkan halaman <i>Doctor's Report List</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor memencet tombol berwarna merah bertuliskan <i>delete</i> 2. Aktor memencet tombol <i>delete</i> (yakin untuk menghapus)
Alternative Flow	Jika aktor memencet tombol <i>cancel</i> , maka laporan batal dihapus dan sistem akan c kembali halaman <i>View Doctor's Report List</i>
Post Condition	Sistem kembali ke halaman <i>View Doctor's Report List</i> (laporan berhasil dihapus)

Tabel 4.28 Skenario Use Case Melihat Laporan Admin

Nama Use Case	Melihat Laporan Admin
Aktor	Direktur
Tujuan	Spesifikasi <i>use case</i> ini dipakai direktur guna melihat laporan yang diterima dari Admin
Pre Condition	Aktor berhasil masuk dan sistem memperlihatkan beranda utama
Main Flow	Aktor memencet menu <i>View Admin's Report List</i> untuk melihat daftar laporan dari <i>admin</i>
Alternative Flow	-
Post Condition	Sistem memperlihatkan beranda <i>Admin's Report List</i>

Tabel 4.29 Skenario Use Case Melihat Laporan Dokter

Nama Use Case	Melihat Laporan Dokter
Aktor	Direktur
Tujuan	Spesifikasi <i>use case</i> ini digunakan direktur untuk melihat laporan yang diterima dari dokter
Pre Condition	Aktor berhasil masuk dan sistem memperlihatkan beranda utama

<i>Main Flow</i>	Aktor memencet menu <i>View Doctor's Report List</i> untuk melihat daftar laporan dari dokter
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem memperlihatkan beranda <i>Doctor's Report List</i>

Tabel 4.30 Skenario Use Case Menambah Admin

Nama Use Case	Menambah Admin
Aktor	Direktur
Tujuan	Spesifikasi <i>use case</i> ini dipakai direktur guna menambah data <i>admin</i>
<i>Pre Condition</i>	Sistem memperlihatkan halaman <i>Admin Registration</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengisi data diri admin yang mencakup nama, <i>username</i>, <i>password</i>, alamat, telepon, serta NIK 2. Aktor memencet tombol <i>admin registration</i> untuk menyimpan data diri <i>admin</i>
<i>Alternative Flow</i>	Jika aktor tidak mengisi keseluruhan kolom yang ada, maka sistem akan memperlihatkan pesan peringatan bahwa ada kolom yang belum terisi
<i>Post Condition</i>	Sistem memperlihatkan pemberitahuan bahwa <i>admin</i> berhasil ditambahkan

Tabel 4.31 Skenario Use Case Menambah Dokter

Nama Use Case	Menambah Dokter
Aktor	Direktur
Tujuan	Spesifikasi <i>use case</i> ini dipakai direktur guna menambah data dokter
<i>Pre Condition</i>	Sistem memperlihatkan halaman <i>Doctor Registration</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengisi data diri dokter yang mencakup nama, <i>username</i>, <i>password</i>, alamat, nomor telepon, SIP, masa berlaku SIP, serta NIK 2. Aktor memencet tombol <i>admin registration</i> untuk menyimpan data diri <i>admin</i>
<i>Alternative Flow</i>	Jika aktor tidak mengisi seluruh kolom yang ada, maka sistem akan memperlihatkan notifikasi peringatan bahwa ada kolom yang belum terisi
<i>Post Condition</i>	Sistem memperlihatkan pemberitahuan bahwa dokter berhasil ditambahkan

Tabel 4.32 Skenario Use Case Menghapus Admin

Nama Use Case	Menghapus Admin
Aktor	Direktur
Tujuan	Spesifikasi <i>use case</i> ini dipakai direktur guna menghapus data <i>admin</i>
Pre Condition	Sistem memperlihatkan halaman <i>Admin's List</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor memencet tombol berwarna merah yang bertuliskan <i>delete</i> untuk menghapus data <i>admin</i> 2. Sistem memperlihatkan peringatan yakin untuk menghapus atau tidak 3. Aktor menekan tombol <i>delete</i> (yakin untuk menghapus)
Alternative Flow	Jika aktor memencet tombol <i>cancel</i> , maka sistem batal menghapus data <i>admin</i> dan sistem kembali ke beranda <i>Admin's List</i>
Post Condition	Sistem kembali ke beranda <i>Admin's List</i> (data berhasil dihapus)

Tabel 4.33 Skenario Use Case Menghapus Dokter

Nama Use Case	Menghapus Dokter
Aktor	Direktur
Tujuan	Spesifikasi <i>use case</i> ini dipakai direktur guna menghapus data diri dokter
Pre Condition	Sistem memperlihatkan halaman <i>Doctor's List</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor memencet tombol berwarna merah yang bertuliskan <i>delete</i> guna menghapus data dokter 2. Sistem memperlihatkan peringatan yakin untuk menghapus atau tidak 3. Aktor menekan tombol <i>delete</i> (yakin untuk menghapus)
Alternative Flow	Jika aktor memencet tombol <i>cancel</i> , maka sistem batal menghapus data dokter dan sistem kembali ke beranda <i>Doctor's List</i>
Post Condition	Sistem kembali ke halaman <i>Doctor's List</i> (data telah dihapus)

Tabel 4.34 Skenario Use Case Melihat Masa Berlaku SIP Dokter

Nama Use Case	Melihat Masa Berlaku SIP Dokter
Aktor	Direktur



Tujuan	Spesifikasi <i>use case</i> ini dipakai direktur guna melihat masa berlaku SIP seluruh dokter yang ada di dalam sistem
<i>Pre Condition</i>	Sistem memperlihatkan halaman utama
<i>Main Flow</i>	Aktor memencet menu <i>View Notification</i>
<i>Alternative Flow</i>	
<i>Post Condition</i>	Sistem memperlihatkan halaman <i>Doctor's Notification List</i> yang berisi informasi/peringatan mengenai masa berlaku SIP milik seluruh dokter

Tabel 4.35 Skenario *Use Case* Mengubah Admin

Nama <i>Use Case</i>	Mengubah Admin
Aktor	Direktur
Tujuan	Spesifikasi <i>use case</i> ini dipakai direktur guna mengubah data diri <i>admin</i>
<i>Pre Condition</i>	Sistem memperlihatkan halaman <i>Admin's List</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memencet tombol berwarna kuning yang bertuliskan <i>edit</i> untuk mengubah data <i>admin</i> 2. Sistem memperlihatkan beranda <i>Edit Admin</i> 3. Aktor mengganti data <i>admin</i>, lalu memencet tombol <i>update admin</i>
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem memperlihatkan pemberitahuan bahwa perubahan berhasil disimpan, lalu sistem kembali ke halaman <i>Admin's List</i>

Tabel 4.36 Skenario *Use Case* Mengubah Dokter

Nama <i>Use Case</i>	Mengubah Dokter
Aktor	Direktur
Tujuan	Spesifikasi <i>use case</i> ini dipakai direktur guna mengubah data diri dokter
<i>Pre Condition</i>	Sistem memperlihatkan halaman <i>Doctor's List</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memencet tombol berwarna kuning yang bertuliskan <i>edit</i> untuk mengubah data dokter 2. Sistem memperlihatkan halaman <i>Edit Doctor</i> 3. Aktor mengganti data dokter, lalu memencet tombol <i>update doctor</i>
<i>Alternative Flow</i>	-

<i>Post Condition</i>	Sistem memperlihatkan pemberitahuan bahwa perubahan berhasil disimpan, lalu sistem kembali ke halaman <i>Doctor's List</i>
-----------------------	--

Tabel 4.37 Skenario Use Case Melihat Daftar Admin

Nama Use Case	Melihat Daftar Admin
Aktor	Direktur
Tujuan	Spesifikasi use case ini digunakan direktur untuk melihat daftar admin
Pre Condition	Sistem memperlihatkan halaman utama
Main Flow	Aktor memencet menu <i>View Admin's List</i>
Alternative Flow	-
Post Condition	Sistem memperlihatkan beranda <i>Admin's List</i>

Tabel 4.38 Skenario Use Case Melihat Daftar Dokter

Nama Use Case	Melihat Daftar Dokter
Aktor	Direktur
Tujuan	Spesifikasi kasus pengujian ini dipakai direktur guna melihat daftar dokter
Pre Condition	Sistem memperlihatkan halaman utama
Main Flow	Aktor memencet menu <i>View Doctor's List</i>
Alternative Flow	-
Post Condition	Sistem memperlihatkan halaman <i>Doctor's List</i>

Tabel 4.39 Skenario Use Case Mencari Admin

Nama Use Case	Mencari Admin
Aktor	Direktur
Tujuan	Spesifikasi kasus pengujian ini dipakai direktur guna mencari data admin
Pre Condition	Aktor berhasil masuk dan sistem memperlihatkan halaman <i>View Admin's List</i>
Main Flow	Aktor meletakkan kursor di kolom pencarian di sebelah kiri atas halaman, lalu mengetikkan nomor ID admin, dan kemudian menekan tombol <i>enter</i> pada keyboard

<i>Alternative Flow</i>	Jika aktor menetik nomor ID yang tidak terdaftar sebagai <i>admin</i> lalu menekan tombol <i>enter</i> , maka sistem akan memperlihatkan halaman kosong (tidak ditemukan data)
<i>Post Condition</i>	Sistem memperlihatkan data <i>admin</i> yang dicari

Tabel 4.40 Skenario Use Case Mencari Dokter

Nama Use Case	Mencari Dokter
Aktor	Direktur
Tujuan	Spesifikasi kasus pengujian ini dipakai direktur guna mencari data dokter
<i>Pre Condition</i>	Aktor berhasil masuk dan sistem menampilkan halaman <i>View Doctor's List</i>
<i>Main Flow</i>	Aktor meletakkan kursor di kolom pencarian di sebelah kiri atas halaman, lalu menetik nomor ID dokter, dan kemudian menekan tombol <i>enter</i> pada <i>keyboard</i>
<i>Alternative Flow</i>	Jika aktor menetik nomor ID yang tidak terdaftar sebagai dokter lalu memencet tombol <i>enter</i> , maka sistem akan memperlihatkan halaman kosong (tidak ditemukan data)
<i>Post Condition</i>	Sistem memperlihatkan data diri dokter yang dicari

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Perancangan akan dilakukan dan digunakan sebagai dasar sebelum melakukan pengimplementasian sistem. Perancangan yang akan digunakan di tahap ini ialah pembuatan rancangan *Sequence Diagram*, rancangan *Class Diagram*, rancangan *Entity Relationship Diagram (ERD)*, rancangan data, dan rancangan antarmuka.

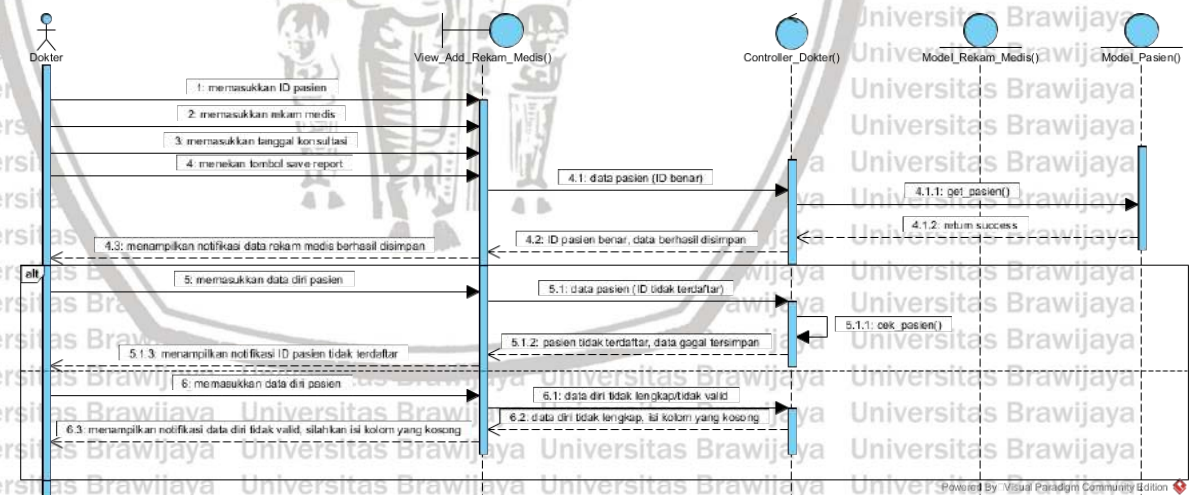
5.1 Perancangan Sistem

Pembentukan desain akan segera dilaksanakan sesudah berakhirnya pengerjaan proses penganalisisan kebutuhan. Berdasarkan hasil dari penganalisisan kebutuhan yang telah dilaksanakan, perancangan akan dikerjakan. Proses perancangan sistem rekam medis Ramdani *Skincare & Spa* ini terbagi ke beberapa tahapan yaitu melakukan pembuatan desain *sequence diagram*, desain *class diagram*, data, serta desain antarmukanya.

5.1.1 Perancangan *Sequence Diagram*

Diagram *sequence* menggambarkan alur urutan proses kejadian serta bagaimana sistem berjalan. Didapatkannya objek-objek sebagai hasil pengidentifikasian kebutuhan dan scenario *use case*. Setiap *use case* yang ada dijelaskan satu persatu oleh setiap diagram *sequence*. Lalu diperoleh 15 diagram *sequence*. Pada Gambar 5.1.1.1 sampai Gambar 5.1.1.3 dijelaskan mengenai perancangan tersebut.

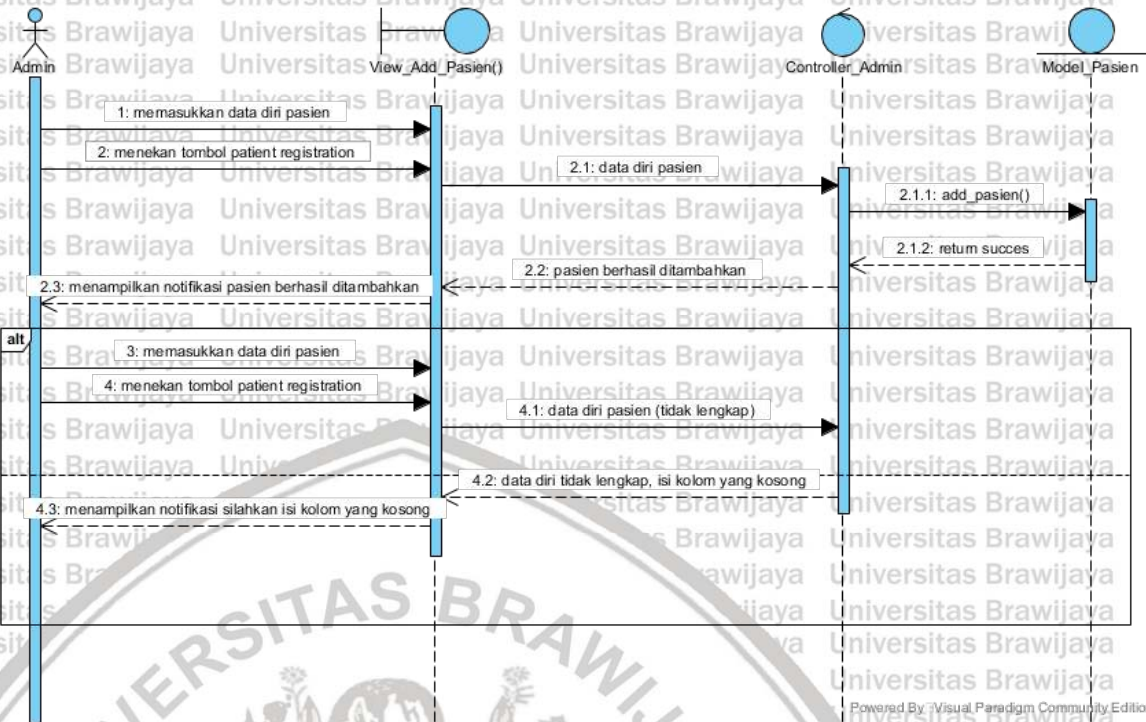
5.1.1.1 *Sequence Diagram* Menambah Data Rekam Medis



Gambar 5.1 *Sequence Diagram* Menambah Data Rekam Medis

Pada *sequence diagram* Menambah Data Rekam Medis, aktor yang memiliki akses adalah dokter. Aktor dapat melakukan penambahan data rekam medis pasien dengan memasukkan ID pasien yang telah terdaftar dan mengisikan riwayat pengobatan serta tanggal konsultasi. Lalu, aktor menekan tombol *save report* dan kemudian akan keluar pemberitahuan penyimpanan data rekam medis berhasil atau tidak.

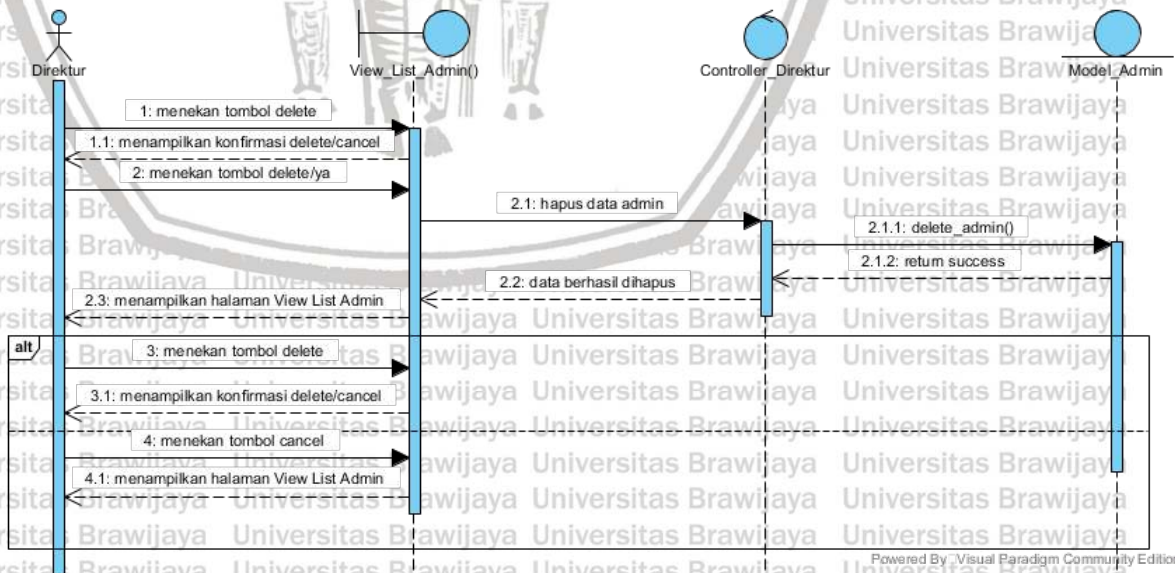
5.1.1.2 Sequence Diagram Menambah Pasien



Gambar 5.2 Sequence Diagram Menambah Pasien

Pada *sequence diagram* Menambah Pasien, aktor yang terlibat adalah dokter. Aktor memasukkan data diri pasien sebagai syarat pendaftaran pada *form* kemudian menekan tombol *patient registration*. Jika berhasil, akan keluar notifikasi pasien berhasil ditambahkan. Jika gagal, akan keluar notifikasi gagal.

5.1.1.3 Sequence Diagram Menghapus Admin



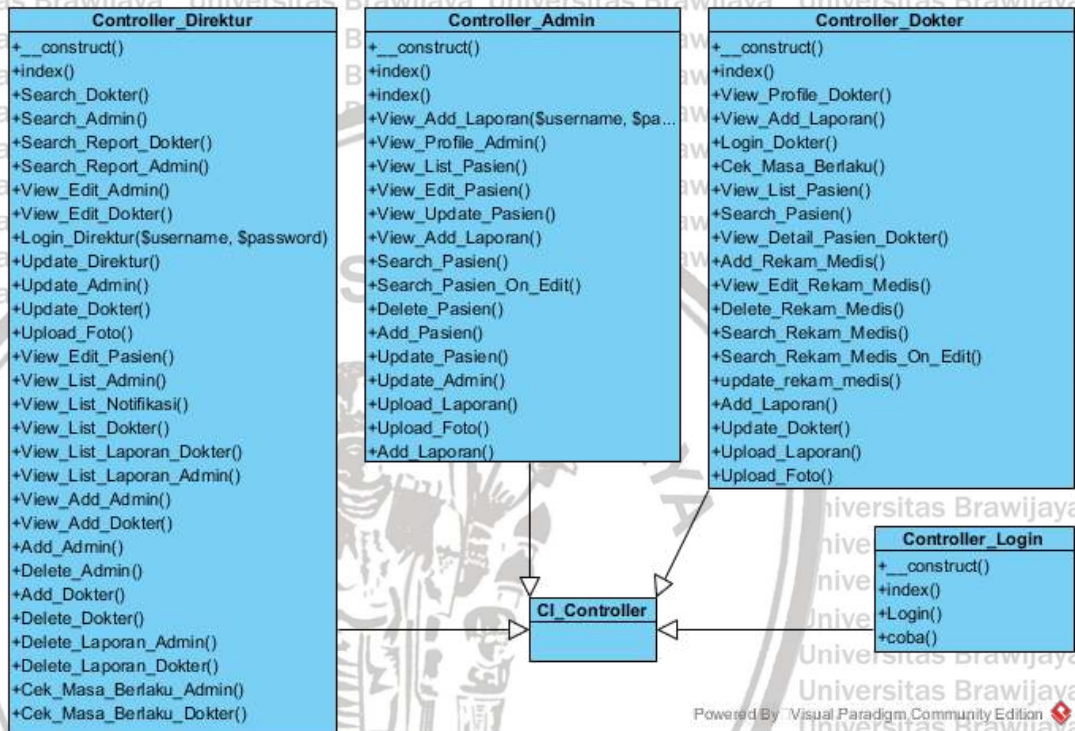
Gambar 5.3 Sequence Diagram Menghapus Admin

Aktor yang terlibat adalah direktur. Untuk menghapus *admin*, maka aktor harus memasukkan ID *admin* dan menekan tombol *enter* agar hasil pencarian

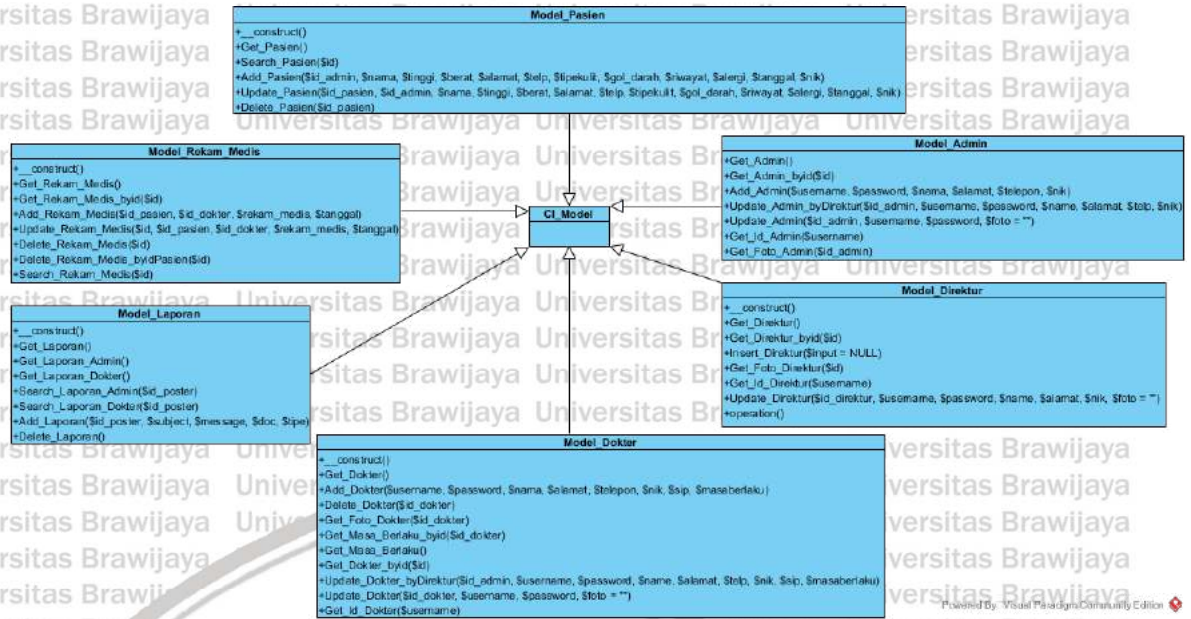
keluar. Lalu aktor memencet tombol *delete* di nama *admin* yang hendak dihapus. Sistem akan memperlihatkan kembali halaman *View Admin's List* sebagai hasil akhir.

5.1.2 Perancangan Class Diagram

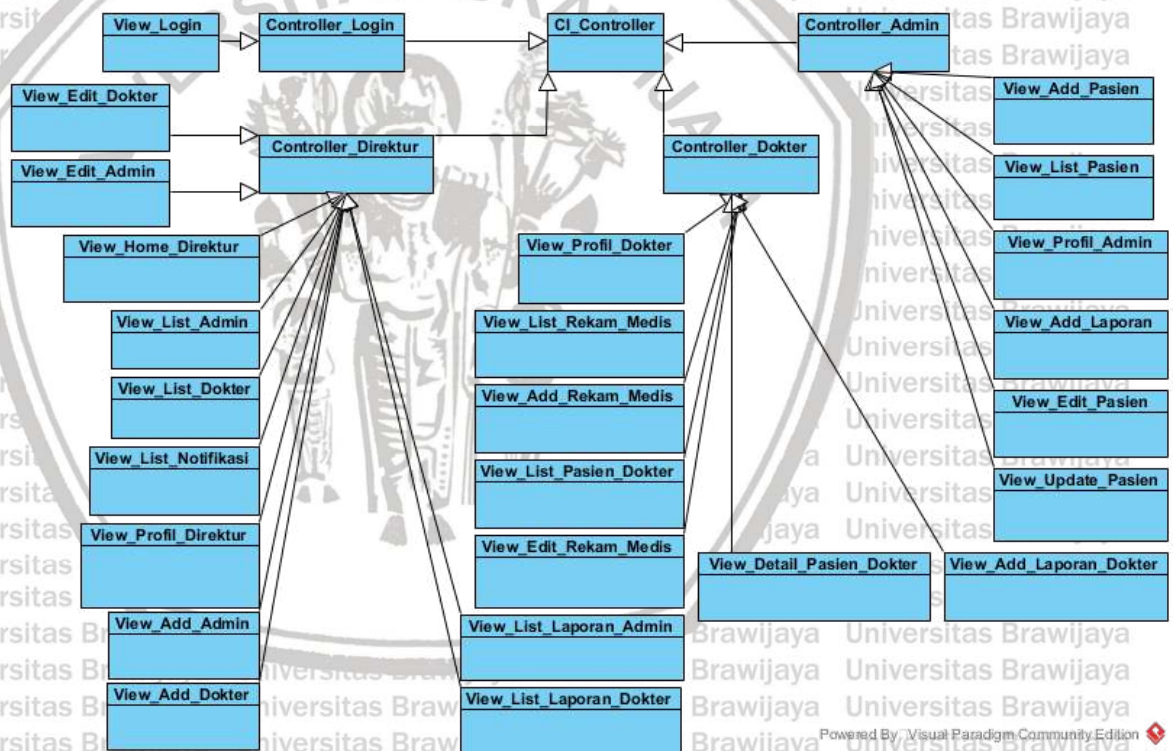
Gambaran berbagai objek yang ada serta koneksi antar objek dijelaskan pada perancangan diagram kelas. *Class diagram* dibagi menjadi tiga perancangan, yaitu rancangan diagram kelas pada *controller*, pada *model*, serta pada *view* seraya memperlihatkan atribut dan operasinya.



Gambar 5.4 Perancangan Class Diagram pada Controller



Gambar 5.5 Perancangan Class Diagram pada Model

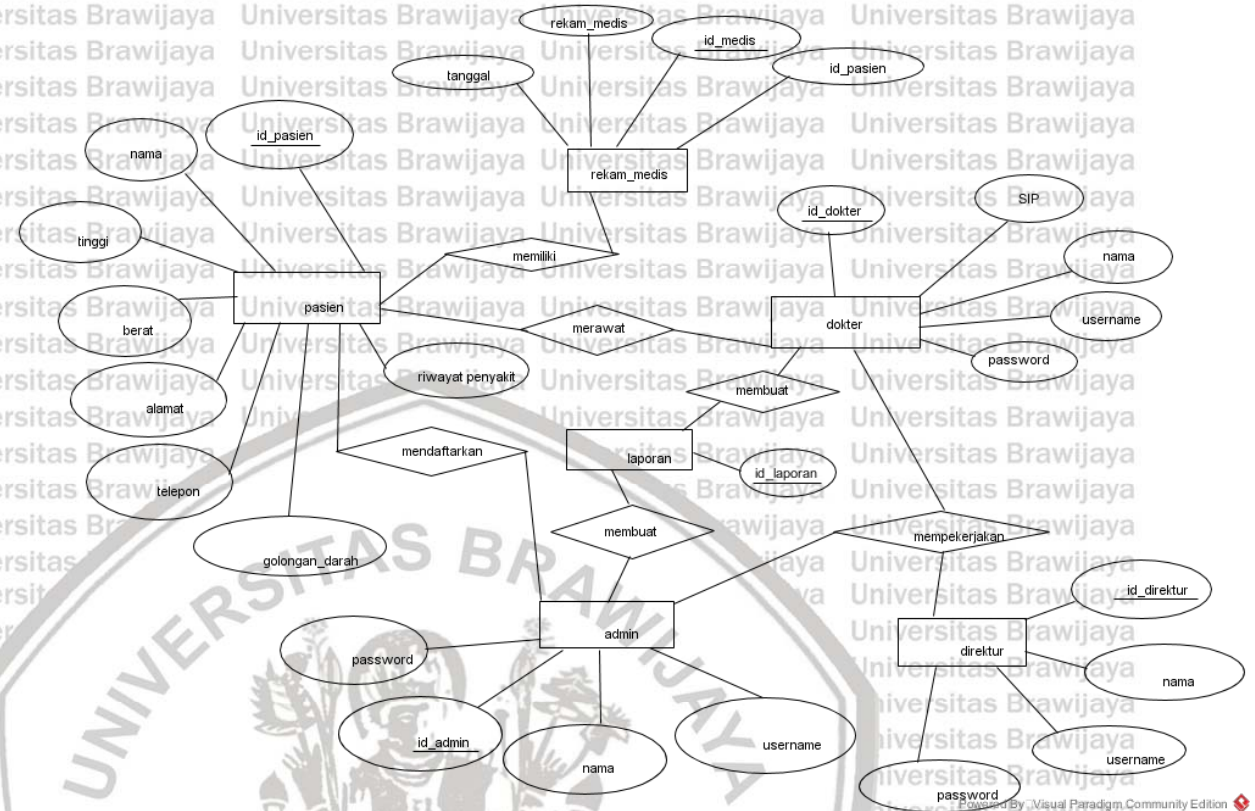


Gambar 5.6 Perancangan Class Diagram pada View

5.1.3 Perancangan ERD (Entity Relationship Diagram)

Basis data digunakan pada sistem ini guna menyimpan data-data rekam medis. Berdasarkan penjelasan tersebut maka dibutuhkan rancangan basis data demi memperlancar proses pembangunan basis data sesuai dengan yang dibutuhkan oleh sistem. Database sistemnya menggunakan MySQL. Ada sejumlah enam tabel yang diterapkan pada sistem, yaitu tabel dengan nama pasien, admin, dokter,

direktur, laporan, dan rekam_medis. Perancangan basis data ERD pada sistem ini mewakili relasi/koneksi antar materi. Desain ERD sistem diterapkan di Gambar 5.7.

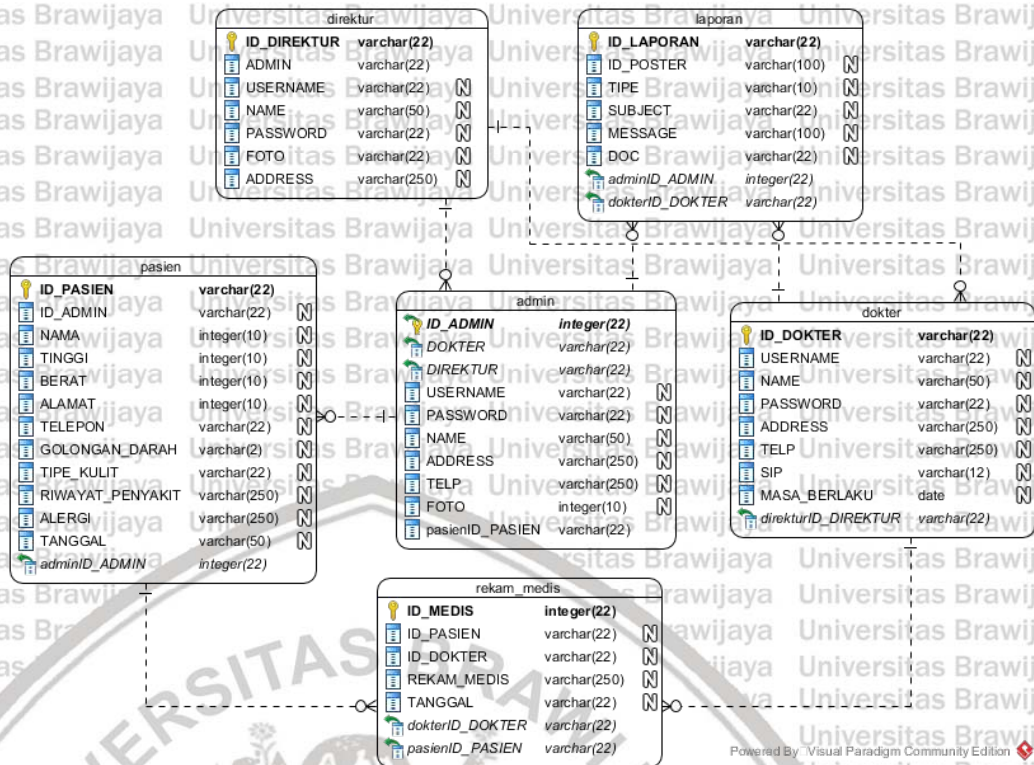


Gambar 5.7 Perancangan Entity Relationship Diagram

5.1.4 Perancangan Data

Tujuan dari pemodelan ini yaitu untuk menyambungkan koneksi antar materi pada sistem. *Physical Data Model* digunakan dalam perancangan ini demi mewakili desain basis data dan menentukan struktur tabelnya, seperti nama setiap kolomnya, tipe data setiap kolomnya, adanya *primary key*, adanya *foreign key*, serta koneksi antar tabel.

Pada perancangan data ini juga akan digambarkan dalam bentuk tabel demi menghasilkan struktur data penyimpanan untuk diimplementasikan pada sistem rekam medis Ramdani Skincare & Spa. Perancangan database sistem rekam medis Ramdani Skincare & Spa ini menggunakan empat entitas yang membentuk sistem. Berikut merupakan atribut dari setiap entitas pada *database* sistem rekam medis Ramdani Skincare & Spa yang akan dijabarkan pada Tabel 5.1 sampai Tabel 5.6. Serta desain *Physical Data Model* sistem rekam medis Ramdani Skincare & Spa yang akan digambarkan di Gambar 5.8.



Gambar 5.8 Perancangan Physical Data Model

Tabel 5.1 Tabel Admin

Nama Atribut	Tipe	Keterangan
id_admin	int(22)	Primary Key dari admin
nama	varchar(50)	Nama admin
username	varchar(22)	Username admin
password	varchar(22)	Password admin
address	varchar(250)	Alamat admin
telp	varchar(250)	Nomor telepon admin
foto	int(10)	Foto admin

Tabel *admin* menyimpan data-data perancangan tabel admin pada database dengan id_admin sebagai Primary Key dari tabel admin.

Tabel 5.2 Tabel Pasien

Nama Atribut	Tipe	Keterangan
id_pasien	varchar(22)	Primary Key dari pasien
id_admin	varchar(22)	Foreign Key dari admin
nama	text	Nama pasien
tinggi	int(200)	Tinggi badan pasien
berat	int(200)	Berat badan pasien
alamat	text	Alamat pasien
telepon	int(15)	Nomor telepon pasien
golongan_darah	varchar(2)	Golongan darah pasien
tipe_kulit	varchar(22)	Jenis kulit pasien

Tabel pasien tersebut di atas dengan id_pasien sebagai *Primary Key* dari tabel pasien menyimpan data-data perancangan tabel pasien.

Tabel 5.3 Tabel Rekam Medis

Nama Atribut	Tipe	Keterangan
id_medis	int(22)	<i>Primary Key</i> dari rekam medis
id_pasien	varchar(22)	<i>Foreign Key</i> dari tabel pasien
id_dokter	varchar(22)	Nomor ID dokter
rekam_medis	varchar(250)	Rekam medis pasien
tanggal	varchar(22)	Tanggal konsultasi

Tabel rekam_medis tersebut di atas dengan id_medis sebagai *Primary Key* dari tabel rekam medis menyimpan data-data jejak rekam medis dan riwayat penyakit serta riwayat pengobatan pasien yang dimasukkan oleh dokter.

Tabel 5.4 Tabel Direktur

Nama Atribut	Tipe	Keterangan
id_direktur	varchar(22)	<i>Primary key</i> dari direktur
username	varchar(22)	<i>Username</i> direktur
nama	varchar(50)	Nama direktur
password	varchar(22)	<i>Password</i> direktur
foto	varchar(22)	Foto direktur
address	varchar(250)	Alamat direktur

Tabel direktur tersebut di atas dengan id_direktur sebagai *Primary Key* dari tabel direktur menyimpan data-data perancangan tabel direktur.

Tabel 5.5 Tabel Dokter

Nama Atribut	Tipe	Keterangan
id_dokter	varchar(22)	<i>Primary key</i> dari dokter
username	varchar(22)	<i>Username</i> dokter
name	varchar(50)	Nama dokter
password	varchar(22)	<i>Password</i> dokter
address	varchar(250)	Alamat dokter
telp	varchar(250)	Nomor telepon dokter
SIP	varchar(12)	Nomor SIP dokter
masa_berlaku	Date	Masa berlaku SIP dokter

Tabel dokter tersebut di atas dengan id_dokter sebagai *Primary Key* dari tabel dokter menyimpan data-data perancangan tabel dokter.

Tabel 5.6 Tabel Laporan

Nama Atribut	Tipe	Keterangan
id_laporan	varchar(22)	<i>Primary key</i> dari laporan
id_poster	varchar(22)	
tipe	varchar(22)	
subject	varchar(22)	Judul laporan

Nama Atribut	Tipe	Keterangan
message	varchar(100)	Isi laporan
doc	varchar(22)	Dokumen laporan

Tabel laporan tersebut di atas dengan id_laporan sebagai *Primary Key* dari tabel laporan menyimpan data-data perancangan tabel laporan.

5.1.5 Perancangan Algoritme

Perancangan algoritme ini membahas perihal rancangan algoritme sebuah sistem diagram kelas yang mana di setiap kelasnya memiliki fungsi algoritme.

Tujuan dilakukannya perancangan algoritme yaitu demi mendapatkan penjabaran algoritme pada fungsi dari *class*.

5.1.5.1 Algoritme Menambah Data Rekam Medis

Tabel 5.7 ialah rancangan algoritme menambah data rekam medis. Pengimplementasian algoritme ini ada pada fungsi Add_Rekam_Medis() yang di dalam kelas Controller_Dokter.

Tabel 5.7 Tabel Algoritme Menambah Data Rekam Medis

1	Mulai
2	Memasukkan ID pasien
3	Memasukkan data rekam medis pasien
4	Memasukkan tanggal konsultasi
5	If (ID = true rekam medis = not null tanggal = not null)
6	Memanggil method add_rekam_medis() untuk menyimpan data
7	Else
8	If (ID = false rekam medis = not null tanggal = not null)
9	Menampilkan peringatan "ID pasien tidak terdaftar"
10	Else
11	Menampilkan pemberitahuan "data diri tidak lengkap, isi kolom yang kosong"
12	End if
13	End if
14	End

5.1.5.2 Algoritme Menambah Pasien

Tabel 5.8 ialah perancangan algoritme menambah pasien. Pengaplikasian algoritme ini ada pada fungsi Add_Pasien() di dalam kelas Controller_Admin.

Tabel 5.8 Tabel Algoritme Menambah Pasien

1	Mulai
---	--------------

2	Memasukkan ID pasien, nama, alamat, tinggi badan, berat badan, golongan darah, tipe kulit, riwayat penyakit, telpon, alergi, tanggal konsultasi/mendaftar, tempat dan tanggal lahir, dan NIK
3	If (semua kolom terisi)
4	Menampilkan pemberitahuan pendaftaran pasien berhasil
5	Else
6	Menampilkan pemberitahuan pendaftaran pasien gagal
7	End If
8	End

5.1.5.3 Algoritme Menghapus Admin

Tabel 5.9 ialah perancangan menghapus admin. Algoritme ini diimplementasikan pada fungsi Delete_Admin() pada kelas Controller_Direktur.

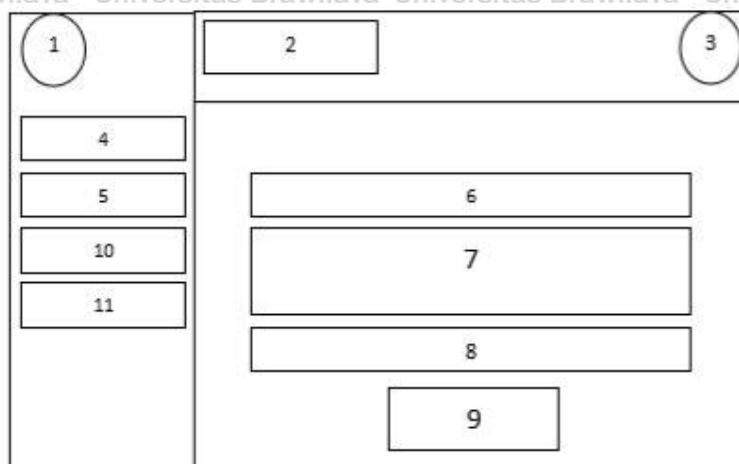
Tabel 5.9 Tabel Algoritme Menghapus Admin

1	Mulai
2	Memasukkan ID admin
3	Menekan tombol enter
4	Menekan tombol delete
5	If (ID admin = terdaftar/benar)
6	Admin terhapus dan kembali ke halaman utama
7	Else
8	Menampilkan peringatan ada yang salah, silahkan coba lagi
9	End If
10	End

5.1.6 Perancangan Antarmuka

Perancangan antarmuka yakni rancangan manifestasi dari sistem yang nantinya dilihat oleh pengguna. Ada beberapa sampel antarmuka sistem yang ditunjukkan dan diterangkan pada Gambar 5.9 sampai Gambar 5.11 dan Tabel 5.10 sampai Tabel 5.12.

5.1.6.1 Perancangan Antarmuka Halaman Menambah Data Rekam Medis



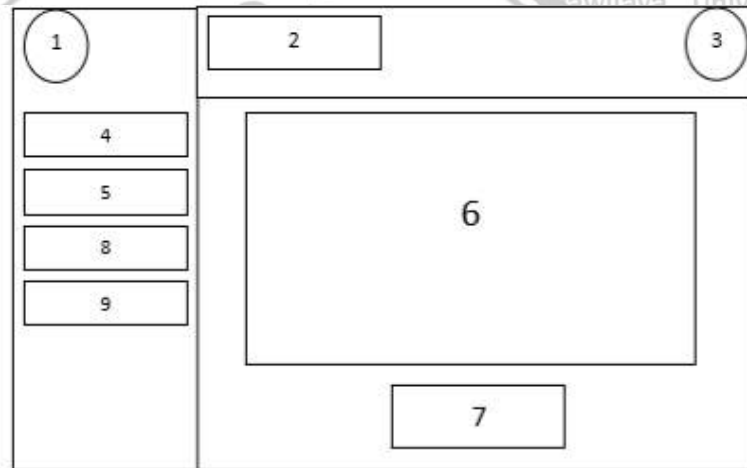
Gambar 5.9 Perancangan Antarmuka Halaman Menambah Data Rekam Medis

Tabel 5.10 Tabel Penjelasan Perancangan Antarmuka Halaman Menambah Data Rekam Medis

No.	Nama Objek	Tipe	Keterangan
1.	Logo aplikasi	Logo <i>button</i>	Memperlihatkan logo aplikasi
2.	Cari	<i>Search field</i>	Memperlihatkan kolom pencarian
3.	Profil	<i>Button</i>	Memperlihatkan tombol profil
4.	<i>Add Medical History</i>	<i>Sidebar menu</i>	Memperlihatkan menu registrasi tambah data rekam medis
5.	<i>View Medical History</i>	<i>Sidebar menu</i>	Memperlihatkan menu lihat data rekam medis
6.	<i>Form Add Medical History (Patient's ID)</i>	<i>Form</i>	Memperlihatkan kolom untuk mengisi ID pasien
7.	<i>Form Add Medical History (Medical Report)</i>	<i>Form</i>	Memperlihatkan kolom untuk mengisi rekam medis pasien
8.	<i>Form Add Medical History (Date)</i>	<i>Form</i>	Memperlihatkan kolom untuk mengisi tanggal konsultasi pasien
9.	<i>Save Report</i>	<i>Button</i>	Memperlihatkan tombol simpan laporan medis

No.	Nama Objek	Tipe	Keterangan
10.	<i>View Patient</i>	<i>Sidebar menu</i>	Memperlihatkan menu lihat daftar pasien
11.	<i>Make Report</i>	<i>Sidebar menu</i>	Memperlihatkan menu membuat laporan kepada direktur

5.1.6.2 Perancangan Antarmuka Halaman Menambah Pasien



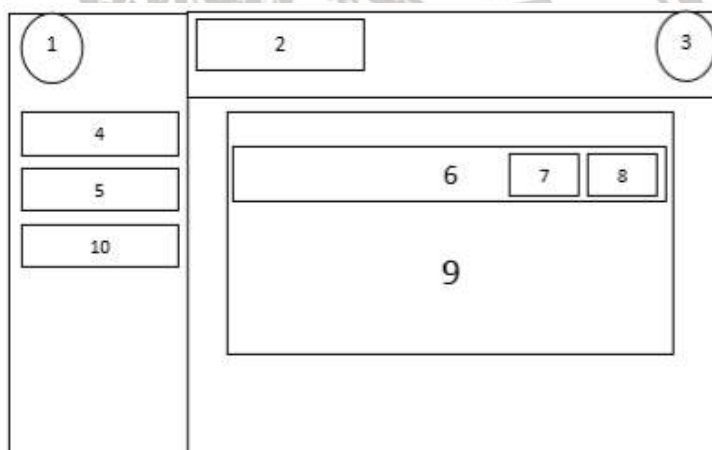
Gambar 5.10 Perancangan Antarmuka Halaman Menambah Pasien

Tabel 5.11 Tabel Penjelasan Perancangan Antarmuka Halaman Menambah Pasien

No.	Nama Objek	Tipe	Keterangan
1.	Logo aplikasi	Logo <i>button</i>	Memperlihatkan logo aplikasi
2.	Cari	<i>Search field</i>	Memperlihatkan kolom pencarian
3.	Profil	<i>Button</i>	Menampilkan tombol profil
4.	<i>Patient Registration</i>	<i>Sidebar menu</i>	Menampilkan menu registrasi pasien
5.	<i>View Patient's List</i>	<i>Sidebar menu</i>	Menampilkan menu lihat daftar pasien

No.	Nama Objek	Tipe	Keterangan
6.	<i>Form Reistration Patient's</i>	<i>Form</i>	Menampilkan kolom untuk mengisi data diri pasien yang terdiri dari nama, NIK, alamat, tempat dan tanggal lahir, nomor telepon, tinggi dan berat badan, golongan darah, jenis kulit, alergi, riwayat penyakit, dan tanggal konsultasi
7.	<i>Registration Patient</i>	<i>Button</i>	Menampilkan tombol untuk menyimpan data diri pasien
8.	<i>Edit Patient</i>	<i>Sidebar menu</i>	Menampilkan menu ubah pasien
9.	<i>Make Report</i>	<i>Sidebar menu</i>	Menampilkan menu membuat laporan kepada direktur

5.1.6.3 Perancangan Antarmuka Halaman Delete Admin



Gambar 5.11 Perancangan Antarmuka Halaman Delete Admin

Tabel 5.12 Tabel Penjelasan Perancangan Antarmuka Halaman Delete Admin

No.	Nama Objek	Tipe	Keterangan
1.	Logo aplikasi	Logo button	Memperlihatkan logo aplikasi
2.	Cari	Search field	Memperlihatkan kolom pencarian
3.	Profil	Button	Memperlihatkan tombol profil
4.	Menu Admin	Sidebar menu	Memperlihatkan menu admin
5.	Menu Dokter	Sidebar menu	Memperlihatkan menu dokter

No.	Nama Objek	Tipe	Keterangan
6.	<i>Form Admin's Data</i>	<i>Form</i>	Memperlihatkan kolom berisi data diri <i>admin</i>
7.	<i>Delete</i>	<i>Button</i>	Memperlihatkan tombol untuk menghapus <i>admin</i>
8.	<i>Edit</i>	<i>Button</i>	Memperlihatkan tombol untuk mengubah data diri <i>admin</i>
9.	<i>List Akun Admin</i>	<i>Sidebar menu</i>	Memperlihatkan halaman daftar <i>admin</i>
10.	Notifikasi	<i>Sidebar menu</i>	Memperlihatkan menu notifikasi

5.2 Implementasi Sistem

Implementasi ini berisi mengenai detail pengimplementasian sistem rekam medis Ramdani Skincare & Spa berdasarkan metodologi dan perancangannya.

5.2.1 Spesifikasi Sistem

Dalam menerapkan hasil desain rancangan untuk mewujudkan sistem ini, dibutuhkan dukungan dari perangkat keras dan lunak.

5.2.1.1 Spesifikasi Perangkat Keras

Detail spesifikasi perangkat keras yang diperlukan dan digunakan untuk mengoperasikan sistem didefinisikan pada Tabel 5.13.

Tabel 5.13 Tabel Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
<i>Model</i>	DELL Inspiron 11 3162
<i>Processor</i>	Intel Celeron <i>Processor</i> N3060
<i>Memory Size</i>	2GB DDR3L 1600MHz
<i>Graphics Processor</i>	Intel HD <i>Graphics</i>
<i>Hard Drive Size</i>	500 GB 5400 RPM

5.2.1.2 Spesifikasi Perangkat Lunak

Detail spesifikasi perangkat lunak yang digunakan untuk menjalankan sistem dijelaskan pada Tabel 5.14.

Tabel 5.14 Tabel Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
<i>Operating System</i>	<i>Windows 10 Pro 64-bit</i>

Nama Komponen	Spesifikasi
<i>Programming Language</i>	<i>PHP, Javascript, CSS, HTML</i>
<i>Programming Environment</i>	<i>Sublime Text 1.4</i>
<i>Framework</i>	<i>CodeIgniter</i>
<i>Database Management System</i>	<i>MySQL versi 5.0.12</i>

5.3 Implementasi Algoritme

Implementasi Algoritme adalah pengubahan algoritme sehingga berubah menjadi kode program yang dapat dimengerti komputer. Pada sistem ini, pembuatan kode menggunakan *PHP*. Berikut merupakan sampel kode program yang digunakan.

5.3.1 Implementasi Algoritme Menambah Data Rekam Medis

Implementasi algoritme menambah data rekam medis pasien dari algoritme fungsi `Add_Rekam_Medis()` pada kelas `Controller_Dokter` dengan bahasa pemrograman *PHP*.

```

Algoritme Menambah Data Rekam Medis
1 public function Add_Rekam_Medis() {
2     $id_dokter = $this->session->userdata('id');
3     $id_pasien = $this->input->post('id_pasien');
4     $rekam_medis = $this->input->
5     post('rekam_medis');
6     $tanggal = $this->input->post('tanggal');
7     if($id_pasien == "" || $rekam_medis == "" || $tanggal ==
8     "" ) {
9         print"<script type=\"text/javascript\">
10    alert('Please Complete the empty textbox');</script>";
11    $this->index();
12    }
13    else {
14        $get_pasien = $this->Model_Pasien->
15    >Get_Pasien_byid($id_pasien);
16    // $cek_pasien = $get_pasien->num_rows();
17    $query = $this->db->query("SELECT * FROM PASIEN
18    WHERE ID_PASIEN = $id_pasien");
19    $cek_pasien = $query->num_rows();
20    if($cek_pasien == 0){
21        print"<script type=\"text/javascript\">alert
22    ('Patient Id not Register');</script>";
23    $this->index();
24    }
25    if($cek_pasien != 0){
26        $save_rekam_medis = $this->Model_Rekam_Medis->
27    >
28    Add_Rekam_Medis($id_pasien,$id_dokter,$rekam_medis,$tanggal);
29    if($save_rekam_medis){
30        print"<script
31    type=\"text/javascript\">alert
32    ('Update Medical History Was Succesfull!');</script>";
33    $this->index();
34    }
35    else {
36        print"<script
37    type=\"text/javascript\">alert

```

```

38 ('Update Medical History Failed');</script>";
39 $this->index();
40 }
41 }
42 }
43 }
    
```

5.3.2 Implementasi Algoritme Menambah Pasien

Pengimplementasi algoritme menambah pasien dari algoritme fungsi Add_Pasien() pada kelas Controller_Admin dengan bahasa pemrograman PHP.

```

Algoritme Menambah Pasien
1 public function Add_Pasien(){
2     $id_admin = $this->session->userdata('id');
3     $nama = $this->input->post('nama');
4     $tinggi = $this->input->post('tinggi');
5     $berat = $this->input->post('berat');
6     $alamat = $this->input->post('alamat');
7     $telp = $this->input->post('telepon');
8     $gol_darah = $this->input->post('gol_darah');
9     $tipekulit = $this->input->post('tipe_kulit');
10    $riwayat = $this->input->post('riwayat');
11    $alergi = $this->input->post('alergi');
12    $tanggal = $this->input->post('tanggal');
13    $tempat_lahir = $this->input->post('tempat_lahir');
14    $tanggal_lahir = $this->input->post('tanggal_lahir');
15    $nik = $this->input->post('nik');
16    $add_pasien = $this->Model_Pasien-
17    >Add_Pasien($id_admin,$nama,$tinggi,$berat,$alamat,$telp,$tipekulit
18    ,$gol_darah,$riwayat,$alergi,$tanggal,$tempat_lahir,$tanggal_lahir,
19    $nik);
20    if($add_pasien){
21        print"<script type=\"text/javascript\">alert('Patient
22    Register Have Been Succses');</script>";
23        $this->index();
24    }
25    else{
26        print"<script type=\"text/javascript\">alert('Patient
27    Register Not Succses');</script>";
28    }
29    }
    
```

5.3.3 Implementasi Algoritme Menghapus Admin

Implementasi algoritme menghapus admin dari algoritme fungsi Delete_Admin() pada kelas Controller_Direktur dengan bahasa pemrograman PHP.

```

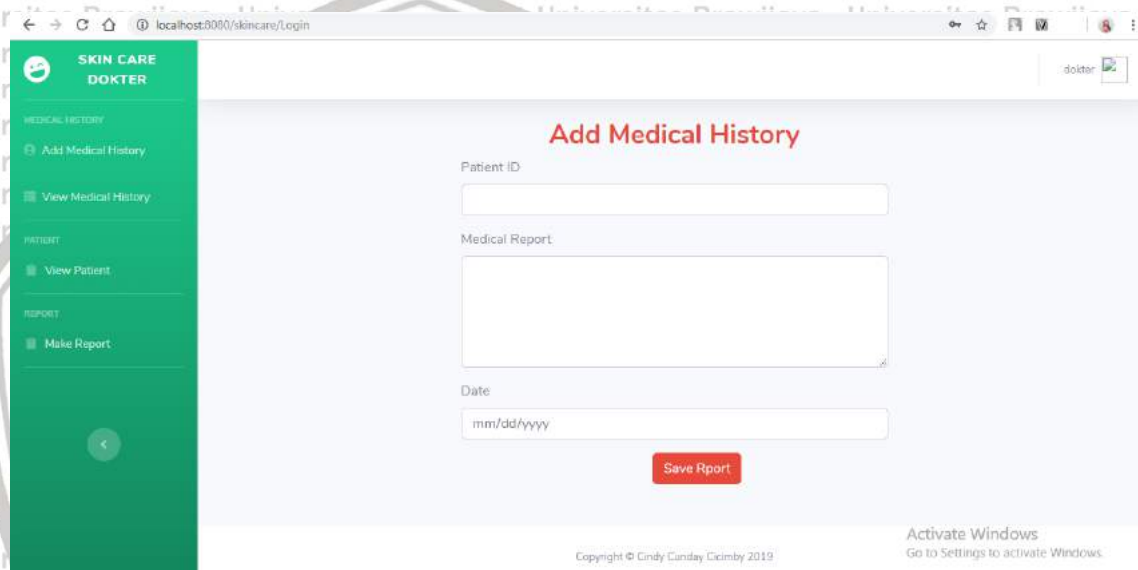
Algoritme Menghapus Admin
1 public function Delete_Admin(){
2     $id_admin = $this->input->post('id_admin');
3     $delete_admin = $this->Model_Admin-
4     >Delete_Admin($id_admin);
5     if($delete_admin){
6         $this->View_List_Admin();
7     }
8     else{
9         print"<script type=\"text/javascript\">alert('Something
10    Wrong, Please Try Again!');</script>";
11    }
12    }
    
```

5.4 Implementasi Antarmuka

Antarmuka sistem ialah media yang aktor gunakan dalam berkomunikasi dengan sistem. Berikut dijelaskan pengimplementasian antarmuka sistem rekam medis Ramdani *Skincare & Spa* dan digambarkan pada Gambar 5.12 sampai Gambar 5.15.

5.4.1 Implementasi Antarmuka Menambah Data Rekam Medis

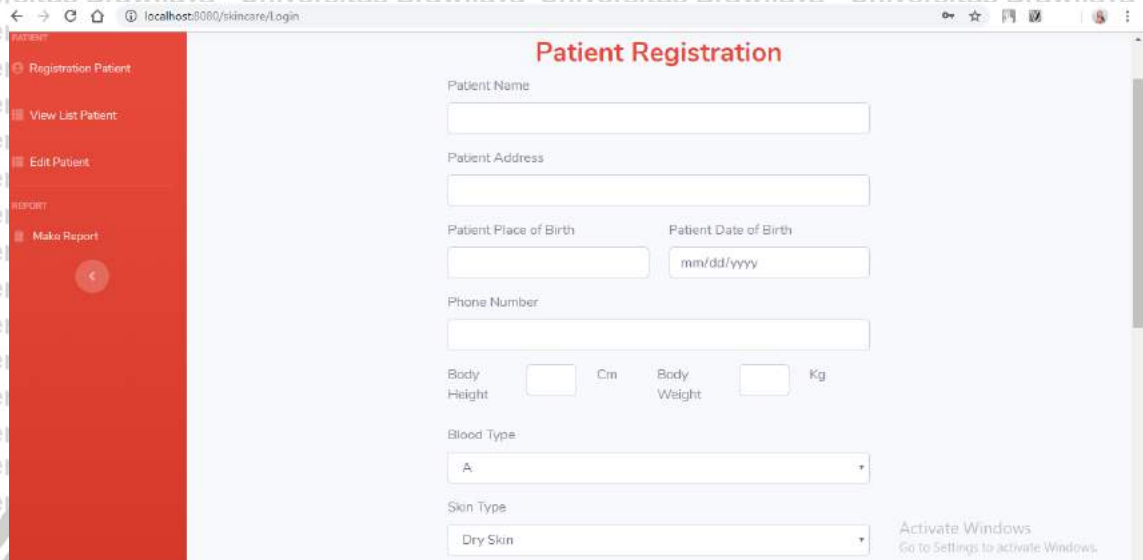
Antarmuka menambah data rekam medis ialah halaman yang tampil di sistem ketika dokter hendak menambah data rekam medis pasien. Gambar 5.12 ialah pengimplementasian antarmuka fungsi menambah data rekam medis.



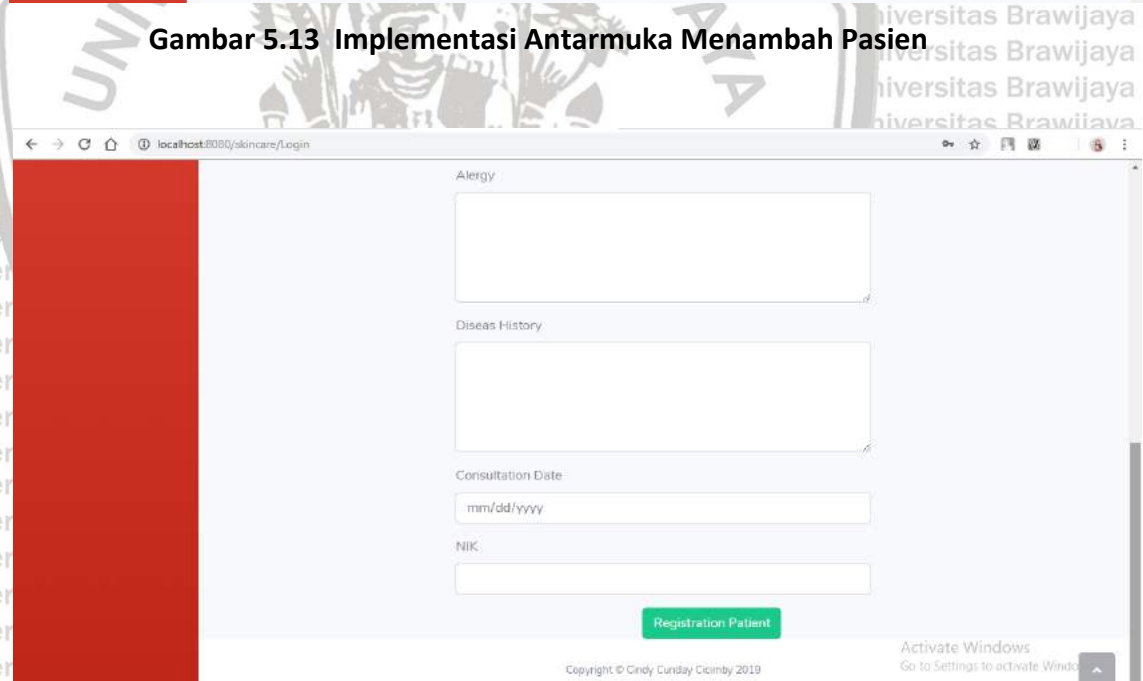
Gambar 5.12 Implementasi Antarmuka Menambah Data Rekam Medis

5.4.2 Implementasi Antarmuka Menambah Pasien

Antarmuka menambah pasien ialah halaman yang tampil pada sistem ketika admin hendak menambah pasien/mendaftarkan pasien baru ke dalam sistem. Gambar 5.13 dan Gambar 5.14 ialah pengimplementasian antarmuka menambah pasien.



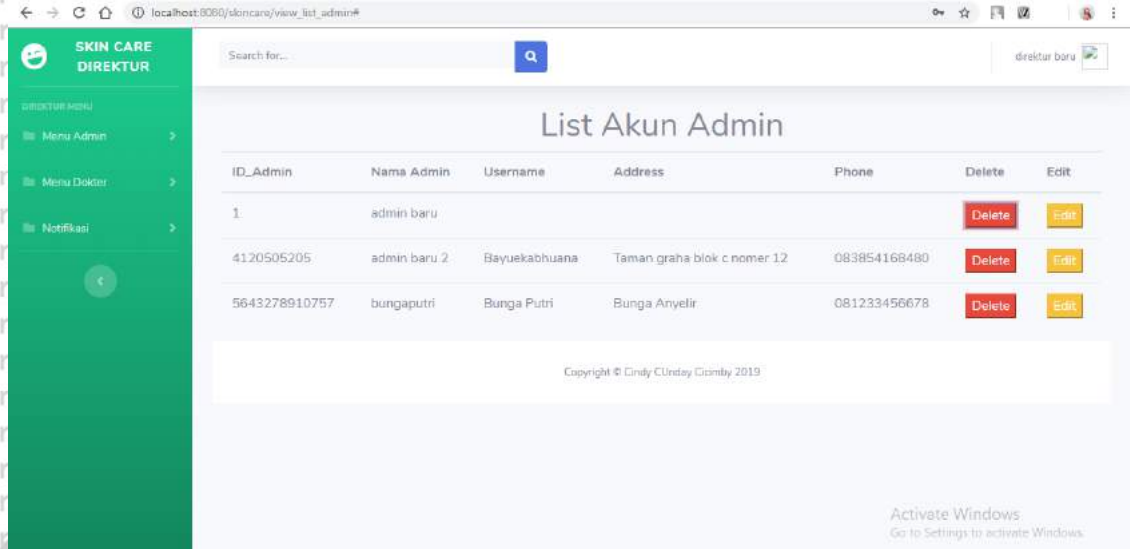
Gambar 5.13 Implementasi Antarmuka Menambah Pasien



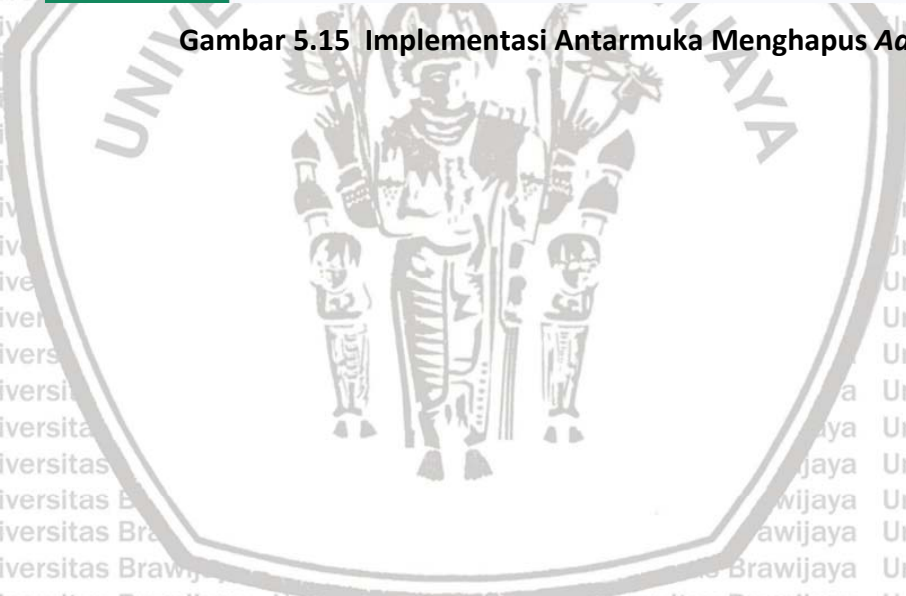
Gambar 5.14 Implementasi Antarmuka Menambah Pasien (Lanjutan)

5.4.3 Implementasi Antarmuka Menghapus Admin

Antarmuka menghapus admin ialah halaman yang tampil pada sistem ketika direktur hendak menghapus *admin* dari sistem. Gambar 5.15 ialah pengimplementasian antarmuka menghapus *admin*.



Gambar 5.15 Implementasi Antarmuka Menghapus Admin



BAB 6 PENGUJIAN

Bab ini membahas mengenai proses uji serta analisis sistem rekam medis Ramdani *Skincare & Spa*. Pengujian ini memiliki tujuan untuk mengetahui kesalahan pada sistem jika ada serta membuktikan jika sistem yang dibuat telah selesai dan berfungsi sesuai dengan kebutuhan pengguna.

6.1 Pengujian Unit

Untuk melakukan pengecekan terhadap detail perancangan sistem, dilakukan pengujian *white-box* yang dimulai dengan struktur desain sistem lalu dibagi menjadi beberapa kasus uji. Untuk menentukan jumlah *cyclomatic complexity* dan jalur independennya, algoritme sistem disusun dalam bentuk *flow graph*.

6.1.1 Pengujian Unit Menambah Data Rekam Medis

Pseudocode dari algoritme menambah data rekam medis dengan fungsi `Add_Rekam_Medis()` dari kelas `Controller_Dokter` disertai *node flowgraph* terlihat pada Tabel 6.1.

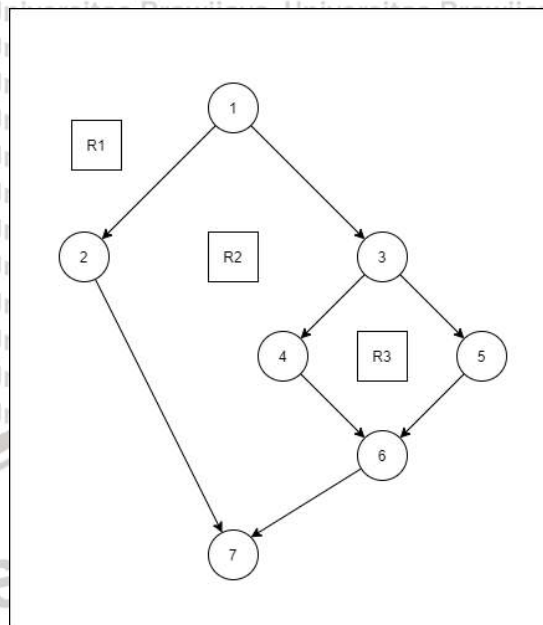
1. *Pseudocode*

Tabel 6.1 Pembentukan Node Algoritme Fungsi `Add_Rekam_Medis()`

1	Mulai
1	Memasukkan ID pasien
1	Memasukkan data rekam medis pasien
1	Memasukkan tanggal konsultasi
2	If (ID = true rekam medis = not null tanggal = not null)
2	Memanggil method <code>Add_Rekam_Medis()</code> untuk menyimpan data
3	Else
4	If (ID = false rekam medis = not null tanggal = not null)
4	Menampilkan peringatan "ID pasien tidak terdaftar"
5	Else
5	Menampilkan pemberitahuan "data diri tidak lengkap, isi kolom yang kosong"
6	End if
7	End if
7	End

2. Basis Path Testing

a. Flow Graph



Gambar 6.1 Flow Graph Add_Rekam_Medis()

b. Cyclomatic Complexity

$$V(G) = R = 3$$

$$V(G) = 8 \text{ edges} - 7 \text{ nodes} + 2 = 3$$

$$V(G) = 2 \text{ predicate nodes} + 1 = 3$$

c. Independent Path

Jalur 1 = 1 – 2 – 7

Jalur 2 = 1 – 3 – 4 – 6 – 7

Jalur 3 = 1 – 3 – 5 – 6 – 7

Berikut test case dan hasilnya dijelaskan pada Tabel 6.2.

Tabel 6.2 Pengujian Unit Fungsi Add_Rekam_Medis()

Nomor Jalur	Prosedur Uji	Hasil yang diinginkan	Hasil yang didapat	Status
1	ID pasien benar, rekam medis terisi, tanggal terisi	Menampilkan notifikasi "menambahkan rekam medis berhasil"	Menampilkan notifikasi "menambahkan rekam medis berhasil"	Valid
2	ID pasien tidak terdaftar/salah, rekam medis	Menampilkan notifikasi "ID	Menampilkan notifikasi "ID	Valid

	terisi, tanggal terisi	pasien tidak terdaftar”	Pasien tidak terdaftar”	
	ID pasien benar, rekam medis terisi, tanggal tidak terisi	Menampilkan notifikasi “ data diri tidak lengkap, isi kolom yang kosong”	Menampilkan notifikasi “ data diri tidak lengkap, isi kolom yang kosong”	Valid

6.1.2 Pengujian Unit Menambah Pasien

Tabel 6.3 ialah penjelasan dari algoritme menambah pasien dengan fungsi Add_Pasien() dari kelas Controller_Admin dengan *node flowgraph*.

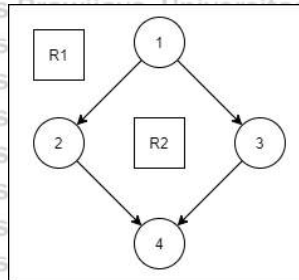
1. Pseudocode

Tabel 6.3 Pembentukan Node Algoritme fungsi Add_Pasien()

1	Mulai
1	Memasukkan ID pasien, nama, alamat, tinggi badan, berat badan, golongan darah, tipe kulit, riwayat penyakit, telpon, alergi, tanggal konsultasi/mendaftar, tempat dan tanggal lahir, dan NIK
2	If (semua kolom terisi)
2	Menampilkan pemberitahuan pendaftaran pasien berhasil
3	Else
3	Menampilkan pemberitahuan pendaftaran pasien gagal
4	End If
4	End

2. Basis Path Testing

a. Flow Graph



Gambar 6.2 Flow Graph Add_Pasien()

b. Cyclomatic Complexity

$$V(G) = 2$$

$$V(G) = 4 \text{ edges} - 4 \text{ nodes} + 2 = 2$$

$$V(G) = 1 \text{ predicate nodes} + 1 = 2$$

c. Independent Path

Jalur 1 = 1 – 2 – 4

Jalur 2 = 1 – 3 – 4

Berikut test case dan hasilnya dijelaskan pada Tabel 6.4.

Tabel 6.4 Pengujian Unit Fungsi Add_Pasien()

Nomor Jalur	Prosedur Uji	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Kolom data diri pasien terisi lengkap	Menampilkan notifikasi “Pasien berhasil ditambahkan”	Menampilkan notifikasi “Pasien berhasil ditambahkan”	Valid
2	Kolom data diri pasien tidak terisi lengkap	Menampilkan notifikasi “Pendaftaran pasien gagal”	Menampilkan notifikasi “Pendaftaran pasien gagal”	Valid

6.1.3 Pengujian Unit Menghapus Admin

Tabel 6.5 merupakan penjelasan dari algoritme proses menghapus admin dengan fungsi Delete_Admin() dari kelas Controller_Direktur dengan *node flowgraph*.



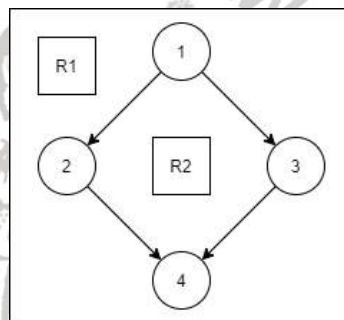
1. Pseudocode

Tabel 6.5 Pembentukan Node Algoritme fungsi Delete_Admin()

1	Mulai
1	Memasukkan ID admin
1	Menekan tombol enter
1	Menekan tombol delete
2	If (ID admin = terdaftar/benar)
2	Admin terhapus dan kembali ke halaman utama
3	Else
3	Menampilkan peringatan ada yang salah, silahkan coba lagi
4	End If
4	End

2. Basis Path Testing

a. Flow Graph



Gambar 6.3 Flow Graph Delete_Admin()

b. Cyclomatic Complexity

$$V(G) = 2$$

$$V(G) = 4 \text{ edges} - 4 \text{ nodes} + 2 = 2$$

$$V(G) = 1 \text{ predicate nodes} + 1 = 2$$

c. Independent Path

$$\text{Jalur 1} = 1 - 2 - 4$$

$$\text{Jalur 2} = 1 - 3 - 4$$

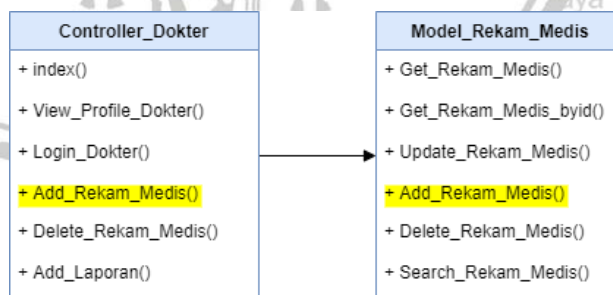
Test case dan hasil dijelaskan pada Tabel 6.6.

Tabel 6.6 Pengujian Unit Fungsi Delete_Admin()

Nomor Jalur	Prosedur Uji	Hasil yang diinginkan	Hasil yang didapat	Status
1	Menghapus admin berhasil	Admin berhasil dihapus dan kembali ke halaman View Admin's List	Admin berhasil dihapus dan kembali ke halaman View Admin's List	Valid
2	Menghapus admin gagal	Menampilkan notifikasi "Ada kesalahan, silahkan ulangi lagi"	Menampilkan notifikasi "Ada kesalahan, silahkan ulangi lagi"	Valid

6.2 Pengujian Integrasi

Untuk menguji kelas yang saling berhubungan, maka dilakukanlah pengujian integrasi. Pengujian ini masuk ke dalam metode pengujian *white-box*. Pengujian ini diuji menggunakan teknik uji *basis path*. Sampel yang digunakan untuk melakukan pengujian integrasi adalah algoritme method `Add_Rekam_Medis()` dari kelas `Controller_Dokter` dengan method `Add_Rekam_Medis()` dari kelas `Model_Rekam_Medis`. Terlihat pada Gambar 6.4 merupakan kelas dari method yang akan dilakukan integrasi.



Gambar 6.4 Method `Add_Rekam_Medis()` dari kelas `Controller` dan `Model`

6.2.1 Pengujian Integrasi Method Add_Rekam_Medis() dengan Method Add_Rekam_Medis(\$id_pasien, \$id_dokter, \$rekam_medis, \$tanggal)

1. Pseudocode

Tabel 6.7 Pembentukan Node Algoritme Fungsi Add_Rekam_Medis() Kelas Dokter_Controller

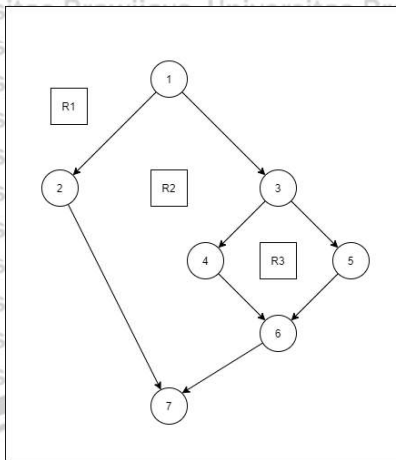
1	Mulai
1	Memasukkan ID pasien
1	Memasukkan data rekam medis pasien
1	Memasukkan tanggal konsultasi
2	If (ID = true rekam medis = not null tanggal = not null)
2	Memanggil method Add_Rekam_Medis() pada kelas Model Rekam Medis untuk menyimpan data
3	Else
4	If (ID = false rekam medis = not null tanggal = not null)
4	Menampilkan peringatan "ID pasien tidak terdaftar"
5	Else
5	Menampilkan pemberitahuan "data diri tidak lengkap, isi kolom yang kosong"
6	End if
7	End if
7	End

Tabel 6.8 Pembentukan Node Algoritme Fungsi Add_Rekam_Medis() Kelas Model_Rekam_Medis

1	Mulai
1	Query untuk memasukan data yang tersimpan pada array ke database rekam_medis (ID_PASIEN, ID_DOKTER, REKAM_MEDIS, TANGGAL)
1	End

2. Basis Path Testing

a. Flow Graph



Gambar 6.5 Flow Graph Add_Rekam_Medis()

b. Cyclomatic Complexity

$$V(G) = R + 1 = 3 + 1 = 4$$

$$V(G) = 8 \text{ edges} - 7 \text{ nodes} + 2 = 3$$

$$V(G) = 2 \text{ predicate nodes} + 1 = 3$$

c. Independent Path

$$\text{Jalur 1} = 1 - 2 - 7$$

$$\text{Jalur 2} = 1 - 3 - 4 - 6 - 7$$

$$\text{Jalur 3} = 1 - 3 - 5 - 6 - 7$$

Untuk test case dan hasilnya dijelaskan pada Tabel 6.2.

Tabel 6.9 Pengujian Unit Fungsi Add_Rekam_Medis()

Nomor Jalur	Prosedur Uji	Hasil yang diinginkan	Hasil yang didapat	Status
1	ID pasien benar, rekam medis terisi, tanggal terisi	Menampilkan notifikasi "menambahkan rekam medis berhasil"	Menampilkan notifikasi "menambahkan rekam medis berhasil"	Valid
2	ID pasien tidak terdaftar/salah, rekam medis terisi, tanggal terisi	Menampilkan notifikasi "ID pasien tidak terdaftar"	Menampilkan notifikasi "ID Pasien tidak terdaftar"	Valid
	ID pasien benar, rekam medis	Menampilkan notifikasi " data	Menampilkan notifikasi " data	Valid

	terisi, tanggal tidak terisi	diri tidak lengkap, isi kolom yang kosong”	diri tidak lengkap, isi kolom yang kosong”
--	------------------------------	--	--

6.3 Pengujian Validasi

Semua persyaratan sistem akan divalidasi dan diuji dengan metode pengujian *black box*. Metode yang digunakan yaitu *State Transition Testing*, dimana akan dilihat perubahan hasil *output* yang disebabkan oleh *input* yang berbeda kondisi. Pengujian validasi dilakukan dengan melihat apakah terjadi kesalahan saat sistem dijalankan atau baik-baik saja. Pengujian ini dilakukan berdasarkan daftar-daftar kebutuhan fungsional yang telah didapatkan dari hasil proses penganalisisan kebutuhan.

Tabel 6.10 Kasus Uji Login Admin

Nama Kasus Uji	Kasus Uji <i>Login</i> sebagai <i>admin</i>
Kode Kebutuhan	SRS-RSS-F-1
Prosedur	<ol style="list-style-type: none"> 1. Aktor memasukkan <i>username = admin</i> dan <i>password = 123456</i> pada formulir <i>login</i> 2. Aktor memilih <i>login</i> sebagai <i>admin</i> 3. Aktor menekan tombol “<i>Login</i>”
Hasil yang diinginkan	Aktor akan terotorisasi dan dapat mengakses sistem sebagai <i>administrator</i>
Hasil	Aktor akan terotorisasi dan dapat mengakses sistem sebagai <i>administrator</i>
Status	Valid

Tabel 6.11 Kasus Uji Login Admin

Alternatif 1: *Password* tidak sesuai

Nama Kasus Uji	Kasus Uji <i>Login</i> sebagai <i>admin</i>
Kode Kebutuhan	SRS-RSS-F-1
Prosedur	<ol style="list-style-type: none"> 1. Aktor memasukkan <i>username = admin</i> dan <i>password</i> yang salah pada formulir <i>login</i> 2. Aktor memilih <i>login</i> sebagai <i>admin</i> 3. Aktor menekan tombol “<i>Login</i>”
Hasil yang diinginkan	Sistem memperlihatkan pesan peringatan “ <i>password is wrong</i> ”
Hasil	Sistem memperlihatkan pesan peringatan “ <i>password is wrong</i> ”
Status	Valid

Tabel 6.12 Kasus Uji Login Admin

Alternatif 2: Salah satu data kosong

Nama Kasus Uji	Kasus Uji <i>Login</i> sebagai <i>admin</i>
Kode Kebutuhan	SRS-RSS-F-1
Prosedur	1. Aktor memasukkan <i>username</i> = <i>admin</i> pada formulir <i>login</i> 2. Aktor menekan tombol " <i>Login</i> "
Hasil yang diinginkan	Sistem memperlihatkan pesan peringatan " <i>please fill username and password</i> "
Hasil	Sistem memperlihatkan pesan peringatan " <i>please fill username and password</i> "
Status	Valid

Tabel 6.13 Kasus Uji Login Admin

Alternatif 3: *Username, password, login by* tidak sesuai

Nama Kasus Uji	Kasus Uji <i>Login</i> sebagai <i>admin</i>
Kode Kebutuhan	SRS-RSS-F-1
Prosedur	1. Aktor memasukkan <i>username</i> = <i>admin</i> , <i>password</i> = 123, dan <i>login by</i> dokter pada formulir <i>login</i> 2. Aktor menekan tombol " <i>Login</i> "
Hasil yang diinginkan	Sistem memperlihatkan pesan peringatan " <i>your username or password is wrong/you're not admin</i> "
Hasil	Sistem memperlihatkan pesan peringatan " <i>your username or password is wrong/you're not admin</i> "
Status	Valid

Tabel 6.14 Kasus Uji Login Dokter

Nama Kasus Uji	Kasus Uji <i>Login</i> sebagai dokter
Kode Kebutuhan	SRS-RSS-F-11
Prosedur	1. Aktor memasukkan <i>username</i> = dokter dan <i>password</i> = 12345 pada formulir <i>login</i> 2. Aktor memilih <i>login</i> sebagai dokter 3. Aktor menekan tombol " <i>Login</i> "
Hasil yang diinginkan	Aktor akan terotorisasi dan dapat mengakses sistem sebagai dokter
Hasil	Aktor akan terotorisasi dan dapat mengakses sistem sebagai dokter
Status	Valid

Tabel 6.15 Kasus Uji Login Dokter

Alternatif 1: *Password* tidak sesuai

Nama Kasus Uji	Kasus Uji <i>Login</i> sebagai dokter
Kode Kebutuhan	SRS-RSS-F-11
Prosedur	<ol style="list-style-type: none"> 1. Aktor memasukkan <i>username</i> = dokter dan <i>password</i> yang salah pada formulir <i>login</i> 2. Aktor memilih <i>login</i> sebagai dokter 3. Aktor menekan tombol "<i>Login</i>"
Hasil yang diinginkan	Sistem memperlihatkan pesan peringatan " <i>password is wrong</i> "
Hasil	Sistem memperlihatkan pesan peringatan " <i>password is wrong</i> "
Status	Valid

Tabel 6.16 Kasus Uji Login Dokter

Alternatif 2: Salah satu data kosong

Nama Kasus Uji	Kasus Uji <i>Login</i> sebagai dokter
Kode Kebutuhan	SRS-RSS-F-11
Prosedur	<ol style="list-style-type: none"> 1. Aktor memasukkan <i>username</i> = dokter pada formulir <i>login</i> 2. Aktor menekan tombol "<i>Login</i>"
Hasil yang diinginkan	Sistem memperlihatkan pesan peringatan " <i>please fill username and password</i> "
Hasil	Sistem memperlihatkan pesan peringatan " <i>please fill username and password</i> "
Status	Valid

Tabel 6.17 Kasus Uji Login Dokter

Alternatif 3: *Username, password, login by* tidak sesuai

Nama Kasus Uji	Kasus Uji <i>Login</i> sebagai dokter
Kode Kebutuhan	SRS-RSS-F-11
Prosedur	<ol style="list-style-type: none"> 1. Aktor memasukkan <i>username</i> = dokter, <i>password</i> = 123, dan <i>login by</i> direktur pada formulir <i>login</i> 2. Aktor menekan tombol "<i>Login</i>"
Hasil yang diinginkan	Sistem memperlihatkan pesan peringatan " <i>your username or password is wrong/you're not doctor</i> "
Hasil	Sistem memperlihatkan pesan peringatan " <i>your username or password is wrong/you're not doctor</i> "

Status	Valid
--------	-------

Tabel 6.18 Kasus Uji Login Direktur

Nama Kasus Uji	Kasus Uji <i>Login</i> sebagai dokter
Kode Kebutuhan	SRS-RSS-F-24
Prosedur	<ol style="list-style-type: none"> 1. Aktor memasukkan <i>username</i> = direktur dan <i>password</i> = 123 pada formulir <i>login</i> 2. Aktor memilih <i>login</i> sebagai direktur 3. Aktor menekan tombol "<i>Login</i>"
Hasil yang diinginkan	Aktor akan terotorisasi dan dapat mengakses sistem sebagai direktur
Hasil	Aktor akan terotorisasi dan dapat mengakses sistem sebagai direktur
Status	Valid

Tabel 6.19 Kasus Uji Login Direktur

Alternatif 1: *Password* tidak sesuai

Nama Kasus Uji	Kasus Uji <i>Login</i> sebagai dokter
Kode Kebutuhan	SRS-RSS-F-24
Prosedur	<ol style="list-style-type: none"> 1. Aktor memasukkan <i>username</i> = direktur dan <i>password</i> yang salah pada formulir <i>login</i> 2. Aktor memilih <i>login</i> sebagai direktur 3. Aktor menekan tombol "<i>Login</i>"
Hasil yang diinginkan	Sistem menayangkan pesan peringatan " <i>password is wrong</i> "
Hasil	Sistem menayangkan pesan peringatan " <i>password is wrong</i> "
Status	Valid

Tabel 6.20 Kasus Uji Login Direktur

Alternatif 2: Salah satu data kosong

Nama Kasus Uji	Kasus Uji <i>Login</i> sebagai dokter
Kode Kebutuhan	SRS-RSS-F-24
Prosedur	<ol style="list-style-type: none"> 1. Aktor memasukkan <i>username</i> = direktur pada formulir <i>login</i> 2. Aktor menekan tombol "<i>Login</i>"
Hasil yang diinginkan	Sistem menayangkan pesan peringatan " <i>please fill username and password</i> "

Hasil	Sistem menayangkan pesan peringatan “ <i>please fill username and password</i> ”
Status	Valid

Tabel 6.21 Kasus Uji Login Direktur

Alternatif 3: *Username, password, login by* tidak sesuai

Nama Kasus Uji	Kasus Uji <i>Login</i> sebagai dokter
Kode Kebutuhan	SRS-RSS-F-24
Prosedur	1. Aktor memasukkan <i>username</i> = direktur, <i>password</i> = 123456, dan <i>login by admin</i> pada formulir <i>login</i> 2. Aktor menekan tombol “ <i>Login</i> ”
Hasil yang diinginkan	Sistem menayangkan pesan peringatan “ <i>your username or password is wrong/you’re not director</i> ”
Hasil	Sistem menayangkan pesan peringatan “ <i>your username or password is wrong/you’re not director</i> ”
Status	Valid

Tabel 6.22 Kasus Uji Mencari Data Rekam Medis

Nama Kasus Uji	Kasus Uji Mencari Data Rekam Medis
Kode Kebutuhan	SRS-RSS-F-12
Prosedur	1. Aktor memasukkan ID pasien yang sudah terdaftar pada kolom pencarian di halaman <i>View List Pasien</i> atau <i>View Rekam Medis</i> . 2. Aktor menekan tombol <i>enter</i>
Hasil yang diinginkan	Sistem memperlihatkan data pasien dengan nomor ID yang dimasukkan
Hasil	Sistem memperlihatkan data pasien dengan nomor ID yang dimasukkan
Status	Valid

Tabel 6.23 Kasus Uji Mencari Data Rekam Medis

Alternatif 1: ID pasien tidak terdaftar

Nama Kasus Uji	Kasus Uji Mencari Data Rekam Medis
Kode Kebutuhan	SRS-RSS-F-12

Prosedur	1. Aktor memasukkan ID pasien yang tidak terdaftar pada kolom pencarian di halaman <i>View List Pasien</i> atau <i>View Rekam Medis</i> . 2. Aktor menekan tombol <i>enter</i> .
Hasil yang diinginkan	Sistem tidak memperlihatkan data apapun
Hasil	Sistem tidak memperlihatkan data apapun
Status	Valid

Tabel 6.24 Kasus Uji Menambah Data Rekam Medis

Nama Kasus Uji	Kasus Uji Menambah Data Rekam Medis
Kode Kebutuhan	SRS-RSS-F-11
Prosedur	1. Aktor memasukkan data rekam medis pasien dengan ID pasien yang sudah terdaftar pada halaman <i>Add Medical History</i> 2. Aktor menekan tombol <i>save</i>
Hasil yang diinginkan	Sistem memperlihatkan tulisan " <i>medical history update was successful</i> "
Hasil	Sistem memperlihatkan tulisan " <i>medical history update was successful</i> "
Status	Valid

Tabel 6.25 Kasus Uji Menambah Data Rekam Medis

Alternatif 1: ID pasien tidak terdaftar

Nama Kasus Uji	Kasus Uji Menambah Data Rekam Medis
Kode Kebutuhan	SRS-RSS-F-11
Prosedur	1. Aktor memasukkan data rekam medis pasien dengan ID pasien yang tidak terdaftar pada halaman <i>Add Medical History</i> 2. Aktor menekan tombol <i>save</i>
Hasil yang diinginkan	Sistem memperlihatkan tulisan " <i>Patient's ID is not registered</i> "
Hasil	Sistem memperlihatkan tulisan " <i>Patient's ID is not registered</i> "
Status	Valid

Tabel 6.26 Kasus Uji Menambah Data Rekam Medis

Alternatif 2: Aktor tidak mengisi kolom *Medical Report* dan *Date*

Nama Kasus Uji	Kasus Uji <i>Logout</i>
Kode Kebutuhan	SRS-RSS-F-2
Prosedur	<ol style="list-style-type: none"> 1. Aktor memasukkan data rekam medis pasien dengan ID pasien yang terdaftar pada halaman <i>Add Medical History</i>, namun aktor tidak mengisi kolom <i>Medical Report</i> dan <i>Date</i> 2. Aktor menekan tombol <i>save report</i>
Hasil yang diinginkan	Sistem memperlihatkan tulisan " <i>Please fill the empty column</i> "
Hasil	Sistem memperlihatkan tulisan " <i>Please fill the empty column</i> "
Status	Valid

Tabel 6.27 Kasus Uji *Logout Admin*

Nama Kasus Uji	Kasus Uji <i>Logout</i> sebagai <i>admin</i>
Kode Kebutuhan	SRS-RSS-F-2
Prosedur	<ol style="list-style-type: none"> 3. <i>Admin</i> memencet tombol <i>logout</i> di pojok kanan atas halaman utama 4. Sistem memperlihatkan sebuah <i>dialog box</i> memastikan apakah <i>admin</i> yakin untuk keluar atau tidak 5. <i>Admin</i> memencet tombol <i>logout</i> untuk keluar dari sistem
Hasil yang diinginkan	Sistem memperlihatkan halaman <i>login</i>
Hasil	Sistem memperlihatkan halaman <i>login</i>
Status	Valid

Tabel 6.28 Kasus Uji *Logout Admin*

Alternatif 1: *Admin* membatalkan proses *logout*

Nama Kasus Uji	Kasus Uji <i>Logout</i> sebagai <i>admin</i>
Kode Kebutuhan	SRS-RSS-F-2
Prosedur	<ol style="list-style-type: none"> 6. <i>Admin</i> memencet tombol <i>logout</i> di pojok kanan atas halaman utama 7. Sistem memperlihatkan sebuah <i>dialog box</i> memastikan apakah <i>admin</i> yakin untuk keluar atau tidak 8. <i>Admin</i> memencet tombol <i>cancel</i> untuk membatalkan

Hasil yang diinginkan	Sistem memperlihatkan halaman utama (<i>dashboard</i>)
Hasil	Sistem memperlihatkan halaman utama (<i>dashboard</i>)
Status	Valid

Tabel 6.29 Kasus Uji Logout Dokter

Nama Kasus Uji	Kasus Uji <i>Logout</i> sebagai dokter
Kode Kebutuhan	SRS-RSS-F-12
Prosedur	9. Dokter memencet tombol <i>logout</i> di pojok kanan atas halaman utama 10. Sistem memperlihatkan sebuah <i>dialog box</i> memastikan apakah dokter yakin untuk keluar atau tidak 11. Dokter memencet tombol <i>logout</i> untuk keluar dari sistem
Hasil yang diinginkan	Sistem memperlihatkan halaman <i>login</i>
Hasil	Sistem memperlihatkan halaman <i>login</i>
Status	Valid

Tabel 6.30 Kasus Uji Logout Dokter

Alternatif 1: Dokter membatalkan proses *logout*

Nama Kasus Uji	Kasus Uji <i>Logout</i> sebagai dokter
Kode Kebutuhan	SRS-RSS-F-12
Prosedur	12. Dokter memencet tombol <i>logout</i> di pojok kanan atas halaman utama 13. Sistem memperlihatkan sebuah <i>dialog box</i> memastikan apakah dokter yakin untuk keluar atau tidak 14. Dokter memencet tombol <i>cancel</i> untuk membatalkan
Hasil yang diinginkan	Sistem memperlihatkan halaman utama (<i>dashboard</i>)
Hasil	Sistem memperlihatkan halaman utama (<i>dashboard</i>)
Status	Valid

Tabel 6.31 Kasus Uji Logout Direktur

Nama Kasus Uji	Kasus Uji <i>Logout</i> sebagai direktur
Kode Kebutuhan	SRS-RSS-F-25

Prosedur	15. Direktur memencet tombol <i>logout</i> di pojok kanan atas halaman utama 16. Sistem memperlihatkan sebuah <i>dialog box</i> memastikan apakah direktur yakin untuk keluar atau tidak 17. Direktur memencet tombol <i>logout</i> untuk keluar dari sistem
Hasil yang diinginkan	Sistem memperlihatkan halaman <i>login</i>
Hasil	Sistem memperlihatkan halaman <i>login</i>
Status	Valid

Tabel 6.32 Kasus Uji Logout Direktur

Alternatif 1: Direktur membatalkan proses *logout*

Nama Kasus Uji	Kasus Uji <i>Logout</i> sebagai direktur
Kode Kebutuhan	SRS-RSS-F-25
Prosedur	18. Direktur memencet tombol <i>logout</i> di pojok kanan atas halaman utama 19. Sistem memperlihatkan sebuah <i>dialog box</i> memastikan apakah direktur yakin untuk keluar atau tidak 20. Direktur memencet tombol <i>cancel</i> untuk membatalkan
Hasil yang diinginkan	Sistem memperlihatkan halaman utama (<i>dashboard</i>)
Hasil	Sistem memperlihatkan halaman utama (<i>dashboard</i>)
Status	Valid

Tabel 6.33 Kasus Uji Mengubah Profil Admin

Nama Kasus Uji	Kasus Uji Mengubah Profil
Kode Kebutuhan	SRS-RSS-F-8
Prosedur	1. <i>Admin</i> memencet tombol <i>Profile</i> di sebelah kanan pojok atas halaman utama 2. Sistem memperlihatkan halaman profil yang berisi <i>username</i> , <i>password</i> , dan foto 3. <i>Admin</i> melakukan perubahan, lalu menekan tombol <i>update profile</i>
Hasil yang diinginkan	Sistem memperlihatkan pemberitahuan bahwa perubahan berhasil disimpan

Hasil	Sistem memperlihatkan pemberitahuan bahwa perubahan berhasil disimpan
Status	Valid

Tabel 6.34 Kasus Uji Mengubah Profil Dokter

Nama Kasus Uji	Kasus Uji Mengubah Profil
Kode Kebutuhan	SRS-RSS-F-19
Prosedur	<ol style="list-style-type: none"> 1. Dokter memencet tombol <i>Profile</i> di sebelah kanan pojok atas halaman utama 2. Sistem memperlihatkan halaman profil yang berisi <i>username, password</i>, dan foto 3. Dokter melakukan perubahan, lalu menekan tombol <i>update profile</i>
Hasil yang diinginkan	Sistem memperlihatkan pemberitahuan bahwa perubahan berhasil disimpan
Hasil	Sistem memperlihatkan pemberitahuan bahwa perubahan berhasil disimpan
Status	Valid

Tabel 6.35 Kasus Uji Mengubah Profil Direktur

Nama Kasus Uji	Kasus Uji Mengubah Profil
Kode Kebutuhan	SRS-RSS-F-37
Prosedur	<ol style="list-style-type: none"> 1. Direktur memencet tombol <i>Profile</i> di sebelah kanan pojok atas halaman utama 2. Sistem memperlihatkan halaman profil yang berisi <i>username, password</i>, dan foto 3. Direktur melakukan perubahan, lalu menekan tombol <i>update profile</i>
Hasil yang diinginkan	Sistem memperlihatkan pemberitahuan bahwa perubahan berhasil disimpan
Hasil	Sistem memperlihatkan pemberitahuan bahwa perubahan berhasil disimpan
Status	Valid

Tabel 6.36 Kasus Uji Mengirim Laporan Admin

Nama Kasus Uji	Kasus Uji Mengirim Laporan
Kode Kebutuhan	SRS-RSS-F-10

Prosedur	<ol style="list-style-type: none"> 1. Admin memencet menu <i>Make Report</i> 2. Admin mengisikan data yang terdiri dari <i>Report Subject, Report Message</i>, dan mengunggah file laporan 3. Admin memencet tombol <i>save report</i>
Hasil yang diinginkan	Sistem memperlihatkan pemberitahuan bahwa proses pengunggahan laporan berhasil dan sistem kembali ke halaman laporan
Hasil	Sistem memperlihatkan pemberitahuan bahwa proses pengunggahan laporan berhasil dan sistem kembali ke halaman laporan
Status	Valid

Tabel 6.37 Kasus Uji Mengirim Laporan Admin

Alternatif 1: Admin tidak mengisi semua kolom

Nama Kasus Uji	Kasus Uji Mengirim Laporan
Kode Kebutuhan	SRS-RSS-F-10
Prosedur	<ol style="list-style-type: none"> 1. Admin memencet menu <i>Make Report</i> 2. Admin tidak mengisi <i>Report Subject, Report Message</i>, dan tidak mengunggah file laporan 3. Admin memencet tombol <i>save report</i>
Hasil yang diinginkan	Sistem memperlihatkan peringatan untuk mengisi kolom <i>Report Subject, Report Message</i> , serta mengunggah file
Hasil	Sistem memperlihatkan peringatan untuk mengisi kolom <i>Report Subject, Report Message</i> , serta mengunggah file
Status	Valid

Tabel 6.38 Kasus Uji Mengirim Laporan Dokter

Nama Kasus Uji	Kasus Uji Mengirim Laporan
Kode Kebutuhan	SRS-RSS-F-23
Prosedur	<ol style="list-style-type: none"> 1. Dokter memencet menu <i>Make Report</i> 2. Dokter mengisikan data yang terdiri dari <i>Report Subject, Report Message</i>, dan mengunggah file laporan 3. Dokter memencet tombol <i>save report</i>
Hasil yang diinginkan	Sistem memperlihatkan pemberitahuan bahwa proses pengunggahan laporan berhasil dan sistem kembali ke halaman laporan

Hasil	Sistem memperlihatkan pemberitahuan bahwa proses pengunggahan laporan berhasil dan sistem kembali ke halaman laporan
Status	Valid

Tabel 6.39 Kasus Uji Mengirim Laporan Dokter
Alternatif 1: Dokter tidak mengisi semua kolom

Nama Kasus Uji	Kasus Uji Mengirim Laporan
Kode Kebutuhan	SRS-RSS-F-23
Prosedur	<ol style="list-style-type: none"> 1. Dokter memencet menu <i>Make Report</i> 2. Dokter tidak mengisi <i>Report Subject</i>, <i>Report Message</i>, dan tidak mengunggah <i>file</i> laporan 3. Dokter menekan tombol <i>save report</i>
Hasil yang diinginkan	Sistem memperlihatkan peringatan untuk mengisi kolom <i>Report Subject</i> , <i>Report Message</i> , serta mengunggah <i>file</i>
Hasil	Sistem memperlihatkan peringatan untuk mengisi kolom <i>Report Subject</i> , <i>Report Message</i> , serta mengunggah <i>file</i>
Status	Valid

Tabel 6.40 Kasus Uji Menambah Data Pasien

Nama Kasus Uji	Kasus Uji Menambah Data Pasien
Kode Kebutuhan	SRS-RSS-F-3
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengisikan data diri pasien baru yang berisi ID pasien (berdasarkan nomor KTP), nama, nomor telepon, tempat dan tanggal lahir, alamat, golongan darah, tinggi dan berat badan, jenis kelamin, jenis kulit (berminyak/kering/kombinasi), alergi, rekam medis, dan tanggal pendaftaran/konsultasi. 2. Aktor memencet tombol <i>Patient Registration</i>
Hasil yang diinginkan	Sistem memperlihatkan pesan bahwa pasien berhasil didaftarkan
Hasil	Sistem memperlihatkan pesan bahwa pasien berhasil didaftarkan
Status	Valid

Tabel 6.41 Kasus Uji Menambah Data Pasien

Alternatif 1: Aktor tidak mengisikan keseluruhan kolom

Nama Kasus Uji	Kasus Uji Menambah Data Pasien
Kode Kebutuhan	SRS-RSS-F-3
Prosedur	<ol style="list-style-type: none"> 1. Aktor tidak mengisikan keseluruhan kolom yang ada pada halaman <i>Patient Registration</i> 2. Aktor memencet tombol <i>Patient Registration</i>
Hasil yang diinginkan	Sistem memperlihatkan peringatan bahwa ada kolom yang belum terisi
Hasil	Sistem memperlihatkan peringatan bahwa ada kolom yang belum terisi
Status	Valid

Tabel 6.42 Kasus Uji Mengubah Data Pasien

Nama Kasus Uji	Kasus Uji Mengubah Data Pasien
Kode Kebutuhan	SRS-RSS-F-4
Prosedur	<ol style="list-style-type: none"> 1. Aktor meletakkan kursor di kolom pencarian di sebelah kiri atas halaman, lalu mengetikkan nomor ID pasien, dan kemudian menekan tombol <i>enter</i> pada <i>keyboard</i> 2. Sistem memperlihatkan data pasien yang dicari 3. Aktor menekan tombol berwarna kuning bertuliskan <i>edit</i> 4. Sistem menampilkan halaman data diri pasien yang akan diubah 5. Aktor melakukan perubahan, lalu menekan tombol <i>update patient data</i>
Hasil yang diinginkan	Sistem memperlihatkan pemberitahuan bahwa perubahan telah berhasil disimpan
Hasil	Sistem memperlihatkan pemberitahuan bahwa perubahan telah berhasil disimpan
Status	Valid

Tabel 6.43 Kasus Uji Mencari Data Pasien pada Admin

Nama Kasus Uji	Kasus Uji Mencari Data Pasien
Kode Kebutuhan	SRS-RSS-F-5
Prosedur	<i>Admin</i> meletakkan kursor di kolom pencarian di sebelah kiri atas halaman, lalu mengetikkan nomor ID

	pasien, dan kemudian menekan tombol <i>enter</i> pada <i>keyboard</i>
Hasil yang diinginkan	Sistem memperlihatkan data pasien yang dicari
Hasil	Sistem memperlihatkan data pasien yang dicari
Status	Valid

Tabel 6.44 Kasus Uji Mencari Data Pasien pada Admin

Alternatif 1: Admin mengetik nomor ID yang tidak terdaftar sebagai pasien

Nama Kasus Uji	Kasus Uji Mencari Data Pasien
Kode Kebutuhan	SRS-RSS-F-5
Prosedur	Admin meletakkan kursor di kolom pencarian di sebelah kiri atas halaman, lalu mengetikkan nomor ID pasien yang tidak terdaftar sebagai pasien, dan kemudian menekan tombol <i>enter</i> pada <i>keyboard</i>
Hasil yang diinginkan	Sistem memperlihatkan halaman kosong (tidak ditemukan data)
Hasil	Sistem memperlihatkan halaman kosong (tidak ditemukan data)
Status	Valid

Tabel 6.45 Kasus Uji Mencari Data Pasien pada Dokter

Nama Kasus Uji	Kasus Uji Mencari Data Pasien
Kode Kebutuhan	SRS-RSS-F-14
Prosedur	Dokter meletakkan kursor di kolom pencarian di sebelah kiri atas halaman, lalu mengetikkan nomor ID pasien, dan kemudian menekan tombol <i>enter</i> pada <i>keyboard</i>
Hasil yang diinginkan	Sistem memperlihatkan data pasien yang dicari
Hasil	Sistem memperlihatkan data pasien yang dicari
Status	Valid

Tabel 6.46 Kasus Uji Mencari Data Pasien pada Dokter

Alternatif 1: Dokter mengetik nomor ID yang tidak terdaftar sebagai pasien

Nama Kasus Uji	Kasus Uji Mencari Data Pasien
Kode Kebutuhan	SRS-RSS-F-14

Prosedur	Dokter meletakkan kursor di kolom pencarian di sebelah kiri atas halaman, lalu mengetikkan nomor ID pasien yang tidak terdaftar sebagai pasien, dan kemudian menekan tombol <i>enter</i> pada <i>keyboard</i>
Hasil yang diinginkan	Sistem memperlihatkan halaman kosong (tidak ditemukan data)
Hasil	Sistem memperlihatkan halaman kosong (tidak ditemukan data)
Status	Valid

Tabel 6.47 Kasus Uji Menghapus Data Pasien

Nama Kasus Uji	Kasus Uji Menghapus Data Pasien
Kode Kebutuhan	SRS-RSS-F-6
Prosedur	<ol style="list-style-type: none"> 1. Aktor meletakkan kursor pada kolom pencarian di sebelah kiri atas halaman, lalu aktor mengetik nomor ID pasien yang akan dihapus 2. Aktor menekan tombol <i>enter</i> pada <i>keyboard</i> 3. Sistem memperlihatkan hasil pencarian 4. Aktor menekan tombol berwarna merah bertuliskan <i>delete</i> 5. Sistem memperlihatkan peringatan yakin untuk menghapus atau tidak 6. Aktor menekan tombol <i>delete</i> (yakin untuk menghapus)
Hasil yang diinginkan	Sistem kembali ke halaman <i>View Patient List</i>
Hasil	Sistem kembali ke halaman <i>View Patient List</i>
Status	Valid

Tabel 6.48 Kasus Uji Menghapus Data Pasien

Alternatif 1: Aktor menekan tombol *cancel* (tidak jadi menghapus)

Nama Kasus Uji	Kasus Uji Menghapus Data Pasien
Kode Kebutuhan	SRS-RSS-F-6
Prosedur	<ol style="list-style-type: none"> 1. Aktor meletakkan kursor pada kolom pencarian di sebelah kiri atas halaman, lalu aktor mengetik nomor ID pasien yang akan dihapus 2. Aktor menekan tombol <i>enter</i> pada <i>keyboard</i> 3. Sistem memperlihatkan hasil pencarian 4. Aktor memencet tombol berwarna merah bertuliskan <i>delete</i>

	5. Sistem memperlihatkan peringatan yakin untuk menghapus atau tidak 6. Aktor memencet tombol <i>cancel</i> (tidak jadi menghapus)
Hasil yang diinginkan	Sistem kembali ke halaman <i>View Patient List</i>
Hasil	Sistem kembali ke halaman <i>View Patient List</i>
Status	Valid

Tabel 6.49 Kasus Uji Melihat Data Pasien pada Admin

Nama Kasus Uji	Kasus Uji Melihat Data Pasien
Kode Kebutuhan	SRS-RSS-F-9
Prosedur	<i>Admin</i> memencet menu <i>View Patient List</i>
Hasil yang diinginkan	Sistem memperlihatkan halaman <i>View Patient List</i>
Hasil	Sistem memperlihatkan halaman <i>View Patient List</i>
Status	Valid

Tabel 6.50 Kasus Uji Melihat Data Pasien pada Dokter

Nama Kasus Uji	Kasus Uji Melihat Data Pasien
Kode Kebutuhan	SRS-RSS-F-17
Prosedur	Dokter memencet menu <i>View Patient List</i>
Hasil yang diinginkan	Sistem memperlihatkan halaman <i>View Patient List</i>
Hasil	Sistem memperlihatkan halaman <i>View Patient List</i>
Status	Valid

Tabel 6.51 Kasus Uji Melihat Profil pada Admin

Nama Kasus Uji	Kasus Uji Melihat Profil
Kode Kebutuhan	SRS-RSS-F-7
Prosedur	<i>Admin</i> memencet tombol <i>Profile</i> di sebelah kanan pojok atas halaman utama

Hasil yang diinginkan	Sistem memperlihatkan halaman profil
Hasil	Sistem memperlihatkan halaman profil
Status	Valid

Tabel 6.52 Kasus Uji Melihat Profil pada Dokter

Nama Kasus Uji	Kasus Uji Melihat Profil
Kode Kebutuhan	SRS-RSS-F-18
Prosedur	Dokter memencet tombol <i>Profile</i> di sebelah kanan pojok atas halaman utama
Hasil yang diinginkan	Sistem memperlihatkan halaman profil
Hasil	Sistem memperlihatkan halaman profil
Status	Valid

Tabel 6.53 Kasus Uji Melihat Profil pada Direktur

Nama Kasus Uji	Kasus Uji Melihat Profil
Kode Kebutuhan	SRS-RSS-F-38
Prosedur	Direktur menekan tombol <i>Profile</i> di sebelah kanan pojok atas halaman utama
Hasil yang diinginkan	Sistem memperlihatkan halaman profil
Hasil	Sistem memperlihatkan halaman profil
Status	Valid

Tabel 6.54 Kasus Uji Mengubah Data Rekam Medis

Nama Kasus Uji	Kasus Uji Mengubah Data Rekam Medis
Kode Kebutuhan	SRS-RSS-F-16
Prosedur	<ol style="list-style-type: none"> 1. Aktor meletakkan kursor pada kolom pencarian di sebelah kiri atas halaman, lalu aktor mengetik nomor ID pasien untuk melihat data rekam medis pasien 2. Aktor memencet tombol <i>enter</i> pada <i>keyboard</i> 3. Sistem memperlihatkan hasil pencarian 4. Aktor memencet tombol berwarna kuning bertuliskan <i>Edit Medical History</i> 5. Sistem menampilkan halaman <i>Edit Medical History</i>



	6. Aktor mengisi perubahan pada data rekam medis pasien, lalu menekan tombol <i>update medical history</i>
Hasil yang diinginkan	Sistem memperlihatkan pesan bahwa data pasien berhasil diubah dan tersimpan di <i>database</i>
Hasil	Sistem memperlihatkan pesan bahwa data pasien berhasil diubah dan tersimpan di <i>database</i>
Status	Valid

Tabel 6.55 Kasus Uji Menghapus Data Rekam Medis

Nama Kasus Uji	Kasus Uji Menghapus Data Rekam Medis
Kode Kebutuhan	SRS-RSS-F-15
Prosedur	<ol style="list-style-type: none"> 1. Aktor meletakkan kursor pada kolom pencarian di sebelah kiri atas halaman, lalu aktor mengetik nomor ID pasien untuk melihat data rekam medis pasien 2. Aktor memencet tombol <i>enter</i> pada <i>keyboard</i> 3. Sistem memperlihatkan hasil pencarian 4. Aktor memencet tombol berwarna merah bertuliskan <i>delete</i> 5. Sistem memperlihatkan peringatan yakin untuk menghapus atau tidak 6. Aktor memencet tombol <i>delete</i> (yakin untuk menghapus)
Hasil yang diinginkan	Sistem kembali ke halaman <i>View Medical History</i>
Hasil	Sistem kembali ke halaman <i>View Medical History</i>
Status	Valid

Tabel 6.56 Kasus Uji Menghapus Data Rekam Medis

Alternatif 1: Aktor menekan tombol *cancel* (tidak jadi menghapus)

Nama Kasus Uji	Kasus Uji Menghapus Data Rekam Medis
Kode Kebutuhan	SRS-RSS-F-15
Prosedur	<ol style="list-style-type: none"> 1. Aktor meletakkan kursor pada kolom pencarian di sebelah kiri atas halaman, lalu aktor mengetik nomor ID pasien untuk melihat data rekam medis pasien 2. Aktor memencet tombol <i>enter</i> pada <i>keyboard</i> 3. Sistem memperlihatkan hasil pencarian 4. Aktor memencet tombol berwarna merah bertuliskan <i>delete</i>

	5. Sistem memperlihatkan peringatan yakin untuk menghapus atau tidak 6. Aktor memencet tombol <i>cancel</i> (tidak jadi menghapus)
Hasil yang diinginkan	Sistem kembali ke halaman <i>View Medical History</i>
Hasil	Sistem kembali ke halaman <i>View Medical History</i>
Status	Valid

Tabel 6.57 Kasus Uji Melihat Data Rekam Medis

Nama Kasus Uji	Kasus Uji Melihat Data Rekam Medis
Kode Kebutuhan	SRS-RSS-F-21
Prosedur	Aktor memilih menu <i>View Medical History</i>
Hasil yang diinginkan	Sistem memperlihatkan halaman <i>View Medical History</i> yang berisi data-data rekam medis pasien
Hasil	Sistem memperlihatkan halaman <i>View Medical History</i> yang berisi data-data rekam medis pasien
Status	Valid

Tabel 6.58 Kasus Uji Melihat Masa Berlaku SIP

Nama Kasus Uji	Kasus Uji Melihat Masa Berlaku SIP
Kode Kebutuhan	SRS-RSS-F-20
Prosedur	Dokter memencet menu <i>View Notification</i>
Hasil yang diinginkan	Sistem memperlihatkan halaman <i>Notification</i> yang berisi informasi/peringatan mengenai masa berlaku SIP milik aktor
Hasil	Sistem memperlihatkan halaman <i>Notification</i> yang berisi informasi/peringatan mengenai masa berlaku SIP milik aktor
Status	Valid

Tabel 6.59 Kasus Uji Mengunduh Laporan Admin

Nama Kasus Uji	Kasus Uji Mengunduh Laporan Admin
Kode Kebutuhan	SRS-RSS-F-44
Prosedur	Aktor memencet tombol <i>download file</i>
Hasil yang diinginkan	<i>File</i> berhasil diunduh dan tersimpan di <i>device</i> milik aktor

Hasil	File berhasil diunduh dan tersimpan di <i>device</i> milik aktor
Status	Valid

Tabel 6.60 Kasus Uji Mengunduh Laporan Dokter

Nama Kasus Uji	Kasus Uji Mengunduh Laporan Dokter
Kode Kebutuhan	SRS-RSS-F-41
Prosedur	Aktor memencet tombol <i>download file</i>
Hasil yang diinginkan	File berhasil diunduh dan tersimpan di <i>device</i> milik aktor
Hasil	File berhasil diunduh dan tersimpan di <i>device</i> milik aktor
Status	Valid

Tabel 6.61 Kasus Uji Menghapus Laporan Admin

Nama Kasus Uji	Kasus Uji Menghapus Laporan Admin
Kode Kebutuhan	SRS-RSS-F-43
Prosedur	1. Aktor memencet tombol berwarna merah bertuliskan <i>delete</i> 2. Aktor tombol <i>delete</i> (yakin untuk menghapus)
Hasil yang diinginkan	Sistem kembali ke halaman <i>View Admin's Report List</i> (laporan berhasil dihapus)
Hasil	Sistem kembali ke halaman <i>View Admin's Report List</i> (laporan berhasil dihapus)
Status	Valid

Tabel 6.62 Kasus Uji Menghapus Laporan Admin

Alternatif 1: Direktur menekan tombol *cancel* (tidak jadi menghapus)

Nama Kasus Uji	Kasus Uji Menghapus Laporan Admin
Kode Kebutuhan	SRS-RSS-F-43
Prosedur	1. Direktur menekan tombol berwarna merah bertuliskan <i>delete</i> 2. Direktur menekan tombol <i>cancel</i> (tidak jadi menghapus)
Hasil yang diinginkan	Sistem kembali ke halaman <i>View Admin's Report List</i> (laporan berhasil dihapus)
Hasil	Sistem kembali ke halaman <i>View Admin's Report List</i> (laporan berhasil dihapus)

Status	Valid
--------	-------

Tabel 6.63 Kasus Uji Menghapus Laporan Dokter

Nama Kasus Uji	Kasus Uji Menghapus Laporan Dokter
Kode Kebutuhan	SRS-RSS-F-40
Prosedur	1. Direktur memencet tombol berwarna merah bertuliskan <i>delete</i> 2. Direktur menekan tombol <i>delete</i> (yakin untuk menghapus)
Hasil yang diinginkan	Sistem kembali ke halaman <i>View Doctor's Report List</i> (laporan berhasil dihapus)
Hasil	Sistem kembali ke halaman <i>View Doctor's Report List</i> (laporan berhasil dihapus)
Status	Valid

Tabel 6.64 Kasus Uji Menghapus Laporan Dokter

Alternatif 1: Direktur menekan tombol *cancel* (tidak jadi menghapus)

Nama Kasus Uji	Kasus Uji Menghapus Laporan Dokter
Kode Kebutuhan	SRS-RSS-F-40
Prosedur	1. Direktur memencet tombol berwarna merah bertuliskan <i>delete</i> 2. Direktur menekan tombol <i>cancel</i> (tidak jadi menghapus)
Hasil yang diinginkan	Sistem kembali ke halaman <i>View Doctor's Report List</i> (laporan berhasil dihapus)
Hasil	Sistem kembali ke halaman <i>View Doctor's Report List</i> (laporan berhasil dihapus)
Status	Valid

Tabel 6.65 Kasus Uji Melihat Laporan Dokter

Nama Kasus Uji	Kasus Uji Melihat Laporan Dokter
Kode Kebutuhan	SRS-RSS-F-39
Prosedur	Aktor memilih menu <i>View Doctor's Report List</i> untuk melihat daftar laporan dari dokter
Hasil yang diinginkan	Sistem memperlihatkan halaman <i>Doctor's Report List</i>
Hasil	Sistem memperlihatkan halaman <i>Doctor's Report List</i>
Status	Valid

Tabel 6.66 Kasus Uji Melihat Laporan Admin

Nama Kasus Uji	Kasus Uji Melihat Laporan <i>Admin</i>
Kode Kebutuhan	SRS-RSS-F-42
Prosedur	Aktor memilih menu <i>View Admin's Report List</i> untuk melihat daftar laporan dari <i>admin</i>
Hasil yang diinginkan	Sistem memperlihatkan halaman <i>Admin's Report List</i>
Hasil	Sistem memperlihatkan halaman <i>Admin's Report List</i>
Status	Valid

Tabel 6.67 Kasus Uji Menambah Admin

Nama Kasus Uji	Kasus Uji Menambah <i>Admin</i>
Kode Kebutuhan	SRS-RSS-F-32
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengisi data diri admin yang terdiri dari nama, <i>username</i>, <i>password</i>, alamat, nomor telepon, dan NIK 2. Aktor memencet tombol <i>admin registration</i> untuk menyimpan data diri <i>admin</i>
Hasil yang diinginkan	Sistem memperlihatkan pemberitahuan bahwa <i>admin</i> berhasil ditambahkan
Hasil	Sistem memperlihatkan pemberitahuan bahwa <i>admin</i> berhasil ditambahkan
Status	Valid

Tabel 6.68 Kasus Uji Menambah Admin

Alternatif 1: Aktor tidak mengisi seluruh kolom

Nama Kasus Uji	Kasus Uji Menambah <i>Admin</i>
Kode Kebutuhan	SRS-RSS-F-32
Prosedur	<ol style="list-style-type: none"> 1. Aktor tidak mengisi keseluruhan kolom yang ada dengan lengkap 2. Aktor memencet tombol <i>admin registration</i> untuk menyimpan data diri <i>admin</i>
Hasil yang diinginkan	Sistem memperlihatkan pesan peringatan bahwa ada kolom yang belum terisi
Hasil	Sistem memperlihatkan pesan peringatan bahwa ada kolom yang belum terisi
Status	Valid

Tabel 6.69 Kasus Uji Menambah Dokter

Nama Kasus Uji	Kasus Uji Menambah Dokter
Kode Kebutuhan	SRS-RSS-F-26
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengisi data diri admin yang terdiri dari nama, <i>username</i>, <i>password</i>, alamat, nomor telepon, SIP, masa berlaku SIP, dan NIK 2. Aktor memencet tombol <i>doctor registration</i> untuk menyimpan data diri dokter
Hasil yang diinginkan	Sistem memperlihatkan pemberitahuan bahwa dokter berhasil ditambahkan
Hasil	Sistem memperlihatkan pemberitahuan bahwa dokter berhasil ditambahkan
Status	Valid

Tabel 6.70 Kasus Uji Menambah Dokter

Alternatif 1: Aktor tidak mengisi seluruh kolom

Nama Kasus Uji	Kasus Uji Menambah Dokter
Kode Kebutuhan	SRS-RSS-F-26
Prosedur	<ol style="list-style-type: none"> 1. Aktor tidak mengisi keseluruhan kolom yang ada dengan lengkap 2. Aktor memencet tombol <i>doctor registration</i> untuk menyimpan data diri dokter
Hasil yang diinginkan	Sistem memperlihatkan pesan peringatan bahwa ada kolom yang belum terisi
Hasil	Sistem memperlihatkan pesan peringatan bahwa ada kolom yang belum terisi
Status	Valid

Tabel 6.71 Kasus Uji Menghapus Dokter

Nama Kasus Uji	Kasus Uji Menghapus Dokter
Kode Kebutuhan	SRS-RSS-F-27
Prosedur	<ol style="list-style-type: none"> 1. Aktor memencet tombol berwarna merah yang bertuliskan <i>delete</i> untuk menghapus data dokter 2. Sistem memperlihatkan peringatan yakin untuk menghapus atau tidak 3. Aktor memencet tombol <i>delete</i> (yakin untuk menghapus)

Hasil yang diinginkan	Sistem kembali ke halaman <i>Doctor's List</i> (data berhasil dihapus)
Hasil	Sistem kembali ke halaman <i>Doctor's List</i> (data berhasil dihapus)
Status	Valid

Tabel 6.72 Kasus Uji Menghapus Dokter

Alternatif 1: Aktor menekan tombol *cancel* (tidak jadi menghapus)

Nama Kasus Uji	Kasus Uji Menghapus Dokter
Kode Kebutuhan	SRS-RSS-F-27
Prosedur	<ol style="list-style-type: none"> 1. Aktor memencet tombol berwarna merah yang bertuliskan <i>delete</i> untuk menghapus data dokter 2. Sistem memperlihatkan peringatan yakin untuk menghapus atau tidak 3. Aktor memencet tombol <i>cancel</i> (tidak jadi menghapus)
Hasil yang diinginkan	Sistem kembali ke halaman <i>Doctor's List</i>
Hasil	Sistem kembali ke halaman <i>Doctor's List</i>
Status	Valid

Tabel 6.73 Kasus Uji Menghapus Admin

Nama Kasus Uji	Kasus Uji Menghapus <i>Admin</i>
Kode Kebutuhan	SRS-RSS-F-33
Prosedur	<ol style="list-style-type: none"> 1. Aktor memencet tombol berwarna merah yang bertuliskan <i>delete</i> untuk menghapus data dokter 2. Sistem memperlihatkan peringatan yakin untuk menghapus atau tidak 3. Aktor memencet tombol <i>delete</i> (yakin untuk menghapus)
Hasil yang diinginkan	Sistem kembali ke halaman <i>Admin's List</i> (data berhasil dihapus)
Hasil	Sistem kembali ke halaman <i>Admin's List</i> (data berhasil dihapus)
Status	Valid

Tabel 6.74 Kasus Uji Menghapus Admin

Alternatif 1: Aktor menekan tombol *cancel* (tidak jadi menghapus)

Nama Kasus Uji	Kasus Uji Menghapus Admin
Kode Kebutuhan	SRS-RSS-F-33
Prosedur	<ol style="list-style-type: none"> 1. Aktor memencet tombol berwarna merah yang bertuliskan <i>delete</i> untuk menghapus data dokter 2. Sistem memperlihatkan peringatan yakin untuk menghapus atau tidak 3. Aktor menekan tombol <i>cancel</i> (tidak jadi menghapus)
Hasil yang diinginkan	Sistem memperlihatkan kembali halaman <i>Admin's List</i>
Hasil	Sistem memperlihatkan kembali halaman <i>Admin's List</i>
Status	Valid

Tabel 6.75 Kasus Uji Melihat Masa Berlaku SIP Dokter

Nama Kasus Uji	Kasus Uji Melihat Masa Berlaku SIP Dokter
Kode Kebutuhan	SRS-RSS-F-31
Prosedur	Direktur memencet menu <i>View Notification</i>
Hasil yang diinginkan	Sistem memperlihatkan halaman <i>Doctor's Notification List</i> yang berisi informasi/peringatan mengenai masa berlaku SIP milik seluruh dokter
Hasil	Sistem memperlihatkan halaman <i>Doctor's Notification List</i> yang berisi informasi/peringatan mengenai masa berlaku SIP milik seluruh dokter
Status	Valid

Tabel 6.76 Kasus Uji Mengubah Admin

Nama Kasus Uji	Kasus Uji Mengubah Admin
Kode Kebutuhan	SRS-RSS-F-34
Prosedur	<ol style="list-style-type: none"> 1. Aktor memencet tombol berwarna kuning yang bertuliskan <i>edit</i> untuk mengubah data <i>admin</i> 2. Sistem memperlihatkan halaman <i>Edit Admin</i> 3. Aktor mengubah data <i>admin</i>, lalu memencet tombol <i>update admin</i>
Hasil yang diinginkan	Sistem memperlihatkan pemberitahuan bahwa perubahan berhasil disimpan, lalu sistem kembali ke halaman <i>Admin's List</i>

Hasil	Sistem memperlihatkan pemberitahuan bahwa perubahan berhasil disimpan, lalu sistem kembali ke halaman <i>Admin's List</i>
Status	Valid

Tabel 6.77 Kasus Uji Mengubah Dokter

Nama Kasus Uji	Kasus Uji Mengubah Dokter
Kode Kebutuhan	SRS-RSS-F-28
Prosedur	<ol style="list-style-type: none"> 1. Aktor memencet tombol berwarna kuning yang bertuliskan <i>edit</i> untuk mengubah data dokter 2. Sistem memperlihatkan halaman <i>Edit Doctor</i> 3. Aktor mengubah data dokter, lalu menekan tombol <i>update doctor</i>
Hasil yang diinginkan	Sistem memperlihatkan pemberitahuan bahwa perubahan berhasil disimpan, lalu sistem kembali ke halaman <i>Doctor's List</i>
Hasil	Sistem memperlihatkan pemberitahuan bahwa perubahan berhasil disimpan, lalu sistem kembali ke halaman <i>Doctor's List</i>
Status	Valid

Tabel 6.78 Kasus Uji Melihat Daftar Dokter

Nama Kasus Uji	Kasus Uji Melihat Daftar Dokter
Kode Kebutuhan	SRS-RSS-F-29
Prosedur	Aktor memencet menu <i>View Doctor's List</i>
Hasil yang diinginkan	Sistem memperlihatkan halaman <i>Doctor's List</i>
Hasil	Sistem memperlihatkan halaman <i>Doctor's List</i>
Status	Valid

Tabel 6.79 Kasus Uji Melihat Daftar Admin

Nama Kasus Uji	Kasus Uji Melihat Daftar Admin
Kode Kebutuhan	SRS-RSS-F-35
Prosedur	Aktor memencet menu <i>View Admin's List</i>
Hasil yang diinginkan	Sistem memperlihatkan halaman <i>Admin's List</i>
Hasil	Sistem memperlihatkan halaman <i>Admin's List</i>
Status	Valid

Tabel 6.80 Kasus Uji Mencari Admin

Alternatif 1: Aktor mengisi ID yang tidak terdaftar sebagai *admin*

Nama Kasus Uji	Kasus Uji Mencari <i>Admin</i>
Kode Kebutuhan	SRS-RSS-F-36
Prosedur	Aktor meletakkan kursor di kolom pencarian di sebelah kiri atas halaman, lalu mengetikkan nomor ID <i>admin</i> , dan kemudian menekan tombol <i>enter</i> pada <i>keyboard</i>
Hasil yang diinginkan	Sistem memperlihatkan data <i>admin</i> yang dicari
Hasil	Sistem memperlihatkan data <i>admin</i> yang dicari
Status	Valid

Tabel 6.81 Kasus Uji Mencari Admin

Alternatif 1: Aktor mengisi ID yang tidak terdaftar sebagai *admin*

Nama Kasus Uji	Kasus Uji Mencari <i>Admin</i>
Kode Kebutuhan	SRS-RSS-F-36
Prosedur	Aktor mengetik nomor ID yang tidak terdaftar sebagai <i>admin</i> lalu menekan tombol <i>enter</i>
Hasil yang diinginkan	Sistem memperlihatkan halaman kosong (tidak ditemukan data)
Hasil	Sistem memperlihatkan halaman kosong (tidak ditemukan data)
Status	Valid

Tabel 6.82 Kasus Uji Mencari Dokter

Nama Kasus Uji	Kasus Uji Mencari <i>Admin</i>
Kode Kebutuhan	SRS-RSS-F-30
Prosedur	Aktor meletakkan kursor di kolom pencarian di sebelah kiri atas halaman, lalu mengetikkan nomor ID dokter, dan kemudian menekan tombol <i>enter</i> pada <i>keyboard</i>
Hasil yang diinginkan	Sistem memperlihatkan data dokter yang dicari
Hasil	Sistem memperlihatkan data dokter yang dicari
Status	Valid

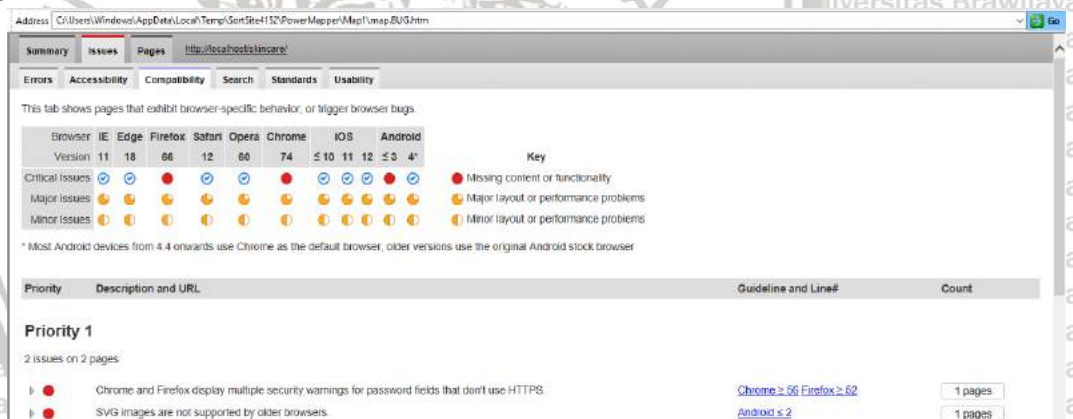
Tabel 6.83 Kasus Uji Mencari Dokter

Alternatif 1: Aktor mengisi ID yang tidak terdaftar sebagai dokter

Nama Kasus Uji	Kasus Uji Mencari Dokter
Kode Kebutuhan	SRS-RSS-F-30
Prosedur	Aktor mengetik nomor ID yang tidak terdaftar sebagai dokter lalu memencet tombol <i>enter</i>
Hasil yang diinginkan	Sistem memperlihatkan halaman kosong (tidak ditemukan data)
Hasil	Sistem memperlihatkan halaman kosong (tidak ditemukan data)
Status	Valid

6.4 Pengujian *Compatibility*

Pengujian ini dilaksanakan menggunakan *software SortSite* versi 5, alat pengujian yang bekerja dengan cara melakukan analisa terhadap struktur *web* guna mengetahui apakah ada bagian dari sistem yang tidak kompatibel dengan suatu *web*.



Gambar 6.6 Hasil Pengujian *Compatibility* Aplikasi

Berdasarkan Gambar 6.4, masalah yang ada pada sistem yang mengindikasikan adanya fungsi yang tidak didukung pada *web* tertentu, serta mengekspos pula adanya masalah utama performa di beberapa *web*.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil pengkajian dari awal sampai dengan akhir, maka dapat disimpulkan:

1. Hasil analisis kebutuhan Sistem Pembangunan Sistem Rekam Medis Ramdani *Skincare & Spa* Malang dengan metode *prototyping* berbasis web, memiliki kebutuhan fungsional sebanyak empat puluh empat dan satu kebutuhan non-fungsional yang menjadi perolehan hasil di tahap ini. Bersama dengan itu, admin, dokter, dan direktur merupakan para aktor yang turut berkontribusi pada sistem. Penganalisisan dari studi literatur yang dilakukan, serta wawancara, dan juga observasi telah memberikan hasil yang disesuaikan dengan kebutuhan yang dibutuhkan oleh Ramdani *Skincare* dan *Spa* Malang.
2. Berdasarkan hasil analisis kebutuhan yang diterapkan pada tahap perancangan, hasil yang diperoleh dari tahap perancangan meliputi, perancangan *class diagram*, *diagram sequence*, algoritme, basis data, dan *interface*.
3. Spesifikasi untuk mengembangkan sistem, pengimplementasian basis data, kode pemrograman, dan antarmuka merupakan hasil yang didapatkan pada tahap implementasi dari Sistem Rekam Medis Ramdani *Skincare* dan *Spa* Malang dengan metode *Prototyping* berbasis Web.

Pengujian validasi, unit, integrasi, dan pengujian *compatibility* merupakan tahapan dalam pengujian yang telah dijalankan. Pengujian unit serta integrasi dengan teknik *whitebox testing* dan untuk teknik *blackbox testing* dilakukan untuk pengujian validasi. Hasil seratus persen valid untuk pengujian validasi terhadap tujuh puluh tiga kasus uji fungsional.

7.2 Saran

Terdapat beberapa saran yang dapat dijadikan acuan untuk penelitian mendatang:


1. Menambahkan sistem monitoring serta evaluasi kelengkapan rekam medis, serta pemberitahuan akan risiko jika tidak mengisi riwayat medis dengan lengkap.
2. Mengadakan pelatihan pengembangan staf demi membangun rekam medis terkomputerisasi secara penuh.
3. Menambahkan kebutuhan untuk dapat melakukan pembayaran dengan satu sistem yang sama.

DAFTAR REFERENSI

- Afrina, M. dan Ibrahim, A. 2015. *Pengembangan sistem informasi SMS Gateway dalam meningkatkan layanan komunikasi sekitar akademika Fakultas Ilmu Komputer Unsri*. S1. Universitas Sriwijaya. Tersedia di: <<https://media.neliti.com/media/publications/131413-ID-pengembangan-sistem-informasi-sms-gatewa.pdf>>[Diakses 13 Februari 2021]
- Ariza, N. dan Fauziah, A. 2009. *Sistem informasi sekolah dasar berbasis SMS Gateway*. Yogyakarta: SNATI.
- Budiardi, Aditya. 2014. *Rancang Bangun Sistem Informasi Rekam Medis Klinik dan Apotek Terintegrasi (Studi Kasus: Klinik Mekarsari Husada Ampel Boyolali)*. S1. Universitas Diponegoro. Tersedia di:<http://eprints.undip.ac.id/59581/1/J2F009019_1_bab1.pdf>[Diakses 3 April 2021]
- Fathansyah. 2012. *Basis Data*. Jakarta: Gramedia.
- Ichron, Muhammad Cholil. 2014. *Sistem Informasi Rekam Medis Pasien pada Pelayanan Kesehatan Klinik Aulia Medica Jepara*. S1. Universitas Muria Kudus. Tersedia di: <http://eprints.umk.ac.id/3591/1/HALAMAN_DEPAN.pdf>[Diakses pada 14 April 2021]
- Istiaji, Rizki Yunika. 2017. *Perancangan Sistem Rekam Medis Informasi Pasien Rawat Jalan menggunakan PHP dan MySQL di Puskesmas Sumber Agung Magetan*. S1. Universitas Muhammadiyah Ponorogo. Tersedia di: <<http://eprints.umpo.ac.id/3422/1/HALAMAN%20DEPAN%20.pdf>>[Diakses 7 Maret 2021]
- Nugroho, B. 2004. *Aplikasi pemrograman web dinamis dengan PHP dan MySQL*. Yogyakarta: Gava Media.
- Rizky, Soetam. 2011. *Konsep Dasar Rekayasa Perangkat Lunak*. Jakarta: Prestasi Pustaka.
- Rosa, A. S., & Shalahuddin, M. 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur & Berorientasi Objek)*. Bandung: Modula.
- Rosa, A. S., & Shalahuddin, M. 2014. *Rekayasa Perangkat Lunak Terstruktur & Berorientasi Objek*. Bandung: Informatika.
- Sari, Desi Novita. 2018. *Sistem Informasi Rekam Medis Pasien Klinik Mulia Klaten Selatan*. S1. Universitas Widya Dharma. Tersedia di:<<http://repository.unwidha.ac.id/1340/1/Desi%20N.S%20fix.pdf>>[Diakses 7 Mei 2021]
- Sidik, B. 2012. *Framework Codeigniter*. Bandung: Penerbit Informatika.
- Sommerville, Ian. 2011. *Software Engineering 9th edition*. Pearson Education, Inc.

LAMPIRAN

A.1 FORMULIR PENDAFTARAN



KARTU REKAM MEDIK PASIEN
RAMDANI SKINCARE
 Jl. Raya Jatikerto No. 78 Malang Telp. (0341) 391800

NO REGISTER

--	--	--	--	--	--	--	--

ALERGI : YA / TIDAK

Nama :

Alamat :

Umur :


No. Tlp :

Tanggal	Anamnesa dan Pemeriksaan Fisik	Diagnosa	Terapi

Lampiran 1 Formulir Pendaftaran



A.2 DATA REKAM MEDIS PASIEN 1



KARTU REKAM MEDIK PASIEN
RAMDANI SKINCARE

Jl. Raya Jatikerto No. 78 Malang Telp. (0341) 391800

NO. REGISTER
16-08-0719


ALERGI : YA / TIDAK

Nama : Nr. Muntas
 Alamat : Jl. ra. kartini Sambisgede
 Umur : 17 Tahun. L/P
 No. Telp : 082 230 407 702

Tanggal	Anamnesa dan Pemeriksaan Fisik	Diagnosis	Terapi
10/1/2019			CC pagi 2 FW B CC B CC 1
17/10/2020			FW.B 3x CC 2 1x AAE 3x Bedak tabur. ^{Pis} <u>warita</u> Pees: <u>whu</u> atau <u>1A</u> * Bahan pees: SB <u>Alungah</u> SB <u>alocan</u> 1 menit. wash tay + air es = <u>happ</u> abut <u>huj</u> . + (senkan ↓ 9 jam. Maken, <u>hand</u> .

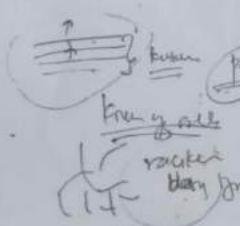
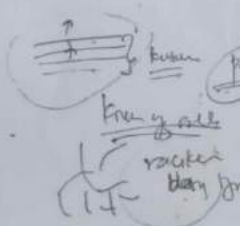
Lampiran 2 Data Rekam Medis Pasien 1

A.3 DATA REKAM MEDIS PASIEN 2


KARTU REKAM MEDIK PASIEN
RAMDANI SKINCARE
 Jl. Raya Jatikerto No. 78 Malang Telp. (0341) 391800


NO. REGISTER
 16-08-5803
 ALERGI: YA/TIDAK

Nama : Nr. Dewi
 Alamat : Krebet
 Umur : 35 Tahun. L/P
 No. Telp : 085 100 671 256

Tanggal	Anamnesa dan Pemeriksaan Fisik	Diagnosis	Terapi
2/8/2020			later stela labat.
	<p>• Abut Lower</p> <p>• tambah hap 3 gm</p> 	<p>• <u>Kor.</u></p> <p>• <u>pasj</u> <u>bekor kulit mati</u></p> <p>• <u>atitri</u></p>	
15/8/2020	<p>• <u>Kor.</u> <u>head</u></p> <p>• <u>krung side</u></p> <p>• <u>rasikan</u> <u>blany gm</u></p> 	<p>• <u>Mantol</u> <u>gum almond</u></p>	<p>• <u>ekspulsi</u></p> <p>• <u>FurB</u> <u>2X/di</u></p> <p>• <u>glue almond</u></p> <p>• <u>SPF 2 hari</u> <u>alca</u></p> <p>• <u>oil almond</u></p> <p>• <u>bedak muruk</u></p>

Lampiran 3 Data Rekam Medis Pasien 2

A.4 DATA REKAM MEDIS PASIEN 3



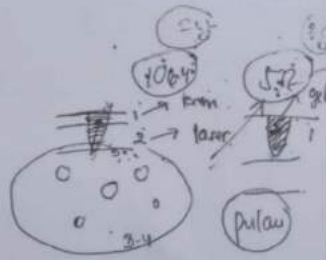
KARTU REKAM MEDIK PASIEN
RAMDANI SKINCARE

Jl. Raya Jatikerto No. 78 Malang Telp. (0341) 391800

NO. REGISTER
16-08-078

ALERGI : YA / TIDAK

Nama : M. Jawarni
 Alamat : Capotomulyo 1/2 kepanjen / Sempol pagak
 Umur : 41 Tahun. L/P (16-7-1973)
 No. Telp : 081.931.844.239

Tanggal	Anamnesa dan Pemeriksaan Fisik	Diagnosis	Terapi
9/4/2017	Senja. mudah berak flak.	Melasma + Acne =	<ul style="list-style-type: none"> - Paket normal - MMT pagak malam - Krim bedak = ditaruh di saya - Cloac cream colate - SPF oil free - Bedak tabur. - Perawatan mgu dep. Retensi + pealy GA 202 + TLA 102 ↓ di flak
23/9/2019		<ul style="list-style-type: none"> - Fw.B - SPF oil free 	<ul style="list-style-type: none"> top syon.
13/1/2020	kpar → 7 hari	<ul style="list-style-type: none"> - Injeksi anti flak. - FPP oil free - Fw.B 	
7/2/2020			<ul style="list-style-type: none"> terima appt di flak syon → ampt FPP oil free

Lampiran 4 Data Rekam Medis Pasien 3