

PENGEMBANGAN SISTEM INFORMASI PERAWATAN LUKA DI PRAKTIK MANDIRI PEDIS CARE MALANG BERBASIS WEB

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Siska Irawati

NIM: 155150201111178



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2020**

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI PERAWATAN LUKA PRAKTIK MANDIRI PEDIS CARE MALANG BERBASIS WEB

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Siska Irawati

NIM: 15515020111178

Skripsi ini telah diuji dan dinyatakan lulus pada

06 Januari 2020

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing 2



Nurudin Santoso, S.T., M.T
NIP: 19740916 200012 1 001



Reza Andria Siregar, S.T, M.Kom.
NIP: 19790621 200604 1 003

Mengetahui

Ketua Jurusan Teknik Informatika



T. Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 19710518 200312 1 001

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 referensi.

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, 6 Januari 2020



Siska Irawati

NIM: 155150201111178

ABSTRAK

Siska Irawati, Pengembangan Sistem informasi Perawatan Luka di Praktik Mandiri Pedis Care Malang Berbasis web

Pembimbing: Nurudin santoso S.T, M.T dan Reza andria siregar, S.T., M.Kom.

Pengelolaan data tentang informasi kesehatan medis merupakan hal utama yang harus dipenuhi oleh setiap rumah sakit maupun tempat klinik kesehatan tertentu khususnya bila memiliki jumlah data yang besar dan kompleks untuk memudahkan manajemen data dan mutu pelayanan kepada kliennya. Akan tetapi pada klinik Pedis Care Malang masih terdapat beberapa kendala seperti belum adanya sistem informasi yang membantu pengolahan data medis, proses dan penyimpanan *file* medis yang tidak efektif dan efisien, akses data yang tidak efektif, sebagai contoh ketika membutuhkan data klien dalam kondisi cepat, harus menghubungi koordinator divisi pelayanan untuk mengirim *file* atau ke tempat praktiknya terlebih dahulu untuk mendapatkan datanya. Untuk menangani permasalahan tersebut, maka dilakukan pengembangan sistem informasi perawatan luka di praktik mandiri pedis care malang berbasis web. Proses pengembangan sistem tersebut menggunakan metode *Rapid Application Development* (RAD). Platform berbasis *website*, kerangka kerja *codeigniter*, database menggunakan MySQL. Perancangan kebutuhan menggunakan *Work Breakdown Structure* (WBS) yang terdiri dari 12 modul utama. Untuk pengujian fungsionalitasnya dilakukan strategi pengujian unit, pengujian integrasi, pengujian validasi dengan hasil 100% valid. Sedangkan pengujian non fungsional menggunakan *compatibility testing* dengan hasil menunjukkan bahwa sistem dapat berjalan dengan baik pada semua versi *browser*.

Kata kunci: sistem informasi, *RAD*, *blackbox testing*, *whitebox testing*, *compatibility testing*.

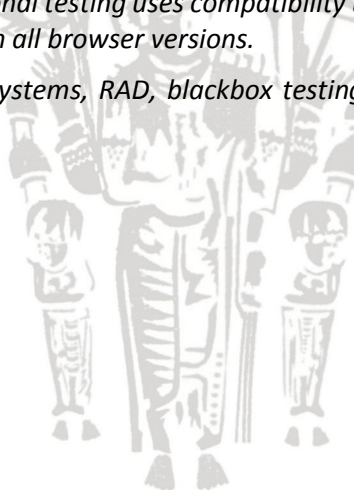
ABSTRACT

Siska Irawati, Development of wound care information systems in malang independent pedis care practice web based.

Supervisors: Nurudin santoso S.T, M.T dan Reza andria siregar, S.T., M.Kom.

Data management about medical health information is the main thing that must be fulfilled by every hospital or health clinic especially if it has a large and complex amount of data to facilitate data management and service quality to its clients. However in Pedis Care Malang clinic there are still some obstacles, such as not having an information system that helps the processing of medical data, processing and storing medical files that are ineffective and inefficient, access to data that is not effective when needing client data in a rush condition must contact the service coordinator of the division to send files or to the place of practice in advance to get data. To overcome this problem, the development of wound care information systems in web-based independent pedis care practices is very poor. The system development process uses the Rapid Application Development method. Website-based platform, CodeIgniter framework, database using MySQL. Designing needs using Work Breakdown Structure (WBS) which consists of 12 main modules. To test functionality, a unit testing strategy is carried out, validation testing with 100% valid results. While non-functional testing uses compatibility testing with results that show that the system can run well in all browser versions.

Keywords: *information systems, RAD, blackbox testing, whitebox testing, compatibility testing.*



DAFTAR ISI

PENGESAHAN Error! Bookmark not defined.

PRAKATA.....iv

ABSTRAK.....v

ABSTRACT.....vi

DAFTAR ISI.....vii

DAFTAR TABEL.....x

DAFTAR GAMBAR.....xviii

DAFTAR LAMPIRANxix

BAB 1 PENDAHULUAN.....1

1.1 Latar Belakang.....1

1.2 Rumusan Masalah.....2

1.3 Tujuan2

1.4 Manfaat.....3

1.5 Batasan Masalah.....3

1.6 Sistematika Pembahasan.....3

BAB 2 LANDASAN KEPUSTAKAAN5

2.1 Kajian Pustaka.....5

2.2 *Work Breakdown Structure (WBS)*.....6

2.3 Pengertian Sistem Informasi.....6

2.3.1 Komponen Sistem Informasi.....7

2.4 Sistem Informasi Kesehatan8

2.5 World Wide Web (WWW)8

2.6 Rekam Medis8

2.7 *Systems Development Life Cycle (SDLC)*.....9

2.7.1 *Rapid Application Development (RAD)*9

2.8 Pemodelan10

2.8.1 UML.....11

2.9 Model View Controller (MVC).....14

2.10 Pengujian Perangkat Lunak14

2.10.1 *White Box Testing*15



2.10.2 Black Box Testing	16
2.10.3 Compatibility Testing	16
2.11 Teknologi Pengembangan Sistem	16
2.11.1 Framework CodeIgniter	16
2.11.2 HTML	17
2.11.3 CSS	17
2.11.4 JavaScript	17
2.11.5 Xampp	18
2.11.6 MySQL	18
BAB 3 METODOLOGI PENELITIAN	19
3.1 Studi Literatur	19
3.2 Rekayasa Kebutuhan	19
3.3 Perancangan Sistem	20
3.4 Implementasi Sistem	22
3.5 Pengujian Sistem	22
3.6 Kesimpulan dan Saran	23
BAB 4 Analisis Kebutuhan	24
4.1 Deskripsi Sistem	24
4.2 Deskripsi Istilah dan Singkatan	24
4.3 Aturan Penomoran	24
4.4 Identifikasi Aktor	25
4.5 Daftar Kebutuhan	25
4.5.1 Kebutuhan Fungsional	26
4.5.2 Spesifikasi Kebutuhan	30
4.5.3 Kebutuhan Non Fungsional	45
4.6 Pemodelan Kebutuhan	45
4.6.1 Pemodelan Use Case Diagram	46
4.6.2 Pemodelan Use Case Scenario	47
BAB 5 PERANCANGAN DAN IMPLEMENTASI	77
5.1 Perancangan Sistem	77
5.1.1 Perancangan Arsitektur	77
5.1.2 Perancangan Data	83

5.1.3 Perancangan Komponen	97
5.1.4 Perancangan Antarmuka	100
5.1.5 Implementasi <i>Database</i>	106
5.1.6 Implementasi Kode	107
5.1.7 Implementasi Antarmuka	122
BAB 6 PENGUJIAN DAN ANALISIS	125
6.1 Pengujian Fungsional	125
6.1.1 Pengujian Unit	125
6.1.2 Pengujian Integrasi	132
6.1.3 Pengujian Validasi	136
6.1.4 Pengujian <i>Compatibility</i>	162
BAB 7 PENUTUP	163
7.1 Kesimpulan	163
7.2 Saran	164
DAFTAR PUSTAKA	165
LAMPIRAN A FORM PENDUKUNG	166

DAFTAR TABEL

Tabel 2.1 Contoh notasi <i>Use Case Diagram</i>	11
Tabel 2.2 Contoh Notasi <i>Use Case Scenario</i>	12
Tabel 2.3 Contoh Notasi <i>Sequence Diagram</i>	13
Tabel 2.4 Contoh Notasi <i>Class Diagram</i>	13
Tabel 3.1 Penjelasan Modul (WBS) Sistem Informasi Praktik Mandiri Pedis Care Malang Berbasis Web	20
Tabel 4.1 Daftar Istilah dan Singkatan	24
Tabel 4.2Daftar Aturan Penomoran.....	24
Tabel 4.3 Identifikasi Aktor Sistem.....	25
Tabel 4.4 Definisi Kebutuhan Fungsional.....	26
Tabel 4.5 Spesifikasi Kebutuhan	31
Tabel 4.6 Definisi Kebutuhan Non Fungsional	45
Tabel 4.7 <i>Use Case Scenario Login</i>	47
Tabel 4.8 <i>Use Case Scenario Logout</i>	47
Tabel 4.9 <i>Use Case Scenario</i> Daftar	48
Tabel 4.10 <i>Use Case Scenario</i> Mengubah <i>Password</i>	48
Tabel 4.11 <i>Use Case Scenario</i> Mengedit Biodata Diri	49
Tabel 4.12 <i>Use Case Scenario</i> Menambah Data Master	49
Tabel 4.13 <i>Use Case Scenario</i> Menghapus Data Master	50
Tabel 4.14 <i>Use Case Scenario</i> Merubah Hak Akses Sistem.....	50
Tabel 4.15 <i>Use Case Scenario</i> Memvalidasi Registrasi	51
Tabel 4.16 <i>Use Case Scenario</i> Menghapus Data Pengguna Sistem	51
Tabel 4.17 <i>Use Case Scenario</i> Menambah Identitas Klien.....	52
Tabel 4.18 <i>Use Case Scenario</i> Menghapus Identitas Klien	52
Tabel 4.19 <i>Use Case Scenario</i> Mengedit Identitas Klien	53
Tabel 4.20 <i>Use Case Scenario</i> Mencetak Identitas Klien	53
Tabel 4.21 <i>Use Case Scenario</i> Menambah <i>Informed Content</i>	54
Tabel 4.22 <i>Use Case Scenario</i> Menghapus <i>Informed Content</i>	54
Tabel 4.23 <i>Use Case Scenario</i> Mengedit <i>Informed Content</i>	55
Tabel 4.24 <i>Use Case Scenario</i> Mencetak <i>Informed Content</i>	55



Tabel 4.25 <i>Use Case Scenario</i> Menambah Pengkajian Umum	56
Tabel 4.26 <i>Use Case Scenario</i> Menghapus Pengkajian Umum	56
Tabel 4.27 <i>Use Case Scenario</i> Mengedit Form Pengkajian Umum	57
Tabel 4.28 <i>Use Case Scenario</i> Mencetak Pengkajian Umum	57
Tabel 4.29 <i>Use Case Scenario</i> Menambah Pengkajian Luka	58
Tabel 4.30 <i>Use Case Scenario</i> Menghapus Data Pengkajian Luka	58
Tabel 4.31 <i>Use Case Scenario</i> Mengedit Form Pengkajian Luka	59
Tabel 4.32 <i>Use Case Scenario</i> Mencetak Pengkajian Luka	59
Tabel 4.33 <i>Use Case Scenario</i> Menambah Data Pengkajian Betes Jensen	60
Tabel 4.34 <i>Use Case Scenario</i> Menghapus Pengkajian Betes Jensen	60
Tabel 4.35 <i>Use Case Scenario</i> Mengedit Form Pengkajian Betes Jensen	61
Tabel 4.36 <i>Use Case Scenario</i> Mencetak Pengkajian Betes Jensen	61
Tabel 4.37 <i>Use Case Scenario</i> Menambah Data Tampilan Klinis Luka	62
Tabel 4.38 <i>Use Case Scenario</i> Menghapus Data Tampilan Klinis Luka	62
Tabel 4.39 <i>Use Case Scenario</i> Mengedit Data Tampilan Klinis Luka	63
Tabel 4.40 <i>Use Case Scenario</i> Mencetak Tampilan Klinis Luka	63
Tabel 4.41 <i>Use Case Scenario</i> Menambah Tindakan	64
Tabel 4.42 <i>Use Case Scenario</i> Menghapus Tindakan	64
Tabel 4.43 <i>Use Case Scenario</i> Mengedit Tindakan	65
Tabel 4.44 <i>Use Case Scenario</i> Mencetak Tindakan	65
Tabel 4.45 <i>Use Case Scenario</i> Menambah Perkembangan Klien	66
Tabel 4.46 <i>Use Case Scenario</i> Menghapus Perkembangan Klien	66
Tabel 4.47 <i>Use Case Scenario</i> Mengedit Form Perkembangan Klien	67
Tabel 4.48 <i>Use Case Scenario</i> Mencetak Form Perkembangan Klien	67
Tabel 4.49 <i>Use Case Scenario</i> Menambah Kunjungan	68
Tabel 4.50 <i>Use Case Scenario</i> Menghapus Kunjungan	68
Tabel 4.51 <i>Use Case Scenario</i> Mengedit Form Kunjungan	69
Tabel 4.52 <i>Use Case Scenario</i> Mencetak Form Kunjungan	69
Tabel 4.53 <i>Use Case Scenario</i> Menambah Pembayaran	70
Tabel 4.54 <i>Use Case Scenario</i> Menghapus Pembayaran	70
Tabel 4.55 <i>Use Case Scenario</i> Mencetak Form Pembayaran	71
Tabel 4.56 <i>Use Case Scenario</i> Menambah Data Penolakan Tindakan	71

Tabel 4.57 <i>Use Case Scenario</i> Menghapus Data Penolakan Tindakan	72
Tabel 4.58 <i>Use Case Scenario</i> Mengedit Form Penolakan Tindakan	72
Tabel 4.59 <i>Use Case Scenario</i> Mencetak Form Penolakan Tindakan	73
Tabel 4.60 <i>Use Case Scenario</i> Menambah Data Rujukan	73
Tabel 4.61 <i>Use Case Scenario</i> Menghapus Data Rujukan	74
Tabel 4.62 <i>Use Case Scenario</i> Mengedit Form Rujukan	74
Tabel 4.63 <i>Use Case Scenario</i> Mencetak Form Rujukan	75
Tabel 4.64 <i>Use Case Scenario</i> Tambah Jadwal	75
Tabel 4.65 <i>Use Case Scenario</i> Hapus Penjadwalan	76
Tabel 5.1 Struktur Tabel <i>Calendar</i>	85
Tabel 5.2 Struktur Tabel daftar_agama	85
Tabel 5.3 Struktur Tabel daftar_bidang	85
Tabel 5.4 Struktur Tabel daftar_dressing	85
Tabel 5.5 Struktur Tabel daftar_rumahsakit	85
Tabel 5.6 Struktur Tabel daftar_status	86
Tabel 5.7 Struktur Tabel dressingklien	86
Tabel 5.8 Struktur Tabel lihatjadwal	86
Tabel 5.9 Struktur Tabel pegawai	86
Tabel 5.10 Struktur Tabel pendaftaran_pegawai	87
Tabel 5.11 Struktur Tabel profil_diklat	87
Tabel 5.12 Struktur Tabel profil_jenjangkarir	87
Tabel 5.13 Struktur Tabel profil_keluarga	88
Tabel 5.14 Struktur Tabel profil_organisasi	88
Tabel 5.15 Struktur Tabel profil_pegawai	88
Tabel 5.16 Struktur Tabel profil_pendidikan	89
Tabel 5.17 Struktur Tabel profil_pengalamankerja	89
Tabel 5.18 Struktur Tabel profil_penghargaan	90
Tabel 5.19 Struktur Tabel profil_prestasi	90
Tabel 5.20 Struktur Tabel profil_sanksi	90
Tabel 5.21 Struktur Tabel riwayatbetesjensen	90
5.22 Struktur Tabel riwayatidentitaklien	91
Tabel 5.23 Struktur Tabel riwayatindformedcontent	91



Tabel 5.24 Struktur Tabel riwayatkunjungan.....	92
Tabel 5.25 Struktur Tabel riwayatpembayaran	92
Tabel 5.26 Struktur Tabel riwayatpengkajianluka	93
Tabel 5.27 Struktur Tabel riwayatpengkajianumum	93
Tabel 5.28 Struktur Tabel riwayatpenolakantindakan	94
Tabel 5.29 Struktur Tabel perkembanganklien.....	95
Tabel 5.30 Struktur Tabel riwayattrujukan	95
Tabel 5.31 Struktur Tabel riwayattindakan	95
Tabel 5.32 Struktur Tabel riwayattampilanklinisluka	96
Tabel 5.33 Struktur Tabel Method do_riwayatbetesjensen.....	98
Tabel 5.34 Struktur Tabel Method do_changepassword().....	99
Tabel 5.35 Struktur Tabel Delete Riwayat <i>Informed Content</i>	100
Tabel 5.36 Komponen Halaman Tambah Riwayat Betes Jensen	101
Tabel 5.37 Komponen Halaman Ubah <i>Password</i>	102
Tabel 5.38 Komponen Halaman <i>Delete Informed Content</i>	103
Tabel 5.39 Spesifikasi Perangkat Keras.....	104
Tabel 5.40 Spesifikasi Perangkat Keras.....	105
Tabel 5.41 Spesifikasi Operasi.....	105
Tabel 5.42 Perintah SQL Membuat Tabel calendar	107
5.43 Perintah SQL Membuat Tabel daftar_agama	107
Tabel 5.44 Perintah SQL Membuat Tabel daftar_bidang	107
Tabel 5.45 Perintah SQL Membuat Tabel daftar_dasarpenerimaan	107
Tabel 5.46 Perintah SQL Membuat Tabel daftar_dressing	107
Tabel 5.47 Perintah SQL Membuat Tabel daftar_rumahsakit	108
Tabel 5.48 Perintah SQL Membuat Tabel daftar_status.....	108
Tabel 5.49 Perintah SQL Membuat Tabel dressingklien	108
Tabel 5.50 Perintah SQL Membuat Tabel lihatjadwal	108
Tabel 5.51 Perintah SQL Membuat Tabel pegawai.....	108
Tabel 5.52 Perintah SQL Membuat Tabel pendaftaran_pegawai.....	109
Tabel 5.53 Perintah SQL Membuat Tabel profil_diklat.....	109
Tabel 5.54 Perintah SQL Membuat Tabel profil_jenjangkarir	109
Tabel 5.55 Perintah SQL Membuat Tabel profil_keluarga	110

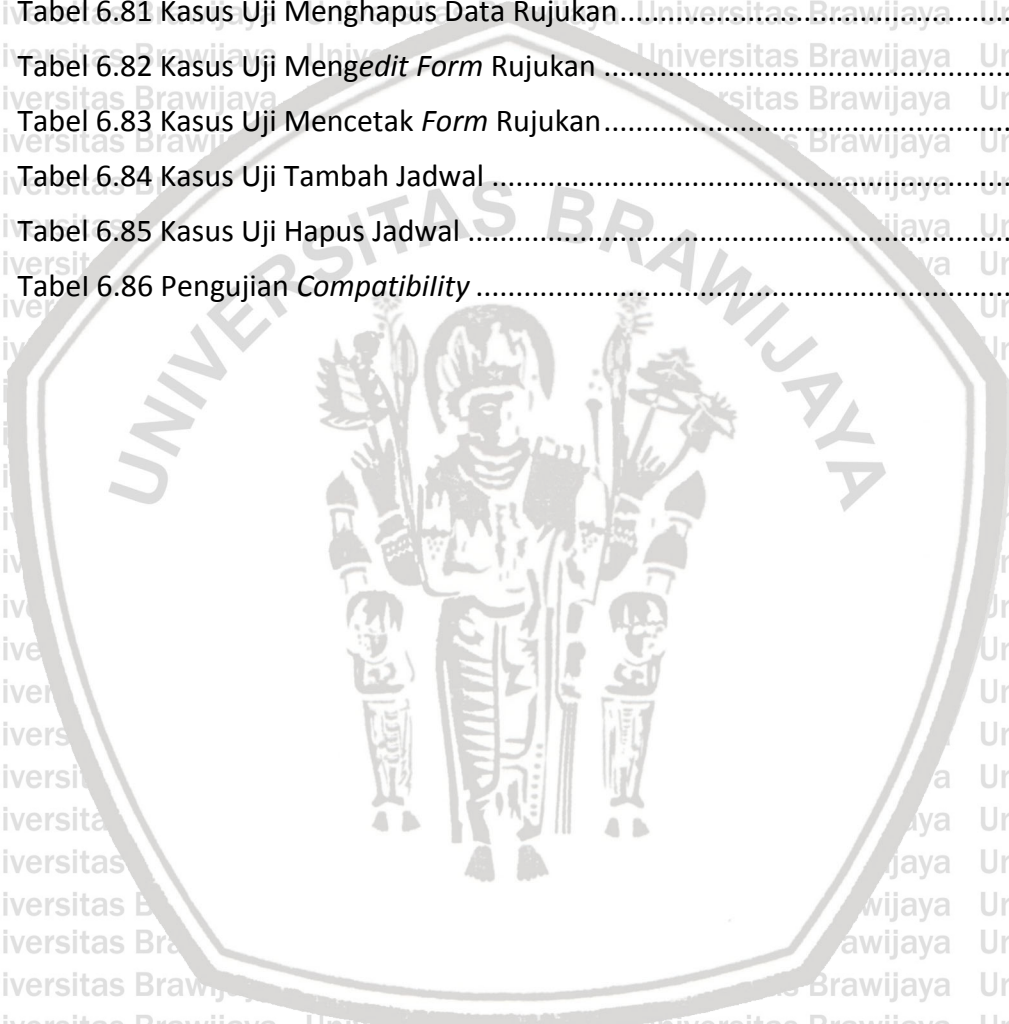


Tabel 5.56 Perintah SQL Membuat Tabel profil_organisasi	110
Tabel 5.57 Perintah SQL Membuat Tabel profil_pegawai	110
Tabel 5.58 Perintah SQL Membuat Tabel profil_pendidikan	111
Tabel 5.59 Perintah SQL Membuat Tabel profil_pengalamankerja	111
Tabel 5.60 Perintah SQL Membuat Tabel profil_penghargaan	111
Tabel 5.61 Perintah SQL Membuat Tabel profil_prestasi	111
Tabel 5.62 Perintah SQL Membuat Tabel profil_sanksi	112
Tabel 5.63 Perintah SQL Membuat Tabel riwayatbetesjensen	112
5.64 Perintah SQL Membuat Tabel riwayatidentitasklien	112
Tabel 5.65 Perintah SQL Membuat Tabel riwayatinformendcontent	113
Tabel 5.66 Perintah SQL Membuat Tabel riwayatkunjugan	113
Tabel 5.67 Perintah SQL Membuat Tabel riwayatpembayaran	114
Tabel 5.68 Perintah SQL Membuat Tabel riwayatpengkajianluka	114
Tabel 5.69 Perintah SQL Membuat Tabel riwayatpengkajianumum	114
Tabel 5.70 Perintah SQL Membuat Tabel riwayatpenolakantindakan	115
Tabel 5.71 Perintah SQL Membuat Tabel riwayatperkembanganklien	115
Tabel 5.72 Perintah SQL Membuat Tabel riwayattrjukan	115
Tabel 5.73 Perintah SQL Membuat Tabel riwayattampilanklinisluka	116
Tabel 5.74 Perintah SQL Membuat Tabel riwayattindakan	117
Tabel 5.75 Implementsi Tambah Riwayat Betes Jensen	117
Tabel 5.76 Implementsi Ubah <i>Password</i>	121
Tabel 5.77 Implementsi Hapus <i>Informed Content</i>	121
Tabel 6.1 Node Fungsi do_riwayatbetesjensen	125
Tabel 6.2 Tase Case Fungsi do_riwayatbetesjensen()	128
Tabel 6.3 Node Fungsi do_changepassword	128
Tabel 6.4 Tase Case Fungsi do_changepassword()	130
Tabel 6.5 Fungsi do_deleteriyawatinformendcontent()	130
Tabel 6.6 Tase Case Fungsi do_deleteriyawatinformendcontent()	131
Tabel 6.7 Algoritme Fungsi do_changepassword	132
Tabel 6.8 Algoritme Fungsi change_password	133
Tabel 6.9 Algoritme Gabungan do_changepassword	134
Tabel 6.10 Tase Case Fungsi do_changepassword()	136

Tabel 6.11 Kasus Uji <i>Login</i>	136
Tabel 6.12 Kasus Uji Alternatif <i>Login</i>	137
Tabel 6.13 Kasus Uji Alternatif <i>Login</i>	137
Tabel 6.14 Kasus Uji <i>Logout</i>	137
Tabel 6.15 Kasus Uji Daftar	137
Tabel 6.16 Kasus Uji Alternatif Daftar.....	138
Tabel 6.17 Kasus Uji Mengubah <i>Password</i>	138
Tabel 6.18 Kasus Uji Alternatif Mengubah <i>Password</i>	139
Tabel 6.19 Kasus Uji Alternatif Mengubah <i>Password</i>	139
Tabel 6.20 Kasus Uji Mengedit Biodata Diri.....	139
Tabel 6.21 Kasus Uji Alternatif Mengedit Biodata Diri	140
Tabel 6.22 Kasus Uji Menambah Data Master.....	140
Tabel 6.23 Kasus Uji Hapus Data Master	140
Tabel 6.24 Kasus Uji Merubah Hak Akses Sistem	141
Tabel 6.25 Kasus Uji Memvalidasi Registrasi.....	141
Tabel 6.26 Kasus Uji Menghapus Data Pengguna Sistem	141
Tabel 6.27 Kasus Uji Menambah Identitas Klien.....	142
Tabel 6.28 Kasus Uji Alternatif Menambah Identitas Klien	142
Tabel 6.29 Kasus Uji Menghapus Identitas Klien	142
Tabel 6.30 Kasus Uji Mengedit Identitas Klien.....	143
Tabel 6.31 Kasus Uji Mencetak Identitas Klien	143
Tabel 6.32 Kasus Uji Menambah <i>Informed Content</i>	143
Tabel 6.33 Kasus Uji Alternatif Menambah <i>Informed Content</i>	144
Tabel 6.34 Kasus Uji Menghapus <i>Informed Content</i>	144
Tabel 6.35 Kasus Uji Mengedit <i>Informed Content</i>	144
Tabel 6.36 Kasus Uji Menambah Pengkajian Umum.....	145
Tabel 6.37 Kasus Uji Alternatif Menambah Pengkajian Umum.....	145
Tabel 6.38 Kasus Uji Menghapus Data Pengkajian Umum	145
Tabel 6.39 Kasus Uji Mengedit <i>Form</i> Pengkajian Umum.....	146
Tabel 6.40 Kasus Uji Mencetak Pengkajian Umum.....	146
Tabel 6.41 Kasus Uji Menambah Pengkajian Luka.....	146
Tabel 6.42 Kasus Uji Alternatif Menambah Pengkajian Luka	147

Tabel 6.43 Kasus Uji Menghapus Data Pengkajian Luka	147
Tabel 6.44 Kasus Uji Mengedit <i>Form</i> Pengkajian Luka	147
Tabel 6.45 Kasus Uji Mencetak Pengkajian Luka	148
Tabel 6.46 Kasus Uji Menambah Betes Jensen	148
Tabel 6.47 Kasus Uji Alternatif Menambah Betes Jensen	148
Tabel 6.48 Kasus Uji Menghapus Bete Jensen	149
Tabel 6.49 Kasus Uji Mengedit Betes Jensen	149
Tabel 6.50 Kasus Uji Mencetak Pengkajian Betes Jensen	150
Tabel 6.51 Kasus Uji Menambah Data Tampilan Klinis Luka	150
Tabel 6.52 Kasus Uji ALternatif Menambah Data Tampilan Klinis Luka	150
Tabel 6.53 Kasus Uji Menghapus Data Tampilan klinis Luka	151
Tabel 6.54 Kasus Uji Mengedit Tampilan Klinis Luka	151
Tabel 6.55 Kasus Uji Mencetak Tampilan Klinis Luka	151
Tabel 6.56 Kasus Uji Menambah Tindakan	152
Tabel 6.57 Kasus Uji Alternatif Menambah Tindakan	152
Tabel 6.58 Kasus Uji Menghapus Tindakan	152
Tabel 6.59 Kasus Uji Mencetak Tindakan	153
Tabel 6.60 Kasus Uji Menambah Perkembangan Klien	153
Tabel 6.61 Kasus Uji Alternatif Menambah Perkembangan Klien	153
Tabel 6.62 Kasus Uji Menghapus Data Perkembangan Klien	154
Tabel 6.63 Kasus Uji Mengedit <i>Form</i> Perkembangan Klien	154
Tabel 6.64 Kasus Uji Mencetak <i>Form</i> Perkembangan Klien	154
Tabel 6.65 Kasus Uji Menambah Kunjungan	155
Tabel 6.66 Kasus Uji Alternatif Menambah Kunjungan	155
Tabel 6.67 Kasus Uji Menghapus Kunjungan	155
Tabel 6.68 Kasus Uji Mengedit <i>Form</i> Kunjungan	156
Tabel 6.69 Kasus Uji Mencetak Kunjungan	156
Tabel 6.70 Kasus Uji Menambah Pembayaran	156
Tabel 6.71 Kasus Uji Alternatif Menambah Pembayaran	157
Tabel 6.72 Kasus Uji Menghapus <i>Form</i> Pembayaran	157
Tabel 6.73 Kasus Uji Mencetak <i>Form</i> Pembayaran	157
Tabel 6.74 Kasus Uji Menambah Penolakan Tindakan	158

Tabel 6.75 Kasus Uji Alternatif Menambah Penolakan Tindakan	158
Tabel 6.76 Kasus Uji Menghapus Penolakan Tindakan	158
Tabel 6.77 Kasus Uji Mengedit <i>Form</i> Penolakan Tindakan	159
Tabel 6.78 Kasus Uji Mencetak penolakan Tindakan	159
Tabel 6.79 Kasus Uji Menambah Data Rujukan	159
Tabel 6.80 Kasus Uji Alternatif Menambah Data Rujukan	160
Tabel 6.81 Kasus Uji Menghapus Data Rujukan	160
Tabel 6.82 Kasus Uji Mengedit <i>Form</i> Rujukan	160
Tabel 6.83 Kasus Uji Mencetak <i>Form</i> Rujukan	161
Tabel 6.84 Kasus Uji Tambah Jadwal	161
Tabel 6.85 Kasus Uji Hapus Jadwal	161
Tabel 6.86 Pengujian <i>Compatibility</i>	162



DAFTAR GAMBAR

Gambar 2.1 <i>Work Breakdown Structure</i> (WBS)	6
Gambar 2.2 Siklus RAD	9
Gambar 3.1 Diagram Alir Metode Penelitian	19
Gambar 3.2 <i>Work Breakdown Structure</i> (WBS) Sistem Informasi Perawatan Luka Praktik mandiri Pedis Care Malang Berbasis Web.	20
Gambar 4.1 <i>Use Case Diagram</i> Sistem Informasi Perawatan Luka Praktik Mandiri Pedis Care Malang Berbasis Web	46
Gambar 5.1 Arsitektur Sistem	77
Gambar 5.2 Sequence Diagram Tambah Riwayat <i>Betes Jensen</i>	78
Gambar 5.3 Sequence Diagram Mengubah <i>Password</i>	79
Gambar 5.4 Sequence Diagram Cetak <i>Betes Jensen</i>	79
Gambar 5.5 Perancangan <i>Class Diagram</i>	80
Gambar 5.6 Model <i>Class Diagram</i>	81
Gambar 5.7 Controller <i>Class Diagram</i>	82
Gambar 5.8 <i>Entity Relationship Diagram</i>	84
Gambar 5.9 Rancangan Antarmuka Halaman Tambah Riwayat <i>Betes Jensen</i> ...	100
Gambar 5.10 Rancangan Antarmuka Halaman Ubah <i>Password</i>	102
Gambar 5.11 Rancangan Antarmuka Delete <i>Informed Content</i>	103
Gambar 5.12 <i>Physical Data Model</i>	106
Gambar 5.13 Implementasi Antarmuka Tambah Riwayat <i>Betes Jensen</i>	123
Gambar 5.14 Implementasi Antarmuka Ubah <i>Password</i>	123
Gambar 5.15 Implementasi Antarmuka Hapus <i>Informed Content</i>	124
Gambar 6.1 <i>Flow Graph</i> Fungsi <i>do_riwayatbetesjensen</i>	127
Gambar 6.2 <i>Flow Graph</i> Fungsi <i>do_changepassword</i> ()	129
Gambar 6.3 <i>Flow Graph</i> Fungsi <i>do_deleteriwayatinformedcontent</i> ()	131
Gambar 6.4 <i>Flow Graph</i> Fungsi <i>do_changepassword</i> ()	133
Gambar 6.5 <i>Flow Graph</i> Fungsi <i>change_password</i> ()	134
Gambar 6.6 <i>Flow Graph</i> Fungsi <i>do_changepassword</i> ()	135
Gambar 6.7 Hasil Pengujian <i>Compatibility</i> Menggunakan Tool <i>SortSite</i>	162

DAFTAR LAMPIRAN

LAMPIRAN A FORM PENDUKUNG.....	85
A.1 Roadmap Pengerjaan Skripsi Awal.....	166
A.2 Roadmap Pengerjaan Skripsi Iterasi 1.....	168
A.3 Perancangan Work Breakdown Structure (WBS) Iterasi 1.....	169
A.4 Roadmap Pengerjaan Skripsi Iterasi 2.....	170
A.5 Perancangan Work Breakdown Structure (WBS) Iterasi 2.....	171
A.6 Roadmap Pengerjaan Skripsi Iterasi 3.....	172
A.7 Perancangan Work Breakdown Structure (WBS) Iterasi 3.....	173
A.8 Roadmap Pengerjaan Skripsi Hasil Akhir.....	175
A.9 Transkrip Wawancara 1.....	176
A.10 Transkrip Wawancara 2.....	178
A.11 Halaman Sampul.....	181
A.12 Halaman Informed Content.....	182
A.13 Halaman Identitas Klien.....	183
A.14 Halaman Daftar Kunjungan.....	184
A.15 Halaman Pengkajian Umum.....	185
A.16 Halaman Pengkajian Luka.....	186
A.17 Halaman Betes Jensen.....	187
A.18 Halaman Tindakan.....	188

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perkembangan dunia sekarang menunjukkan bahwa teknologi informasi sudah sangat berkembang dengan cepat dan pesat sehingga secara tidak langsung mempengaruhi beragam aktivitas manusia. Dengan adanya dukungan sarana teknologi yang canggih memberikan berbagai manfaat salah satunya berbagai bentuk informasi lebih mudah untuk didapatkan. Perkembangan teknologi bisa berguna dengan baik bila mendapatkan dukungan yang baik juga dari sumber daya manusianya. Tidak bisa dipungkiri bahwa dunia kesehatan merupakan factor penting yang ada pada aspek kehidupan di dunia ini. Aspek-aspek yang ada di dalam dunia kesehatan sudah seharusnya memberi pelayanan yang sesuai dan tepat sasaran agar meningkatkan kepuasan terhadap kliennya (Afdaliyah, 2019). Pelayanan seharusnya memberikan keuntungan bagi kesehatan pada pihak praktik mandiri terkait. Untuk itu, guna meningkatkan mutu pelayanan kesehatan pada praktik mandiri yang terkait, sistem informasi diperlukan untuk membantu dalam mengolah data klinis yang akurat, cepat dan tepat bagi pelayanan medis (Kristiawan, 2012). Salah satu indikasi keberhasilan rumah sakit atau tempat praktik mandiri dalam meningkatkan kepercayaan masyarakat yang melakukan pemeriksaan dan kontrol adalah jumlah kunjungan rawat jalan yang tinggi, dimana sebagian besar klien tersebut akan kembali untuk melakukan pemeriksaan dan kontrol ditempat tersebut terutama bagi klien dengan penyakit kronis atau penyakit kritis lainnya.

Pedis Care adalah sebuah tempat praktik mandiri dimana proses *system* pengolahan kunjungan klien rawat jalan masih dilakukan secara manual. Tempat praktik mandiri ini melayani dibidang perawatan luka dengan metode dan teknologi terkini untuk membantu mengatasi luka kronis dan memperoleh kesehatan yang lebih baik. Pedis Care didirikan tahun 2015. Pelayanan yang tersedia pada praktik mandiri Pedis Care meliputi perawatan kaki dan food SPA, konseling diabetes, dan gaya hidup sehat, perawatan stoma, perawatan di rumah atau home care, cek gula, cek asam urat, cek kolesterol, dan cek tekanan darah. Praktik mandiri Pedis Care ini lokasinya di Jalan Mayjend Panjaitan no. 68 A Malang jawa timur dan memiliki cabang tepatnya di jalan Trunojoyo No 11, Rejoso, Junrejo, Kota Batu.

Berdasarkan hasil observasi di lapangan, dampak dari banyaknya klien yang mendaftar mengakibatkan petugas mengalami kendala dalam melakukan pencatatan data klien yang masih manual. Pada alur pemeriksaan klien rawat jalan di praktik mandiri pedis care, prosedur pertama adalah pendaftaran. Pendaftaran klien dilakukan oleh seorang petugas dengan menyiapkan *form* yang harus diisi berdasarkan data klien yang bersangkutan, data tersebut meliputi *form informed content*, identitas klien dan menganalisa diagnosa klien dengan mengisi beberapa *form* pengkajian seperti *form* pengkajian umum, *form* pengkajian luka, *form* pengkajian betes Jensen, *form* pengkajian tampilan klinis luka. Kemudian untuk

form yang sudah diisi oleh bagian pelayanan tersebut akan disimpan dalam rak khusus. Kendala selanjutnya yang terjadi pada bagian rekam medis klien yang dimana data klien direkap dari *form* kertas menjadi *soft file* dalam waktu 1 minggu masing-masing bagian pelayanan akan menyerahkan hasil rekam medis *hard copy* yang dibuat ke koordinator divisi pelayanan dan koordinator divisi pelayanan akan menyalin kembali data rekam medisnya ke *google form* sehingga pemborosan waktu terjadi di dalamnya. Kendala ke tiga yaitu sulitnya berbagi data antar divisi karna banyak diantara pegawai biasanya membutuhkan data klien dalam kondisi terburu-buru. Untuk mendapatkan data klien tersebut bagian divisi pelayanan dapat menempuh 2 cara yaitu menghubungi koordinator divisi pelayanan untuk meminta *soft filenya* atau datang ke tempat praktik untuk melihat data klien *hard copynya* secara langsung.

Dari permasalahan yang sudah didapatkan, diperlukan suatu sistem informasi yang dapat mengelola data klien di Praktik Mandiri Pedis Care Malang. Sistem informasi tersebut memiliki beberapa fitur pengolahan data yang memuat menambah data, mengedit data, menghapus data, mencetak, dan menampilkan riwayat data yang dimasukkan serta penyimpanan data yang lebih efisien. Selain itu fitur tersebut memudahkan pengaksesan data pada tiap anggota divisi. Untuk penggambaran fitur yang lebih lengkapnya dapat di lihat dalam perancangan *Work Breakdown Structure* (WBS). Penelitian dilakukan menggunakan metode *Rapid Application Development* (RAD) sebagai metode pengembangan sistem informasi. Menurut Rosa A. S. dan M. Shalanudin *Rapid Application Development* merupakan model proses pengembangan *software* yang waktu pengerjaannya berjangka pendek.

1.2 Rumusan Masalah

Berdasarkan identifikasi dan batasan masalah yang ada, dapat dibuat beberapa rumusan masalah yaitu:

1. Bagaimana hasil rekayasa kebutuhan dari pengembangan sistem informasi perawatan luka di praktik mandiri pedis care malang berbasis web?
2. Bagaimana hasil perancangan dari pengembangan sistem informasi perawatan luka di praktik mandiri pedis care malang berbasis web?
3. Bagaimana hasil implementasi dari pengembangan sistem informasi perawatan luka di praktik mandiri pedis care malang berbasis web?
4. Bagaimana hasil pengujian dari pengembangan sistem informasi perawatan luka di praktik mandiri pedis care malang berbasis web?

1.3 Tujuan

1. Rekayasa kebutuhan menghasilkan elisitasi kebutuhan, spesifikasi kebutuhan, identifikasi aktor, deskripsi sistem dan pemodelan kebutuhan yang terdapat use case diagram dan use case scenario pada pengembangan sistem informasi perawatan luka di praktik mandiri pedis care malang berbasis web.

2. Merancang sistem perangkat lunak sesuai persyaratan untuk pengembangan sistem informasi perawatan luka di praktik mandiri pedis care malang berbasis web.
3. Mengimplementasikan rancangan pengembangan sistem informasi perawatan luka di praktik mandiri pedis care malang berbasis web.
4. Pengujian modul pengembangan sistem informasi perawatan luka di praktik mandiri pedis care malang berbasis web.

1.4 Manfaat

Melalui pengembangan sistem informasi rekam medis di praktik mandiri Pedis Care berbasis web ini, maka diharapkan penelitian ini dapat bermanfaat bagi praktik mandiri Pedis Care terutama bagi pegawai yang bekerja di dalamnya. Manfaat dari sistem ini adalah:

1. Membantu mendapatkan kemudahan dalam pengolahan laporan data klinis maupun non klinis.
2. Dapat meminimalisir kehilangan dan kerusakan data.
3. Memudahkan pencarian *history* data klien.

1.5 Batasan Masalah

Adapun batasan masalah dari penelitian ini sehingga mencegah terjadinya penyimpangan dari tujuan yang sudah direncanakan :

1. Sistem hanya diterapkan di praktik mandiri Pedis Care Malang.
2. Sistem hanya dibangun untuk membantu proses manajemen pengolahan data perawatan luka, administrasi pembayaran perawatan dan penjadwalan klien berobat.
3. Sistem akan dibangun dengan menggunakan *platform* web.
4. Sistem dibangun menggunakan aplikasi web server xampp dan bahasa pemrograman PHP, HTML dan CSS.
5. Sistem menggunakan *database* MySQL .
6. *Framework* yang digunakan yaitu *CodeIgniter* dan CSSnya menggunakan Bootstrap.

1.6 Sistematika Pembahasan

Sistematika pembahasan yang digunakan peneliti adalah:

BAB I PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian yang diharapkan dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan tentang dan menguraikan teori-teori dari penelitian terdahulu yang dikaji sebagai bahan pendukung untuk mendasari penelitian.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan metode dan langkah kerja yang dilakukan dalam proses perancangan, analisis kebutuhan, dan implementasi yang menjadi objek studi kasus skripsi.

BAB IV ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang analisis kebutuhan yang dibutuhkan dan proses pengembangan sistem informasi perawatan luka di praktik mandiri pedis care malang berbasis web.

BAB V IMPLEMENTASI

Bab ini menjelaskan tentang implementasi dari pengembangan sistem informasi rekam medis di praktik mandiri pedis care malang berbasis web berdasarkan perancangan.

BAB VI PENGUJIAN

Bab ini menjelaskan tentang teknik pengujian yang akan dilakukan serta analisis hasil pengujian yang dilakukan pada sistem.

BAB VII KESIMPULAN DAN SARAN

Dalam bab ini merupakan bagian akhir dari penelitian yang mengungkapkan tentang kesimpulan yang diperoleh perancangan sistem serta saran-saran yang dapat diberikan dari hasil penelitian tersebut.

BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepastakan membahas tentang kajian pustaka serta dasar teori yang dijadikan penunjang dalam penelitian. Penelitian terdahulu yang akan digunakan sebagai pembandingan terkait dengan sistem informasi klinik praktik mandiri akan dibahas pada kajian pustaka. Dasar teori akan dijabarkan dalam bahasan di bawah ini pada Tabel 2.1.

2.1 Kajian Pustaka

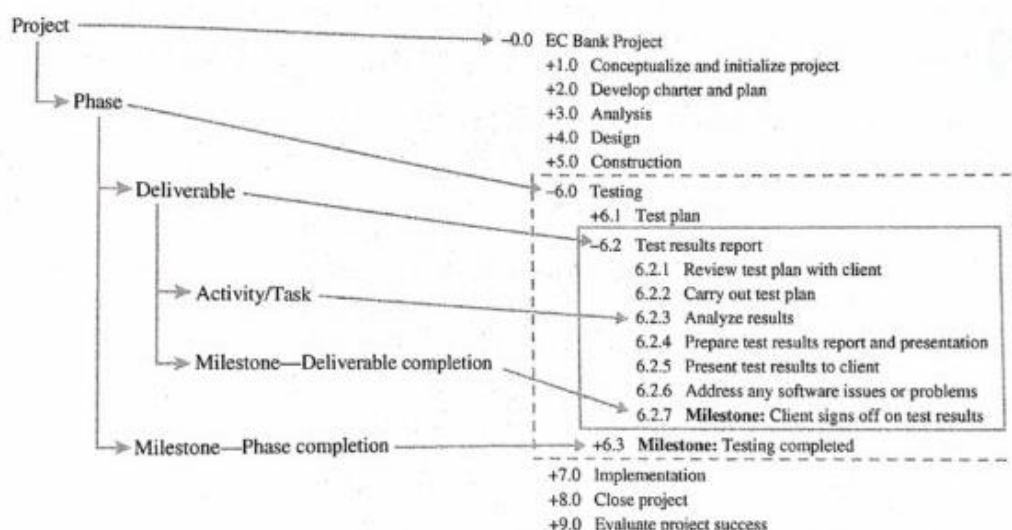
Tabel 2.1 Contoh notasi *Use Case Diagram*

No	Judul	Penulis dan Tahun Penelitian	Perbedaan	
			Penulis Sebelumnya	Penulis Sekarang
1	Sistem Informasi Rekam Medis Pada Klinik Paramitha Medika	Afrizal Havid, 2017.	Penelitian dilakukan berfokus pada objek rekam medis penyakit umum di rumah sakit.	Penelitian dilakukan berfokus pada objek rekam medis perawatan luka di praktik mandiri.
2	<i>Health Information System (HIS) Concept and Technology</i>	Mohd. Nabil Almunawar, Muhammad Anshari, 2012.	Tujuan penelitian untuk meringkas secara singkat informasi kesehatan masa lalu dan saat ini Pengamatan diambil berdasarkan konsep dan terminologi yang berkaitan dengan kesehatan bidang sistem informasi, sejarah beberapa generasi, dan diakhiri dengan tren perkembangan terkini dari alat dan teknologi dalam menciptakan dan mengelola HIS.	Tujuan penelitian untuk memperoleh rekayasa kebutuhan, perancangan, implementasi dan pengujian pada sistem yang akan dikembangkan.
3	Pembangunan Sistem Informasi	Nur Aini, 2019.	Pengujian sistem menggunakan blackbox testing dan User	Pengujian sistem menggunakan pengujian unit,

Perpustakaan Berbasis Web menggunakan Metode <i>Rapid Application Development</i> (RAD) (Studi pada : SMK Negeri 11 Malang)	Acceptance Testing (UAT)	pengujian integrasi, pengujian validasi dan pengujian <i>compatibility</i>
--	--------------------------	--

2.2 Work Breakdown Structure (WBS)

Work Breakdown Structure (WBS) adalah daftar kegiatan atau target dari ruang lingkup suatu proyek yang terorganisir dan biasa dibuat dengan menggunakan *project management tools*. Menurut (Satzinger, et al., 2012) ada dua pendekatan umum untuk membuat WBS, yaitu berdasarkan tujuan proyek atau berdasarkan timeline proyek. Pendekatan pertama dilakukan dengan mengidentifikasi seluruh tujuan yang harus diselesaikan sesuai dengan iterasi yang telah dibuat. Kemudian WBS mengidentifikasi setiap tugas yang diperlukan untuk membuat setiap tujuan. Sedangkan pendekatan yang kedua, setiap tugas dikerjakan sesuai dengan urutan timeline dari aktifitas yang diperlukan untuk mencapai tujuan akhir. Pada Gambar 2.1 adalah contoh gambar dari *Work Package* dan *Work Breakdown Structure* (WBS) (Marchewka, 2015):



Gambar 2.1 Work Breakdown Structure (WBS)

Sumber: Marchewka (2015)

Cara membuat sebuah *Work Breakdown Structure*(WBS) dapat dilihat dibawah ini.

1. *The WBS should support the project's MOV*

WBS harus mencakup tugas atau kegiatan yang diizinkan untuk tujuan proyek yang dilaksanakan.

2. *The WBS should be deliverable oriented*

Fokus dari proyek harus menghasilkan sesuatu, bukan hanya menyelesaikan sebuah kegiatan spesifik tertentu.

3. *The level of detail should support planning and control*

WBS memberikan sebuah jembatan antara ruang lingkup proyek dan rencana proyek, yaitu jadwal dan anggaran.

4. *Developing the WBS should involve the people who will be doing the work.*

Untuk memastikan bahwa WBS telah sesuai dengan tingkat kerincian yang diinginkan adalah dengan memastikan orang-orang yang memiliki pekerjaan tersebut telah terlibat dalam pengerjaan proyek itu.

2.3 Pengertian Sistem Informasi

Menurut O'Brian dikutip oleh Yakub (2012:17), definisi sistem informasi yaitu kombinasi antara orang-orang, *hardware*, *software*, dan jaringan komunikasi didalam organisasi yang bertujuan untuk mengumpulkan informasi ataupun merubah dan menyebarkannya. Perangkat lunak dapat dianggap berkualitas apabila dapat memenuhi kebutuhan pelanggan (*customer*), dan sesuai dengan kebutuhan pelanggan (*customer*). Setelah dikembangkannya perangkat lunak tersebut, masih diperlukannya sosialisasi perangkat lunak dengan diadakan pelatihan-pelatihan secara bertahap, dikarenakan untuk merubah kebiasaan di dalam sebuah lingkungan menjadi sesuatu yang baru tidaklah gampang, dan para pengguna perangkat lunak tersebut memiliki karakter yang berbeda-beda (A.S and Shalanudin, 2015).

2.3.1 Komponen Sistem Informasi

Menurut Yakub (2012: 20) Sistem informasi adalah susunan yang terdiri dari beberapa komponen-komponen. Komponen-komponen tersebut adalah:

1. Blok Masukan (*Input Block*), Input memiliki data yang masuk ke dalam sistem informasi, dan juga metode untuk menangkap data yang masuk tersebut.
2. Blok Model (*Model Block*), merupakan kombinasi dari prosedur, logika, dan model matematik yang memanipulasi data *input* dan data yang tersimpan pada basis data.
3. Blok Keluaran (*Output Block*), yaitu keluaran berupa informasi yang berkualitas dan dokumentasi untuk pengguna sistem dan semua tingkatan manajemen.
4. Blok Teknologi (*Technology Block*), digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dari sistem secara keseluruhan. Terdiri dari teknis (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok Basis Data (*Database Block*), yaitu kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan pada *hardware* komputer dan untuk memanipulasinya menggunakan *software*.

2.4 Sistem Informasi Kesehatan

Pelayanan rekam medis merupakan kegiatan pelayanan *professional* yang berorientasi pada kebutuhan informasi kesehatan dan mempunyai dasar dari ilmu dan teknologi, sehingga dapat digunakan untuk pemberi layanan kesehatan termasuk admin dan pihak manajemen pada instansi kesehatan. Pada era global saat ini setiap tenaga kesehatan, termasuk perekam medis dan informasi kesehatan dituntut untuk lebih *professional* pada pelayanan kesehatan seperti diagnosa dan terapi kepada klien, sehingga kualitas maupun keterjangkauan akan pelayanan kesehatan tersebut dapat meningkat (Susanto and Sugiharto, 2017).

Penggunaan sistem informasi tersebut terdiri dari tiga tahap berupa tahapan pemasukan data, kemudian pemrosesan data, dan terakhir pengeluaran informasi (Hatta, 2012).

2.5 World Wide Web (WWW)

Aplikasi yang menggunakan *protocol* HTTP (*Hyper Text Transfer Protocol*) perangkat lunak dan menggunakan *browser* sebagai perangkat lunak, yang didalamnya berisi dokumen multimedia yang berupa teks, gambar, suara, animasi dan video disebut Web.

Web adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan *protocol* HTTP (*Hyper Text Transfer Protocol*) dan untuk mengakses menggunakan perangkat lunak yang disebut *browser* (Arief, 2011). *Website* semakin berkembang di era industri 4.0. *Website* terdiri dari halaman (*page*) dan juga beberapa halaman atau disebut *homepage*. Posisi *homepage* terletak dibagian teratas yang di bawahnya terdiri dari halaman-halaman yang saling terkait. Halaman di bawah *homepage* berisi suatu *hyperlink* yang merujuk ke halaman lain yang disebut *child page* (Gregorius, 2000:30).

Terdapat 2 komponen utama dalam pengembangan *website* *Front-end* dan *Back-end*. *Front-end* yaitu bagian yang berinteraksi langsung dengan *user*, sedangkan *back-end* bagian yang mengolah data. Kedua komponen tersebut merupakan komponen utama yang perlu diperhatikan.

2.6 Rekam Medis

Suatu berkas yang di dalamnya berisi catatan terkait identitas, hasil pemeriksaan, pengobatan, tindakan yang harus dilakukan serta pelayanan apa saja yang telah diberikan kepada klien disebut dengan Rekam Medis (Pemenkes no 269, 2008). Rekam medis berfungsi untuk merekam informasi data sosial klien yang bertujuan untuk proses kesembuhan dan kesehatan klien. Dokumen tersebut dapat dipakai kapanpun saat dibutuhkan kembali.

2.7 Systems Development Life Cycle (SDLC)

Systems Development Life Cycle (SDLC) menurut Turban, Rainer dan Potter (2010, p690), yaitu kerangka kerja terstruktur untuk digunakan pada proyek sistem informasi yang terdiri dari berbagai proses untuk pengembangan sistem informasi, dimana di dalamnya terdapat beberapa metode yang digunakan yaitu *V model*, *Waterfall*, *prototyping*, *spiral*, *formal method*, *extreme programming*, dan *rapid application development (RAD)*. Metode yang digunakan pada pengembangan perangkat lunak ini adalah *Rapid Application Development* atau disingkat dengan RAD.

2.7.1 Rapid Application Development (RAD)

Rapid Application Development (RAD) merupakan model dari proses pengembangan perangkat lunak yang bersifat *incremental* dan dapat digunakan untuk waktu pengerjaan dalam jangka pendek. Jangka waktu yang telah ditentukan ini disebut dengan *timebox*. Hasil dari perangkat lunak ini berkembang selama proses pengerjaan pembangunan *Rapid Application Development (RAD)* dimana didasarkan pada umpan balik dari klien (Gunadarma, 2015). *Rapid Application Development (RAD)* merupakan gabungan antara teknik terstruktur dengan teknik *prototyping* dan teknik pengembangan *joint application*. Sehingga pengembangan menggunakan model ini sangat sesuai dengan jangka waktu yang relatif lebih cepat (Kendal, 2010).



Gambar 2.2 Siklus RAD

Sumber: Kendall (2010)

Beberapa kelemahan dari model *Rapid Application Development (RAD)*:

1. RAD memerlukan SDM yang besar guna pembentukan tim-tim dalam pengembangan komponen di dalamnya, sehingga untuk pembuatan sistem perangkat lunak dengan skala besar kurang maksimal.
2. Jika dalam proses pengembangan perangkat lunak dengan model RAD tidak ada keputusan atau persetujuan secara cepat, maka proyek ini akan gagal dikarenakan tidak kepastian dalam mendefinisikan kebutuhan pelanggan (*customer*) atau *user*.

3. Apabila sistem perangkat lunak yang akan dibuat tidak bisa dimodulkan, maka model ini tidak dapat digunakan karena terlalu banyak campur tangan antar tim.
4. Model ini tidak cocok digunakan untuk sistem perangkat lunak dengan resiko teknis yang tinggi, sebagai contoh penggunaan teknologi baru yang masih belum dikuasai dan belum banyak dikenal oleh pengembang.

Model *Rapid Application Development* sesuai dengan proyek dengan kriteria sebagai berikut:

1. Mempunyai anggota tim yang berpengalaman dalam pengembangan perangkat lunak yang sejenis.
2. Pengembang memiliki komponen-komponen sistem yang dapat digunakan dalam proyek tersebut.

Berdasarkan konsep model RAD ini, maka muncullah model-model yang berdasar atau memodifikasi model RAD salah satunya adalah pengembangan perangkat lunak “Tangkas” atau disebut dengan *Agile Software* yang berfokus kepada interaksi antar anggota tim dan pelanggan, sehingga pengembangan berjalan sangat cepat apabila terdapat perubahan yang terjadi. Beberapa contoh pengembangan perangkat lunak “tangkas” adalah *scrum* dan pemrograman ekstrim.

1. Pengembangan *scrum* (semua tim).

Peranan setiap tim dalam proyek telah ditentukan, dan bekerja secara *overlapping* atau tumpang tindih sesuai dengan kebutuhan sumber daya yang tersedia pada proyek perangkat lunak tersebut agar kecepatan semakin meningkat dan semakin fleksibel.

2. Pengembangan pemrograman ekstrim (*Extreme Programming*).

Ciri khas dari model ini berupa komunikasi intens yang dilakukan setiap hari atau disetiap kejadian kurang jelas yang ditemukan oleh tim. Kemudian antara tim pengembang, pelanggan, *user*, maupun sesama *programmer* terdapat komunikasi yang dilakukan secara langsung guna mendukung adanya umpan balik (*feedback*) dari setiap anggota tim.

2.8 Pemodelan

Pemodelan merupakan gambaran dari realita yang disederhanakan dan direpresentasikan ke dalam bentuk pemetaan menggunakan aturan tertentu. Secara umum, pemodelan dipakai untuk melakukan perencanaan suatu hal yang bertujuan untuk meminimalisasi kegagalan maupun resiko yang mungkin terjadi. Pada proses pengembangan perangkat lunak, pemodelan ditujukan untuk memudahkan langkah selanjutnya dari pengembangan sebuah sistem sehingga dapat menjadi lebih terencana.

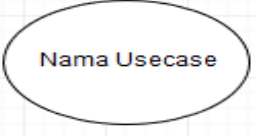
2.8.1 UML

Unified Modelling Language atau UML merupakan bahasa standar yang digunakan dalam pembangunan rancangan perangkat lunak dan untuk menggambarkan melalui rancangan dan membangun rancangan tersebut, dokumen artefak dari *software intensive system* (Booch, 2005:7). Tujuannya adalah agar dapat memberikan bahasa pemodelan visual kepada pengguna, memberikan model yang siap digunakan sehingga terjadi pertukaran model secara mudah serta dapat menghasilkan bahasa pemodelan yang dimengerti oleh manusia maupun mesin. UML terdiri dari 13 diagram dan terbagi menjadi 3 kategori. Ada beberapa jenis UML yang sering digunakan dalam pemograman berorientasi objek yaitu *class diagram*, *use case diagram* dan *sequence diagram*.

2.8.1.1 Usecase Diagram

Use case diagram digunakan untuk memberikan sebuah gambaran dari interaksi yang terjadi antara pengguna dengan sistem dalam *Unified Modeling Language* atau sering disebut dengan UML (Shelly dan Rosenblatt, 2012:p151). Tujuannya adalah membuat sebuah model bisnis proses berdasarkan sudut pandang pengguna sistem. Pada *Use case Diagram* penggunaan use case tidak akan dijelaskan secara detail, namun hanya gambaran hubungan antara *use case*, aktor dan sistem yang digambarkan secara singkat. Pada *Usecase Diagram* digunakan untuk mengetahui fungsi-fungsi apa saja yang terdapat pada sistem yang akan dibangun. Syarat penamaan pada use case adalah nama didefinisikan secara sederhana mungkin dan mudah untuk dipahami (A.S and Shalanudin, 2015) seperti yang terlihat pada Tabel 2.2.

Tabel 2.2 Contoh notasi Use Case Diagram

No	Nama Simbol	Deskripsi
1	<i>Use case</i> 	Merupakan fungsionalitas yang memiliki bentuk sistem sebagai unit-unit yang saling bertukar pesan
2	Aktor 	Merupakan representasi aktor dalam aplikasi

3	Asosiasi	Merupakan representasi komunikasi antar aktor atau <i>usecase</i>
4	Ektensi/ <i>extend</i> 	Merupakan relasi <i>use case</i> yang dapat berdiri sendiri tanpa bergantung pada <i>use case</i> lain
5	Generalisasi	Merupakan hubungan generalisasi (umum-khusus)
6	<i>Include</i> 	Merupakan relasi <i>use case</i> yang fungsinya dapat berjalan ketika <i>basecasenya</i> dijalankan duluan

Sumber: A.S and Shalanudin (2015)

2.8.1.2 Usecase Scenario

Usecase scenario adalah sebuah *usecase* yang berisikan skenario yang terjadi pada sebuah *use case diagram*. Tabel 2.3 di bawah merupakan contoh dari *use case scenario*.

Tabel 2.3 Contoh Notasi Use Case Scenario

<i>Flow of events</i> yaitu aliran-aliran <i>event</i>	
<i>Objective</i>	Merupakan tujuan yang akan dicapai
<i>Actor</i>	Aktor yang bersangkutan
<i>Pre-Condition</i>	Berisi kondisi-kondisi sebelumnya
<i>Main flow</i>	Berisi aliran utama
<i>Alternative flow</i>	Berisi aliran-aliran <i>alternative</i>

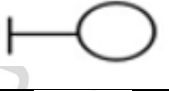
<i>Post-Condition</i>	Berisi kondisi-kondisi setelahnya
-----------------------	-----------------------------------

Sumber: A.S and Shalanudin (2015)

2.8.1.3 Sequence Diagram

Sequence diagram merupakan diagram interaksi yang bertujuan untuk menjelaskan eksekusi sebuah skenario dan interaksi yang terjadi antar objek pada jangka waktu tertentu (Booch, et al.2007). *Sequence Diagram* digunakan sebagai penggambaran scenario berupa rangkaian langkah-langkah yang dilakukan sebagai sebuah respon untuk menghasilkan output tertentu. Untuk penjelasan mengenai gambar, nama dan keterangan, dapat dilihat pada Tabel 2.4.

Tabel 2.4 Contoh Notasi Sequence Diagram

No	Gambar	Nama	Keterangan
1		Aktor	Menggambarkan orang yang akan melakukan interaksi dengan sistem
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Merupakan pemisah antara sistem dengan kondisi luar lingkungan sistem
4		<i>Control Class</i>	Merupakan penghubung antara model dan view dalam proses pengembangan
5		<i>A focus of control & a life line</i>	Tempat memulai dan berakhirnya sebuah pesan
6		<i>Message</i>	Simbol pengiriman pesan

Sumber: Booch (2007)

2.8.1.4 Class Diagram

Class diagram merupakan sekumpulan objek yang terdiri dari struktur umum, *behavior* umum, relasi umum, dan kata yang umum (*semantic*). Setiap *class* dapat ditentukan menggunakan cara memeriksa objek-objek yang terdapat pada *sequence diagram* dan *collaboration diagram*. Suatu *class* digambarkan sebagai sebuah bujur sangkar yang dibagi menjadi tiga bagian ruangan, yang terdiri dari nama *class*, atribut beserta tipenya, dan *method* beserta tipenya. (Whitten.L.Jeffery.et.al,2004). Untuk penjelasan dari setiap simbol dan keterangan yang terdapat dalam *class diagram* dapat dilihat pada Tabel 2.5.

Tabel 2.5 Contoh Notasi Class Diagram

Gambar	Nama	Keterangan
--------	------	------------

	Asosiasi	Relasi antar klas
	Asosiasi berarah	Relasi antar klas yaitu klas yang digunakan oleh klas yang lain
	Generalisasi	Relasi antar klas (umum-khusus)
	Dependency	Relasi antar klas dengan arti kebergantungan antar klas
	Agregasi	Relasi antar klas (semua bagian)

Sumber: Booch (2007)

2.9 Model View Controller (MVC)

Model View Controller (MVC) merupakan metode yang digunakan dalam pembangunan aplikasi menggunakan cara membagi antara data (*model*) dari tampilan sistem (*view*) dan bagaimana cara untuk memprosesnya (*controller*). Pada MVC akan memisahkan pengembangan aplikasi berdasarkan dari komponen pembangun utama dari suatu aplikasi, yaitu seperti memanipulasi data, tampilan atau *user interface*, dan bagian yang menjadi *control* pada suatu aplikasi.

1. *Model*, merupakan bagian yang berhubungan secara langsung dengan *database* untuk manipulasi data, menangani validasi pada bagian *controller*, dan tidak bisa berhubungan langsung dengan *view*.
2. *View*, merupakan bagian yang mengatur *presentation logic*. *View* memiliki fungsi untuk menerima dan merepresentasikan data kepada pengguna. Pada aplikasi web, *view* biasanya berupa *template* HTML yang diatur oleh *controller*.
3. *Controller*, merupakan bagian yang mengatur hubungan antara *model* dengan *view*. *Controller* memiliki fungsi untuk menerima *request* dan data dari pengguna, lalu menentukan proses apa yang akan dilakukan oleh aplikasi.

2.10 Pengujian Perangkat Lunak

Menurut IEEE 610.12, "Rekayasa perangkat lunak adalah sebuah studi dan aplikasi dari sebuah pendekatan kuantitatif, disiplin, dan sistematis kepada pengembangan, operasi dan pemeliharaan perangkat lunak yang ke semuanya itu merupakan aplikasi rekayasa yang berkaitan dengan perangkat lunak". Maka dari itu diperlukan suatu pengujian berupa *software testing* yang mempunyai definisi sebagai proses dalam pengembangan perangkat lunak yang bertujuan untuk mengidentifikasi kualitas dari perangkat lunak tersebut. Prosedurnya melalui proses penemuan kesalahan dari perangkat lunak, kemudian dilakukan pencatatan dari hasilnya, setelah itu evaluasi dari setiap aspek dan fitur-fitur yang terdapat di dalamnya.

Evaluasi dilakukan terhadap konfigurasi perangkat lunak termasuk di dalamnya spesifikasi kebutuhan, deskripsi rancangan, dan program yang dihasilkan. Berikutnya dilakukan perbandingan antara hasil dari evaluasi tersebut dengan hasil uji awal yang diharapkan. Bila dalam perbandingan tersebut terdapat kesenjangan antara hasil dari evaluasi dengan uji yang diharapkan, maka perlu dilakukan pengujian ulang setelah dilakukan perbaikan terhadap kesalahan yang ditemukan tadi. Pengujian perangkat lunak dikatakan berhasil apabila hasil dari pengujian memiliki nilai probabilitas yang tinggi dalam menemukan kesalahan yang belum pernah ditemukan sebelumnya. Ada dua teknik yang bisa dilakukan untuk pengujian perangkat lunak, contohnya adalah teknik *black box testing* dan *white box testing*.

2.10.1 White Box Testing

White box testing atau disebut juga dengan *glass box testing* menggunakan struktur kontrol, dimana struktur tersebut termasuk dalam *component level design* guna menghasilkan suatu *test cases*. *Test cases* yang dihasilkan dengan menggunakan metode *white box testing* adalah sebagai berikut:

1. Seluruh *independent path* yang terdapat dalam suatu modul harus dilakukan paling tidak satu kali dan harus terjamin.
2. Percobaan semua *logical decision* dan pemberian hasil nilai *true* atau *false* yang sesuai berdasarkan kondisi tertentu.
3. Percobaan untuk semua *loops* yang disesuaikan dengan batasan yang ada.
4. Percobaan pada struktur data Internal sebagai bukti bahwa struktur tersebut valid.

White box testing digunakan untuk melakukan pengujian terhadap *logic* yang ada dalam program secara internal. Untuk dapat melakukan pengujian ini, penguji harus mengetahui terlebih dahulu proses kerja produk secara internal. Kemudian, pengujian dapat dilakukan untuk mengetahui apakah seluruh produk yang dibuat saling berhubungan. Pengujian prosedur dalam *white box testing* dilakukan secara detail termasuk pada jalur-jalur logis (*logical path*) yang dilalui perangkat lunak dan kolaborasi antar komponen akan diujikan melalui pengulangan atau *loops* kondisi (*conditions*) tertentu. Teknik yang digunakan dalam *white box testing* adalah *basis path testing*.

Pada skripsi menggunakan *basis path testing*. Dengan menggunakan metode *basis path*, hasil yang diharapkan berupa pengukuran kompleksitas logis (*logical complexity*) dari rancangan secara prosedural. Selanjutnya pengukuran tersebut dapat dijadikan acuan untuk penentuan dasar dari kumpulan jalur yang akan dieksekusi. *Test cases* yang dihasilkan dapat menjamin bahwa setiap statement yang terdapat dalam program paling tidak dieksekusi satu kali selama pengujian dilakukan.

2.10.2 Black Box Testing

Black box testing, atau *behavioural testing*, digunakan sebagai pendeteksi kesesuaian fungsi tertentu dari suatu produk dengan tujuan awal dari rancangan produk tersebut. Pengujian ini dilaksanakan untuk penemuan *error* saat skrip telah dieksekusi, sehingga pengujian ini dikhususkan untuk kebutuhan perangkat lunak. Tujuan dari adanya metode *Black box testing* dapat dituliskan dibawah ini (Pressman, 2010):

1. Fungsi salah atau fungsi yang tidak lengkap,
2. *Interface* yang *error*,
3. *Error* saat akses basis data eksternal atau akses data struktur,
4. *Error* pada *behavior* atau *performance*, dan
5. *Error* pada inisialisasi dan *termination*.

Pressman (2010) menjabarkan bahwa *black box testing* merupakan pendekatan yang bersifat sebagai pelengkap teknik *white box testing* dengan penemuan *error* yang berbeda dari teknik *white box testing*. *Black box testing* dilaksanakan pada langkah terakhir dikarenakan sifat pengujiannya yang tidak bergantung pada struktur kontrol.

Melainkan berfokus pada domain informasi, *black box testing* dilaksanakan dengan mendesain dan pembuatan kumpulan *test case* yang nantinya akan menghasilkan penemuan *error* sebanyak mungkin pada level validasi perangkat lunak (*software validation level*) dengan waktu yang minim. Pada *test case* yang telah dibuat harus bisa dilakukan uji coba terhadap fungsinya. Langkah berikutnya adalah validasi pada operasional sepenuhnya pada saat yang bersamaan dengan pencarian *error* berikutnya yang mungkin terjadi. Pengujian dilakukan pada *interface* dengan cara pemberian kondisi *input* yang dijalankan secara menyeluruh sesuai dengan kebutuhan dari fungsi program tersebut.

2.10.3 Compatibility Testing

Compatibility testing dideskripsikan sebagai pengujian yang berguna untuk pemeriksaan kemampuan perangkat lunak tersebut saat dijalankan pada *hardware*, sistem operasi, aplikasi, ataupun lingkungan jaringan lainnya. Fungsi dari *compatibility testing* adalah menentukan set lingkungan tersebut dapat dijalankan dengan perangkat lunak yang dikembangkan. Aspek kompatibilitasnya dikatakan baik apabila perangkat lunak tersebut dapat dijalankan pada berbagai macam jenis perangkat yang berbeda.

2.11 Teknologi Pengembangan Sistem

Suatu teknologi yang digunakan sebagai acuan pengembangan sistem dalam perangkat lunak komputer.

2.11.1 Framework CodeIgniter

CodeIgniter merupakan sebuah *framework* PHP yang bersifat *open source* dengan penggunaan konsep MVC (*Model, View, Controller*) yang diterapkan pada

aplikasi berbasis web. Tujuannya adalah untuk mempermudah pengembang dalam pembuatan aplikasi *website* secara cepat dan mudah. *CodeIgniter* adalah proyek *open source* yang dikembangkan oleh masyarakat, dengan beberapa tempat bagi anggota komunitas untuk berkumpul dan berukar ide (CodeIgneter, 2019).

2.11.2 HTML

HTML (*Hyper Text Markup Language*) merupakan *symbol/tag* yang ditulis didalam sebuah *file*, berfungsi sebagai penampil halaman pada web *browser*. Setiap *file* harus diawali dan diakhiri dengan *tag* HTML seperti *b*, *i*, *u* dan lainnya. Tetapi ada beberapa *tag* yang tidak diakhiri dengan tanda (Astamal, 2006).

Beberapa fungsi dari HTML:

1. Membuat halaman web, setiap halaman web pasti dibuat dengan menggunakan HTML.
2. Untuk dasar dalam sebuah *website*. Jika *website* tidak memiliki HTML sebagai pondasi, maka kita tidak bisa mengimplementasikan bahasa lainnya seperti bahasa untuk mendesain *website* (CSS), bahasa untuk penambahan perilaku *website* (*Javascript*), dan bahasa pemrograman server *website* (PHP).
3. Sebagai penanda teks di halaman web, contohnya ketika kita memberikan tanda pada sebuah teks menjadi bergaris bawah maka ditambahkan menggunakan *tag* html `<u>`.
4. Sebagai penanda bagian atau elemen pada halaman web. *Website* memiliki beberapa bagian yaitu *navigasi*, *header*, *main* dan *footer* yang dapat kita berikan tanda pada setiap elemen tersebut dengan menggunakan HTML.
5. Berguna untuk penampilan informasi yang berupa bentuk tabel.
6. Sebagai penambahan objek seperti video, audio ataupun gambar dalam halaman web.

2.11.3 CSS

Cascading Style Sheet (CSS) adalah bahasa pemrograman web yang mengendalikan komponen dalam web sehingga menjadi lebih terstruktur dan seragam (Wahyu Sya'ban, 2010). Oleh karena itu CSS dapat digunakan sebagai alat untuk merubah ukuran gambar, serta warna bagian tubuh pada teks, warna tabel, warna dan ukuran border, warna *hyperlink*, warna *mouse over*, spasi antar *paragraph*, spasi antar teks, *margin* kiri, *margin* kanan, *margin* atas dan bawah, serta parameter lainnya.

2.11.4 JavaScript

JavaScript adalah bahasa *scripting* yang digunakan pada hampir seluruh *brower* yang populer seperti *Mozilla Firefox*, *Opera*, *Netscape*. Kode dari JavaScript

dapat digunakan pada halaman web yang memakai *tag* SCRIPT. *JavaScript* merespon perintah *user* secara cepat sehingga halaman web menjadi lebih responsif (Sunyoto, 2007:17).

2.11.5 Xampp

XAMPP singkatan dari X (empat sistem operasi apapun), Apache, MariaDB, PHP, dan Perl adalah server *localhost* yang berdiri sendiri. Program ini merupakan web server yang mudah digunakan dan dapat melayani tampilan halaman web yang bersifat dinamis, tersedia dalam bentuk GNU *General public license* dan bebas.

Pada umumnya, bagian Xampp yang digunakan adalah:

1. Htdocs, sebagai folder tempat peletakan berkas-berkas yang akan dijalankan (PHP, HTML, dan *script* lainnya).
2. PHPMyAdmin berfungsi sebagai temat untuk mengelola basis data MySQL yang terdapat pada komputer. Langkahnya adalah dengan memasukkan alamat <http://localhost/phpmyadmin> pada *browser* sehingga muncul halaman PHPMyAdmin.

Kontrol Panel, berfungsi sebagai pengelolaan layanan (*service*) XAMPP contohnya memulai layanan ataupun menghentikan layanan.

2.11.6 MySQL

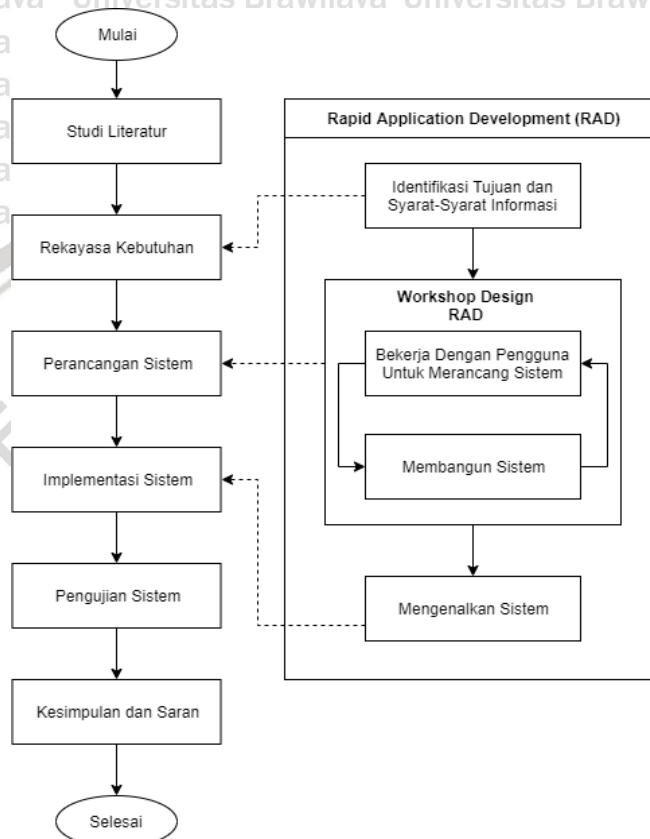
MySQL merupakan *software* sistem manajemen basis data yang sangat sering digunakan *programmer* web sebagai pembangunan aplikasi web dimana sumbernya berupa basis data dan dapat dikelola sesuai tujuan aplikasi web tersebut (Ariasari, 2010).

Keuntungan dari penggunaan MySQL adalah:

1. MySQL adalah *Database Management System* (DBMS).
2. *Database* yang dapat diakses oleh individu sesuai keperluannya secara gratis dan tidak perlu untuk melakukan pembelian lisensi.
3. MySQL adalah *database client*.
4. *Database* yang dapat menyimpan data yang berukuran sangat besar hingga mencapai ukuran *gigabyte*.
5. Dalam pembuatan dan *update* tabel, MySQL dapat dijalankan dengan baik dan cepat.

BAB 3 METODOLOGI PENELITIAN

Pada tahap ini dijelaskan terkait metodologi yang dipakai untuk mengerjakan penelitian “Pengembangan Sistem informasi Perawatan Luka Di Praktik Mandiri Pedis Care Malang Berbasis Web”. Adapun tahapan-tahapan berdasarkan metode *Rapid Application Development* (RAD) seperti yang ditampilkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metode Penelitian

3.1 Studi Literatur

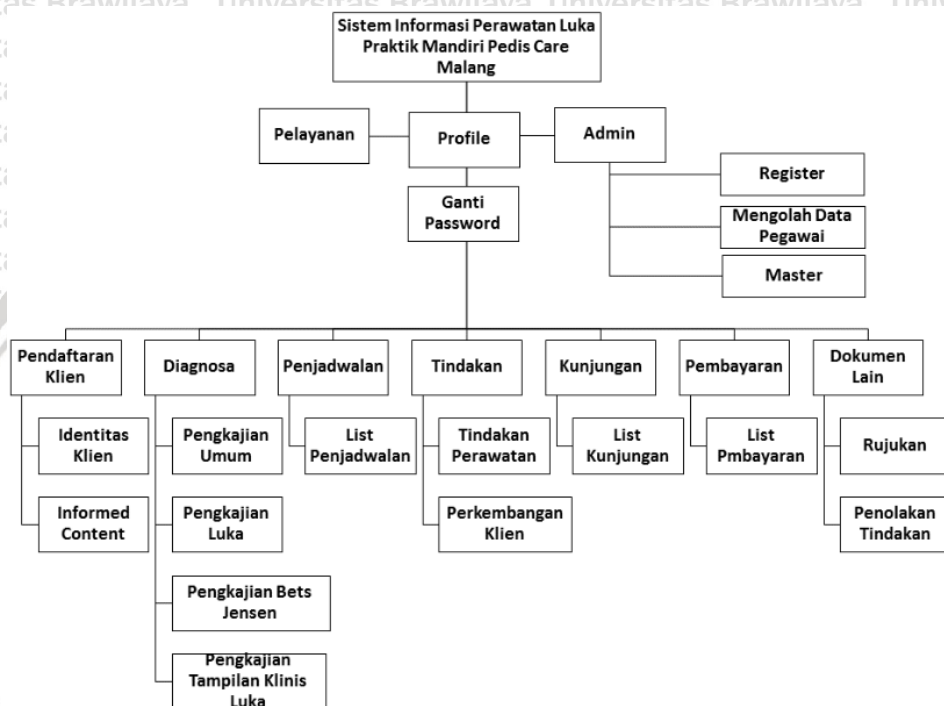
Studi literatur merupakan penjelasan tentang beberapa paper yang terkait dengan penelitian ini. Teori pendukung yang diperoleh dari jurnal, buku, dan beberapa penelitian sebelumnya yang terkait dengan penulisan penelitian yang akan di lakukan.

3.2 Rekayasa Kebutuhan

Rekayasa kebutuhan pada penelitian ini berisi proses elisitasi kebutuhan, yang dimana nanti akan menjadi fungsionalitas sistem, deskripsi sistem, identifikasi aktor. Selain itu terdapat permodelan kebutuhan di dalamnya yaitu terdapat *use case diagram* dan *use case scenario*.

3.3 Perancangan Sistem

Perancangan pada penelitian ini akan menjelaskan mengenai gambaran dari sistem yang akan dibuat dari perancangan arsitektur, perancangan data, perancangan komponen dan perancangan antarmuka. Dan untuk merancang sistem dengan pengguna, penulis menggunakan design perancangan berupa *Work Breakdown Structure* (WBS) untuk menentukan modul-modul utama yang nantinya akan diimplementasikan ke pembuatan sistem seperti yang terlihat pada Gambar 3.2.



Gambar 3.2 Work Breakdown Structure (WBS) Sistem Informasi Perawatan Luka Praktik mandiri Pedis Care Malang Berbasis Web.

Pada Gambar 3.2 yang merupakan gambaran rancangan WBS untuk fitur aplikasi sistem informasi perawatan luka praktik mandiri Pedis Care Malang. Berdasarkan *requirement* dipetakan aplikasi dengan fitur dan kegunaannya. Pada sistem informasi perawatan luka praktik mandiri pedis care malang terdapat 12 modul, yaitu pendaftaran klien, diagnosa, tindakan, penjadwalan, kunjungan, pembayaran, dan dokumen lain seperti penolakan tindakan dan rujukan, *profile*, dan ganti *password*. Tabel 3.1 di bawah ini merupakan penjelasan modul dari perancangan *Work Breakdown Structure* yang sudah dibuat.

Tabel 3.1 Penjelasan Modul (WBS) Sistem Informasi Praktik Mandiri Pedis Care Malang Berbasis Web

No	Modul	Sub Modul	Penjelasan
1	Pendaftaran Klien	Identitas klien	Fitur untuk mengelola identitas klien baru yang belum terdaftar.

		<i>Informed content</i>	Fitur untuk mengelola data perjanjian antara klien atau keluarga dengan tempat praktik tempat dimana klien bersedia dirawat.
2	Diagnosa	Pengkajian umum	Fitur untuk mengelola data klien mengenai riwayat penyakit sebelumnya yang pernah diderita.
		Pengkajian luka	Fitur untuk mengelola riwayat penyakit klien, datanya meliputi tipe penyakitnya, faktor penghambat penyembuhan dan alat yang diperlukan untuk menangani penyakit klien yang bersangkutan.
		Pengkajian betes jensen	Fitur untuk mengelola riwayat data penyakit klien, datanya meliputi gambaran luka klien seperti tipe luka, kedalaman luka.
		Tampilan klinis luka	Fitur untuk mengelola riwayat tampilan luka klien seperti warna lukanya, lebar lukanya dan bisa untuk mengupload kondisi luka klien pada saat itu.
3	Penjadwalan		Fitur untuk melihat semua penjadwalan klien yang akan berobat.
4	Tindakan	Tindakan perawatan	Fitur untuk mengelola data tindakan yang dilakukan pada klien yang berobat.
		Perkembangan klien	Fitur untuk mengelola data perkembangan klien ketika berobat.
5	kunjungan		Fitur untuk mengelola maksud klien berkunjung datang ke tempat praktik.
6	Pembayaran		Fitur untuk mengelola data administrasi pembayaran berobat klien.
7	Dokumen lain	Rujukan	Fitur untuk mengelola data rujukan klien.
		Penolakan tindakan	Fitur untuk mengelola permintaan penolakan tindakan dari klien atau

			keluarga untuk tidak meneruskan perawatan di tempat praktik mandiri pedis care malang.
8	Ganti <i>password</i>		Fitur untuk mengelola penggantian <i>password</i> dengan merubah <i>password</i> lama ke <i>password</i> yang baru.
9	Profile		Fitur untuk melihat riwayat detail profil akun dan bisa dilakukan perubahan juga didalamnya.
10	Register		Fitur untuk mendaftar akun baru supaya <i>user</i> dapat masuk ke sistem.
11	Pegawai		Terdiri dari beberapa fitur, diantaranya terdapat fitur hapus akun pegawai, rubah hak akses pegawai, <i>reset password</i> dan tambah sanksi.
12	Master		Fitur untuk mengelola data seperti <i>list status</i> , <i>list Pendidikan</i> , <i>list agama</i> , <i>list bidang kerja</i> , <i>list dressing</i> , <i>list rumah sakit</i> .

3.4 Implementasi Sistem

Implementasi sistem perawatan luka menggunakan *framework CodeIgneter*. Pada implementasi berisi spesifikasi sistem, implementasi data, implementasi kode program dan implementasi antarmuka. Pada spesifikasi sistem berisi spesifikasi perangkat keras, spesifikasi perangkat lunak dan sistem operasi yang digunakan dalam proses implementasi pengembangan sistem. Implementasi data yang digunakan diambil dari model *Entity Relationship Diagram*. Implementasi kode program dibuat berdasarkan perancangan komponen yang sudah dibuat. Dan implementasi antarmuka dibuat berdasarkan kerangka perancangan antarmuka sebelumnya.

3.5 Pengujian Sistem

Pengujian yang digunakan untuk menguji sistem aplikasi perawatan luka ini yaitu pengujian unit, pengujian integrasi, pengujian validasi dan pengujian *compatibility*. Pada tahap pengujian unit didapatkan melalui metode *white box testing* dengan teknik *basis path testing*, begitu pula dengan pengujian integrasi yang menggunakan metode *white box testing* dengan teknik *basis path testing*. Kemudian terdapat pengujian validasi dilakukan dengan metode *black box testing* yang nantinya akan didapatkan berdasarkan kasus uji dari kebutuhan fungsional.

Sedangkan untuk pengujian *compatibility* adalah pengujian yang dilakukan untuk menyeleksi apakah sistemnya mampu untuk dijalankan disemua *browser*.

3.6 Kesimpulan dan Saran

Tahap selanjutnya yaitu kesimpulan, untuk pengambilan kesimpulan didasarkan pada proses rekayasa kebutuhan sampai ke tahap pengujian sistem. Setelah mengambil kesimpulan, dituliskan saran untuk menjadi bahan pertimbangan penelitian selanjutnya.



BAB 4 ANALISIS KEBUTUHAN

4.1 Deskripsi Sistem

Sistem dibangun dengan *platform website* dan *frameworknya* menggunakan *CodeIgniter*. Memiliki fungsi untuk mengelola data medis klien, penjadwalan berobat beserta administrasi pembayarannya. Sistem dibangun menggunakan *database MySQL*. Aktor yang terlibat dalam sistem yaitu divisi pelayanan sebagai pengelola data terhadap klien yang berobat dan admin untuk mengelola data apabila terdapat pembaruan atau penghapusan data pada sistem yang menyangkut data pegawai, registrasi dan pengolahan data master.

4.2 Deskripsi Istilah dan Singkatan

Daftar istilah dan singkatan dari pengembangan sistem informasi perawatan luka di praktik mandiri pedis care malang berbasis web, dapat dijelaskan pada Tabel 4.1.

Tabel 4.1 Daftar Istilah dan Singkatan

SIPCAM	Merupakan singkatan dari Sistem Informasi Pedis Care Malang.
<i>Database</i>	Sebuah sitem yang memudahkan untuk menyimpan, mengelola dan mengambil data yang besar.
<i>Use Case Diagram</i>	Diagram yang menjelaskan peran aktor terhadap skema yang dikembangkan.
<i>Use Case Scenario</i>	Deskripsi Scenario Dan <i>Use Case Diagram</i>
<i>Sequence Diagram</i>	Diagram yang menjelaskan alur kerja sistem pada suatu method. Terdiri dari <i>Boundary</i> , <i>Entity</i> , <i>Controller</i> .
<i>Class Diagram</i>	Diagram yang mengilustrasikan <i>class</i> yang ada di sistem.
Pengujian <i>White Box</i>	Pengujian sistem secara detail dengan menguji kode program pada sistem yang telah dirancang.
Pengujian <i>Black Box</i>	Pengujian yang bertujuan untuk menguji keseluruhan fungsionalitas dari sistem.

4.3 Aturan Penomoran

Tahap selanjutnya yaitu aturan penomoran, yang nantinya digunakan untuk menentukan kebutuhan pengembangan sistem informasi perawatan luka di praktik mandiri pedis care malang berbasis web seperti yang dijelaskan Tabel 4.2 di bawah ini.

Tabel 4.2 Daftar Aturan Penomoran

Auran Penomoran	Keterangan
-----------------	------------

SIPCAM	Singkatan dari aplikasi Sistem Informasi Pedis Care Malang
F	Representasi Kebutuhan Fungsional
NF	Representasi kebutuhan non fungsional
NOMOR	Nomor urut kebutuhan

Sebagai contoh : SIPCAMP_F_1 representasi kebutuhan fungsional sistem informasi Pedis Care Malang dengan nomor urut 1.

4.4 Identifikasi Aktor

Berdasarkan identifikasi permasalahan yang telah diuraikan di latar belakang, aktor yang terlibat dalam sistem yaitu divisi pelayanan, admin dan *user*. Aktor-aktor tersebut didapat dari hasil observasi dan wawancara dengan *staff* pekerja di tempat praktik mandiri pedis care malang. Observasi dilakukan dengan cara terjun ke tempat yang bersangkutan dan mengamati secara langsung aktivitas yang terdapat di tempat praktek. Pada Tabel 4.3 terdapat identifikasi aktor berdasarkan penjelasan perannya.

Tabel 4.3 Identifikasi Aktor Sistem

Aktor	Deskripsi
Divisi pelayanan	Merupakan aktor yang dapat mengelola data medis mulai dari pendaftaran klien, diagnosa klien dan tindakan yang diberikan serta pengolahan dokumen-dokumen lain seperti dokumen rujukan dan penolakan tindakan dan penjadwalan berobat klien.
Admin	Merupakan aktor yang berperan dalam mengelola data pegawai, data masteryang meliputi <i>list</i> status pegawai, <i>list</i> status Pendidikan, <i>list</i> agama, <i>list</i> bidang kerja, <i>list</i> rumah sakit, <i>list</i> dressing.
User	Aktor yang belum pernah masuk ke dalam sistem (hanya dapat melihat halaman <i>login</i>).

4.5 Daftar Kebutuhan

Berdasarkan daftar kebutuhan yang terdapat di analisis kebutuhan pengembangan sistem informasi perawatan luka praktik mandiri pedis care malang berbasis web terdapat 2 kebutuhan yaitu kebutuhan fungsional dan non fungsional. Dengan adanya daftar kebutuhan berfungsi untuk memudahkan menyusun perencanaan pengembangan sistem.

4.5.1 Kebutuhan Fungsional

Sistem informasi perawatan luka praktik mandiri pedis care malang ini memiliki kebutuhan utama yang mendeskripsikan layanan apa saja yang dapat disediakan. Pada Tabel 4.4 dijelaskan deskripsi kebutuhan fungsional yang sudah teridentifikasi. Ada 59 definisi kebutuhan yang teridentifikasi pada tahap analisis.

Tabel 4.4 Definisi Kebutuhan Fungsional

No	Kode Kebutuhan	Deskripsi Kebutuhan	Nama Use Case
1	SIPCAM_F_1	Sistem mempunyai fungsi <i>login</i> untuk mengizinkan pengguna mengakses sistem sebagai pemilik dan pegawai.	<i>Login</i>
2	SIPCAM_F_2	Sistem terdapat fungsi <i>logout</i> untuk keluar dari sistem.	<i>Logout</i>
3	SIPCAM_F_3	Sistem terdapat fungsi daftar untuk calon pegawai baru yang belum mempunyai akun.	Daftar
4	SIPCAM_F_4	Sistem dapat merubah <i>password</i> .	Ubah <i>password</i> .
5	SIPCAM_F_5	Sistem harus dapat mengedit halaman biodata diri.	Mengedit biodata diri.
6	SIPCAM_F_6	Sistem dapat menambah data master sistem.	Menambah data master.
7	SIPCAM_F_7	Sistem dapat menghapus data master sistem.	Menghapus data master.
8	SIPCAM_F_8	Sistem dapat merubah hak akses pengguna sistem.	Merubah hak akses sistem.
9	SIPCAM_F_9	Sistem dapat memvalidasi registrasi dari pegawai baru.	Memvalidasi registrasi.
10	SIPCAM_F_10	Sistem dapat menghapus data pegawai.	Menghapus data pengguna sistem.
11	SIPCAM_F_11	Sistem harus dapat menambah data identitas klien baru pada sub menu identitas klien.	Menambah identitas klien
12	SIPCAM_F_12	Sistem harus dapat menghapus data identitas klien tertentu pada sub menu identitas klien.	Menghapus identitas klien

13	SIPCAM_F_13	Sistem harus dapat mengedit data identitas klien pada sub menu identitas klien.	Mengedit identitas klien
14	SIPCAM_F_14	Sistem harus dapat mencetak <i>form</i> identitas klien pada sub menu identitas klien.	Mencetak identitas klien.
15	SIPCAM_F_15	Sistem harus dapat menambah data <i>informed content</i> pada sub menu <i>informed content</i> .	Menambah <i>informed content</i> .
16	SIPCAM_F_16	Sistem harus dapat menghapus data identitas klien tertentu pada sub menu <i>informed content</i> .	Menghapus <i>informed content</i> .
17	SIPCAM_F_17	Sistem harus dapat mengedit data <i>informed content</i> pada sub menu <i>informed content</i> .	Mengedit <i>informed content</i> .
18	SIPCAM_F_18	Sistem harus dapat mencetak <i>form informed content</i> pada sub menu <i>informed content</i> .	Mencetak <i>informed content</i> .
19	SIPCAM_F_19	Sistem harus dapat menambah data pengkajian umum pada sub menu pengkajian umum.	Menambah data pengkajian umum.
20	SIPCAM_F_20	Sistem harus dapat menghapus data pengkajian umum tertentu pada sub menu pengkajian umum.	Menghapus data pengkajian umum.
21	SIPCAM_F_21	Sistem harus dapat mengedit data pengkajian umum pada sub menu pengkajian umum.	Mengedit <i>form</i> pengkajian umum.
22	SIPCAM_F_22	Sistem harus dapat mencetak <i>form</i> pengkajian umum pada sub menu pengkajian umum.	Mencetak pengkajian umum.
23	SIPCAM_F_23	Sistem harus dapat menambah data pengkajian luka pada sub menu pengkajian luka.	Menambah data pengkajian luka.
24	SIPCAM_F_24	Sistem harus dapat menghapus data pengkajian luka tertentu pada sub menu pengkajian luka.	Menghapus data pengkajian luka.
25	SIPCAM_F_25	Sistem harus dapat mengedit data pengkajian luka pada sub menu pengkajian luka.	Mengedit <i>form</i> pengkajian luka.

26	SIPCAM_F_26	Sistem harus dapat mencetak <i>form</i> pengkajian luka pada sub menu pengkajian luka.	Mencetak pengkajian luka.
27	SIPCAM_F_27	Sistem harus dapat menambah data pengkajian betes jensen pada sub menu pengkajian betes Jensen.	Menambah data pengkajian betes Jensen.
28	SIPCAM_F_28	Sistem harus dapat menghapus data pengkajian betes jensen tertentu pada sub menu pengkajian betes Jensen.	Menghapus data pengkajian betes Jensen.
29	SIPCAM_F_29	Sistem harus dapat mengedit data pengkajian betes jensen pada sub menu pengkajian betes Jensen.	Mengedit form pengkajian betes Jensen.
30	SIPCAM_F_30	Sistem harus dapat mencetak <i>form</i> pengkajian betes jensen pada sub menu pengkajian betes Jensen.	Mencetak pengkajian betes Jensen.
31	SIPCAM_F_31	Sistem harus dapat menambah data tampilan klinis luka pada sub menu tampilan klinis luka.	Menambah data tampilan klinis luka.
32	SIPCAM_F_32	Sistem harus dapat menghapus data tampilan klinis luka tertentu pada sub menu tampilan klinis luka.	Menghapus data tampilan klinis luka.
33	SIPCAM_F_33	Sistem harus dapat mengedit data tampilan klinis luka pada sub menu tampilan klinis luka.	Mengedit data tampilan klinis luka.
34	SIPCAM_F_34	Sistem harus dapat mencetak <i>form</i> tampilan klinis luka pada sub menu tampilan klinis luka.	Mencetak tampilan klinis luka.
35	SIPCAM_F_35	Sistem harus dapat menambah data tindakan yang pernah dilakukan pada sub menu tindakan.	Menambah tindakan.
36	SIPCAM_F_36	Sistem harus dapat menghapus data tindakan tertentu pada sub menu tindakan.	Menghapus tindakan.

37	SIPCAM_F_37	Sistem harus dapat mengedit data tindakan tertentu pada sub menu tindakan.	Mengedit tindakan.
38	SIPCAM_F_38	Sistem harus dapat mencetak form tindakan pada sub menu tindakan.	Mencetak tindakan
39	SIPCAM_F_39	Sistem harus dapat menambah data perkembangan klien pada sub menu perkembangan klien.	Menambah data perkembangan klien
40	SIPCAM_F_40	Sistem harus dapat menghapus data perkembangan klien tertentu pada sub menu perkembangan klien.	Menghapus data perkembangan klien.
41	SIPCAM_F_41	Sistem harus dapat mengedit data perkembangan klien tertentu pada sub menu perkembangan klien.	Mengedit form perkembangan klien.
42	SIPCAM_F_42	Sistem harus dapat mencetak form perkembangan klien pada sub menu perkembangan klien.	Mencetak form perkembangan klien.
43	SIPCAM_F_43	Sistem harus dapat menambah data kunjungan pada menu kunjungan.	Menambah kunjungan.
44	SIPCAM_F_44	Sistem harus dapat menghapus data kunjungan tertentu pada menu kunjungan.	Menghapus kunjungan.
45	SIPCAM_F_45	Sistem harus dapat mengedit data kunjungan tertentu pada menu kunjungan.	Mengedit kunjungan.
46	SIPCAM_F_46	Sistem harus dapat mencetak form kunjungan pada menu kunjungan.	Mencetak kunjungan.
47	SIPCAM_F_47	Sistem dapat menambah data pembayaran pada menu pembayaran.	Menambah pembayaran
48	SIPCAM_F_48	Sistem harus menghapus data pembayaran tertentu pada menu pembayaran.	Menghapus pembayaran

49	SIPCAM_F_49	Sistem harus dapat mencetak <i>form</i> pembayaran pada menu pembayaran.	Mencetak <i>form</i> pembayaran.
50	SIPCAM_F_50	Sistem harus dapat menambah data penolakan tindakan pada sub menu penolakan tindakan.	Menambah data penolakan tindakan.
51	SIPCAM_F_51	Sistem harus dapat menghapus data penolakan tindakan pada sub menu penolakan tindakan.	Menghapus data penolakan tindakan.
52	SIPCAM_F_52	Sistem harus dapat mengedit <i>form</i> penolakan tindakan tertentu pada sub menu penolakan tindakan.	Mengedit <i>form</i> penolakan tindakan.
53	SIPCAM_F_53	Sistem harus dapat mencetak <i>form</i> penolakan tindakan pada sub menu penolakan tindakan.	Mencetak <i>form</i> penolakan tindakan.
54	SIPCAM_F_54	Sistem harus dapat menambah data rujukan pada sub menu rujukan.	Menambah data rujukan.
55	SIPCAM_F_55	Sistem harus dapat menghapus data rujukan pada sub menu rujukan.	Menghapus data rujukan.
56	SIPCAM_F_56	Sistem harus dapat mengedit <i>form</i> rujukan tertentu pada sub menu rujukan.	Mengedit <i>form</i> rujukan.
57	SIPCAM_F_57	Sistem harus dapat mencetak <i>form</i> rujukan pada sub menu rujukan.	Mencetak <i>form</i> rujukan.
58	SIPCAM_F_58	Sistem harus dapat menambah data penjadwalan klien.	Tambah penjadwalan
59	SIPCAM_F_59	Sistem harus dapat menghapus data penjadwalan klien.	Hapus Penjadwalan

4.5.2 Spesifikasi Kebutuhan

Pada Tabel 4.5 merupakan spesifikasi kebutuhan fungsional yang sudah teridentifikasi pada bagian analisis kebutuhan sebelumnya. Spesifikasi kebutuhan adalah pendetailan dari definisi kebutuhan.

Tabel 4.5 Spesifikasi Kebutuhan

No	Kode Definisi Kebutuhan	Kode Spesifikasi Kebutuhan	Spesifikasi Kebutuhan	Aktor
1	SIPCAM_F_1	SIPCAM_S_1	Sistem harus menyediakan <i>form</i> yang berisi <i>field</i> email dan <i>password</i> .	Divisi Pelayanan, Admin, User
		SIPCAM_S_2	<i>Field</i> email atau <i>password</i> tidak boleh kosong.	
		SIPCAM_S_3	Input pada <i>field</i> email dibatasi minimal 10 karakter dan maksimal 30 karakter.	
		SIPCAM_S_4	Input pada <i>field password</i> dibatasi minimal 8 maksimal 30 karakter.	
		SIPCAM_S_5	Sistem menyediakan <i>button</i> sign in.	
2	SIPCAM_F_2	SIPCAM_S_6	Sistem harus menyediakan <i>button</i> logout untuk keluar dari sistem.	Divisi Pelayanan, Admin
3	SIPCAM_F_3	SIPCAM_S_7	Sistem harus menyediakan <i>button</i> daftar untuk masuk ke <i>form</i> pendaftaran.	Admin
		SIPCAM_S_8	Sistem menyediakan <i>form</i> pendaftaran yang dimana didalamnya berisi data <i>username</i> , email, <i>password</i> , retype <i>password</i> , nama lengkap dan gelar, nama panggilan status pegawai, divisi, dasar	

			penerimaan, tanggal lahir, tanggal masuk.	
		SIPCAM_S_9	Sistem harus menyediakan <i>button</i> sign up untuk mendaftarkan data yang sudah di isi pada <i>form</i> pendaftaran.	
4	SIPCAM_F_4	SIPCAM_S_10	Sistem menyediakan <i>form</i> ubah <i>password</i> yang didalamnya terdapat field-field yang harus diisi diantaranya adalah field <i>password</i> lama, field <i>password</i> baru dan field retype new <i>password</i> .	Divisi Pelayanan, Admin
		SIPCAM_S_11	Sistem menyediakan <i>button</i> save untuk menyimpan <i>password</i> baru.	
5	SIPCAM_F_5	SIPCAM_S_12	Sistem menyediakan <i>button</i> edit profile	Divisi Pelayanan, Admin
		SIPCAM_S_13	Sistem menyediakan <i>form</i> profile data diri untuk pengguna dapat mengedit data.	
6	SIPCAM_F_6	SIPCAM_S_19	Sistem menyediakan <i>button</i> tambah untuk menambah data master.	Admin
		SIPCAM_S_20	Sistem menyediakan <i>form</i> untuk memasukkan data yang ingin ditambah.	
7	SIPCAM_F_7	SIPCAM_S_21	Sistem menyediakan <i>button</i> hapus untuk menghapus data.	Admin
8	SIPCAM_F_8	SIPCAM_S_22	Sistem menyediakan <i>button</i> aksi untuk	Admin

			mengubah hak akses pengguna.	
9	SIPCAM_F_9	SIPCAM_S_23	Sistem menyediakan <i>button</i> aksi untuk memvalidasi registrasi.	Admin
10	SIPCAM_F_10	SIPCAM_S_24	Sistem menyediakan <i>button</i> hapus untuk menghapus pengguna sistem.	Admin
11	SIPCAM_F_11	SIPCAM_S_27	Sistem menyediakan <i>form</i> identitas klien untuk mengisi data identitas klien.	Divisi pelayanan
		SIPCAM_S_28	Sistem menyediakan <i>button</i> submit untuk menyimpan data yang sudah dimasukkan.	
12	SIPCAM_F_12	SIPCAM_S_29	Sistem menyediakan <i>button</i> hapus untuk menghapus data identitas klien.	Divisi pelayanan
			Sistem menyediakan <i>form</i> pemberitahuan sebelum penghapusan dilakukan.	
13	SIPCAM_F_13	SIPCAM_S_30	Sistem menyediakan <i>button edit</i> .	Divisi pelayanan
		SIPCAM_S_31	Sistem menyediakan <i>form edit</i> untuk mengedit data identitas klien didalamnya.	
		SIPCAM_S_32	Sistem menyediakan <i>button save</i> .	
14	SIPCAM_F_14	SIPCAM_S_35	Sistem menyediakan <i>button</i> print untuk	

			mencetak data identitas <i>klien</i> .	Divisi pelayanan
		SIPCAM_S_36	Sistem menampilkan form data identitas <i>klien</i> yang dipilih untuk di cetak atau print.	
15	SIPCAM_F_15	SIPCAM_S_39	Sistem menyediakan <i>button</i> tambah untuk menambah data <i>informed content</i> .	Divisi pelayanan
		SIPCAM_S_40	Sistem menyediakan form baru untuk mengisi data <i>informed content</i> .	
			Sistem menyediakan <i>button</i> submit untuk menyimpan data yang sudah diinputkan ke <i>database</i> .	
16	SIPCAM_F_16	SIPCAM_S_41	Sistem menyediakan <i>button</i> hapus untuk menghapus data <i>informed content</i> .	Divisi pelayanan
17	SIPCAM_F_17	SIPCAM_S_42	Sistem menyediakan <i>button edit</i> untuk mengedit form <i>informed content</i> .	Divisi pelayanan
		SIPCAM_S_43	Sistem menyediakan form edit <i>informed content</i> untuk tempat mengedit data yang sudah tersedia didalamnya.	
		SIPCAM_S_44	Sistem menyediakan <i>button save</i> untuk menyimpan data.	
18	SIPCAM_F_18	SIPCAM_S_47	Sistem menyediakan <i>button print</i> untuk	

			mencetak data <i>informed content</i> .	Divisi pelayanan
		SIPCAM_S_48	Sistem menampilkan <i>form data informed content</i> yang dipilih untuk di cetak atau print.	
19	SIPCAM_F_19	SIPCAM_S_51	Sistem menyediakan <i>button</i> tambah untuk menambah data pengkajian umum.	Divisi pelayanan
		SIPCAM_S_52	Sistem menyediakan <i>form</i> baru untuk mengisi data pengkajian umum.	
			Sistem menyediakan <i>button</i> submit untuk menyimpan data yang sudah dimasukkan ke <i>database</i> .	
20	SIPCAM_F_20	SIPCAM_S_53	Sistem menyediakan <i>button</i> hapus untuk menghapus data pengkajian umum.	Divisi pelayanan
21	SIPCAM_F_21	SIPCAM_S_54	Sistem menyediakan <i>button edit</i> untuk mengedit <i>form</i> pengkajian umum.	Divisi pelayanan
		SIPCAM_S_55	Sistem menyediakan <i>form edit</i> untuk tempat mengedit data diagnosa pengkajian umum yang sudah tersedia didalamnya.	
		SIPCAM_S_56	Sistem menyediakan <i>button save</i> untuk menyimpan data.	

22	SIPCAM_F_22	SIPCAM_S_59	Sistem menyediakan <i>button</i> print untuk mencetak data pengkajian umum.	Divisi pelayanan
		SIPCAM_S_60	Sistem menampilkan <i>form</i> data pengkajian umum yang dipilih untuk di cetak atau print.	
23	SIPCAM_F_23	SIPCAM_S_63	Sistem menyediakan <i>button</i> tambah untuk menambah data pengkajian luka.	Divisi pelayanan
		SIPCAM_S_64	Sistem menyediakan <i>form</i> baru untuk mengisi data pengkajian luka.	
			Sistem menyediakan <i>button</i> submit untuk menyimpan data yang sudah dimasukkan ke <i>database</i> .	
24	SIPCAM_F_24	SIPCAM_S_65	Sistem menyediakan <i>button</i> hapus untuk menghapus data pengkajian luka.	Divisi pelayanan
25	SIPCAM_F_25	SIPCAM_S_66	Sistem menyediakan <i>button</i> edit untuk mengedit <i>form</i> pengkajian luka.	Divisi pelayanan
		SIPCAM_S_67	Sistem menyediakan <i>form</i> edit untuk tempat mengedit data diagnosa pengkajian luka yang sudah tersedia didalamnya.	
		SIPCAM_S_68	Sistem menyediakan <i>button</i> save untuk menyimpan data.	

26	SIPCAM_F_26	SIPCAM_S_71	Sistem menyediakan <i>button</i> print untuk mencetak data pengkajian luka.	Divisi pelayanan
		SIPCAM_S_72	Sistem menampilkan <i>form</i> data pengkajian luka yang dipilih untuk di cetak atau print.	
27	SIPCAM_F_27	SIPCAM_S_75	Sistem menyediakan <i>button</i> tambah untuk menambah data pengkajian betes Jensen.	Divisi pelayanan
		SIPCAM_S_76	Sistem menyediakan <i>form</i> baru untuk mengisi data pengkajian betes Jensen.	
			Sistem menyediakan <i>button</i> submit untuk menyimpan data yang sudah dimasukkan ke <i>database</i> .	
28	SIPCAM_F_28	SIPCAM_S_77	Sistem menyediakan <i>button</i> hapus untuk menghapus data pengkajian betes Jensen.	Divisi pelayanan
29	SIPCAM_F_29	SIPCAM_S_78	Sistem menyediakan <i>button edit</i> untuk mengedit <i>form</i> pengkajian betes Jensen.	Divisi pelayanan
		SIPCAM_S_79	Sistem menyediakan <i>form edit</i> untuk tempat mengedit data diagnosa pengkajian betes Jensen yang sudah tersedia didalamnya.	

		SIPCAM_S_80	Sistem menyediakan <i>button save</i> untuk menyimpan data.	
30	SIPCAM_F_30	SIPCAM_S_83	Sistem menyediakan <i>button print</i> untuk mencetak data pengkajian betes Jensen.	Divisi pelayanan
		SIPCAM_S_84	Sistem menampilkan <i>form</i> data pengkajian betes Jensen yang dipilih untuk di cetak atau print.	
31	SIPCAM_F_31	SIPCAM_S_87	Sistem menyediakan <i>button</i> tambah untuk menambah data tampilan klinis luka.	Divisi pelayanan
		SIPCAM_S_88	Sistem menyediakan <i>form</i> baru untuk mengisi data tampilan klinis luka.	
			Sistem menyediakan <i>button submit</i> untuk menyimpan data yang sudah dimasukkan ke <i>database</i> .	
32	SIPCAM_F_32	SIPCAM_S_89	Sistem menyediakan <i>button</i> hapus untuk menghapus data tampilan klinis luka.	Divisi pelayanan
33	SIPCAM_F_33	SIPCAM_S_90	Sistem menyediakan <i>button edit</i> untuk mengedit <i>form</i> tampilan klinis luka.	Divisi pelayanan
		SIPCAM_S_91	Sistem menyediakan <i>form edit</i> untuk tempat mengedit data tampilan klinis luka yang sudah tersedia didalamnya.	

		SIPCAM_S_92	Sistem menyediakan <i>button save</i> untuk menyimpan data.	
34	SIPCAM_F_34	SIPCAM_S_95	Sistem menyediakan <i>button print</i> untuk mencetak data tampilan klinis luka.	Divisi pelayanan
		SIPCAM_S_96	Sistem menampilkan <i>form</i> data tampilan klinis luka yang dipilih untuk di cetak atau print.	
35	SIPCAM_F_35	SIPCAM_S_99	Sistem menyediakan <i>button</i> tambah untuk menambah data tindakan.	Divisi pelayanan
		SIPCAM_S_100	Sistem menyediakan <i>form</i> baru untuk mengisi data tindakan.	
			Sistem menyediakan <i>button submit</i> untuk menyimpan data yang sudah dimasukkan ke <i>database</i> .	
36	SIPCAM_F_36	SIPCAM_S_101	Sistem menyediakan <i>button</i> hapus untuk menghapus data tindakan.	Divisi pelayanan
37	SIPCAM_F_37	SIPCAM_S_102	Sistem menyediakan <i>button edit</i> untuk mengedit <i>form</i> tindakan.	Divisi pelayanan
		SIPCAM_S_103	Sistem menyediakan <i>form edit</i> untuk tempat mengedit data tindakan yang sudah tersedia didalamnya.	

		SIPCAM_S_104	Sistem menyediakan <i>button save</i> untuk menyimpan data.	
38	SIPCAM_F_38	SIPCAM_S_107	Sistem menyediakan <i>button print</i> untuk mencetak data tindakan.	Divisi pelayanan
		SIPCAM_S_108	Sistem menampilkan <i>form</i> data tindakan yang dipilih untuk di cetak atau print.	
39	SIPCAM_F_39	SIPCAM_S_111	Sistem menyediakan <i>button</i> tambah untuk menambah data perkembangan <i>klien</i> .	Divisi pelayanan
		SIPCAM_S_112	Sistem menyediakan <i>form</i> baru untuk mengisi data perkembangan <i>klien</i> dan <i>button</i> submit.	
			Sistem menyediakan <i>button</i> submit untuk menyimpan data yang sudah dimasukkan ke <i>database</i> .	
40	SIPCAM_F_40	SIPCAM_S_113	Sistem menyediakan <i>button</i> hapus untuk menghapus data perkembangan <i>klien</i> .	Divisi pelayanan
41	SIPCAM_F_41	SIPCAM_S_114	Sistem menyediakan <i>button edit</i> untuk mengedit <i>form</i> perkembangan <i>klien</i> .	Divisi pelayanan
		SIPCAM_S_115	Sistem menyediakan <i>form edit</i> untuk tempat mengedit data perkembangan <i>klien</i> yang sudah tersedia didalamnya.	

		SIPCAM_S_116	Sistem menyediakan <i>button save</i> untuk menyimpan data.	
42	SIPCAM_F_42	SIPCAM_S_119	Sistem menyediakan <i>button print</i> untuk mencetak data perkembangan klien.	Divisi pelayanan
		SIPCAM_S_120	Sistem menampilkan <i>form</i> data perkembangan klien yang dipilih untuk di cetak atau print.	
43	SIPCAM_F_43	SIPCAM_S_123	Sistem menyediakan <i>button</i> tambah untuk menambah data kunjungan.	Divisi pelayanan
		SIPCAM_S_124	Sistem menyediakan <i>form</i> baru untuk mengisi data kunjungan.	
			Sistem menyediakan <i>button submit</i> untuk menyimpan data yang sudah dimasukkan ke <i>database</i> .	
44	SIPCAM_F_44	SIPCAM_S_125	Sistem menyediakan <i>button</i> hapus untuk menghapus data kunjungan.	Divisi pelayanan
45	SIPCAM_F_45	SIPCAM_S_126	Sistem menyediakan <i>button edit</i> untuk mengedit <i>form</i> kunjungan.	Divisi pelayanan
		SIPCAM_S_127	Sistem menyediakan <i>form edit</i> untuk tempat mengedit data kunjungan yang sudah tersedia didalamnya.	

		SIPCAM_S_128	Sistem menyediakan <i>button save</i> untuk menyimpan data.	
46	SIPCAM_F_46	SIPCAM_S_131	Sistem menyediakan <i>button print</i> untuk mencetak data kunjungan.	Divisi pelayanan
		SIPCAM_S_132	Sistem menampilkan <i>form</i> data kunjungan yang dipilih untuk di cetak atau print.	
47	SIPCAM_F_47	SIPCAM_S_135	Sistem menyediakan <i>button</i> tambah untuk menambah data administrasi pembayaran perawatan.	Divisi pelayanan
		SIPCAM_S_136	Sistem menyediakan <i>form</i> baru untuk mengisi data administrasi pembayaran	
			Sistem menyediakan <i>button submit</i> untuk menyimpan data yang sudah dimasukkan.	
48	SIPCAM_F_48	SIPCAM_S_137	Sistem menyediakan <i>button</i> hapus untuk menghapus data administrasi pembayaran.	Divisi pelayanan
49	SIPCAM_F_49	SIPCAM_S_143	Sistem menyediakan <i>button print</i> untuk mencetak data administrasi pembayaran.	Divisi pelayanan
		SIPCAM_S_144	Sistem menampilkan <i>form</i> data yang	

			dipilih untuk di cetak atau print.	
50	SIPCAM_F_50	SIPCAM_S_159	Sistem menyediakan <i>button</i> tambah untuk menambah data penolakan tindakan.	Divisi pelayanan
		SIPCAM_S_160	Sistem menyediakan <i>form</i> baru untuk mengisi data penolakan tindakan.	
		SIPCAM_S_160	Sistem menyediakan <i>button</i> submit untuk menyimpan data yang sudah dimasukkan ke <i>database</i> .	
51	SIPCAM_F_79	SIPCAM_S_161	Sistem menyediakan <i>button</i> hapus untuk menghapus data penolakan tindakan.	Divisi pelayanan
52	SIPCAM_F_52	SIPCAM_S_162	Sistem menyediakan <i>button edit</i> untuk mengedit <i>form</i> penolakan tindakan.	Divisi pelayanan
		SIPCAM_S_163	Sistem menyediakan <i>form edit</i> untuk tempat mengedit data penolakan tindakan yang sudah tersedia didalamnya.	
		SIPCAM_S_164	Sistem menyediakan <i>button save</i> untuk menyimpan data.	
53	SIPCAM_F_53	SIPCAM_S_167	Sistem menyediakan <i>button</i> print untuk mencetak data penolakan tindakan.	Divisi pelayanan
		SIPCAM_S_168	Sistem menampilkan <i>form</i> data penolakan tindakan yang dipilih	

			untuk di cetak atau print.	
54	SIPCAM_F_54	SIPCAM_S_171	Sistem menyediakan <i>button</i> tambah untuk menambah data rujukan.	Divisi pelayanan
		SIPCAM_S_172	Sistem menyediakan <i>form</i> baru untuk mengisi data rujukan.	
		SIPCAM_S_173	Sistem menyediakan <i>button</i> submit untuk menyimpan data yang sudah dimasukkan ke <i>database</i> .	
55	SIPCAM_F_55	SIPCAM_S_173	Sistem menyediakan <i>button</i> hapus untuk menghapus data rujukan.	Divisi pelayanan
56	SIPCAM_F_86	SIPCAM_S_174	Sistem menyediakan <i>button edit</i> untuk mengedit <i>form</i> rujukan.	Divisi pelayanan
		SIPCAM_S_175	Sistem menyediakan <i>form edit</i> untuk tempat mengedit data rujukan yang sudah tersedia didalamnya.	
		SIPCAM_S_176	Sistem menyediakan <i>button save</i> untuk menyimpan data.	
57	SIPCAM_F_57	SIPCAM_S_179	Sistem menyediakan <i>button</i> print untuk mencetak data rujukan.	Divisi pelayanan
		SIPCAM_S_180	Sistem menampilkan <i>form</i> data rujukan yang dipilih untuk di cetak atau print.	

58	SIPCAM_F_58	SIPCAM_S_180	Sistem menyediakan <i>button</i> tambah untuk menambah data penjadwalan.	Divisi pelayanan
		SIPCAM_S_180	Sistem menyediakan <i>form</i> baru untuk mengisi data jadwal berobat klien.	
		SIPCAM_S_180	Sistem menyediakan <i>button</i> submit untuk menyimpan data yang sudah dimasukkan ke <i>database</i> .	
59	SIPCAM_F_59	SIPCAM_S_180	Sistem menyediakan <i>button</i> hapus untuk menghapus data rujukan.	Divisi pelayanan

4.5.3 Kebutuhan Non Fungsional

Pada Tabel 4.6 dijelaskan definisi kebutuhan non fungsional dari sistem yang akan dikembangkan nantinya.

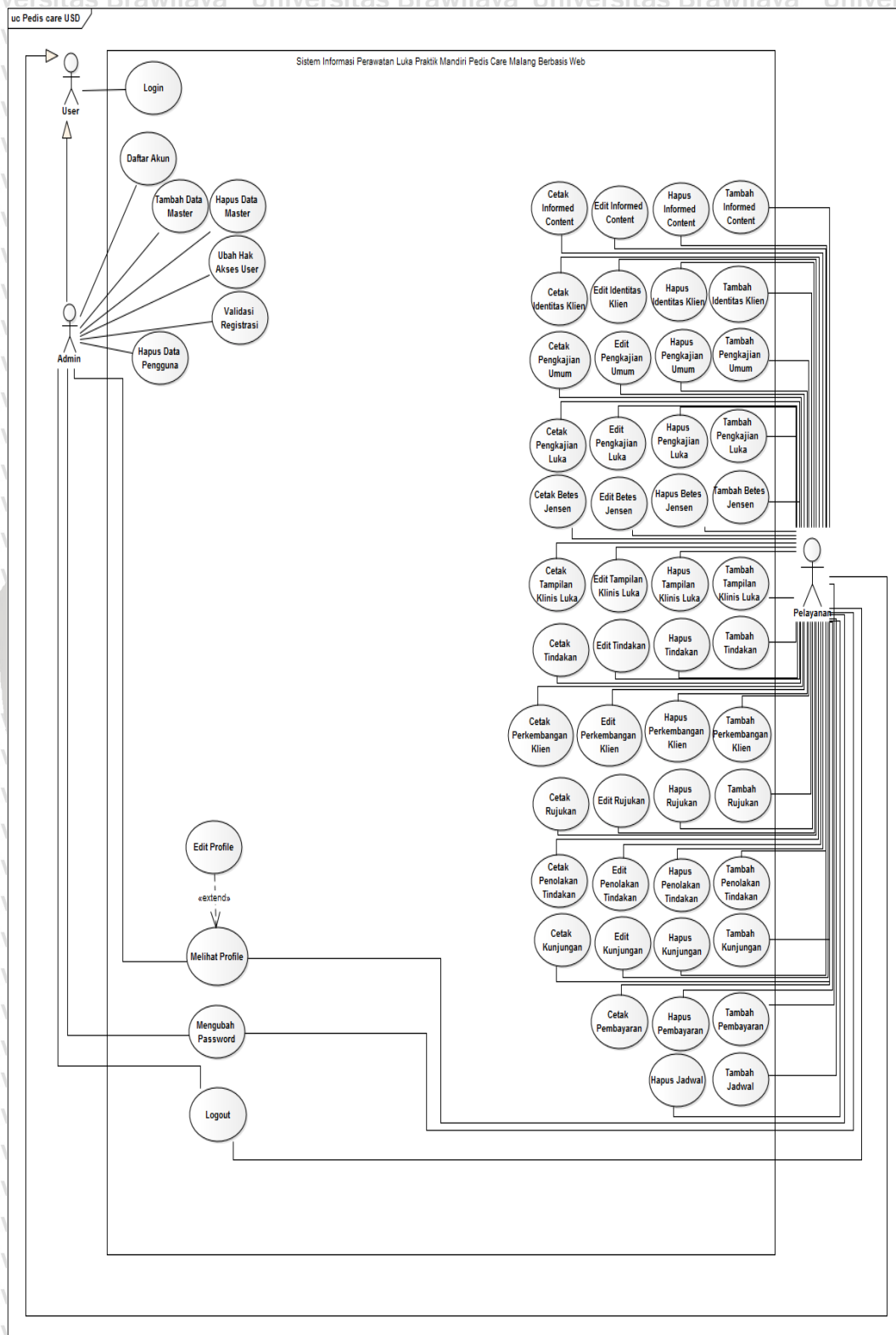
Tabel 4.6 Definisi Kebutuhan Non Fungsional

No	Kode	Parameter	Deskripsi
1	SIPCAM_NF_1	<i>Compatibility</i>	Sistem harus dapat dijalankan secara sempurna (tanpa kehilangan fungsionalitas) di berbagai web <i>browser</i> yaitu <i>Google Chrome, Firefox, Opera</i> .

4.6 Pemodelan Kebutuhan

Pemodelan kebutuhan pada langkah ini akan dijabarkan tentang uraian *use case diagram* dan *use case scenario* yang dimana berfungsi untuk mempermudah pengembang dalam mengerjakan perancangan sistem.

4.6.1 Pemodelan Use Case Diagram



Gambar 4.1 Use Case Diagram Sistem Informasi Perawatan Luka Praktik Mandiri Pedis Care Malang Berbasis Web

4.6.2 Pemodelan Use Case Scenario

Use case scenario digunakan untuk menjelaskan lebih rinci terhadap *use case* yang telah didefinisikan sebelumnya di atas. *Use case scenario* bertujuan untuk menjelaskan tentang pra kondisi, pasca kondisi dan alternative jika ada. penjelasan lebih rinci dapat dilihat pada Tabel 4.7 sampai dengan Tabel 4.65.

4.6.2.1 Use Case Scenario Login

Use case scenario pada fungsi *login* dituliskan pada Tabel 4.7 berikut:

Tabel 4.7 Use Case Scenario Login

Nama use case	Login
Objective	Aktor berhasil <i>login</i> untuk bisa menggunakan layanan-layanan yang tersedia di menu sistem berdasarkan hak aksesnya.
Aktor	Divisi pelayanan, admin, user.
Pre-condition	Halaman <i>login</i> telah ditampilkan
Mainflow	1. Aktor mengisi kolom <i>username</i>. 2. Aktor mengisi kolom <i>password</i>. 3. Aktor menekan <i>button login</i>.
Alternative Flow	1. Apabila terdapat salah satu kolom dari <i>username</i> dan <i>password</i> tidak diisi, maka sistem akan keluar notifikasi <i>please fill out this field</i> dalam sistem. 2. Jika memasukkan <i>username</i> atau <i>password</i> salah, maka sistem akan memunculkan peringatan <i>username</i> atau <i>password</i> salah.
Post-condition	Sistem berhasil masuk ke sistem dan menampilkan halaman <i>dashboard</i> .

4.6.2.2 Use Case Scenario Logout

Use case scenario pada fungsi *logout* dituliskan pada Tabel 4.8 berikut:

Tabel 4.8 Use Case Scenario Logout

Nama use case	Logout
Objective	Aktor dapat keluar dari akun untuk berhenti mengakses sistem.
Aktor	Divisi pelayanan, admin.
Pre-condition	Aktor berada pada halaman <i>dashboard</i> atau menu utama.
Mainflow	1. Aktor menekan <i>icon</i> foto atau nama pengguna yang berada dipojok kanan atas. 2. Aktor menekan <i>button sign out</i>.

<i>Alternative Flow</i>	-
<i>Post-condition</i>	Berhasil menjalankan fungsi <i>logout</i> dan keluar sistem

4.6.2.3 Use Case Scenario Daftar

Use case scenario pada fungsi daftar dituliskan pada Tabel 4.9 berikut:

Tabel 4.9 Use Case Scenario Daftar

Nama <i>use case</i>	Daftar
<i>Objective</i>	Berhasil mendaftarkan akun baru untuk pengguna yang belum terdaftar di sistem.
Aktor	Admin.
<i>Pre-condition</i>	Aktor berada di halaman <i>login</i> .
<i>Mainflow</i>	1. Aktor menekan <i>link</i> daftar di sini. 2. Aktor mengisi <i>field-field</i> yang ada dalam <i>form</i>. 3. Aktor menekan <i>button sign up</i>.
<i>Alternative Flow</i>	Jika <i>field-field</i> ada yang tidak diisi, <i>button sign up</i> otomatis non aktif.
<i>Post-condition</i>	Sistem berhasil dan menunjukkan notifikasi "Pendaftaran telah diajukan!" pada halaman daftar akun baru.

4.6.2.4 Use Case Scenario Mengubah Password

Use case scenario pada fungsi mengubah *password* dituliskan pada Tabel 4.10 berikut:

Tabel 4.10 Use Case Scenario Mengubah Password

Nama <i>use case</i>	Mengubah <i>password</i> .
<i>Objective</i>	Berhasil mengubah <i>password</i> lama dengan <i>password</i> baru.
Aktor	Divisi pelayanan, admin.
<i>Pre-condition</i>	Aktor dalam kondisi sudah <i>login</i> dan berada pada halaman menu ganti <i>password</i> .
<i>Mainflow</i>	1. Aktor mengisi <i>field password</i> lama. 2. Aktor mengisi <i>field new password</i>. 3. Aktor mengisi <i>field retype new password</i>. 4. Aktor menekan <i>button save</i>.
<i>Alternative Flow</i>	1. Sistem menampilkan pesan error "ubah <i>password</i> gagal" apabila data yang diisikan tidak sesuai. 2. Sistem menunjukkan notifikasi "<i>Please fill out this field</i>"

<i>Post-condition</i>	Berhasil mengganti <i>password</i> dan data <i>password</i> baru yang sudah dimasukkan akan tersimpan di <i>database</i> .
-----------------------	--

4.6.2.5 Use Case Scenario Mengedit Biodata Diri

Use case scenario pada fungsi mengedit biodata diri dituliskan pada Tabel 4.11 berikut:

Tabel 4.11 Use Case Scenario Mengedit Biodata Diri

Nama use case	Mengedit biodata diri.
<i>Objective</i>	Sistem dapat mengubah biodata diri pegawai sesuai <i>input</i> yang diberikan.
Aktor	Divisi pelayanan, divisi marketing, admin.
<i>Pre-condition</i>	Aktor dalam kondisi sudah <i>login</i> dan berada pada halaman menu <i>profile</i> .
<i>Mainflow</i>	1. Aktor menekan <i>button edit</i>. 2. Aktor memilih bagian kolom yang ingin diedit. 3. Aktor menekan <i>button submit</i>.
<i>Alternative Flow</i>	Sistem menunjukkan notifikasi peringatan apabila data yang diisikan tidak sesuai dengan <i>format</i> inputan.
<i>Post-condition</i>	Berhasil memperbarui biodata dan data baru yang sudah diinputkan akan tersimpan di <i>database</i> .

4.6.2.6 Use Case Scenario Menambah Data Master

Use case scenario pada fungsi menambah master dituliskan pada Tabel 4.12 berikut:

Tabel 4.12 Use Case Scenario Menambah Data Master

Nama use case	Menambah data master
<i>Objective</i>	Aktor dapat menambah data master untuk datanya masuk dalam pilihan <i>form</i> tertentu dalam pilihan <i>combo box</i> .
Aktor	Admin.
<i>Pre-condition</i>	Aktor telah <i>login</i> dan berada pada halaman <i>master</i> .
<i>Mainflow</i>	1. Aktor memilih salah satu bagian menu yang akan ditambah diantaranya terdapat menu status, pendidikan, agama, bidang kerja, <i>list</i> rumah sakit. 2. Aktor menekan <i>button</i> tambah data. 3. Aktor mengisi data yang ingin ditambah. 4. Aktor menekan <i>button</i> simpan.

<i>Alternative Flow</i>	-
<i>Post-condition</i>	Sistem berhasil menambah data master dan tersimpan di database.

4.6.2.7 Use Case Scenario Menghapus Data Master

Use case scenario pada fungsi menghapus data master dituliskan pada Tabel 4.13 berikut:

Tabel 4.13 Use Case Scenario Menghapus Data Master

<i>Nama use case</i>	Menghapus data master.
<i>Objective</i>	Sistem dapat menghapus data di master.
Aktor	Admin.
<i>Pre-condition</i>	Aktor dalam kondisi sudah <i>login</i> dan berada pada halaman master.
<i>Mainflow</i>	1. Aktor memilih salah satu bagian menu yang akan dihapus diantaranya terdapat menu status, Pendidikan, agama, bidang kerja, <i>list</i> rumah sakit. 2. Aktor menekan <i>button</i> “icon tempat sampah” untuk menghapus data.
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Sistem berhasil menghapus data master dan tersimpan di database.

4.6.2.8 Use Case Scenario Merubah Hak Akses Sistem

Use case scenario pada fungsi merubah hak akses sistem dituliskan pada Tabel 4.14 berikut:

Tabel 4.14 Use Case Scenario Merubah Hak Akses Sistem

<i>Nama use case</i>	Merubah hak akses sistem.
<i>Objective</i>	Sistem dapat merubah hak akses pengguna untuk penempatan divisi baru.
Aktor	Admin.
<i>Pre-condition</i>	Aktor dalam kondisi sudah <i>login</i> dan berada pada halaman pegawai.
<i>Mainflow</i>	1. Aktor menekan <i>icon</i> gembok. 2. Aktor memilih <i>button</i> akses untuk pengguna. 3. Aktor menekan <i>button</i> simpan.

<i>Alternative Flow</i>	-
<i>Post-condition</i>	Sistem berhasil merubah hak akses.

4.6.2.9 Use Case Scenario Memvalidasi Registrasi

Use case scenario pada fungsi memvalidasi registrasi dituliskan pada Tabel 4.15 berikut:

Tabel 4.15 Use Case Scenario Memvalidasi Registrasi

Nama use case	Memvalidasi registrasi.
<i>Objective</i>	Memvalidasi registrasi yang sudah diajukan ke dalam sistem.
Aktor	Admin.
<i>Pre-condition</i>	Aktor dalam kondisi sudah <i>login</i> dan berada pada <i>form</i> daftar karyawan baru.
<i>Mainflow</i>	1. Aktor menekan <i>button</i> aksi terima atau tidak. 2. Aktor menekan <i>button combo box</i> akses.
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Sistem berhasil memvalidasi registrasi pegawai baru.

4.6.2.10 Use Case Scenario Menghapus Data Pengguna Sistem

Use case scenario pada fungsi menghapus pengguna sistem dituliskan pada Tabel 4.16 berikut:

Tabel 4.16 Use Case Scenario Menghapus Data Pengguna Sistem

Nama use case	Menghapus pengguna sistem.
<i>Objective</i>	Admin dapat menghapus pengguna sistem supaya tidak dapat masuk dan berhenti mengakses sistem.
Aktor	Admin.
<i>Pre-condition</i>	Aktor dalam kondisi sudah <i>login</i> dan berada pada halaman pegawai.
<i>Mainflow</i>	1. Aktor menekan <i>button</i> "icon tempat sampah" 2. Aktor muncul notifikasi "apakah yakin ingin menghapus pegawai" 3. Aktor menekan <i>button</i> "ya"
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Sistem sukses melakukan penghapusan pengguna sistem.

4.6.2.11 Use Case Scenario Menambah Identitas Klien

Use case scenario pada fungsi menambah identitas klien dituliskan pada Tabel 4.17 berikut:

Tabel 4.17 Use Case Scenario Menambah Identitas Klien

Nama use case	Menambah identitas klien.
Objective	Divisi pelayanan dapat menambahkan data klien baru yang bersedia untuk dirawat di tempat praktik.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada pada halaman <i>dashboard</i> .
Mainflow	1. Aktor menekan menu pendaftaran klien. 2. Aktor memilih sub menu identitas klien. 3. Aktor mengisi kolom yang tersedia pada <i>form</i> identitas klien. 4. Aktor menekan <i>button</i> submit untuk menyimpan data klien baru ke <i>database</i>.
Alternative Flow	Sistem menampilkan peringatan " <i>Please fill out this field</i> " apabila ada <i>field</i> yang tidak terisi.
Post-condition	Sistem berhasil melakukan tambah data identitas klien dan tersimpan di <i>database</i> .

4.6.2.12 Use Case Scenario Menghapus Identitas Klien

Use case scenario pada fungsi menghapus identitas klien dituliskan pada Tabel 4.18 berikut:

Tabel 4.18 Use Case Scenario Menghapus Identitas Klien

Nama use case	Menghapus identitas klien.
Objective	Sistem dapat menghapus riwayat identitas klien.
Aktor	Pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor menekan menu pendaftaran klien. 2. Aktor menekan sub menu riwayat identitas klien. 3. Aktor menekan <i>button</i> "<i>icon</i> tempat sampah". 4. Jika muncul popup pilih <i>button</i> "<i>ya</i>".
Alternative Flow	
Post-condition	Sistem berhasil melakukan penghapusan data identitas klien.

4.6.2.13 Use Case Scenario Mengedit Identitas Klien

Use case scenario pada fungsi mengedit identitas klien dituliskan pada Tabel 4.19 berikut:

Tabel 4.19 Use Case Scenario Mengedit Identitas Klien

Nama use case	Mengedit identitas klien.
Objective	Sistem dapat mengedit <i>form</i> riwayat identitas klien.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor menekan menu pendaftaran klien. 2. Aktor menekan sub menu riwayat identitas klien. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Alternative Flow	-
Post-condition	Sistem berhasil melakukan <i>edit</i> data identitas klien dan menyimpannya dalam <i>database</i> .

4.6.2.14 Use Case Scenario Mencetak Identitas Klien

Use case scenario pada fungsi mencetak identitas klien dituliskan pada Tabel 4.20 berikut:

Tabel 4.20 Use Case Scenario Mencetak Identitas Klien

Nama use case	Mencetak identitas klien.
Objective	Sistem dapat mencetak <i>form</i> identitas klien pada <i>list</i> riwayat identitas klien.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> sistem.
Mainflow	1. Aktor menekan menu pendaftaran klien. 2. Aktor menekan sub menu riwayat identitas klien. 3. Aktor menekan <i>button print</i>. 4. Aktor menekan <i>button keyboard</i> ctrl + p 5. Aktor menekan <i>print</i>.
Alternative Flow	-
Post-condition	Sistem berhasil melakukan cetak <i>form</i> identitas klien.

4.6.2.15 Use Case Scenario Menambah Informed Content

Use case scenario pada fungsi menambah *informed content* dituliskan pada Tabel 4.21 berikut:

Tabel 4.21 Use Case Scenario Menambah Informed Content

Nama use case	Menambah <i>informed content</i> .
Objective	Sistem dapat menambahkan data <i>informed content</i> atau lembar persetujuan dirawat ditempat praktik pedis care.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> sistem.
Mainflow	1. Aktor menekan menu pendaftaran <i>klien</i>. 2. Aktor menekan sub menu <i>informed content</i>. 3. Aktor mengisi kolom yang tersedia pada <i>form informed content</i>. 4. Aktor menekan <i>button</i> submit untuk menyimpan data <i>informed content</i> ke <i>database</i>.
Alternative Flow	Sistem menunjukkan notifikasi " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Post-condition	Sistem berhasil menambah data <i>informed content</i> dan tersimpan di <i>database</i> .

4.6.2.16 Use Case Scenario Menghapus Informed Content

Use case scenario pada fungsi menghapus *informed content* dituliskan pada Tabel 4.22 berikut:

Tabel 4.22 Use Case Scenario Menghapus Informed Content

Nama use case	Menghapus <i>informed content</i> .
Objective	Sistem dapat menghapus riwayat <i>informed content</i> .
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>form</i> riwayat <i>informed content</i> .
Mainflow	1. Aktor menekan <i>button</i> "<i>icon tempat sampah</i>". 2. Jika muncul pop up pilih <i>button</i> "<i>ya</i>".
Alternative Flow	-
Post-condition	Sistem berhasil menghapus data <i>informed content</i> .

4.6.2.17 Use Case Scenario Mengedit Informed Content

Use case scenario pada fungsi mengedit *informed content* dituliskan pada Tabel 4.23 berikut:

Tabel 4.23 Use Case Scenario Mengedit Informed Content

Nama use case	Mengedit <i>informed content</i> .
Objective	Sistem dapat mengedit <i>form</i> riwayat <i>informed content</i> .
Aktor	Divisi pelayanan
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor menekan menu pendaftaran klien. 2. Aktor menekan sub menu riwayat <i>informed content</i>. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Alternative Flow	-
Post-condition	Sistem berhasil melakukan <i>edit informed content</i> dan menyimpannya dalam <i>database</i> .

4.6.2.18 Use Case Scenario Mencetak Informed Content

Use case scenario pada fungsi mencetak *informed content* dituliskan pada Tabel 4.24 berikut:

Tabel 4.24 Use Case Scenario Mencetak Informed Content

Nama use case	Mencetak <i>informed content</i> .
Objective	Dapat mencetak <i>form informed content</i> pada <i>list</i> riwayat <i>informed content</i> .
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor menekan menu pendaftaran klien. 2. Aktor menekan sub menu riwayat <i>informed content</i>. 3. Aktor menekan <i>button print</i>. 4. Aktor menekan <i>button keyboard</i> ctrl + p. 5. Aktor menekan <i>button print</i>.
Alternative Flow	-
Post-condition	Sistem berhasil mencetak <i>form informed content</i> .

4.6.2.19 Use Case Scenario Menambah Data Pengkajian Umum

Use case scenario pada fungsi menambah data pengkajian umum dituliskan pada Tabel 4.25 berikut:

Tabel 4.25 Use Case Scenario Menambah Pengkajian Umum

Nama <i>use case</i>	Menambah data pengkajian umum.
<i>Objective</i>	Sistem dapat menambahkan data pengkajian umum <i>klien</i> sebagai diagnosa.
Aktor	Divisi pelayanan.
<i>Pre-condition</i>	Aktor telah <i>login</i> dan berada pada halaman <i>dashboard</i> .
<i>Mainflow</i>	1. Aktor dalam kondisi sudah <i>login</i> dan berhasil masuk ke <i>dashboard</i>. 2. Aktor memilih menu diagnosa klien. 3. Aktor memilih sub menu pengkajian umum. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> pengkajian umum. 5. Aktor menekan <i>button submit</i> untuk menyimpan data pengkajian umum ke <i>database</i>.
<i>Alternative Flow</i>	Sistem menampilkan catatan " <i>Please fill out this field</i> " apabila ada kolom tidak terisi.
<i>Post-condition</i>	Sistem berhasil menambah data pengkajian umum dan tersimpan di <i>database</i> .

4.6.2.20 Use Case Scenario Menghapus Data Pengkajian Umum

Use case scenario pada fungsi menghapus data pengkajian umum dituliskan pada Tabel 4.26 berikut:

Tabel 4.26 Use Case Scenario Menghapus Pengkajian Umum

Nama <i>use case</i>	Menghapus data pengkajian umum.
<i>Objective</i>	Sistem dapat menghapus riwayat data pengkajian umum tertentu.
Aktor	Divisi pelayanan.
<i>Pre-condition</i>	Aktor telah <i>login</i> dan berada pada halaman <i>dashboard</i> .
<i>Mainflow</i>	1. Aktor menekan menu Diagnosa klien. 2. Aktor menekan sub menu riwayat pengkajian umum. 3. Aktor menekan <i>button "icon tempat sampah"</i>. 4. Jika muncul popup pilih <i>button "ya"</i>.
<i>Alternative Flow</i>	

<i>Post-condition</i>	Sistem berhasil menghapus diagnosa pengkajian umum.
-----------------------	---

4.6.2.21 Use Case Scenario Mengedit Form Pengkajian Umum

Use case scenario pada fungsi mengedit form pengkajian umum dituliskan pada Tabel 4.27 berikut:

Tabel 4.27 Use Case Scenario Mengedit Form Pengkajian Umum

Nama use case	Mengedit <i>form</i> pengkajian umum.
<i>Objective</i>	Sistem mengedit <i>form</i> riwayat pengkajian umum.
Aktor	Divisi pelayanan
<i>Pre-condition</i>	Aktor telah <i>login</i> dan berada pada halaman <i>dashboard</i> .
<i>Mainflow</i>	1. Aktor menekan menu diagnosa klien. 2. Aktor menekan sub menu riwayat pengkajian umum. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Sistem berhasil merubah data <i>form</i> pengkajian umum dan menyimpannya dalam <i>database</i> .

4.6.2.22 Use Case Scenario Mencetak Pengkajian Umum

Use case scenario pada fungsi mencetak pengkajian umum dituliskan pada Tabel 4.28 berikut:

Tabel 4.28 Use Case Scenario Mencetak Pengkajian Umum

Nama use case	Mencetak pengkajian umum.
<i>Objective</i>	Sistem dapat mencetak <i>form</i> pengkajian umum tertentu pada <i>list</i> riwayat pengkajian umum.
Aktor	Divisi pelayanan.
<i>Pre-condition</i>	Aktor telah <i>login</i> dan berada pada halaman <i>dashboard</i> .
<i>Mainflow</i>	1. Aktor menekan menu diagnosa klien. 2. Aktor menekan sub menu riwayat pengkajian umum. 3. Aktor menekan <i>button print</i>. 4. Aktor menekan <i>button keyboard</i> ctrl + p. 5. Aktor menekan <i>print</i>.
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Sistem berhasil mencetak <i>form</i> pengkajian umum.

4.6.2.23 Use Case Scenario Menambah Data Pengkajian Luka

Use case scenario pada fungsi menambah data pengkajian luka dituliskan pada Tabel 4.29 berikut:

Tabel 4.29 Use Case Scenario Menambah Pengkajian Luka

Nama use case	Menambah data pengkajian luka.
Objective	Sistem dapat menambahkan data pengkajian luka <i>klien</i> sebagai diagnosa.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu pengkajian luka. 3. Aktor mengisi kolom yang tersedia pada <i>form</i> pengkajian luka. 4. Aktor menekan <i>button</i> submit untuk menyimpan data pengkajian luka ke <i>database</i>.
Alternative Flow	Sistem menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Post-condition	Berhasil menambah data pengkajian luka dan tersimpan di <i>database</i> .

4.6.2.24 Use Case Scenario Menghapus Data Pengkajian Luka

Use case scenario pada fungsi menghapus data pengkajian luka dituliskan pada Tabel 4.30 berikut:

Tabel 4.30 Use Case Scenario Menghapus Data Pengkajian Luka

Nama use case	Menghapus data pengkajian luka.
Objective	Sistem dapat menghapus riwayat pengkajian luka.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu Diagnosa klien. 2. Aktor memilih sub menu riwayat pengkajian luka. 3. Aktor menekan <i>button</i> "<i>icon</i> tempat sampah". 4. Jika muncul popup pilih <i>button</i> "<i>ya</i>".
Alternative Flow	-
Post-condition	Sistem berhasil menghapus data diagnosa pengkajian luka.

4.6.2.25 Use Case Scenario Mengedit form Pengkajian Luka

Use case scenario pada fungsi mengedit form pengkajian luka dituliskan pada Tabel 4.31 berikut:

Tabel 4.31 Use Case Scenario Mengedit Form Pengkajian Luka

Nama use case	Mengedit form pengkajian luka.
Objective	Sistem dapat mengedit form riwayat pengkajian luka.
Aktor	Divisi pelayanan
Pre-condition	Aktor dalam kondisi sudah login dan berada di dashboard.
Mainflow	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu riwayat pengkajian luka. 3. Aktor menekan button edit. 4. Akan muncul form edit dan mengisi bagian yang ingin diedit. 5. Aktor menekan button save.
Alternative Flow	-
Post-condition	Sistem berhasil merubah form pengkajian luka dan menyimpannya dalam database.

4.6.2.26 Use Case Scenario Mencetak Pengkajian Luka

Use case scenario pada fungsi mencetak pengkajian luka dituliskan pada Tabel 4.32 berikut:

Tabel 4.32 Use Case Scenario Mencetak Pengkajian Luka

Nama use case	Mencetak pengkajian luka.
Objective	Sistem dapat mencetak form pengkajian luka pada list riwayat pengkajian luka.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah login dan di dashboard.
Mainflow	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu riwayat pengkajian luka. 3. Aktor menekan button print. 4. Aktor mekan button keyboard ctrl + p. 5. Aktor mekan print.
Alternative Flow	-
Post-condition	Sistem berhasil mencetak form pengkajian luka.

4.6.2.27 Use Case Scenario Menambah Data Pengkajian Betes Jensen

Use case scenario pada fungsi menambah data pengkajian betes jensen dituliskan pada Tabel 4.33 berikut:

Tabel 4.33 Use Case Scenario Menambah Data Pengkajian Betes Jensen

Nama <i>use case</i>	Menambah data pengkajian betes Jensen.
<i>Objective</i>	Sistem dapat menambahkan data pengkajian betes Jensen sebagai diagnosa.
Aktor	Divisi pelayanan.
<i>Pre-condition</i>	Aktor dalam kondisi sudah <i>login</i> dan berada pada halaman pengkajian betes Jensen.
<i>Mainflow</i>	1. Aktor mengisi kolom yang tersedia pada halaman pengkajian betes Jensen. 2. Aktor menekan <i>button submit</i> untuk menyimpan data pengkajian betes Jensen ke <i>database</i>.
<i>Alternative Flow</i>	Sistem menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
<i>Post-condition</i>	Sistem berhasil menambah data betes Jensen dan tersimpan di <i>database</i> .

4.6.2.28 Use Case Scenario Menghapus Data Pengkajian Betes Jensen

Use case scenario pada fungsi menghapus data pengkajian betes jensen dituliskan pada Tabel 4.34 berikut:

Tabel 4.34 Use Case Scenario Menghapus Pengkajian Betes Jensen

Nama <i>use case</i>	Menghapus data pengkajian betes jensen.
<i>Objective</i>	Sistem dapat menghapus data riwayat pengkajian betes jensen.
Aktor	Divisi pelayanan.
<i>Pre-condition</i>	Aktor telah <i>login</i> dan berada pada halaman <i>dashboard</i> .
<i>Mainflow</i>	1. Aktor memilih menu Diagnosa klien. 2. Aktor memilih sub menu riwayat betes jensen. 3. Aktor menekan <i>button</i> "<i>icon tempat sampah</i>". 4. Jika muncul popup pilih <i>button</i> "<i>ya</i>".
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Sistem berhasil menghapus diagnosa pengkajian betes jensen.

4.6.2.29 Use Case Scenario Mengedit Form Pengkajian Betes Jensen

Use case scenario pada fungsi mengedit form pengkajian betes jensen dituliskan pada Tabel 4.35 berikut:

Tabel 4.35 Use Case Scenario Mengedit Form Pengkajian Betes Jensen

Nama use case	Mengedit form pengkajian betes jensen.
Objective	Sistem dapat mengedit form riwayat pengkajian betes Jensen.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah login dan berada di dashboard.
Mainflow	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu riwayat pengkajian betes jensen. 3. Aktor menekan <i>button edit</i>. 4. Akan muncul form edit dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Alternative Flow	-
Post-condition	Sistem berhasil mengedit form pengkajian betes jensen dan menyimpannya dalam database.

4.6.2.30 Use Case Scenario Mencetak Pengkajian Betes Jensen

Use case scenario pada fungsi mencetak pengkajian betes jensen dituliskan pada Tabel 4.36 berikut:

Tabel 4.36 Use Case Scenario Mencetak Pengkajian Betes Jensen

Nama use case	Mencetak pengkajian betes jensen.
Objective	Sistem dapat mencetak form pengkajian pengkajian betes jensen pada list riwayat pengkajian betes jensen.
Aktor	Divisi pelayanan.
Pre-condition	Aktor telah login dan berada pada halaman dashboard.
Mainflow	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu riwayat pengkajian betes jensen. 3. Aktor menekan <i>button print</i>. 4. Aktor menekan <i>button keyboard ctrl + p</i>. 5. Aktor menekan <i>print</i>.
Alternative Flow	-
Post-condition	Sistem berhasil mencetak form pengkajian betes jensen.

4.6.2.31 Use Case Scenario Menambah Data Tampilan Klinis Luka

Use case scenario pada fungsi menambah data tampilan klinis luka dituliskan pada Tabel 4.37 berikut:

Tabel 4.37 Use Case Scenario Menambah Data Tampilan Klinis Luka

Nama use case	Menambah data tampilan klinis luka.
Objective	Sistem dapat menambahkan data tampilan klinis luka sebagai diagnosa.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu tampilan klinis luka. 3. Aktor mengisi kolom yang tersedia pada <i>form</i> tampilan klinis luka. 4. Aktor menekan <i>button</i> submit untuk menyimpan data tampilan klinis luka ke <i>database</i>.
Alternative Flow	Sistem muncul catatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Post-condition	Sistem berhasil menambah data tampilan klinis luka dan tersimpan di <i>database</i> .

4.6.2.32 Use Case Scenario Menghapus Data Tampilan Klinis Luka

Use case scenario pada fungsi menghapus data tampilan klinis luka dituliskan pada Tabel 4.38 berikut:

Tabel 4.38 Use Case Scenario Menghapus Data Tampilan Klinis Luka

Nama use case	Menghapus data tampilan klinis luka.
Objective	Sistem dapat menghapus data tampilan klinis luka.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu "<i>Diagnosa klien</i>". 2. Aktor memilih sub menu riwayat betes Jensen. 3. Aktor menekan <i>button</i> "<i>icon</i> tempat sampah". 4. Jika muncul popup pilih <i>button</i> "<i>ya</i>".
Alternative Flow	-
Post-condition	Sistem berhasil menghapus diagnosa tampilan klinis luka.

4.6.2.33 Use Case Scenario Mengedit Data Tampilan Klinis Luka

Use case scenario pada fungsi mengedit data tampilan klinis luka dituliskan pada Tabel 4.39 berikut:

Tabel 4.39 Use Case Scenario Mengedit Data Tampilan Klinis Luka

Nama use case	Mengedit data tampilan klinis luka.
Objective	Sistem dapat mengedit <i>form</i> tampilan klinis luka.
Aktor	Divisi pelayanan
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu riwayat tampilan klinis luka. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Alternative Flow	-
Post-condition	Sistem berhasil mengedit <i>form</i> tampilan klinis luka dan menyimpannya dalam <i>database</i> .

4.6.2.34 Use Case Scenario Mencetak Tampilan Klinis Luka

Use case scenario pada fungsi mencetak tampilan klinis luka dituliskan pada Tabel 4.40 berikut:

Tabel 4.40 Use Case Scenario Mencetak Tampilan Klinis Luka

Nama use case	Mencetak tampilan klinis luka.
Objective	Sistem dapat mencetak <i>form</i> tampilan klinis luka pada <i>list</i> riwayat tampilan klinis luka.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu riwayat tampilan klinis luka. 3. Aktor menekan <i>button print</i>. 4. Aktor menekan <i>button keyboard</i> ctrl + p. 5. Aktor menekan <i>print</i>.
Alternative Flow	-
Post-condition	Sistem berhasil mencetak <i>form</i> tampilan klinis luka.

4.6.2.35 Use Case Scenario Menambah Tindakan

Use case scenario pada fungsi menambah tindakan dituliskan pada Tabel 4.41 berikut:

Tabel 4.41 Use Case Scenario Menambah Tindakan

Nama use case	Menambah tindakan.
Objective	Sistem dapat menambahkan tindakan pada <i>list</i> riwayat tindakan.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu tindakan. 2. Aktor memilih sub menu tindakan. 3. Aktor mengisi kolom yang tersedia pada <i>form</i> tindakan. 4. Aktor menekan <i>button submit</i> untuk menyimpan data tindakan ke <i>database</i>.
Alternative Flow	Sistem menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Post-condition	Sistem berhasil menambah data tindakan dan tersimpan di <i>database</i> .

4.6.2.36 Use Case Scenario Menghapus Tindakan

Use case scenario pada fungsi menghapus tindakan dituliskan pada Tabel 4.42 berikut:

Tabel 4.42 Use Case Scenario Menghapus Tindakan

Nama use case	Menghapus tindakan
Objective	Sistem dapat menghapus data tindakan pada <i>list</i> riwayat tindakan.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu tindakan. 2. Aktor memilih sub menu riwayat tindakan. 3. Aktor menekan <i>button</i> "<i>icon tempat sampah</i>". 4. Jika muncul popup pilih <i>button</i> "<i>ya</i>".
Alternative Flow	-
Post-condition	Sistem berhasil menghapus tindakan dalam sistem.

4.6.2.37 Use Case Scenario Mengedit Tindakan

Use case scenario pada fungsi mengedit tindakan dituliskan pada Tabel 4.43 berikut:

Tabel 4.43 Use Case Scenario Mengedit Tindakan

Nama use case	Mengedit tindakan.
Objective	Sistem mengedit <i>form</i> tindakan.
Aktor	Divisi pelayanan
Pre-condition	Aktor telah <i>login</i> dan berada pada halaman <i>dashboard</i> .
Mainflow	1. Aktor memilih menu tindakan. 2. Aktor memilih sub menu riwayat tindakan. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Alternative Flow	-
Post-condition	Sistem berhasil mengedit <i>form</i> tindakan dan menyimpannya dalam <i>database</i> .

4.6.2.38 Use Case Scenario Mencetak Tindakan

Use case scenario pada fungsi mencetak tindakan dituliskan pada Tabel 4.44 berikut:

Tabel 4.44 Use Case Scenario Mencetak Tindakan

Nama use case	Mencetak tindakan.
Objective	Sistem dapat mencetak <i>form</i> tindakan tertentu pada <i>list</i> riwayat tindakan.
Aktor	Divisi pelayanan.
Pre-condition	Aktor telah <i>login</i> dan berada pada halaman <i>dashboard</i> .
Mainflow	1. Aktor memilih menu tindakan. 2. Aktor memilih sub menu riwayat tindakan. 3. Aktor menekan <i>button print</i>. 4. Aktor menekan <i>button</i> keyboard ctrl + p. 5. Aktor menekan <i>print</i>.
Alternative Flow	-
Post-condition	Sistem berhasil mencetak <i>form</i> tindakan.

4.6.2.39 Use Case Scenario Menambah Data Perkembangan Klien

Use case scenario pada fungsi menambah data perkembangan klien dituliskan pada Tabel 4.45 berikut:

Tabel 4.45 Use Case Scenario Menambah Perkembangan Klien

Nama use case	Menambah data perkembangan klien.
Objective	Sistem dapat menambahkan data perkembangan klien pada <i>list</i> riwayat perkembangan klien.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu perkembangan klien. 2. Aktor memilih sub menu perkembangan klien. 3. Aktor mengisi kolom yang tersedia pada <i>form</i> perkembangan klien. 4. Aktor menekan <i>button submit</i> untuk menyimpan data perkembangan <i>klien</i> ke <i>database</i>.
Alternative Flow	Sistem menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Post-condition	Sistem berhasil menambah data perkembangan <i>klien</i> dan tersimpan di <i>database</i> .

4.6.2.40 Use Case Scenario Menghapus Data Perkembangan Klien

Use case scenario pada fungsi menghapus data perkembangan klien dituliskan pada Tabel 4.46 berikut:

Tabel 4.46 Use Case Scenario Menghapus Perkembangan Klien

Nama use case	Menghapus data perkembangan klien.
Objective	Sistem dapat menghapus data perkembangan klien pada <i>list</i> riwayat perkembangan klien.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu perkembangan klien. 2. Aktor memilih sub menu riwayat perkembangan klien. 3. Aktor menekan <i>button "icon tempat sampah"</i>. 4. Jika muncul popup pilih <i>button "ya"</i>.
Alternative Flow	
Post-condition	Sistem berhasil menghapus data perkembangan klien.

4.6.2.41 Use Case Scenario Mengedit Form Perkembangan Klien

Use case scenario pada fungsi mengedit *form* perkembangan klien dituliskan pada Tabel 4.47 berikut:

Tabel 4.47 Use Case Scenario Mengedit Form Perkembangan Klien

Nama use case	Mengedit <i>form</i> perkembangan klien.
Objective	Sistem dapat mengedit <i>form</i> perkembangan klien
Aktor	Divisi pelayanan
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu perkembangan klien. 2. Aktor memilih sub menu riwayat perkembangan klien. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Alternative Flow	-
Post-condition	Sistem berhasil mengedit <i>form</i> perkembangan klien dan menyimpannya dalam <i>database</i> .

4.6.2.42 Use Case Scenario Mencetak Form Perkembangan Klien

Use case scenario pada fungsi mencetak form perkembangan klien dituliskan pada Tabel 4.48 berikut:

Tabel 4.48 Use Case Scenario Mencetak Form Perkembangan Klien

Nama use case	Mencetak <i>form</i> perkembangan klien.
Objective	Sistem dapat mencetak <i>form</i> perkembangan klien tertentu pada <i>list</i> riwayat perkembangan klien.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu perkembangan klien. 2. Aktor menekan sub menu riwayat perkembangan klien. 3. Aktor menekan <i>button print</i>. 4. Aktor menekan <i>button keyboard</i> ctrl + p. 5. Aktor menekan <i>print</i>.
Alternative Flow	-
Post-condition	Sistem berhasil mencetak <i>form</i> perkembangan klien.

4.6.2.43 Use Case Scenario Menambah Kunjungan

Use case scenario pada fungsi menambah kunjungan dituliskan pada Tabel 4.49 berikut:

Tabel 4.49 Use Case Scenario Menambah Kunjungan

Nama use case	Menambah kunjungan.
Objective	Sistem dapat menambahkan data kunjungan pada <i>list</i> riwayat kunjungan.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu kunjungan. 2. Aktor memilih sub menu kunjungan. 3. Aktor mengisi kolom yang tersedia pada <i>form</i> kunjungan. 4. Aktor menekan <i>button submit</i> untuk menyimpan data kunjungan ke <i>database</i>.
Alternative Flow	Sistem menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Post-condition	Sistem berhasil menambah data kunjungan dan tersimpan di <i>database</i> .

4.6.2.44 Use Case Scenario Menghapus Kunjungan

Use case scenario pada fungsi menghapus kunjungan dituliskan pada Tabel 4.50 berikut:

Tabel 4.50 Use Case Scenario Menghapus Kunjungan

Nama use case	Menghapus kunjungan.
Objective	Sistem dapat menghapus data kunjungan pada <i>list</i> riwayat kunjungan.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu kunjungan. 2. Aktor menekan sub menu riwayat kunjungan. 3. Aktor menekan <i>button</i> "<i>icon tempat sampah</i>". 4. Jika muncul popup pilih <i>button</i> "<i>ya</i>".
Alternative Flow	-
Post-condition	Sistem berhasil menghapus data kunjungan.

4.6.2.45 Use Case Scenario Mengedit Kunjungan

Use case scenario pada fungsi mengedit kunjungan dituliskan pada Tabel 4.51 berikut:

Tabel 4.51 Use Case Scenario Mengedit Form Kunjungan

Nama use case	Mengedit kunjungan.
Objective	Sistem dapat mengedit <i>form</i> kunjungan.
Aktor	Divisi pelayanan
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu kunjungan. 2. Aktor menekan sub menu riwayat kunjungan. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Alternative Flow	-
Post-condition	Sistem berhasil mengedit <i>form</i> kunjungan dan menyimpannya dalam <i>database</i> .

4.6.2.46 Use Case Scenario Mencetak Kunjungan

Use case scenario pada fungsi mencetak kunjungan dituliskan pada Tabel 4.52 berikut:

Tabel 4.52 Use Case Scenario Mencetak Form Kunjungan

Nama use case	Mencetak kunjungan.
Objective	Sistem dapat mencetak <i>form</i> kunjungan tertentu pada <i>list</i> riwayat kunjungan.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu kunjungan. 2. Aktor menekan sub menu riwayat kunjungan. 3. Aktor menekan <i>button print</i>. 4. Aktor menekan <i>button keyboard</i> ctrl + p. 5. Aktor menekan <i>print</i>.
Alternative Flow	-
Post-condition	Sistem berhasil mencetak <i>form</i> kunjungan.

4.6.2.47 Use Case Scenario Menambah Pembayaran

Use case scenario pada fungsi menambah pembayaran dituliskan pada Tabel 4.53 berikut:

Tabel 4.53 Use Case Scenario Menambah Pembayaran

Nama use case	Menambah pembayaran.
Objective	Sistem dapat menambahkan data pembayaran pada <i>list</i> riwayat pembayaran.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu pembayaran. 2. Aktor memilih sub menu pembayaran. 3. Aktor mengisi kolom yang tersedia pada <i>form</i> pembayaran. 4. Aktor menekan <i>button submit</i> untuk menyimpan data pembayaran ke <i>database</i>.
Alternative Flow	Sistem menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Post-condition	Sistem berhasil menambah data pembayaran dan tersimpan di <i>database</i> .

4.6.2.48 Use Case Scenario Menghapus Pembayaran

Use case scenario pada fungsi menghapus pembayaran dituliskan pada Tabel 4.54 berikut:

Tabel 4.54 Use Case Scenario Menghapus Pembayaran

Nama use case	Menghapus pembayaran.
Objective	Sistem dapat menghapus data pembayaran pada <i>list</i> riwayat pembayaran.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu pembayaran. 2. Aktor menekan sub menu riwayat pembayaran. 3. Aktor menekan <i>button</i> "<i>icon</i> tempat sampah". 4. Jika muncul pop up pilih <i>button</i> "<i>ya</i>".
Alternative Flow	
Post-condition	Sistem berhasil melakukan penghapusan data pembayaran dalam sistem.

4.6.2.49 Use Case Scenario Mencetak Form Pembayaran

Use case scenario pada fungsi mencetak *form* pembayaran dituliskan pada Tabel 4.55 berikut:

Tabel 4.55 Use Case Scenario Mencetak Form Pembayaran

Nama use case	Mencetak <i>form</i> pembayaran.
Objective	Sistem dapat mencetak <i>form</i> konsultasi tertentu pada <i>list</i> riwayat pembayaran.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu pembayaran. 2. Aktor menekan sub menu riwayat pembayaran. 3. Aktor menekan <i>button print</i>. 4. Aktor menekan <i>button keyboard</i> ctrl + p. 5. Aktor menekan <i>print</i>.
Alternative Flow	-
Post-condition	Sistem berhasil mencetak <i>form</i> pembayaran.

4.6.2.50 Use Case Scenario Menambah Data Penolakan Tindakan

Use case scenario pada fungsi menambah data penolakan tindakan dituliskan pada Tabel 4.56 berikut:

Tabel 4.56 Use Case Scenario Menambah Data Penolakan Tindakan

Nama use case	Menambah data penolakan tindakan.
Objective	Sistem dapat menambahkan data penolakan tindakan pada <i>list</i> riwayat penolakan tindakan.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu dokumen. 2. Aktor memilih sub menu penolakan tindakan. 3. Aktor mengisi kolom yang tersedia pada <i>form</i> penolakan tindakan. 4. Aktor menekan <i>button submit</i> untuk menyimpan data penolakan tindakan ke <i>database</i>.
Alternative Flow	Sistem menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Post-condition	Sistem berhasil menambah data penolakan tindakan dan tersimpan di <i>database</i> .

4.6.2.51 Use Case Scenario Menghapus Data Penolakan Tindakan

Use case scenario pada fungsi menghapus data penolakan tindakan dituliskan pada Tabel 4.57 berikut:

Tabel 4.57 Use Case Scenario Menghapus Data Penolakan Tindakan

Nama use case	Menghapus data penolakan tindakan.
Objective	Sistem dapat menghapus data penolakan tindakan pada <i>list</i> riwayat penolakan tindakan.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu dokumen. 2. Aktor menekan sub menu riwayat penolakan tindakan. 3. Aktor menekan <i>button</i> "icon tempat sampah". 4. Jika muncul popup pilih <i>button</i> "ya".
Alternative Flow	-
Post-condition	Aktor berhasil menghapus data penolakan tindakan.

4.6.2.52 Use Case Scenario Mengedit Form Penolakan Tindakan

Use case scenario pada fungsi mengedit *form* penolakan tindakan dituliskan pada Tabel 4.58 berikut:

Tabel 4.58 Use Case Scenario Mengedit Form Penolakan Tindakan

Nama use case	Mengedit <i>form</i> penolakan tindakan.
Objective	Sistem dapat mengedit <i>form</i> penolakan tindakan.
Aktor	Divisi pelayanan
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu dokumen. 2. Aktor menekan sub menu riwayat penolakan tindakan. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Alternative Flow	-
Post-condition	Sistem berhasil mengedit <i>form</i> penolakan tindakan dan menyimpannya dalam <i>database</i> .

4.6.2.53 Use Case Scenario Mencetak Form Penolakan Tindakan

Use case scenario pada fungsi mencetak form penolakan tindakan dituliskan pada Tabel 4.59 berikut:

Tabel 4.59 Use Case Scenario Mencetak Form Penolakan Tindakan

Nama use case	Mencetak form penolakan tindakan.
Objective	Sistem dapat mencetak form penolakan tindakan pada list riwayat penolakan tindakan.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah login dan berada di dashboard.
Mainflow	1. Aktor memilih menu dokumen. 2. Aktor menekan sub menu riwayat penolakan tindakan. 3. Aktor menekan button print. 4. Aktor menekan button keyboard ctrl + p. 5. Aktor menekan print.
Alternative Flow	-
Post-condition	Aktor berhasil mencetak form penolakan tindakan.

4.6.2.54 Use Case Scenario Menambah Data Rujukan

Use case scenario pada fungsi menambah data rujukan dituliskan pada Tabel 4.60 berikut:

Tabel 4.60 Use Case Scenario Menambah Data Rujukan

Nama use case	Menambah data rujukan.
Objective	Sistem dapat menambahkan data rujukan pada list riwayat rujukan.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah login dan berada di dashboard.
Mainflow	1. Aktor memilih menu dokumen. 2. Aktor memilih sub menu rujukan. 3. Aktor mengisi kolom yang tersedia pada form rujukan. 4. Aktor menekan button submit untuk menyimpan data rujukan ke database.
Alternative Flow	Sistem menampilkan peringatan "Please fill out this field" apabila ada kolom yang tidak terisi.
Post-condition	Sistem berhasil menambah data rujukan dan tersimpan di database.

4.6.2.55 Use Case Scenario Menghapus Data Rujukan

Use case scenario pada fungsi menghapus data rujukan dituliskan pada Tabel 4.61 berikut:

Tabel 4.61 Use Case Scenario Menghapus Data Rujukan

Nama use case	Menghapus data rujukan.
Objective	Sistem dapat menghapus data rujukan pada <i>list</i> riwayat rujukan.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu dokumen. 2. Aktor menekan sub menu riwayat rujukan. 3. Aktor menekan <i>button</i> "icon tempat sampah". 4. Jika muncul popup pilih <i>button</i> "ya".
Alternative Flow	-
Post-condition	Sistem berhasil menghapus data rujukan.

4.6.2.56 Use Case Scenario Mengedit Form Rujukan

Use case scenario pada fungsi mengedit *form* rujukan dituliskan pada Tabel 4.62 berikut:

Tabel 4.62 Use Case Scenario Mengedit Form Rujukan

Nama use case	Mengedit <i>form</i> rujukan.
Objective	Sistem dapat mengedit <i>form</i> rujukan.
Aktor	Divisi pelayanan.
Pre-condition	Aktor telah <i>login</i> dan berada pada halaman <i>dashboard</i> .
Mainflow	1. Aktor memilih menu dokumen. 2. Aktor menekan sub menu riwayat rujukan. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Alternative Flow	-
Post-condition	Sistem berhasil mengedit <i>form</i> rujukan dan menyimpannya dalam <i>database</i> .

4.6.2.57 Use Case Scenario Mencetak Form Rujukan

Use case scenario pada fungsi mencetak form rujukan dituliskan pada Tabel 4.63 berikut:

Tabel 4.63 Use Case Scenario Mencetak Form Rujukan

Nama use case	Mencetak form rujukan.
Objective	Sistem dapat mencetak form rujukan tertentu pada list riwayat rujukan.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah login dan berada di dashboard.
Mainflow	1. Aktor memilih menu dokumen. 2. Aktor menekan sub menu riwayat rujukan. 3. Aktor menekan button print. 4. Aktor menekan button keyboard ctrl + p. 5. Aktor menekan print.
Alternative Flow	-
Post-condition	Sistem berhasil mencetak form rujukan.

4.6.2.58 Use Case Scenario Tambah Jadwal

Use case scenario pada fungsi tambah jadwal dituliskan pada Tabel 4.64 berikut:

Tabel 4.64 Use Case Scenario Tambah Jadwal

Nama use case	Tambah jadwal
Objective	Sistem dapat membuat schedule penjadwalan berobat klien.
Aktor	Divisi pelayanan.
Pre-condition	Aktor telah login dan berada pada halaman dashboard.
Mainflow	1. Aktor memilih menu penjadwalan klien. 2. Aktor memilih sub menu lihat jadwal. 3. Aktor mengisi kolom yang tersedia pada form lihat jadwal. 4. Aktor menekan button submit untuk menyimpan data rujukan ke database.
Alternative Flow	-
Post-condition	Sistem berhasil menambah data jadwal dan tersimpan di database.

4.6.2.59 Use Case Scenario Hapus Penjadwalan

Use case scenario pada fungsi hapus penjadwalan dituliskan pada Tabel 4.65 berikut:

Tabel 4.65 Use Case Scenario Hapus Penjadwalan

Nama use case	Hapus penjadwalan
Objective	Sistem dapat menghapus data penjadwalan pada <i>list</i> riwayat penjadwalan.
Aktor	Divisi pelayanan.
Pre-condition	Aktor dalam kondisi sudah <i>login</i> dan berada di <i>dashboard</i> .
Mainflow	1. Aktor memilih menu penjadwalan klien 2. Aktor menekan sub menu riwayat penjadwalan 3. Aktor menekan <i>button</i> “<i>icon</i> sampah”. 4. Jika muncul notifikasi “Anda yakin ingin menghapus riwayat data ini?” pilih <i>button</i> “<i>ya</i>”.
Alternative Flow	-
Post-condition	Aktor berhasil menghapus data penjadwalan pada sistem.

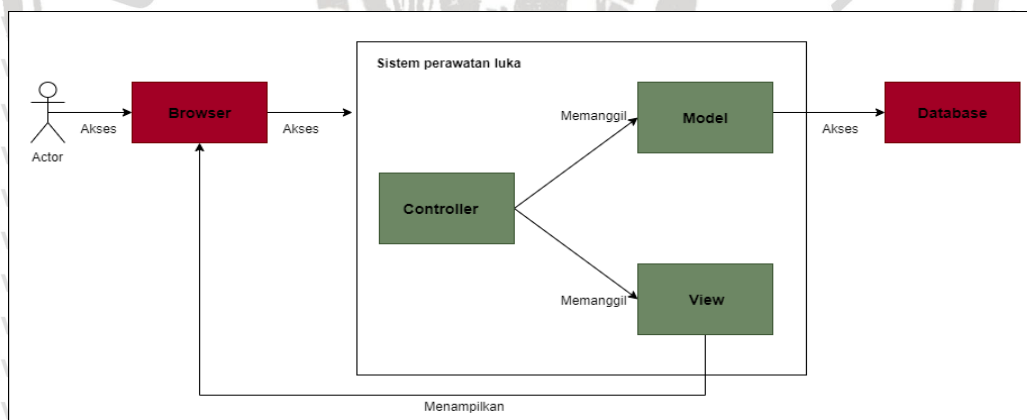
BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan Sistem

Perancangan sistem ini dikembangkan berdasar pada studi literatur dan perancangan kebutuhan yang sudah dilaksanakan sebelumnya. Langkah perancangan pada penelitian ini meliputi perancangan arsitektur sistem, perancangan data, perancangan komponen dan perancangan antarmuka. Pada perancangan arsitektur terdiri dari perancangan *sequence diagram*, perancangan *class diagram*.

5.1.1 Perancangan Arsitektur

Perancangan arsitektur merupakan gambaran model yang mudah dipahami untuk menjelaskan bagaimana suatu sistem di strukturkan dan bagaimana komponen-komponen saling bekerja sama didalamnya. Elemen-elemen perancangan ini juga memberikan gambaran keseluruhan aplikasi yang akan dikembangkan. Tujuan perancangan arsitektur adalah untuk merepresentasikan komponen-komponen program dan struktur data yang diperlukan dalam pengembangan sistem (Pressman, 2010). Proses perancangan arsitektur sistem merupakan hubungan penetapan struktur dasar kerangka kerja dalam pengembangan sistem (Sommerville, 2003). Perancangan arsitektur aplikasi perawatan luka pada penelitian ini dapat dilihat dalam Gambar 5.1.



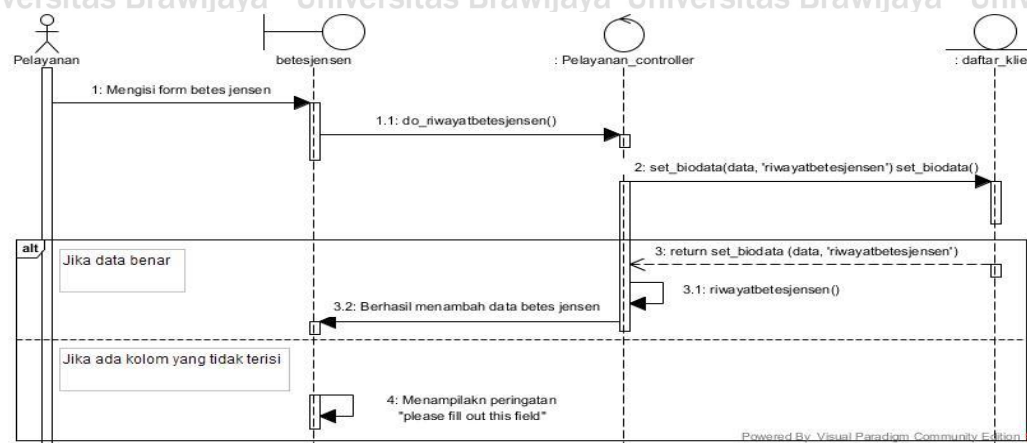
Gambar 5.1 Arsitektur Sistem

Gambar 5.1 dijelaskan, sistem dapat diakses oleh aktor melalui *browser* yang akan memproses sistem dengan menetapkan *framework CodeIgniter*. Pada *framework CodeIgniter* menerapkan pola perancangan model, view dan controller (MVC). *Controller* berisi fungsi-fungsi logika pengolahan data untuk memanggil model dan view. Kemudian, pada model berisi berbagai *query* yang dialamatkan untuk mengakses sebuah *database*. Pada view akan menampilkan data pada *browser* yang telah diolah sebelumnya melalui *controller*.

5.1.1.1 Perancangan Sequence Diagram

Sequence diagram adalah gambaran tahap demi tahap untuk menghasilkan sesuatu sesuai dengan *use case diagram* yang telah dilakukan sebelumnya (Yasin, 2012). Perancangan *sequence diagram* pada penelitian ini hanya menggambarkan 3 *use case* pada *sequence diagram*. Perancangan *sequence diagram* dapat dijelaskan dalam Gambar 5.2, Gambar 5.3, Gambar 5.4

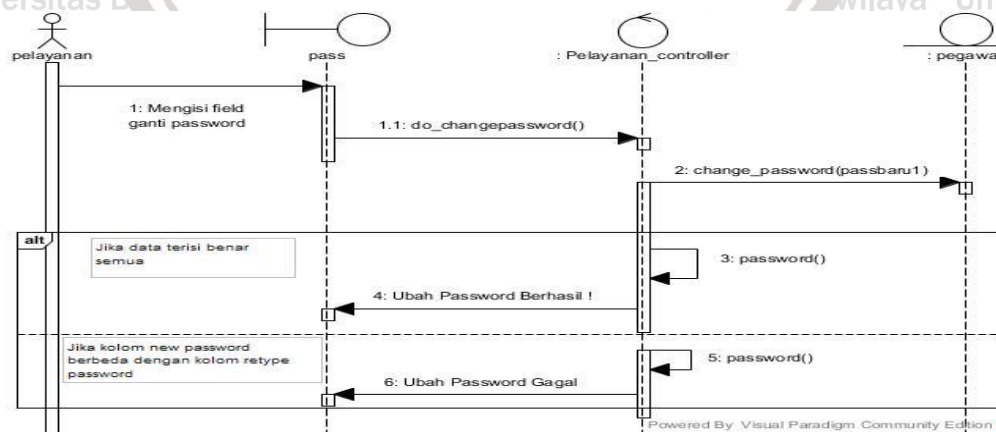
1. Sequence Diagram Tambah Riwayat Betes Jensen



Gambar 5.2 Sequence Diagram Tambah Riwayat Betes Jensen

Menambah riwayat betes Jensen dimulai dari aktor divisi pelayanan mengakses halaman betesjensen.php. kemudian ada *form* yang harus diisi untuk melakukan penambahan data riwayat betes Jensen dengan memanggil method *do_riwayatbetesjensen()* yang ada di *Pelayanan_controller*. Kemudian *Pelayanan_controller* akan memanggil fungsi *set_biodata* yang berasal dari model *daftar_klien* yang nantinya akan menyimpan ke databasenya dan nanti nilainya akan di return. Jika data sudah benar dan terisi semua maka berhasil menambah data betes jensen dan mengalihkan ke halaman riwayatbetesjensen. Sedangkan jika terdapat kolom yang tidak diisi maka akan menampilkan pesan "*please fill out this field*".

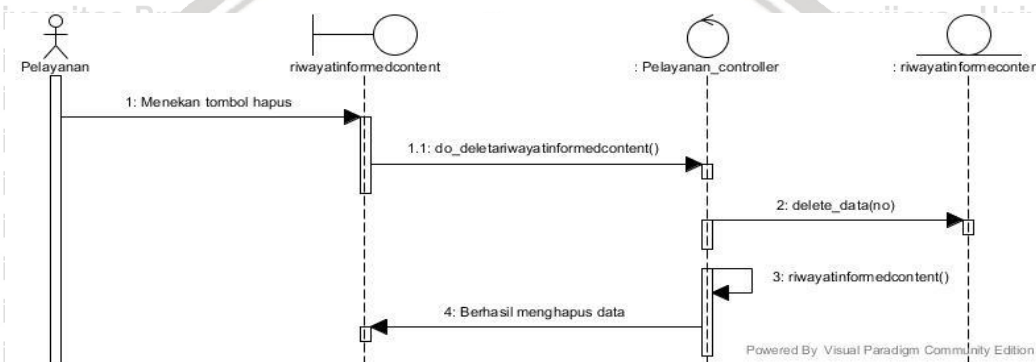
2. Sequence Diagram Mengubah Password



Gambar 5.3 Sequence Diagram Mengubah Password

Mengubah *password* dimulai dari aktor divisi pelayanan mengakses halaman *pass.php*. kemudian terdapat beberapa *field* yang harus diisi untuk melakukan penggantian *password*, yaitu *field password*, *new password* dan *retype new password* dengan memanggil method *do_changepassword()* yang ada di Pelayanan_controller. Kemudian Pelayanan_controller akan memanggil fungsi *change_password(passbaru1)* yang berasal dari model pegawai yang nantinya akan menyimpan ke databasenya. Jika data sudah benar dan terisi semua maka akan menampilkan pesan “ubah password berhasil !” dan akan dialihkan ke halaman *password*. Sedangkan jika isi kolom *new password* berbeda dengan kolom *retype new password* maka akan menampilkan pesan “Ubah password gagal !” dan akan dialihkan ke halaman *password*.

3. Sequence Diagram Delete Informed Content

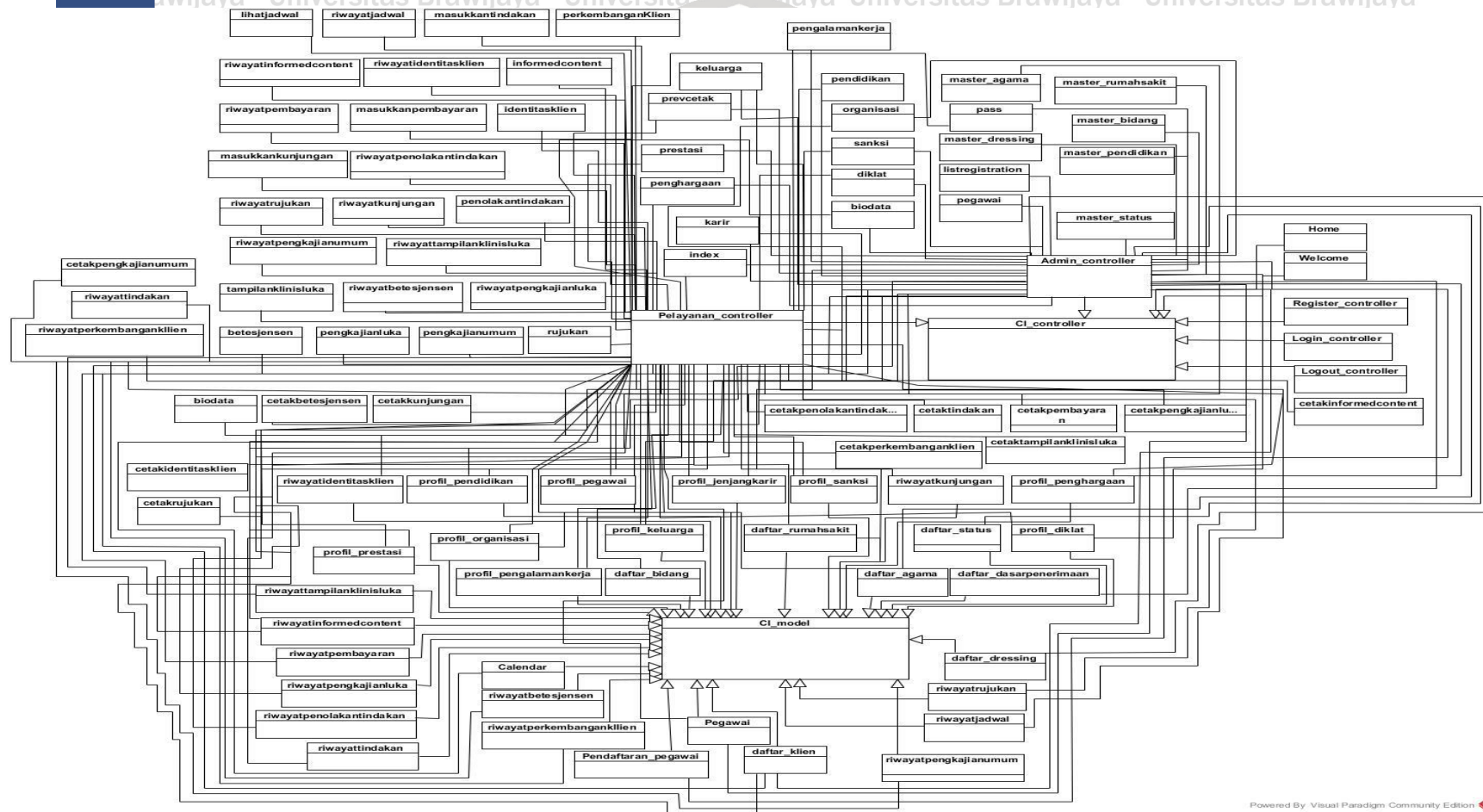


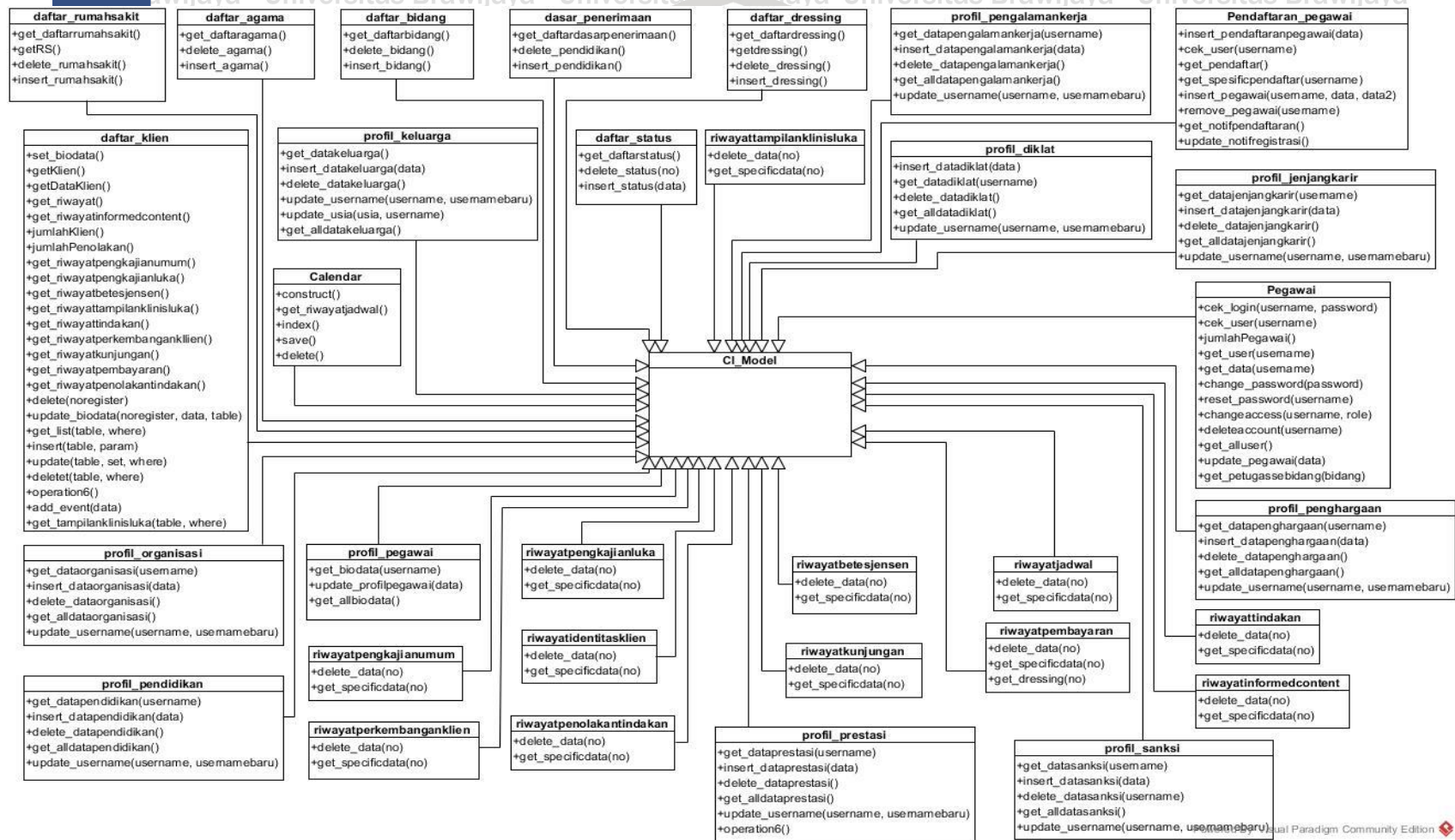
Gambar 5.4 Sequence Diagram Cetak Betes Jensen

Menghapus data *form informed content* dimulai dari aktor divisi pelayanan mengakses halaman *riwayatinformcontent.php* kemudian klik tombol hapus. Kemudian Pelayanan_controller akan memanggil fungsi *delete_data(no)* yang berasal dari model *riwayatinformcontent* yang nantinya akan menyimpan ke databasenya. Kemudian akan dialihkan ke halaman *riwayatinformcontent* jika berhasil menghapus data.

5.1.1.2 Perancangan Class Diagram

Terdapat sejumlah 7 klas *controller* pada pemodelan *class diagram* sistem informasi perawatan luka yaitu Admin_controller, Login_controller, Logout_controller, Pelayanan_controller, Register_controller, Welcome. Sedangkan pada klas model terdapat 34 Calendar, daftar_agama, daftar_bidang, daftar_dasarpenerimaan, daftar_dressing, daftar_klien, daftar_rumahsakit, daftar_status, Pegawai, Pendaftaran_pegawai, profil_diklat, profil_jenjangkarir, profil_keluarga, profil_organisasi, profil_pegawai, profil_pendidikan, profil_pengalamankerja, profil_penghargaan, profil_prestasi, profil_sanksi, riwayatbetesjensen, riwayatidentitasklien, riwayatinformcontent, riwayatjadwal, riwayatkunjungan, riwayatpembayaran, riwayatpengkajianluka, riwayatpengkjianumum, riwayatpenolakantindakan, riwayatperkembanganklien, riwayatrujukan, riwayattampilanklinisluka, riwayattindakan.





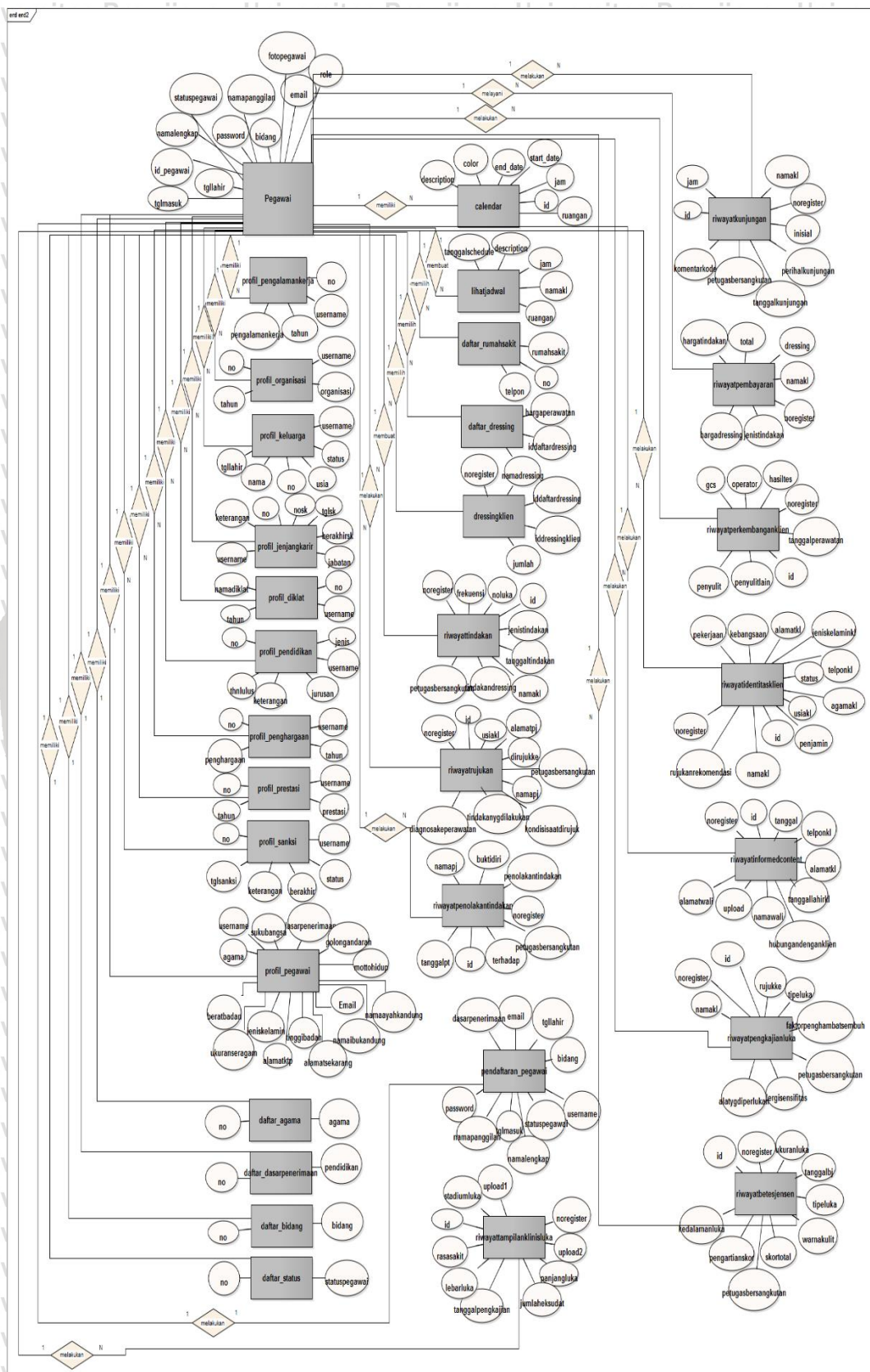
Gambar 5.6 Model Class Diagram



5.1.2 Perancangan Data

Tahap ini adalah pedoman penyusunan data yang digunakan untuk pembuatan implementasi basis data nanti, perlu dilakukan penyusunan data Gambar 5.6 adalah perancangan data yang di rancang dengan *Entity Relationship Diagram* (ERD). Terdapat 33 tabel untuk merancang data yaitu calendar, profil_pegawai, daftar_agama, daftar_bidang, daftar_dasarpenerimaan, daftar_dressing, daftar_rumahsakit, daftar_status, dressingklien, lihatjadwal, pegawai, pendaftaran_pegawai, profil_diklat, profil_keluarga, profil_organisasi, profil_prestasi, profil_sanksi, riwayatpengkajianumum, profil_pendidikan, riwayatindakan, profil_pengalamankerja, riwayatrujukan, riwayatidentitasklien, profil_penghargaan, riwayatpenolakantindakan, riwayatkunjungan, riwayatampilanklinisluka, riwayatpembayaran, riwayatbetesjensen, riwayatinformedcontent, riwayatperkembanganklien, riwayatpengkajianluka.





Tabel 5.1 Struktur Tabel *Calendar*

No	Atribut	Tipe	Panjang	Keterangan
1	id	Int	11	Id sebagai <i>primary key</i> .
2	ruangan	Varchar	20	Menyimpan ruangan.
3	jam	Time	-	Waktu dibuatnya jadwal berobat klien.
4	description	Text	-	Keterangan deskripsi.
5	color	Varchar	24	
6	Start_date	date	-	Keterangan tanggal awal pertemuan.
7	End_date	date	-	Keterangan tanggal akhir pertemuan.

Tabel 5.2 Struktur Tabel *daftar_agama*

No	Atribut	Tipe	Panjang	Keterangan
1	no	Int	255	no sebagai <i>primary key</i> .
2	agama	Varchar	20	Menyimpan agama.

Tabel 5.3 Struktur Tabel *daftar_bidang*

No	Atribut	Tipe	Panjang	Keterangan
1	no	Int	255	no sebagai <i>primary key</i> .
2	bidang	Varchar	20	Menyimpan bidang atau divisi kerja.

Tabel 5.4 Struktur Tabel *daftar_dasarpenerimaan*

No	Atribut	Tipe	Panjang	Keterangan
1	no	Int	255	no sebagai <i>primary key</i> .
2	pendidikan	Varchar	20	Menyimpan Pendidikan.

Tabel 5.4 Struktur Tabel *daftar_dressing*

No	Atribut	Tipe	Panjang	Keterangan
1	iddaftardressing	Int	20	iddaftardressing sebagai <i>primary key</i> .
2	namadressing	Varchar	50	Menyimpan namadressing.
3	hargaperawatan	int	50	Menyimpan hargaperawatan.

Tabel 5.5 Struktur Tabel *daftar_rumahsakit*

No	Atribut	Tipe	Panjang	Keterangan
----	---------	------	---------	------------

1	no	Int	8	no sebagai <i>primary key</i> .
2	rumahsakit	Varchar	50	Menyimpan rumahsakit.

Tabel 5.6 Struktur Tabel daftar_status

No	Atribut	Tipe	Panjang	Keterangan
1	no	Int	255	no sebagai <i>primary key</i> .
2	statuspegawai	Varchar	20	Menyimpan statuspegawai.

Tabel 5.7 Struktur Tabel dressingklien

No	Atribut	Tipe	Panjang	Keterangan
1	idaddressingklien	Int	11	idaddressingklien sebagai <i>primary key</i> .
2	iddaftardressing	Int	11	Menyimpan iddaftardressing.
3	noregister	Int	11	Menyimpan noregister.
4	jumlah	Int	50	Menyimpan jumlah.

Tabel 5.8 Struktur Tabel lihatjadwal

No	Atribut	Tipe	Panjang	Keterangan
1	namakl	Int	255	no sebagai <i>primary key</i> .
2	ruangan	Varchar	20	Menyimpan Pendidikan.
3	jam	Time	-	Waktu penjadwalan bertemu.
4	description	Varchar	150	Menyimpan deskripsi pertemuan.
5	tanggalschedule	Date	-	Tanggal penjadwalan berobat.

Tabel 5.9 Struktur Tabel pegawai

No	Atribut	Tipe	Panjang	Keterangan
1	id_pegawai	int	11	Id-pegawai sebagai <i>primary key</i> .
2	username	Int	10	Menyimpan username.
3	email	Varchar	55	Menyimpan email.
4	password	Varchar	255	Menyimpan password.
5	namalengkap	Varchar	255	Menyimpan namalengkap.
6	namapanggilan	Varchar	20	Menyimpan namapanggilan.
7	foto_pegawai	Varchar	200	Menyimpan foto_pegawai.

8	statuspegawai	Varchar	10	Menyimpan statuspegawai.
9	bidang	Varchar	20	Menyimpan bidang.
10	tgllahir	date	-	Menyimpan tgllahir pegawai.
11	tglmasuk	date	-	Menyimpan tglmasuk pegawai.
12	tglpensiun	varchar	10	Menyimpan tglpensiun.

Tabel 5.10 Struktur Tabel pendaftaran_pegawai

No	Atribut	Tipe	Panjang	Keterangan
1	username	Int	10	Menyimpan username.
2	email	Varchar	55	Menyimpan email.
3	password	Varchar	255	Menyimpan password.
4	namalengkap	Varchar	255	Menyimpan namalengkap.
5	namapanggilan	Varchar	20	Menyimpan namapanggilan.
6	statuspegawai	Varchar	10	Menyimpan statuspegawai.
7	bidang	Varchar	20	Menyimpan bidang.
8	tgllahir	date	-	Menyimpan tgllahir pegawai.
9	tglmasuk	date	-	Menyimpan tglmasuk pegawai.
10	tglpensiun	varchar	10	Menyimpan tglpensiun.
11	dasarpenerimaan	Varchar	10	Menyimpan dasarpenerimaan.
12	terbaca	tinyint	1	Menyimpan terbaca.

Tabel 5.11 Struktur Tabel profil_diklat

No	Atribut	Tipe	Panjang	Keterangan
1	no	Int	255	no sebagai <i>primary key</i> .
2	username	Int	10	Menyimpan username.
3	namadiklat	Varchar	50	Menyimpan namadiklat.
4	tahun	Varchar	4	Menyimpan tahun.

Tabel 5.12 Struktur Tabel profil_jenangkarir

No	Atribut	Tipe	Panjang	Keterangan
1	no	Int	255	no sebagai <i>primary key</i> .
2	username	Int	10	Menyimpan username.

3	jabatan	Varchar	30	Menyimpan jabatan.
4	tglsk	Date	-	Menyimpan tglsk.
5	nosk	Varchar	20	Menyimpan nosk.
6	berakhirsk	date	-	Menyimpan waktu berakhirsk.
7	keterangansk	Varchar	50	Menyimpan keterangank.

Tabel 5.13 Struktur Tabel profil_keluarga

No	Atribut	Tipe	Panjang	Keterangan
1	no	Int	255	no sebagai <i>primary key</i> .
2	username	Int	10	Menyimpan username.
3	status	Varchar	20	Menyimpan status.
4	nama	Varchar	50	Menyimpan nama.
5	tgllahir	Date	-	Menyimpan tgllahir.
6	usia	int	2	Menyimpan usia.

Tabel 5.14 Struktur Tabel profil_organisasi

No	Atribut	Tipe	Panjang	Keterangan
1	no	Int	255	no sebagai <i>primary key</i> .
2	username	Int	10	Menyimpan username.
3	organisasi	Varchar	30	Menyimpan organisasi.
4	tahun	Varchar	4	Menyimpan tahun.

Tabel 5.15 Struktur Tabel profil_pegawai

No	Atribut	Tipe	Panjang	Keterangan
1	username	Varchar	10	Username sebagai <i>primary key</i> .
2	jeniskelamin	Varchar	9	Menyimpan jeniskelamin.
3	dasarpenerimaan	Varchar	4	Menyimpan dasarpenerimaan.
4	agama	Varchar	10	Menyimpan agama.
5	sukubangsa	Varchar	10	Menyimpan sukubangsa.
6	golongandarah	Varchar	1	Menyimpan golongandarah.
7	tingggibadan	Varchar	6	Menyimpan tingggibadan.
8	beratbadan	Varchar	6	Menyimpan beratbadan

9	hobi	Varchar	100	Menyimpan hobi.
10	mottohidup	Varchar	100	Menyimpan mottohidup.
11	noktp	Varchar	16	Menyimpan noktp.
12	alamatktip	Varchar	100	Menyimpan alamatktip.
13	alamatsekarang	Varchar	100	Menyimpan alamatsekarang.
14	email	Varchar	50	Menyimpan email.
15	status	Varchar	10	Menyimpan status.
16	nokk	Varchar	17	Menyimpan nokk.
17	bpjs	Varchar	11	Menyimpan bpjs.
18	nosepatu	Varchar	2	Menyimpan nosepatu.
19	ukuranseragam	Varchar	5	Menyimpan ukuranseragam.
20	namaibukandung	Varchar	30	Menyimpan namaibukandung.
21	namaayahkandung	Varchar	30	Menyimpan namaayahkandung.
22	keahlianhusus	Varchar	100	Menyimpan keahlianhusus.
23	bahasadikuasai	Varchar	50	Menyimpan bahasadikuasai.
24	hasilmedical	Varchar	50	Menyimpan hasilmedical.

Tabel 5.16 Struktur Tabel profil_pendidikan

No	Atribut	Tipe	Panjang	Keterangan
1	no	Int	255	no sebagai <i>primary key</i> .
2	username	Varchar	10	Menyimpan username.
3	Jenis	Varchar	20	Menyimpan jenis.
4	keterangan	Varchar	50	Menyimpan keterangan.
5	jurusan	Varchar	20	Menyimpan jurusan.
6	thnlulus	Varchar	4	Menyimpan thnlulus.

Tabel 5.17 Struktur Tabel profil_pengalamankerja

No	Atribut	Tipe	Panjang	Keterangan
1	no	Int	255	no sebagai <i>primary key</i> .
2	username	Varchar	10	Menyimpan username.
3	pengalamankerja	Varchar	100	Menyimpan pengalamankerja.

4	tahun	Varchar	4	Menyimpan tahun.
---	-------	---------	---	------------------

Tabel 5.18 Struktur Tabel profil_penghargaan

No	Atribut	Tipe	Panjang	Keterangan
1	no	Int	255	no sebagai <i>primary key</i> .
2	username	Varchar	10	Menyimpan username.
3	penghargaan	Varchar	30	Menyimpan penghargaan.
4	tahun	Varchar	4	Menyimpan tahun.

Tabel 5.19 Struktur Tabel profil_prestasi

No	Atribut	Tipe	Panjang	Keterangan
1	no	Int	255	no sebagai <i>primary key</i> .
2	username	Varchar	10	Menyimpan username.
3	prestasi	Varchar	30	Menyimpan prestasi.
4	tahun	Varchar	4	Menyimpan tahun.

Tabel 5.20 Struktur Tabel profil_sanksi

No	Atribut	Tipe	Panjang	Keterangan
1	no	Int	255	no sebagai <i>primary key</i> .
2	username	Varchar	10	Menyimpan username.
3	tglsanksi	date	-	Menyimpan tglsanksi.
4	keterangan	Varchar	50	Menyimpan keterangan.
5	berakhir	date	-	Menyimpan berakhir.
6	status	Varchar	20	Menyimpan status.

Tabel 5.21 Struktur Tabel riwayatbetesjensen

No	Atribut	Tipe	Panjang	Keterangan
1	noregister	Varchar	20	noregister sebagai <i>primary key</i> .
2	Tanggalbj	date	-	Waktu pengkajian betes Jensen.
3	namakl	Varchar	30	Menyimpan namakl.
4	ukuranluka	Varchar	5	Menyimpan ukuranluka.
5	kedalamanluka	Varchar	5	Menyimpan kedalamanluka.
6	tipeluka	Varchar	5	Menyimpan tipeluka.

7	goa	Varchar	5	Menyimpan goa.
8	tipeeksudat	Varchar	5	Menyimpan tipeeksudat.
9	warnakulit	Varchar	5	Menyimpan warnakulit.
10	jaringanedema	Varchar	5	Menyimpan jaringanedema.
11	jaringangranulasi	Varchar	5	Menyimpan jaringangranulasi.
12	epitelisasi	Varchar	5	Menyimpan epitelisasi.
13	skortotal	Varchar	5	Menyimpan skortotal.
14	petugasbersangkutan	Varchar	30	Menyimpan petugasbersangkutan.
15	jumlaheksudat	Varchar	50	Menyimpaneksudat.
16	pengartianskor	Varchar	50	Menyimpan pengartianskor.

5.22 Struktur Tabel riwayatidentitaklien

No	Atribut	Tipe	Panjang	Keterangan
1	noregister	Varchar	20	noregister sebagai <i>primary key</i> .
2	namakl	Varchar	25	Menyimpan namakl
3	tanggallahirkl	date	-	Menyimpan tanggallahirkl.
4	usiakl	Varchar	5	Menyimpan usiakl.
5	jeniskelaminkl	Varchar	13	Menyimpan jeniskelaminkl.
6	alamatkl	Varchar	50	Menyimpan alamatkl
7	telponkl	Varchar	13	Menyimpan telponkl.
8	agamakl	Varchar	25	Menyimpan agamakl.
9	penjamin	Varchar	20	Menyimpan penjaminkl.
10	rujukanrekomendasi	Varchar	30	Menyimpan rujukanrekomendasi.
11	status	Varchar	5	Menyimpan status.
12	kebangsaan	Varchar	15	Menyimpan kebangsaan.
13	pekerjaan	Varchar	30	Menyimpan pekerjaan.

Tabel 5.23 Struktur Tabel riwayatindformedcontent

No	Atribut	Tipe	Panjang	Keterangan
----	---------	------	---------	------------

1	noregister	Varchar	20	noregister sebagai <i>primary key</i> .
2	tanggal	date	-	Tanggal pelaksanaan pengisian <i>form informed content</i>
3	tanggallahirkl	date	-	Menyimpan tanggallahirkl.
4	alamatkl	varchar	100	Menyimpan alamatkl.
5	telponkl	Int	15	Menyimpan telponkl.
6	namawali	Varchar	25	Menyimpan namawali.
7	alamatwali	Varchar	50	Menyimpan alamatwali.
8	telponwali	Varchar	15	Menyimpan telponwali.
9	tanggallahirwali	date	-	Menyimpan tanggallahirwali.
10	hubungandengankl	Varchar	20	Menyimpan hubungandengankl.
11	diagnosakl	Varchar	70	Menyimpan diagnosakl.
12	upload	Varchar	100	Menyimpan upload gambar.

Tabel 5.24 Struktur Tabel riwayatkunjugan

No	Atribut	Tipe	Panjang	Keterangan
1	noregister	Varchar	20	noregister sebagai <i>primary key</i> .
2	namakl	Varchar	30	Menyimpan namakl
3	perihalkunjugan	Varchar	100	Menyimpan perihalkunjugan.
4	jam	time	-	Menyimpan jam.
5	tanggalkunjugan	date	-	Menyimpan tanggal kunjungan.
6	inisial	Varchar	50	Menyimpan inisial.
7	komentarkode	Varchar	100	Menyimpan komentarkode
8	petugasbersangkutan	Varchar	30	Menyimpan petugasbersangkutan.

Tabel 5.25 Struktur Tabel riwayatpembayaran

No	Atribut	Tipe	Panjang	Keterangan
----	---------	------	---------	------------

1	noregister	Int	11	noregister sebagai <i>primary key</i> .
2	namakl	Varchar	30	Menyimpan namakl
3	jenistindakan	Varchar	20	Menyimpan jenistindakan.
4	dressing	Varchar	255	Menyimpan dressing.
5	hargadressing	Int	11	Menyimpan hargadressing.
6	hargatindakan	Int	11	Menyimpan hargatindakan.
7	total	Int	11	Menyimpan total.

Tabel 5.26 Struktur Tabel riwayatpengkajianluka

No	Atribut	Tipe	Panjang	Keterangan
1	noregister	Varchar	20	Noregister sebagai <i>primary key</i> .
2	namakl	Varchar	30	Menyimpan namakl.
3	faktorpenghambatsembuh	Varchar	100	Menyimpan faktorpenghambatsembuh.
4	alergisensifitas	Varchar	100	Menyimpan alergisensifitas.
5	rujukke	Varchar	50	Menyimpan rujukke.
6	alatygdiinginkan	Varchar	100	Menyimpan alatygdiinginkan.
7	tipeluka	Varchar	50	Menyimpan tipeluka.
8	petugasbersangkutan	Varchar	30	Menyimpan petugasbersangkutan.

Tabel 5.27 Struktur Tabel riwayatpengkajianumum

No	Atribut	Tipe	Panjang	Keterangan
1	noregister	Varchar	30	noregister sebagai <i>primary key</i> .
2	doktermerawatmerujuk	Varchar	30	Menyimpan doktermerawatmerujuk.
3	dokterkonsultan	Varchar	30	Menyimpan dokterkonsultan.
4	diagnosa	Varchar	100	Menyimpan diagnosa.
5	tanggaldirawat	date	-	Menyimpan tanggaldirawat.

6	adviceterapi	Varchar	50	Menyimpan adviceterapi
7	pernahdirawat	Varchar	5	Menyimpan pernahdirawat.
8	pernahoperasi	Varchar	5	Menyimpan pernahoperasi.
9	penyakitkeluarga	Varchar	50	Menyimpan penyaktkeluarga.
10	sistempencernaan	Varchar	70	Menyimpan sistempencernaan.
11	sistemrespirasi	Varchar	70	Menyimpan sistemrespirasi.
12	sistemkardiovaskuler	Varchar	70	Menyimpan sistemkardiovaskuler.
13	keluhanutama	Varchar	100	Menyimpan keluhanutama.
14	keluhanumum	Varchar	100	Menyimpan keluhanumum.
15	td	Varchar	20	Menyimpan td.
16	rr	Varchar	20	Menyimpan rr.
17	n	Varchar	20	Menyimpan n.
18	s	Varchar	5	Menyimpan s.
19	tinggibadan	Varchar	5	Menyimpan tinggibadan.
20	beratbadan	Varchar	3	Menyimpan beratbadan.
21	penyakitlain	Varchar	30	Menyimpan penyakilain.
22	petugasbersangkutan	Varchar	30	Menyimpan petugasbersangkutan.

Tabel 5.28 Struktur Tabel riwayatpenolakantindakan

No	Atribut	Tipe	Panjang	Keterangan
1	noregister	Varchar	30	Noregister sebagai <i>primary key</i>
2	tanggalpt	date	-	Menyimpan tanggalpt.
3	namapj	Varchar	30	Menyimpan namapj.
4	buktidiri	Varchar	25	Menyimpan buktidiri.
5	penolakantindakan	Varchar	50	Menyimpan penolakantindakan.

6	terhadap	Varchar	65	Menyimpan terhadap.
7	dirawatdi	Varchar	30	Menyimpan dirawatdi.
8	petugasbersangkutan	Varchar	30	Menyimpan petugasbersangkutan.

Tabel 5.29 Struktur Tabel perkebanganklien

No	Atribut	Tipe	Panjang	Keterangan
1	noregister	Varchar	30	Noregister sebagai <i>primary key</i> .
2	tanggalperawatan	date	-	Menyimpan tanggalpt.
3	operator	Varchar	30	Menyimpan namapj.
4	hasiltes	Varchar	50	Menyimpan buktidiri.
5	penyulit	Varchar	15	Menyimpan penolakantindakan.
6	penyulitlain	Varchar	50	Menyimpan terhadap.
7	gcs	Varchar	70	Menyimpan dirawatdi.

Tabel 5.30 Struktur Tabel riwayatrujukan

No	Atribut	Tipe	Panjang	Keterangan
1	noregister	Varchar	30	noregister sebagai <i>primary key</i> .
2	usiakl	Varchar	5	Menyimpan usiakl.
3	namapj	Varchar	25	Menyimpan namapj
4	alamatpj	Varchar	30	Menyimpan alamatpj.
5	diagnosakeperawatan	Varchar	100	Menyimpan diagnosakeperawatan.
6	dirujukke	Varchar	40	Menyimpan dirujukke.
7	kondisisaatdirujuk	Varchar	80	Menyimpan kondisisaatdirujuk.
8	tindakanygdilakukan	Varchar	50	Menyimpan tindakanygdilakukan.
9	petugasbersangkutan	Varchar	30	Menyimpan petugasbersangkutan.

Tabel 5.31 Struktur Tabel riwayatindakan

No	Atribut	Tipe	Panjang	Keterangan
----	---------	------	---------	------------

1	noregister	Varchar	30	noregister sebagai <i>primary key</i> .
2	namakl	Varchar	30	Menyimpan namakl
3	tanggaltindakan	Date	-	Menyimpan tanggaltindakan.
4	noluka	Varchar	10	Menyimpan noluka
5	tindakandressing	Varchar	70	Menyimpan tindakandressing.
6	frekuensi	Varchar	70	Menyimpan frekuensi.
7	evaluasi	Varchar	100	Menyimpan evaluasi
8	jenistindakan	Varchar	20	Menyimpan jenistindakan.
9	petugasbersangkutan	Varchar	30	Menyimpan petugasbersangkutan.

Tabel 5.32 Struktur Tabel riwayattampilanklinisluka

No	Atribut	Tipe	Panjang	Keterangan
1	noregister	Varchar	30	noregister sebagai <i>primary key</i> .
2	noluka	Varchar	25	Menyimpan noluka
3	tanggalpengkajian	Date	-	Menyimpan tanggalpengkajian.
4	analgetikpredressing	Varchar	50	Menyimpan analgetikpredressing.
5	panjangluka	Varchar	5	Menyimpan panjangluka.
6	lebarluka	Varchar	5	Menyimpan lebarluka.
7	kedalamanluka	Varchar	5	Menyimpan kedalamanluka.
8	datemininglubanggoa	Varchar	70	Menyimpan datemininglubanggoa.
9	tracking	Varchar	70	Menyimpan tracking.
10	tepiluka	Varchar	70	Menyimpan tepiluka.
11	nekrotik	Varchar	5	Menyimpan nekrotik.
12	slough	Varchar	5	Menyimpan slough.
13	granulasi	Varchar	5	Menyimpan granulasi.
14	epitelisasi	Varchar	5	Menyimpan epitelisasi.

15	hypergranulasi	Varchar	5	Menyimpan hypergranulasi.
16	infeksi	Varchar	5	Menyimpan infeksi.
17	tampilanwarnalainnya	Varchar	50	Menyimpan tampilan warnalainnya.
18	serous	Varchar	70	Menyimpan serous.
19	blood	Varchar	70	Menyimpan blood.
20	haemoserous	Varchar	70	Menyimpan haemoserous.
21	purulent	Varchar	70	Menyimpan purulent.
22	haemopurulent	Varchar	70	Menyimpan haemopurulent.
23	odour	Varchar	70	Menyimpan odour.
24	jumlahheksudat	Varchar	100	Menyimpan jumlahheksudat.
25	kulitsekitarluka	Varchar	10	Menyimpan kulitsekitarluka.
26	rasasakit	Varchar	15	Menyimpan rasasakit.
27	frekuensi	Varchar	10	Menyimpan frekuensi.
28	stadiumluka	Varchar	5	Menyimpan stadiumluka.
29	upload1	Varchar	255	Menyimpan upload1.
30	upload2	Varchar	255	Menyimpan upload2.
31	petugasbersangkutan	Varchar	30	Menyimpan petugasbersangkutan.

5.1.3 Perancangan Komponen

Perancangan komponen untuk pengembangan *software* mendeskripsikan rincian algoritma pada tiap-tiap komponen (Pressman, 2010) tahap ini menggambarkan tentang penggunaan algoritma yang diimplementasikan pada klas-klas yang sudah dideskripsikan. Tujuan adanya tahap ini untuk menyesuaikan rancangan perangkat lunak yang sudah digambarkan dengan perangkat lunak yang dibuat. Beberapa algoritma yang dipaparkan dalam perancangan komponen ini merupakan sampel dari beberapa fungsi diantaranya tambah betes jensen, mengubah *password*, menghapus *informed content*.

5.1.3.1 Perancangan Komponen Method Tambah Betes Jensen

Pada algoritme tambah betes jensen mendefinisikan pseudocode method `do_rwayatbetesjensen()` yang digunakan untuk memasukkan data kajian tentang luka *klien* dan kemudian akan disimpan dalam *database*. Terdapat beberapa *text*

field yang harus dimasukkan di dalamnya diantaranya adalah tanggal pengkajian betes Jensen, noregister, ukuran luka, kedalaman luka, tipe luka, goa, tipeeksudat, warna kulit, jaringanedema, jaringangranulasi, epitelisasi, petugasbersangkutan, jumlaheksudat, skor total dan pengatian skor. Pada Tabel 5.1 terdapat struktur tabel seperti yang terlihat di bawah ini.

Tabel 5.33 Struktur Tabel Method do_riwayatbetesjensen

No	Method do_riwayatbetesjensen
1	Input data variabel tanggalbj
2	Input data variabel noregister
3	Input data variabel ukuranluka
4	Input data variabel kedalamanluka
5	Input data variabel tipeluka
6	Input data variabel goa
7	Input data variabel tipeeksudat
8	Input data variable warnakulit
9	Input data variable jaringanedema
10	Input data variable jaringangranulasi
11	Input data variable epitelisasi
12	Input data variable skortotal
13	Input data variable pengartianskor
14	Input data variable petugasbersangkutan
15	Input data variable jumlaheksudat
16	Mengambil class model daftar_klien
17	Inisialisasi variable array data
18	Menyimpan data dari variable tanggalbj ke kolom table
19	database
20	Menyimpan data dari variable noregister ke kolom table
21	database
22	Menyimpan data dari variable ukuranluka ke kolom table
23	database
24	Menyimpan data dari variable kedalamanluka ke kolom table
25	database
26	Menyimpan data dari variable tipeluka ke kolom table
27	database
28	Menyimpan data dari variable goa ke kolom table database
29	Menyimpan data dari variable tipeeksudat ke kolom table
30	database
31	Menyimpan data dari variable warnakulit ke kolom table
32	database
33	Menyimpan data dari variable jaringanedema ke kolom table
34	database
35	Menyimpan data dari variable jaringangranulasi ke kolom
36	table database
37	Menyimpan data dari variable epitelisasi ke kolom table
38	database
39	Menyimpan data dari variable skortotal ke kolom table
40	database
41	Menyimpan data dari variable pengartianskor ke kolom table
42	database
43	Menyimpan data dari variable petugasbersangkutan ke kolom
44	table database
45	Menyimpan data dari variable jumlaheksudat ke kolom table
46	database
47	
48	

49	Inisialisasi variable data dengan index skorttotal dari
50	penjumlahan variable data dengan index ukuranluka,
51	kedalamanluka, tipeluka, goa, tipeekudat, warnakulit,
52	jaringanedema, epitelisasi, jumlahheksudat
53	Jika nilai variable data dengan index skorttotal <=15
54	maka inisialisasi variable data dengan index pengartianskor
55	= "Jaringan Sehat"
56	Jika nilai variable data dengan index skorttotal <= 30
57	maka inisialisasi variable data dengan index pengartianskor
58	= "Regenerasi Luka"
59	Jika nilai variable data dengan index skorttotal <= 55
60	maka inisialisasi variable data dengan index pengartianskor
61	= "Degenerasi Luka"
62	Inisialisasi variable berhasil memanggil method set_biodata
63	dengan parameter variable data dan nama Tabel
64	riwayatbetesjensen

5.1.3.2 Perancangan Komponen Method Ubah Password

Pada algoritme ubah *password* mendefinisikan *pseudocode* method `do_changepassword()` yang digunakan untuk mengganti *password* lama aktor menjadi *password* baru dan kemudian akan disimpan dalam *database*. Terdapat beberapa *text field* yang harus dimasukkan di dalamnya diantaranya adalah *password*, *new password* dan *retype new password*. Pada Tabel 5.2 terdapat struktur tabel ubah *password*.

Tabel 5.34 Struktur Tabel Method `do_changepassword()`

No	Method <code>do_changepassword()</code>
1	Mulai method <code>do_changepassword</code>
2	Inisialisasi variable <i>user</i> untuk mendapatkan data <i>user</i> daam
3	<i>row</i>
4	Inisialisasi variable <i>passasli</i> untuk mendapatkan <i>password</i>
5	<i>user</i>
6	Inisialisasi variable <i>passlama</i> untuk mendapatkan <i>password</i>
7	lama dari <i>form</i> sebelumnya
8	<i>Passlama</i> melakukan hashing
9	Inisialisasi variable <i>passbaru1</i> untuk mendapatkan <i>password</i>
10	baru 1 dari <i>form</i>
11	<i>Passbaru1</i> melakukan hashing
12	Inisialisasi variable <i>passbaru2</i> untuk mendapatkan <i>paword</i>
13	baru 2 dari <i>form</i>
14	<i>Passbaru2</i> melakukan hashing
15	If <i>passasli</i> == <i>passlama</i> && <i>passbaru1</i> == <i>passbaru2</i>
16	Memanggil method <code>change_password</code> dengan parameter <i>passbaru1</i>
17	Mengirim pesan mulai <i>session flashdata</i>
18	Mengalihkan ke halaman <i>password</i>
19	Else
20	Mengirim pesan melalui <i>session flashdata</i>
21	Mengalihkan ke halaman <i>password</i>
22	End if
23	selesai

5.1.3.3 Perancangan Komponen Method Delete Riwayat Informed Content

Pada algoritme tambah betes jensen mendefinisikan *pseudocode* method `do_deleteriwayatinformcontent()` yang digunakan menghapus data *informed content*. Pada Tabel 5.3 terdapat struktur tabel ubah *password*.

Tabel 5.35 Struktur Tabel Delete Riwayat *Informed Content*

No	Method <code>do_deleteriwayatinformcontent</code>
1	Mulai method <code>do_deleteriwayatinformcontent()</code>
2	Memanggil model <code>riwayatinformcontent</code>
3	Variable <code>no</code> memanggil uri dari segmen 3
4	Memanggil <code>delete_data</code> dengan parameter <code>no</code>
5	Mengalihkan ke halaman <code>riwayatinformcontent</code>
6	selesai

5.1.4 Perancangan Antarmuka

Pada tahap ini terdapat gambaran rancangan antarmuka atau tampilan dari sistem yang dikembangkan. Perancangan ini berfungsi sebagai pedoman dalam pengembangan sistem informasi perawatan luka praktik mandiri pedis care malang berbasis web. Terdapat 3 sampel yang diambil dalam proses pengembangan sistem ini yaitu fungsi tambah riwayat betes jensen, ubah *password* dan *delete riwayat informed content*.

5.1.4.1 Perancangan Antarmuka Tambah Riwayat Betes Jensen

No	Column 2	Column 3
1	Cell Content 1	Cell Content 1
2	Cell content 2	Cell Content 1

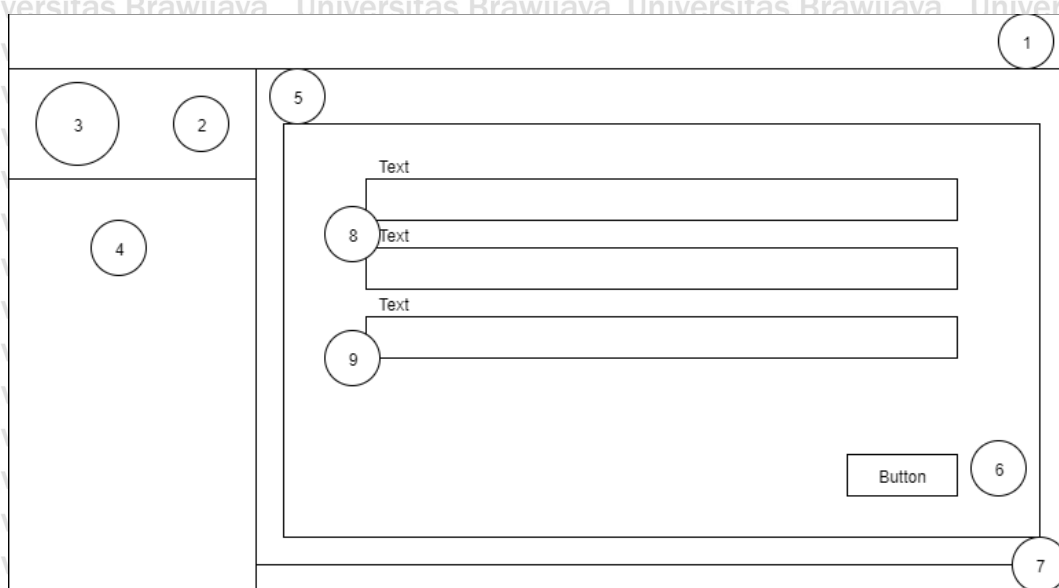
Gambar 5.9 Rancangan Antarmuka Halaman Tambah Riwayat Betes Jensen

Gambar 5.9 merupakan gambaran antarmuka halaman tambah riwayat betes jensen untuk aktor divisi pelayanan. Halaman tambah riwayat betes Jensen bertujuan untuk menambah data pengkajian luka yang lebih mendalam tentang keadaan luka *klien* pada saat itu jugakemudian disimpan dalam dalam *database*. Pada Tabel 5.4 dijelaskan komponen halaman tambah riwayat betes jensen seperti yang terlihat di bawah ini.

Tabel 5.36 Komponen Halaman Tambah Riwayat Betes Jensen

No	Nama Objek	Keterangan
1	Header	Header ditampilkan yang dimana nanti akan berisi <i>user icon</i> dan nama pengguna yang dapat diklik untuk melakukan proses <i>sign out</i> atau melihat profil.
2	Label “nama identitas member”	Merupakan informasi nama pemilik akun pada sistem yang telah terdaftar.
3	Image “logo Pediscare”	Merupakan simbol identitas sistem bahwa sistem tersebut milik tempat praktik mandiri pediscare dan hanya digunakan untuk pegawai yang terlibat dalam tempat tersebut.
4	Sidebar	Merupakan kategori menu pilihan yang terdapat dalam sistem.
5	Label “Ubah Password”	Merupakan informasi nama <i>form</i> yang sedang disinggahi.
6	form	Merupakan jendela penulisan dokumen.
7	Button “submit”	Button yang berfungsi menyimpan data yang sudah dimasukkan.
8	footer	Menampilkan footer dari sistem.
9	Column	Menampilkan penjelasan informasi untuk pengisian beberapa <i>field</i> tertentu.
10	Text field	Merupakan kolom tempat memasukkan informasi berdasarkan nama label.
11	Label	Merupakan informasi nama <i>text field</i> yang nantinya akan dimasukkan.
12	Drop Down	Merupakan tempat pemilihan alah satu menu tertentu yang sudah dijadikan menu pilihan.

5.1.4.2 Perancangan Antarmuka Ubah Password



Gambar 5.10 Rancangan Antarmuka Halaman Ubah Password

Gambar 5.10 adalah susunan antarmuka halaman ubah *password* untuk aktor divisi pelayanan. Halaman mengganti *password* berfungsi untuk mengubah *password* lama aktor menjadi *password* baru untuk meningkatkan keamanan akun. Pada Tabel 5.5 dijelaskan komponen halaman ubah *password* seperti yang terlihat di bawah ini.

Tabel 5.37 Komponen Halaman Ubah Password

No	Nama Objek	Keterangan
1	Header	Header ditampilkan yang dimana nanti akan berisi <i>user icon</i> dan nama pengguna yang dapat diklik untuk melakukan proses <i>sign out</i> atau melihat profil.
2	Label “nama identitas member”	Merupakan informasi nama pemilik akun pada sistem yang telah terdaftar.
3	Image “logo Pediscare”	Merupakan simbol identitas sistem bahwa sistem tersebut milik tempat praktik mandiri pediscare dan hanya digunakan untuk pegawai yang terlibat dalam tempat tersebut.
4	Sidebar	Merupakan kategori menu pilihan yang terdapat dalam sistem.
5	Label “Ubah Password”	Merupakan informasi nama <i>form</i> yang sedang disinggahi.

6	Button “submit”	Button yang berfungsi menyimpan data yang sudah dimasukkan.
7	footer	Menampilkan <i>footer</i> dari sistem.
8	Label	Merupakan informasi nama <i>text field</i> yang nantinya akan dimasukkan.
9	Text field	Merupakan kolom tempat memasukkan informasi berdasarkan nama label

5.1.4.3 Perancangan Antarmuka Delete *Informed Content*

Gambar 5.11 Rancangan Antarmuka Delete *Informed Content*

Gambar 5.11 adalah susunan antarmuka halaman *informed content* untuk aktor divisi pelayanan. Di halaman ini terdapat fungsi *delete* riwayat *informed content* yang teletak di *icon* yang berbentuk sampah. Pada Tabel 5.6 dijelaskan komponen halaman *delete informed content* seperti yang terlihat di bawah ini.

Tabel 5.38 Komponen Halaman Delete *Informed Content*

No	Nama Objek	Keterangan
1	Header	Header ditampilkan yang dimana nanti akan berisi <i>user icon</i> dan nama pengguna yang dapat diklik untuk melakukan proses <i>sign out</i> atau melihat profil.
2	Label “nama identitas member”	Merupakan informasi nama pemilik akun pada sistem yang telah terdaftar.
3	Image “logo Pediscare”	Merupakan simbol identitas sistem bahwa sistem tersebut milik tempat praktik mandiri pediscare dan hanya

		digunakan untuk pegawai yang terlibat dalam tempat tersebut.
4	Sidebar	Merupakan kategori menu pilihan yang terdapat dalam sistem.
5	Label "Form Riwayat Informed Content"	Merupakan informasi nama <i>form</i> yang sedang disinggahi.
6	Tabel daftar riwayat Informed Content	Menampilkan daftar riwayat pengkajian umum beserta terdapat <i>button</i> aksi di dalamnya.
8	footer	Menampilkan <i>footer</i> dari sistem.
9	Button "Edit"	Merupakan <i>button</i> yang digunakan untuk mengedit data <i>form Informed Content</i> .
10	Button"hapus"	Merupakan <i>button</i> yang digunakan untuk menghapus data <i>form informed Content</i> .
11	Button"print"	Merupakan <i>button</i> yang digunakan untuk mencetak <i>form</i> pengkajian umum.

a. Spesifikasi Perangkat Keras

Pembuatan sistem informasi perawatan luka pedis care malang menggunakan perangkat keras, yaitu perangkat laptop dengan spesifikasi yang di jabarkan dalam Tabel 5.39.

Tabel 5.39 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
System Model	HP 14 Notebook PC
Processor	AMD A4-5000 APU with Radeon™ HD Graphics (4CPUs), ~1.5GHz
Memory	2048 MB
Display	AMD Radeon HD 8330
Hard Disk	HGST HTS 545050A7E680 SATA Disk Device
Keyboard	Standard PS/2 Keyboard
Mouse/Touchpad	PS/2 Compatible Mouse

b. Spesifikasi Perangkat Lunak

Dalam pembuatan sistem informasi perawatan luka pedis care malang menggunakan perangkat lunak dengan spesifikasi yang di jabarkan dalam Tabel 5.40.

Tabel 5.40 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
Editor Perancangan	Enterprise Architect Version 12.1, Visual Paradigm 14.0 Community Edition, draw io.
Editor Pemrograman	Sublime Text Version 3.2.1
Framework dan Bahasa Pemrograman	PHP, CSS, JS, CodeIgniter
DBMS	MySQL
Aplikasi Pendukung Sistem	Xampp
Browser	Chrome
Database Server	PHPMyAdmin

c. Spesifikasi Sistem Operasi

Pembuatan sistem informasi perawatan luka di praktik mandiri pedis care malang menggunakan sistem operasi dengan spesifikasi yang di jabarkan dalam Tabel 5.41.

Tabel 5.41 Spesifikasi Operasi

Nama Komponen	Spesifikasi
Sistem Operasi	HP 14 Notebook PC

Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya
Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya
Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya
Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya
Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya
Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya
Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya
ntasi Database rsitas Brawijaya Unive rsitas Brawijaya Universitas Brawijaya Universitas Brawijaya



Gambar 5.12 Physical Data Model

5.1.6 Implementasi Kode

Tabel 5.42 Perintah SQL Membuat Tabel calendar

```
CREATE TABLE `calendar` (
  `id` int(11) NOT NULL,
  `ruangan` varchar(20) NOT NULL,
  `jam` time(6) NOT NULL,
  `title` varchar(126) DEFAULT NULL,
  `description` text,
  `color` varchar(24) DEFAULT NULL,
  `start_date` date DEFAULT NULL,
  `end_date` date DEFAULT NULL,
  `create_at` datetime DEFAULT NULL,
  `create_by` varchar(64) DEFAULT NULL,
  `modified_at` datetime DEFAULT NULL,
  `modified_by` varchar(64) DEFAULT NULL
)
```

5.43 Perintah SQL Membuat Tabel daftar_agama

```
CREATE TABLE `daftar_agama` (
  `no` int(255) NOT NULL,
  `agama` varchar(20) NOT NULL
)
```

Tabel 5.44 Perintah SQL Membuat Tabel daftar_bidang

```
CREATE TABLE `daftar_bidang` (
  `no` int(255) NOT NULL,
  `bidang` varchar(20) NOT NULL
)
```

Tabel 5.45 Perintah SQL Membuat Tabel daftar_dasarpenerimaan

```
CREATE TABLE `daftar_dasarpenerimaan` (
  `no` int(255) NOT NULL,
  `pendidikan` varchar(20) NOT NULL
)
```

Tabel 5.46 Perintah SQL Membuat Tabel daftar_dressing

```
CREATE TABLE `daftar_dressing` (
  `iddaftardressing` int(20) NOT NULL,
  `namadressing` varchar(50) NOT NULL,
  `hargaperawatan` int(50) NOT NULL
)
```

Tabel 5.47 Perintah SQL Membuat Tabel daftar_rumahsakit

```
CREATE TABLE `daftar_rumahsakit` (
  `no` int(8) NOT NULL,
  `rumahsakit` varchar(50) NOT NULL,
  `telpon` varchar(15) NOT NULL
)
```

Tabel 5.48 Perintah SQL Membuat Tabel daftar_status

```
CREATE TABLE `daftar_status` (
  `no` int(255) NOT NULL,
  `statuspegawai` varchar(20) NOT NULL
)
```

Tabel 5.49 Perintah SQL Membuat Tabel dressingklien

```
CREATE TABLE `dressingklien` (
  `iddressingklien` int(11) NOT NULL,
  `iddaftardressing` int(11) NOT NULL,
  `noregister` int(11) NOT NULL,
  `jumlah` int(50) NOT NULL
)
```

Tabel 5.50 Perintah SQL Membuat Tabel lihatjadwal

```
CREATE TABLE `lihatjadwal` (
  `namakl` varchar(60) NOT NULL,
  `ruangan` varchar(10) NOT NULL,
  `jam` time NOT NULL,
  `description` varchar(150) NOT NULL,
  `tanggalschedule` date NOT NULL
)
```

Tabel 5.51 Perintah SQL Membuat Tabel pegawai

```
CREATE TABLE `pegawai` (
  `username` varchar(10) NOT NULL,
  `email` varchar(55) NOT NULL,
  `password` varchar(255) NOT NULL,
  `namalengkap` varchar(255) NOT NULL,
  `namapanggilan` varchar(20) NOT NULL,
  `foto_pegawai` varchar(200) NOT NULL,
  `role` varchar(11) NOT NULL,
  `statuspegawai` varchar(10) NOT NULL,
  `bidang` varchar(20) NOT NULL,
)
```

```

`tgllahir` date NOT NULL,
`tglmasuk` date NOT NULL,
`tglpensiun` varchar(10) NOT NULL,
`tglbebascuti` date NOT NULL,
`sisacuti` int(8) NOT NULL,
`id_pegawai` int(11) NOT NULL
)

```

Tabel 5.52 Perintah SQL Membuat Tabel pendaftaran_pegawai

```

CREATE TABLE `pendaftaran_pegawai` (
  `username` varchar(10) NOT NULL,
  `email` varchar(55) NOT NULL,
  `password` varchar(255) NOT NULL,
  `namalengkap` varchar(255) NOT NULL,
  `namapanggilan` varchar(20) NOT NULL,
  `statuspegawai` varchar(10) NOT NULL,
  `bidang` varchar(20) NOT NULL,
  `tgllahir` date NOT NULL,
  `tglmasuk` date NOT NULL,
  `dasarpenerimaan` varchar(10) NOT NULL,
  `terbaca` tinyint(1) NOT NULL
)

```

Tabel 5.53 Perintah SQL Membuat Tabel profil_diklat

```

CREATE TABLE `profil_diklat` (
  `no` int(255) NOT NULL,
  `username` varchar(10) NOT NULL,
  `namadiklat` varchar(50) DEFAULT NULL,
  `tahun` varchar(4) DEFAULT NULL
)

```

Tabel 5.54 Perintah SQL Membuat Tabel profil_jenjangkarir

```

CREATE TABLE `profil_jenjangkarir` (
  `no` int(255) NOT NULL,
  `username` varchar(10) NOT NULL,
  `jabatan` varchar(30) NOT NULL,
  `tglsk` date NOT NULL,
  `nosk` varchar(20) NOT NULL,
  `berakhirsk` date NOT NULL,

```

```
`keterangan` varchar(50) NOT NULL
)
```

Tabel 5.55 Perintah SQL Membuat Tabel profil_keluarga

```
CREATE TABLE `profil_keluarga` (
  `no` int(255) NOT NULL,
  `username` varchar(10) NOT NULL,
  `status` varchar(20) NOT NULL,
  `nama` varchar(50) NOT NULL,
  `tgllahir` date DEFAULT NULL,
  `usia` int(2) NOT NULL
)
```

Tabel 5.56 Perintah SQL Membuat Tabel profil_organisasi

```
CREATE TABLE `profil_organisasi` (
  `no` int(255) NOT NULL,
  `username` varchar(10) NOT NULL,
  `organisasi` varchar(30) NOT NULL,
  `tahun` varchar(4) NOT NULL
)
```

Tabel 5.57 Perintah SQL Membuat Tabel profil_pegawai

```
CREATE TABLE `profil_pegawai` (
  `username` varchar(10) NOT NULL,
  `jeniskelamin` varchar(9) NOT NULL,
  `dasarpenerimaan` varchar(4) NOT NULL,
  `agama` varchar(10) NOT NULL,
  `sukubangsa` varchar(10) NOT NULL,
  `golongandarah` varchar(1) NOT NULL,
  `tinggibadan` varchar(6) NOT NULL,
  `beratbadan` varchar(6) NOT NULL,
  `hobi` varchar(100) NOT NULL,
  `mottohidup` varchar(100) NOT NULL,
  `noktp` varchar(16) NOT NULL,
  `alamatktp` varchar(100) NOT NULL,
  `alamatsekarang` varchar(100) NOT NULL,
  `email` varchar(50) NOT NULL,
  `status` varchar(10) NOT NULL,
  `nokk` varchar(17) NOT NULL,
  `bpjs` varchar(11) NOT NULL,
)
```

```

`nosepatu` varchar(2) NOT NULL,
`ukuranseragam` varchar(5) NOT NULL,
`namaibukandung` varchar(30) NOT NULL,
`namaayahkandung` varchar(30) NOT NULL,
`keahliankhusus` varchar(100) NOT NULL,
`bahasadikuasai` varchar(50) NOT NULL,
`hasilmedical` varchar(50) NOT NULL
)

```

Tabel 5.58 Perintah SQL Membuat Tabel profil_pendidikan

```

CREATE TABLE `profil_pendidikan` (
  `no` int(255) NOT NULL,
  `username` varchar(10) NOT NULL,
  `jenis` varchar(20) NOT NULL,
  `keterangan` varchar(50) NOT NULL,
  `jurusan` varchar(20) NOT NULL,
  `thnlulus` varchar(4) DEFAULT NULL
)

```

Tabel 5.59 Perintah SQL Membuat Tabel profil_pengalamankerja

```

CREATE TABLE `profil_pengalamankerja` (
  `no` int(255) NOT NULL,
  `username` varchar(10) NOT NULL,
  `pengalamankerja` varchar(100) NOT NULL,
  `tahun` varchar(4) NOT NULL
)

```

Tabel 5.60 Perintah SQL Membuat Tabel profil_penghargaan

```

CREATE TABLE `profil_penghargaan` (
  `no` int(255) NOT NULL,
  `username` varchar(10) NOT NULL,
  `penghargaan` varchar(30) NOT NULL,
  `tahun` varchar(4) NOT NULL
)

```

Tabel 5.61 Perintah SQL Membuat Tabel profil_prestasi

```

CREATE TABLE `profil_prestasi` (
  `no` int(255) NOT NULL,
  `username` varchar(10) NOT NULL,
  `prestasi` varchar(30) NOT NULL,
  `tahun` varchar(4) NOT NULL
)

```

Tabel 5.62 Perintah SQL Membuat Tabel profil_sanksi

```
CREATE TABLE `profil_sanksi` (
  `no` int(255) NOT NULL,
  `username` varchar(10) NOT NULL,
  `tglsanksi` date NOT NULL,
  `keterangan` varchar(50) NOT NULL,
  `berakhir` date NOT NULL,
  `status` varchar(20) NOT NULL
)
```

Tabel 5.63 Perintah SQL Membuat Tabel riwayatbetesjensen

```
CREATE TABLE `riwayatbetesjensen` (
  `id` int(11) NOT NULL,
  `noregister` varchar(20) NOT NULL,
  `namakl` varchar(30) NOT NULL,
  `ukuranluka` varchar(5) NOT NULL,
  `kedalamanluka` varchar(5) NOT NULL,
  `tipeluka` varchar(5) NOT NULL,
  `goa` varchar(5) NOT NULL,
  `tipeeksudat` varchar(5) NOT NULL,
  `warnakulit` varchar(5) NOT NULL,
  `jaringanedema` varchar(5) NOT NULL,
  `jaringangranulasi` varchar(5) NOT NULL,
  `epitelisasi` varchar(5) NOT NULL,
  `skortotal` varchar(5) NOT NULL,
  `petugasbersangkutan` varchar(30) NOT NULL,
  `jumlaheksudat` varchar(50) NOT NULL,
  `tanggalbj` date NOT NULL,
  `pengartianskor` varchar(50) NOT NULL
)
```

5.64 Perintah SQL Membuat Tabel riwayatidentitasklien

```
CREATE TABLE `riwayatidentitasklien` (
  `id` int(11) NOT NULL,
  `noregister` varchar(30) NOT NULL,
  `tanggalahirkl` date NOT NULL,
  `usiakl` varchar(5) NOT NULL,
  `jeniskelaminkl` varchar(13) NOT NULL,

```

```

alamatkl` varchar(50) NOT NULL,
`telponkl` varchar(13) NOT NULL,
`agama` varchar(25) NOT NULL,
`penjamin` varchar(20) NOT NULL,
`rujukanrekomendasi` varchar(30) NOT NULL,
`namakl` varchar(25) NOT NULL,
`status` varchar(15) NOT NULL,
`kebangsaan` varchar(15) NOT NULL,
`pekerjaan` varchar(30) NOT NULL
)

```

Tabel 5.65 Perintah SQL Membuat Tabel riwayatinformendcontent

```

CREATE TABLE `riwayatinformendcontent` (
  `id` int(11) NOT NULL,
  `noregister` varchar(20) NOT NULL,
  `namakl` varchar(30) NOT NULL,
  `Tanggallahirk1` date NOT NULL,
  `alamatkl` varchar(100) NOT NULL,
  `telponkl` int(15) NOT NULL,
  `tanggal` date NOT NULL,
  `namawali` varchar(25) NOT NULL,
  `alamatwali` varchar(50) NOT NULL,
  `telponwali` varchar(15) NOT NULL,
  `hubungandengankl` varchar(20) NOT NULL,
  `diagnosakl` varchar(70) NOT NULL,
  `tanggallahirwali` date NOT NULL,
  `upload` varchar(100) NOT NULL
)

```

Tabel 5.66 Perintah SQL Membuat Tabel riwayatkunjungan

```

CREATE TABLE `riwayatkunjungan` (
  `id` int(11) NOT NULL,
  `noregister` varchar(30) NOT NULL,
  `namakl` varchar(30) NOT NULL,
  `tanggalkunjungan` date NOT NULL,
  `inisial` varchar(50) NOT NULL,
  `komentarkode` varchar(100) NOT NULL,
  `petugasbersangkutan` varchar(30) NOT NULL,
  `perihalkunjungan` varchar(100) NOT NULL,
  `jam` time NOT NULL
)

```

Tabel 5.67 Perintah SQL Membuat Tabel riwayatpembayaran

```
CREATE TABLE `riwayatpembayaran` (
  `noregister` int(11) NOT NULL,
  `namakl` varchar(30) NOT NULL,
  `jenistindakan` varchar(20) NOT NULL,
  `dressing` varchar(225) NOT NULL,
  `hargadressing` int(11) NOT NULL,
  `hargatindakan` int(11) NOT NULL,
  `total` int(11) NOT NULL
)
```

Tabel 5.68 Perintah SQL Membuat Tabel riwayatpengkajianluka

```
CREATE TABLE `riwayatpengkajianluka` (
  `namakl` varchar(30) NOT NULL,
  `faktorpenghambatsembuh` varchar(100) NOT NULL,
  `alergisensifitas` varchar(100) NOT NULL,
  `rujukke` varchar(50) NOT NULL,
  `alatygdiperlukan` varchar(100) NOT NULL,
  `tipeluka` varchar(50) NOT NULL,
  `petugasbersangkutan` varchar(30) NOT NULL,
  `noregister` varchar(20) NOT NULL,
  `id` int(11) NOT NULL
)
```

Tabel 5.69 Perintah SQL Membuat Tabel riwayatpengkajianumum

```
CREATE TABLE `riwayatpengkajianumum` (
  `id` int(11) NOT NULL,
  `noregister` varchar(30) NOT NULL,
  `doktermerawatmerujuk` varchar(30) NOT NULL,
  `dokterkonsultan` varchar(30) NOT NULL,
  `diagnosa` varchar(100) NOT NULL,
  `tanggaldirawat` date NOT NULL,
  `adviceterapi` varchar(50) NOT NULL,
  `petugasbersangkutan` varchar(30) NOT NULL,
  `pernahdirawat` varchar(5) NOT NULL,
  `pernahoperasi` varchar(5) NOT NULL,
  `penyakitkeluarga` varchar(50) NOT NULL,
  `sistempencernaan` varchar(70) NOT NULL,
)
```

```

`sistemrespirasi` varchar(70) NOT NULL,
`sistemkardiovaskuler` varchar(70) NOT NULL,
`keluhanutama` varchar(100) NOT NULL,
`keluhanumum` varchar(100) NOT NULL,
`td` varchar(20) NOT NULL,
`rr` varchar(20) NOT NULL,
`n` varchar(20) NOT NULL,
`s` varchar(5) NOT NULL,
`tinggibadan` varchar(5) NOT NULL,
`beratbadan` varchar(3) NOT NULL,
`penyakitlain` varchar(30) NOT NULL
)

```

Tabel 5.70 Perintah SQL Membuat Tabel riwayatpenolakantindakan

```

CREATE TABLE `riwayatpenolakantindakan` (
  `id` int(11) NOT NULL,
  `tanggalpt` date NOT NULL,
  `namapj` varchar(30) NOT NULL,
  `buktidiri` varchar(25) NOT NULL,
  `penolakantindakan` varchar(50) NOT NULL,
  `terhadap` varchar(65) NOT NULL,
  `dirawatdi` varchar(30) NOT NULL,
  `petugasbersangkutan` varchar(30) NOT NULL,
  `noregister` varchar(30) NOT NULL
)

```

Tabel 5.71 Perintah SQL Membuat Tabel riwayatperkembanganklien

```

CREATE TABLE `riwayatperkembangankllien` (
  `id` int(11) NOT NULL,
  `noregister` varchar(30) NOT NULL,
  `tanggalperawatan` date NOT NULL,
  `operator` varchar(30) NOT NULL,
  `hasiltes` varchar(50) NOT NULL,
  `penyulit` varchar(15) NOT NULL,
  `penyulitlain` varchar(50) NOT NULL,
  `gcs` varchar(70) NOT NULL
)

```

Tabel 5.72 Perintah SQL Membuat Tabel riwayatrujukan

```

CREATE TABLE `riwayatrujukan` (

```

```

`id` int(11) NOT NULL,
`usiakl` varchar(5) NOT NULL,
`namapj` varchar(25) NOT NULL,
`alamatpj` varchar(30) NOT NULL,
`diagnosakeperawatan` varchar(100) NOT NULL,
`dirujukke` varchar(40) NOT NULL,
`kondisisaatdirujuk` varchar(80) NOT NULL,
`tindakanygdilakukan` varchar(50) NOT NULL,
`petugasbersangkutan` varchar(30) NOT NULL,
`noregister` varchar(30) NOT NULL
)

```

Tabel 5.73 Perintah SQL Membuat Tabel riwayatampilanklinisluka

```

CREATE TABLE `riwayatampilanklinisluka` (
`id` int(11) NOT NULL,
`noregister` varchar(30) NOT NULL,
`noluka` varchar(25) NOT NULL,
`tanggalpengkajian` date NOT NULL,
`analgetikpredressing` varchar(50) NOT NULL,
`panjangluka` varchar(5) NOT NULL,
`lebarluka` varchar(5) NOT NULL,
`kedalamanluka` varchar(5) NOT NULL,
`datemininglubanggoa` varchar(70) NOT NULL,
`tracking` varchar(70) NOT NULL,
`tepiluka` varchar(70) NOT NULL,
`nekrotik` varchar(5) NOT NULL,
`slough` varchar(5) NOT NULL,
`granulasi` varchar(5) NOT NULL,
`epitelisasi` varchar(5) NOT NULL,
`hypergranulasi` varchar(5) NOT NULL,
`infeksi` varchar(5) NOT NULL,
`tampilanwarnalainnya` varchar(50) NOT NULL,
`serous` varchar(70) NOT NULL,
`blood` varchar(70) NOT NULL,
`haemoserous` varchar(70) NOT NULL,
`purulent` varchar(70) NOT NULL,
`haempurulent` varchar(70) NOT NULL,
`odour` varchar(70) NOT NULL,
`jumaheksudat` varchar(100) NOT NULL,

```

```
`kulitsekitarluka` varchar(10) NOT NULL,
`rasasakit` varchar(15) NOT NULL,
`frekuensi` varchar(10) NOT NULL,
`stadiumluka` varchar(5) NOT NULL,
`petugasbersangkutan` varchar(30) NOT NULL,
`upload1` varchar(255) NOT NULL,
`upload2` varchar(255) NOT NULL
)
```

Tabel 5.74 Perintah SQL Membuat Tabel riwayatindakan

```
CREATE TABLE `riwayatindakan` (
`namakl` varchar(30) NOT NULL,
`tanggalindakan` date NOT NULL,
`noluka` varchar(10) NOT NULL,
`tindakandressing` varchar(70) NOT NULL,
`frekuensi` varchar(70) NOT NULL,
`evaluasi` varchar(100) NOT NULL,
`jenistindakan` varchar(20) NOT NULL,
`petugasbersangkutan` varchar(30) NOT NULL,
`id` int(11) NOT NULL,
`noregister` varchar(20) NOT NULL
)
```

5.1.6.1 Implementasi Kode Tambah Riwayat Betes Jensen

Tabel 5.43 menjelaskan implementasi kode tambah riwayat betes jensen, dimana pada klas *controller* terdapat fungsi `do_riwayatbetesjensen()` yang berguna sebagai jembatan antara klas *view* dan klas *model*.

Tabel 5.75 Implementasi Tambah Riwayat Betes Jensen

No	Pelayanan_controller.php
1	<code>function do_riwayatbetesjensen(){</code>
2	<code> \$tanggalbj = \$this->input->post('tanggalbj');</code>
3	<code> \$noregister = \$this->input->post("kl");</code>
4	<code> \$ukuranluka = (int) \$this->input-</code>
5	<code> >post('ukuranluka');</code>
6	<code> \$kedalamanluka = (int) \$this->input-</code>
7	<code> >post('kedalamanluka');</code>
8	<code> \$tipeluka = (int) \$this->input-</code>
9	<code> >post('tipeluka');</code>
10	<code> \$goa = (int) \$this->input->post('goa');</code>
11	<code> \$tipeeksudat = (int) \$this->input-</code>
12	<code> >post('tipeeksudat');</code>
13	<code> \$warnakulit = (int) \$this->input-</code>
14	<code> >post('warnakulit');</code>

```

15      $jaringanedema = (int) $this->input-
16      >post('jaringanedema');
17      $jaringangranulasi = (int) $this->input-
18      >post('jaringangranulasi');
19      $epitelisasi = (int) $this->input-
20      >post('epitelisasi');
21      $skorttotal = $this->input->post('skorttotal');
22      $pengartianskor = $this->input-
23      >post('pengartianskor');
24      $petugasbersangkutan = $this->input-
25      >post('petugasbersangkutan');
26      $jumlaheksudat = $this->input-
27      >post('jumlaheksudat');
28
29      $this->load->model('daftar_klien');
30
31      $data = array(
32          'tanggalbj' => $tanggalbj,
33          'noregister' => $noregister,
34          'ukuranluka' => (int) $ukuranluka,
35          'kedaIamanluka' => (int) $kedaIamanluka,
36          'tipeluka' => (int) $tipeluka,
37          'goa' => (int) $goa,
38          'tipeeksudat' => (int) $tipeeksudat,
39          'warnakulit' => (int) $warnakulit,
40          'jaringanedema' => (int) $jaringanedema,
41          'jaringangranulasi' => (int)
42          $jaringangranulasi,
43          'epitelisasi' => (int) $epitelisasi,
44          'skorttotal' => (int) $skorttotal,
45          'pengartianskor' => $pengartianskor,
46          'petugasbersangkutan' =>
47          $petugasbersangkutan,
48          'jumlaheksudat' => (int) $jumlaheksudat,
49      );
50
51      $error = "";
52      (int) $data['skorttotal'] = ((int)
53      (($data['ukuranluka'] + $data['kedaIamanluka'] +
54      $data['tipeluka'] + $data['goa'] + $data['tipeeksudat'] +
55      $data['warnakulit'] + $data['jaringanedema'] +
56      $data['jaringangranulasi'] + $data['epitelisasi'] +
57      $data['jumlaheksudat'])));
58      if ($data['skorttotal'] >= 0 &&
59      $data['skorttotal'] <= 15) {
60          $data['pengartianskor'] = "Jaringan
61      Sehat";
62      } else if ($data['skorttotal'] >= 16 &&
63      $data['skorttotal'] <= 30) {
64          $data['pengartianskor'] = "Regenerasi
65      Luka";
66      } else if ($data['skorttotal'] >= 31 &&
67      $data['skorttotal'] <= 55) {
68          $data['pengartianskor'] = "Degenerasi
69      Luka";
70      }
71      $berhasil = $this->daftar_klien-
72      >set_biodata($data, 'riwayatbetesjensen');
73      $this->index();

```

```

74         }
75
76         function do_riwayatampilanklinisluka()
77         {
78             $noregister = $this->input->post('kl');
79             $noluka = $this->input->post('noluka');
80             $tanggalpengkajian = $this->input-
81 >post('tanggalpengkajian');
82             $analgetikpredressing = $this->input-
83 >post('analgetikpredressing');
84             $panjangluka = $this->input-
85 >post('panjangluka');
86             $lebarluka = $this->input->post('lebarluka');
87             $kedatamanluka = $this->input-
88 >post('kedatamanluka');
89             $datemininglubanggoa = $this->input-
90 >post('datemininglubanggoa');
91             $tracking = $this->input->post('tracking');
92             $tepiluka = $this->input->post('tepiluka');
93             $nekrotik = $this->input->post('nekrotik');
94             $slough = $this->input->post('slough');
95             $granulasi = $this->input->post('granulasi');
96             $epitelisasi = $this->input-
97 >post('epitelisasi');
98             $hypergranulasi = $this->input-
99 >post('hypergranulasi');
100             $infeksi = $this->input->post('infeksi');
101             $tampilanwarnalainnya = $this->input-
102 >post('tampilanwarnalainnya');
103             $serous = $this->input->post('serous');
104             $blood = $this->input->post('blood');
105             $haemoserous = $this->input-
106 >post('haemoserous');
107             $purulent = $this->input->post('purulent');
108             $haempurulent = $this->input-
109 >post('haempurulent');
110             $odour = $this->input->post('odour');
111             $jumlaheksudat = $this->input-
112 >post('jumlaheksudat');
113             $kulitsekitarluka = $this->input-
114 >post('kulitsekitarluka');
115             $rasasakit = $this->input->post('rasasakit');
116             $frekuensi = $this->input->post('frekuensi');
117             $stadiumluka = $this->input-
118 >post('stadiumluka');
119             $petugasbersangkutan = $this->input-
120 >post('petugasbersangkutan');
121             $upload1 = "";
122
123             if (!empty($_FILES["upload1"]["type"])) {
124                 $sourcePath =
125 $_FILES['upload1']['tmp_name'];
126                 $targetPath = 'assets/uploads/';
127                 $_FILES['upload1']['name'];
128                 if (move_uploaded_file($sourcePath,
129 $targetPath)) {
130                     $upload1 =
131 $_FILES['upload1']['name'];
132                 }

```

```

133     }
134
135     $upload2 = "";
136
137     if (!empty($_FILES["upload2"]["type"])) {
138         $sourcePath =
139         $_FILES['upload2']['tmp_name'];
140         $targetPath = 'assets/uploads/';
141         $_FILES['upload2']['name'];
142         if (move_uploaded_file($sourcePath,
143         $targetPath)) {
144             $upload2 =
145             $_FILES['upload2']['name'];
146         }
147     }
148
149     $this->load->model('daftar_klien');
150
151     $data = array(
152         'noregister' => $noregister,
153         'noluka' => $noluka,
154         'tanggalpengkajian' =>
155         $tanggalpengkajian,
156         'analgetikpredressing' =>
157         $analgetikpredressing,
158         'panjangluka' => $panjangluka,
159         'lebarluka' => $lebarluka,
160         'kedalamanluka' => $kedalamanluka,
161         'datemininglubanggoa' =>
162         $datemininglubanggoa,
163         'tracking' => $tracking,
164         'tepiluka' => $tepiluka,
165         'nekrotik' => $nekrotik,
166         'slough' => $slough,
167         'granulasi' => $granulasi,
168         'epitelisasi' => $epitelisasi,
169         'hypergranulasi' => $hypergranulasi,
170         'infeksi' => $infeksi,
171         'tampilanwarnalainnya' =>
172         $tampilanwarnalainnya,
173         'serous' => $serous,
174         'blood' => $blood,
175         'haemoserous' => $haemoserous,
176         'purulent' => $purulent,
177         'haemopurulent' => $haemopurulent,
178         'odour' => $odour,
179         'jumlahheksudat' => $jumlahheksudat,
180         'kulitsekitarluka' => $kulitsekitarluka,
181         'rasasakit' => $rasasakit,
182         'frekuensi' => $frekuensi,
183         'stadiumluka' => $stadiumluka,
184         'upload1' => $upload1,
185         'upload2' => $upload2,
186         'petugasbersangkutan' =>
187         $petugasbersangkutan,
188     );
189
190     $error = "";
191

```

192	\$berhasil = \$this->daftar_klien-
193	>set_biodata(\$data, 'riwayatampilanklinisluka');
194	\$this->index();
195	}

5.1.6.2 Implementasi Kode Ubah Password

Tabel 5.44 menjelaskan implementasi kode ubah *password*, dimana pada kelas *controller* terdapat fungsi `do_changepassword()` yang berguna sebagai jembatan antara klas *view* dan klas *model*.

Tabel 5.76 Implementasi Ubah Password

No	Pelayanan_controller.php
1	function do_changepassword(){
2	\$user = \$this->pegawai->get_user(\$this->session-
3	>userdata('username'));
4	\$passasli = \$user->password;
5	
6	\$passlama = \$this->input->post('passlama');
7	\$passlama = hash('sha256', sha1(\$passlama));
8	\$passbaru1 = \$this->input->post('passbaru1');
9	\$passbaru1 = hash('sha256', sha1(\$passbaru1));
10	\$passbaru2 = \$this->input->post('passbaru2');
11	\$passbaru2 = hash('sha256', sha1(\$passbaru2));
12	
13	if (\$passasli == \$passlama && \$passbaru1 ==
14	\$passbaru2) {
15	\$this->pegawai-
16	>change_password(\$passbaru1);
17	\$this->session->set_flashdata('ubahpass',
18	' <div style="max-width:50%; text-align:center; margin-
19	left:auto; margin-right:auto;"><h5
20	style="color:green;">Ubah Password Berhasil!!!</h5></div>');
21	
22	redirect('Pelayanan_controller/password');
23	} else {
24	\$this->session->set_flashdata('ubahpass',
25	' <div style="max-width:50%; text-align:center; margin-
26	left:auto; margin-right:auto;"><h5
27	style="color:red;">Ubah Password Gagal!!!</h5></div>');//6
28	redirect('Pelayanan_controller/password');
29	}
30	}

5.1.6.3 Implementasi Hapus Informed Content

Tabel 5.45 menjelaskan implementasi kode tambah *informed content*, dimana pada klas *controller* terdapat fungsi `do_deletariwayatinformcontent()` yang berguna sebagai jembatan antara klas *view* dan kelas *model*.

Tabel 5.77 Implementasi Hapus Informed Content

No	Pelayanan_controller.php
1	function do_deletariwayatinformcontent(){
2	\$this->load->model('riwayatinformcontent');

```

3      $no = $this->uri->segment(3);
4      $this->riwayatinformcontent->delete_data($no);
5
6      redirect('Pelayanan_controller/riwayatinformcontent')
7  ;
8  }

```

5.1.7 Implementasi Antarmuka

Tahap ini diperoleh dari hasil perancangan antarmuka yang diimplementasikan dengan cara visual. Penerapan antarmuka di sistem ini tergantung pada susunan yang dirancang pada tahapan rancangan antarmuka.

5.1.7.1 Antarmuka Tambah Riwayat Betes Jensen

Berdasarkan penerapan dari perancangan antarmuka yang sudah dibuat sebelumnya, dihasilkan implementasi antarmuka pada modul “Tambah Riwayat Betes Jensen” yang terdiri dari beberapa *text field* yang harus diisi, *button choose file* untuk *upload file* gambar dan *button submit* untuk menyimpan data ke *database*.

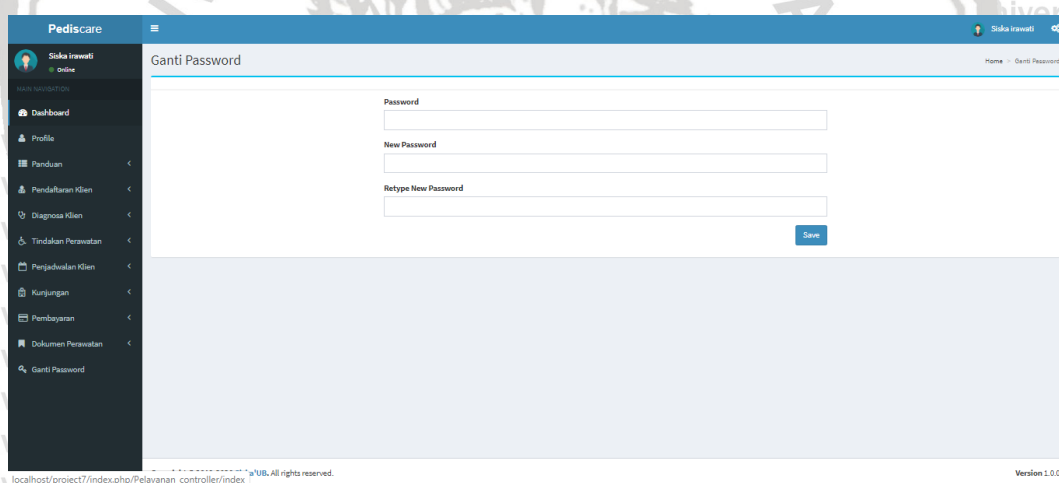
No	Items	Pengkajian
1	Ukuran Luka	1 = P X L < 4 cm 2 = P X L 4 < 16 cm 3 = P X L 16 < 36 cm 4 = P X L 36 < 80 cm 5 = P X L > 80 cm
2	Kedalaman	1 = Tidak ada erythema pada kulit sehat. 2 = Kehilangan kulit "partial thickness" melibatkan epidermis dan atau dermis. 3 = Kehilangan kulit "full thickness" melibatkan kerusakan / nekrosis.

	4 = Tidak jelas akibat nekrosis. 5 = Kehilangan Kulit "Full thickness dengan kerusakan yang luas, jaringan nekrosis atau kerusakan otot, tulang atau struktur yang mendukung.
3 Tepi Luka	1 = Samar, tidak terlihat jelas. 2 = Batas tepi terlihat, menyatu dengan dasar luka. 3 = Jelas, tidak menyatu dengan dasar luka. 4 = Jelas, tidak menyatu dengan dasar luka, tebal. 5 = Jelas, fibrotic, parut tebal/ hiperkeratonic.
4 GDA	1 = Tidak ada. 2 = GDA < 2 cm di area masapun. 3 = GDA 2-4 cm > 50 % pinggir luka. 4 = GDA 2-4 cm > 50 % pinggir luka. 5 = GDA > 4 cm di area masapun.
5 Tipe Eksudat	1 = Tidak ada. 2 = Bloody. 3 = Serosanguineous. 4 = Serosus. 5 = Purulent.
6 Jumlah Eksudat	1 = Kering. 2 = Moist. 3 = Sedikit. 4 = Sedang. 5 = Banyak.
7 Warna Kulit Sekitar Luka	1 = Pink atau normal. 2 = Merah terang ketika ditekan. 3 = Putih atau pucat atau hipopigmentasi. 4 = Merah gelap / abu-abu. 5 = Hitam atau hyperpigmentasi.
8 Jaringan Yang Edema	1 = No swelling atau edema. 2 = Non pitting edema kurang dari 4 cm disekitar luka. 3 = Non pitting edema kurang dari 4 cm disekitar luka. 4 = Pitting edema kurang dari 4 cm disekitar luka. 5 = Kneptasi atau pitting edema > 4 cm.

Gambar 5.13 Implementasi Antarmuka Tambah Riwayat Betes Jensen

5.1.7.2 Antarmuka Ubah *Password*

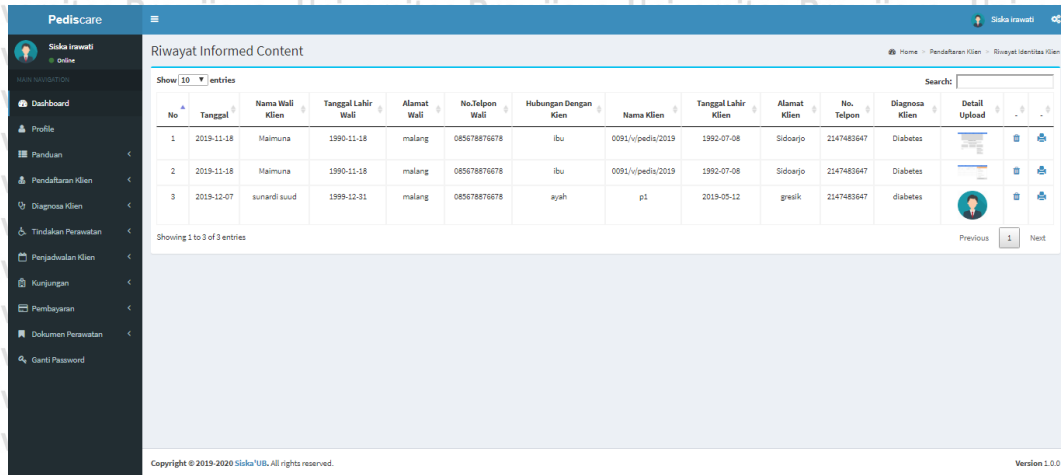
Berikut hasil penerapan dari perancangan dan implementasi antarmuka pada modul “Ubah *Password*” yang terdiri dari beberapa *text field* antara lain *text field password*, *new password* dan *retype new password*.



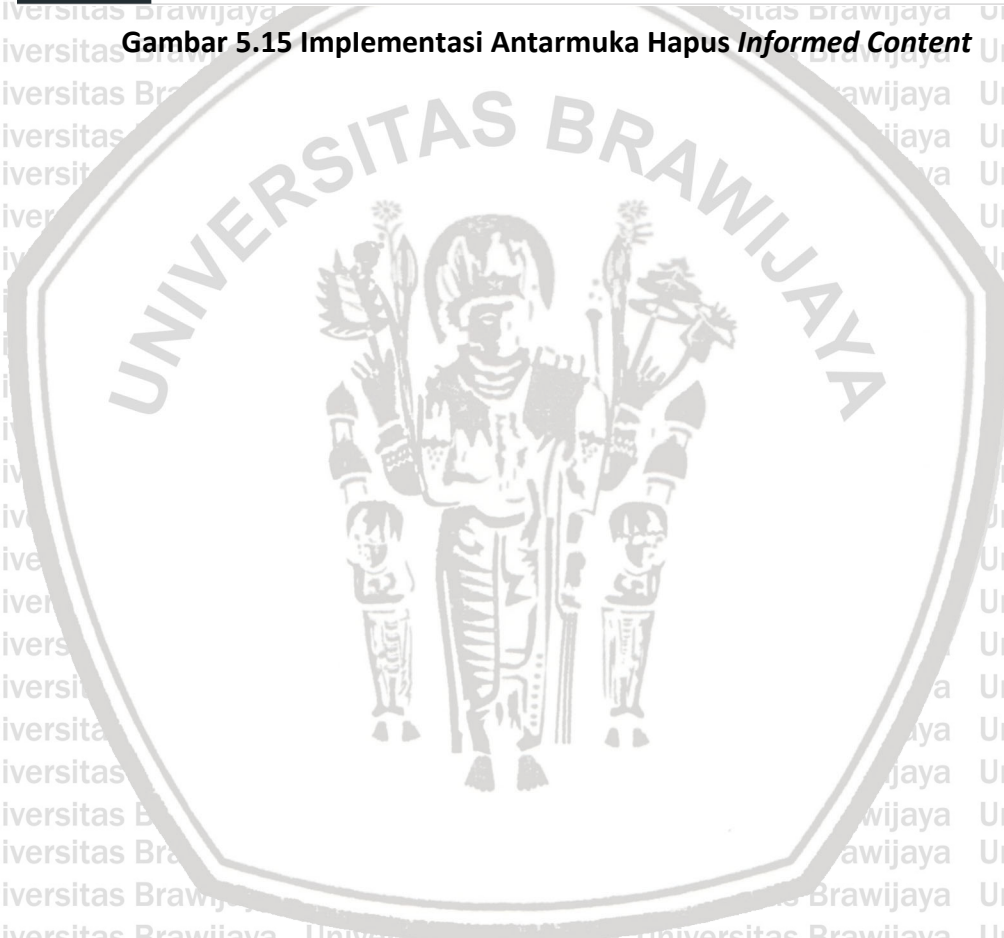
Gambar 5.14 Implementasi Antarmuka Ubah *Password*

5.1.7.3 Antarmuka Hapus *Informed Content*

Berikut hasil penerapan dari perancangan dan implementasi antarmuka pada modul “Hapus *Informed Content*” yang terdiri dari beberapa *list data informed content* yang sudah pernah disimpan sebelumnya, *button* dengan *icon edit*, *button* dengan *icon delete* dan *button* dengan *icon cetak*.



Gambar 5.15 Implementasi Antarmuka Hapus *Informed Content*



BAB 6 PENGUJIAN DAN ANALISIS

6.1 Pengujian Fungsional

Pada tahap ini pengujian perangkat lunak mempunyai peran yang sangat penting untuk menjamin kualitasnya. Pengujian dilakukan untuk mengurangi kesalahan sedini mungkin serta meningkatkan kualitas dan mempertahankan kepuasan pengguna. Pengujian ini terdiri dari beberapa tingkat pengujian yang dilakukan pada penelitian ini, yaitu tingkat unit, integrasi dan validasi. Penelitian ini juga menguji tingkat *compatibility* pada sistem.

6.1.1 Pengujian Unit

Pengujian unit dilakukan dengan cara melihat ke dalam modul dengan meneliti *pseudocode* program, kemudian akan dilakukan analisis apakah masukan dan keluaran dari unit program telah tepat dan sesuai harapan. *Scenario* pengujian unit yang dilakukan pada laporan penelitian ini mengambil 3 sampel unit yaitu `do_riwayatbetesjensen()`, `do_changepassword()` dan `deletainformedcontent()`.

6.1.1.1 Pengujian Fungsi `do_riwayatbetesjensen`

Fungsi `do_riwayatbetesjensen` merupakan fungsi yang digunakan oleh aktor divisi pelayanan untuk menambah data baru mengenai spesifik data luka klien yang dirawat di praktik mandiri pedis care malang. Tabel 6.1 akan menunjukkan fungsi `do_riwayatbetesjensen` beserta *node flow graph*nya.

Tabel 6.1 Node Fungsi `do_riwayatbetesjensen`

Kode	Node
Mulai method <code>do_riwayatbetesjensen()</code>	1
Input data variabel tanggalbj	
Input data variabel noregister	
Input data variabel ukuranluka	
Input data variabel kedalamanluka	
Input data variabel tipeluka	
Input data variabel goa	
Input data variabel tipeeksudat	
Input data variable warnakulit	
Input data variable jaringanedema	
Input data variable jaringangranulasi	
Input data variable epitelisasi	
Input data variable skortotal	
Input data variable pengartianskor	
Input data variable petugasbersangkutan	
Input data variable jumlaheksudat	
Mengambil class model daftar_klien	2
Inisialisasi variable array data	3
Menyimpan data dari variable tanggalbj ke kolom table database	4
Menyimpan data dari variable noregister ke kolom table database	
Menyimpan data dari variable ukuranluka ke kolom table database	

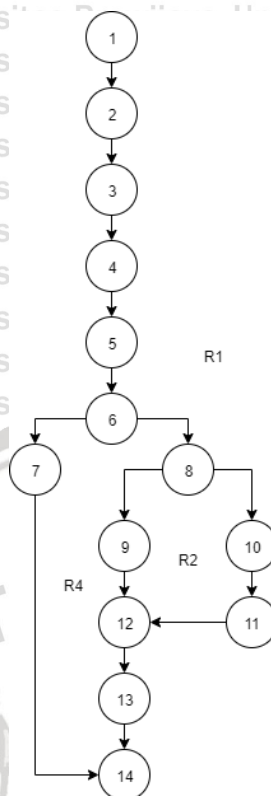
```

Menyimpan data dari variable kedaTamanluka ke kolom table
database
Menyimpan data dari variable tipeluka ke kolom table
database
Menyimpan data dari variable goa ke kolom table database
Menyimpan data dari variable tipeeksudat ke kolom table
database
Menyimpan data dari variable warnakulit ke kolom table
database
Menyimpan data dari variable jaringanedema ke kolom table
database
Menyimpan data dari variable jaringangranulasi ke kolom
table database
Menyimpan data dari variable epitelisasi ke kolom table
database
Menyimpan data dari variable skorttotal ke kolom table
database
Menyimpan data dari variable pengartianskor ke kolom table
database
Menyimpan data dari variable petugasbersangkutan ke kolom
table database
Menyimpan data dari variable jumlaheksudat ke kolom table
database
Inisialisasi variable data dengan index skorttotal dari
penjumlahan variable data dengan index ukuranluka,
kedaTamanluka, tipeluka, goa, tipeeksudat, warnakulit,
jaringanedema, epitelisasi, jumlaheksudat
Jika nilai variable data dengan index skorttotal <=15
maka inisialisasi variable data dengan index
pengartianskor = "Jaringan Sehat"
Jika nilai variable data dengan index skorttotal <= 30
maka inisialisasi variable data dengan index
pengartianskor = "Regenerasi Luka"
Jika nilai variable data dengan index skorttotal <= 55
maka inisialisasi variable data dengan index
pengartianskor = "Degenerasi Luka"
end if
Inisialisasi variable berhasilI memanggil method
set_biodata dengan parameter variable data dan nama Tabel
riwayatbetesjensen
Mengalihkan ke halaman riwayatbetesjensen
selesai

```

5
6
7
8
9
10
11
12
13
14

Dalam pembentukan node fungsi `do_riwayatbetesjensen` terdapat 14 node yang terbentuk. Setelah ditemukan node yang terbentuk, langkah selanjutnya adalah membuat *flow graph* dari fungsi `do_riwayatbetesjensen` berdasarkan node-node yang telah ditentukan. Di bawah ini Gambar 6.1 merupakan *Flow graph* dari fungsi `do_riwayatbetesjensen`.



Gambar 6.1 Flow Graph Fungsi do_rwayatbetesjensen

Berdasarkan *flow graph* fungsi *do_rwayatbetesjensen* dalam Gambar 6.1 didapatkan nilai *cyclomatic complexity* $V(G)$ yang dihitung pada perhitungan berikut ini.

$$V(G) = 3$$

$$V(G) = E - N + 2$$

$$= 13 - 12 + 2$$

$$= 3$$

$$V(G) = P + 1$$

$$= 2 + 1$$

$$= 3$$

Dari perhitungan *cyclomatic complexity* didapatkan 3 jalur independen, diantaranya :

- Jalur 1 = 1-2-3-4-5-6-7-12-13-14
- Jalur 2 = 1-2-3-4-5-6-8-9-12-13-14
- Jalur 3 = 1-2-3-4-5-6-8-10-11-12-13-14

Dari jalur independen yang sudah dibuat selanjutnya dilakukan pengujian *test case* atau kasus uji di masing-masing jalurnya. Tabel 6.2 di bawah merupakan *test case* pada tiap-tiap jalur independen.

Tabel 6.2 Tase Case Fungsi do_rwayatbetesjensen()

No	Jalur	Perosedur Uji	Ekspektasi Pengujian	Hasil pengujian	Status
1	1-2-3-4-5-6-7-12-13-14	Menguji method do_rwayatbetesjensen dengan memasukkan semua data yang dengan minimal skor 0 dan maksimal skor 15	Data akan tersimpan ke <i>database</i> kemudian akan dialihkan ke halaman <i>dashboard</i>	Data berhasil tersimpan di <i>database</i> kemudian dialihkan ke halaman <i>dashboard</i>	Valid
2	1-2-3-4-5-6-8-9-12-13-14	Menguji method do_rwayatbetesjensen dengan memasukkan semua data yang dengan minimal skor 16 dan maksimal skor 30	Data akan tersimpan ke dalam <i>database</i> dan akan dialihkan ke halaman <i>dashboard</i>	Data berhasil tersimpan di <i>database</i> dan dialihkan ke halaman <i>dashboard</i>	Valid
3	1-2-3-4-5-6-8-10-11-12-13-14	Menguji method do_rwayatbetesjensen dengan memasukkan semua data yang dengan minimal skor 31 dan maksimal skor 55	Data akan tersimpan ke dalam <i>database</i> dan akan dialihkan ke halaman <i>dashboard</i>	Data berhasil tersimpan di <i>database</i> dan dialihkan ke halaman <i>dashboard</i>	Valid

6.1.1.2 Pengujian Fungsi do_changepassword()

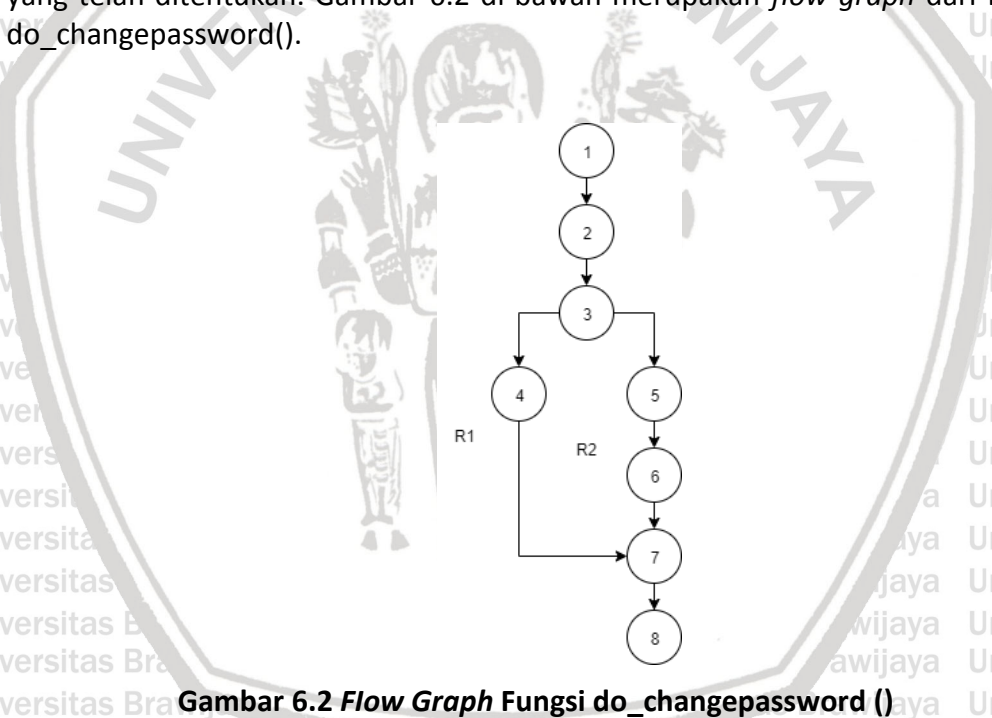
Fungsi do_changepassword merupakan fungsi yang memungkinkan pengguna untuk mengubah *password* lama menjadi *password* baru. Tabel 6.3 akan menunjukkan fungsi do_changepassword beserta dengan node *flow graph*nya.

Tabel 6.3 Node Fungsi do_changepassword

Method do_changepassword()	Node
Mulai method do_changepassword	1
Inisialisasi variable <i>user</i> untuk mendapatkan data <i>user</i> daam row	2
Inisialisasi variable <i>passasli</i> untuk mendapatkan password <i>user</i>	
Inisialisasi variable <i>passlama</i> untuk mendapatkan password lama dari <i>form</i> sebelumnya	

Passlama melakukan hashing	
Inisialisasi variable passbaru1 untuk mendapatkan password baru 1 dari form	
Passbaru1 melakukan hashing	
Inisialisasi variable passbaru2 untuk mendapatkan password baru 2 dari form	
Passbaru2 melakukan hashing	
If passasli == passlama && passbaru1 == passbaru2	3
Memanggil method change_password dengan parameter passbaru1	4
Mengirim pesan mulai session flashdata	
Mengalihkan ke halaman password	5
Else	6
Mengirim pesan melalui session flashdata	
Mengalihkan ke halaman password	7
End if	8
selesai	

Dalam pembentukan node fungsi do_changepassword terdapat 8 node yang terbentuk. Setelah menemukan node yang terbentuk, langkah selanjutnya adalah membuat *flow graph* dari fungsi do_changepassword berdasarkan node-node yang telah ditentukan. Gambar 6.2 di bawah merupakan *flow graph* dari fungsi do_changepassword().



Gambar 6.2 Flow Graph Fungsi do_changepassword ()

Berdasarkan *flow graph* fungsi do_changepassword dalam Gambar 6.2 didapatkan nilai *cyclomatic complexity* $V(G)$ yang dihitung pada perhitungan berikut ini.

$$V(G) = 2$$

$$V(G) = E - N + 2$$

$$= 8 - 8 + 2$$

$$= 2$$

$$V(G) = P + 1$$

$$= 1+1$$

$$= 2$$

Dari perhitungan *cyclometry complexity* dihasilkan 2 jalur independen, diantaranya :

- Jalur 1 = 1-2-3-4-7-8
- Jalur 2 = 1-2-3-5-6-7-8

Dari jalur independen yang sudah dibuat selanjutnya dilakukan pengujian *test case* atau kasus uji di masing-masing jalurnya. Tabel 6.4 di bawah merupakan *test case* pada fungsi `do_changepassword()`.

Tabel 6.4 Tase Case Fungsi `do_changepassword()`

No	Jalur	Perosedur Uji	Ekspektasi Pengujian	Hasill pengujian	Status
1	1-2-3-4-7-8	Menguji method <code>do_changepassword</code> dengan memasukkan semua data	<i>Password</i> akan berubah dan dialihkan ke halaman ganti <i>password</i>	<i>Password</i> berhasil berubah dan dialihkan ke halaman ganti <i>password</i>	valid
2	1-2-3-5-6-7-8	Menguji method <code>do_changepassword</code> dengan memasukkan data yang salah	<i>Password</i> akan gagal tersimpan	<i>Password</i> gagal tersimpan	valid

6.1.1.3 Pengujian Fungsi `do_deletariwayatinformmedcontent()`

Fungsi `do_deletariwayatinformmedcontent` merupakan fungsi yang digunakan oleh aktor divisi pelayanan untuk menghapus data *informed content* klien. Tabel 6.5 akan menunjukkan fungsi `do_deletariwayatinformmedcontent` beserta dengan *node flow graph*nya.

Tabel 6.5 Fungsi `do_deleteriyawatinformedcontent()`

Method <code>do_deletariwayatinformmedcontent</code>	Node
Mulai method <code>do_deletariwayatinformmedcontent()</code>	1
Memanggil model <code>riwayatinformmedcontent</code>	2
Variable <code>no</code> memanggil uri dari segmen 3	
Memanggil <code>delete_data</code> dengan parameter <code>no</code>	
Mengalihkan ke halaman <code>riwayatinformmedcontent</code>	3
selesai	4

Pada pembentukan node fungsi `do_deletariwayatinformmedcontent` terdapat 4 node yang terbentuk. Setelah menemukan node yang terbentuk, langkah

selanjutnya adalah membuat *flow graph* dari fungsi `do_deletariwayatinformcontent()` berdasarkan *node-node* yang telah ditentukan. Gambar 6.3 di bawah merupakan *flow graph* dari fungsi `do_deletariwayatinformcontent()`.



Gambar 6.3 Flow Graph Fungsi `do_deletariwayatinformcontent()`

Berdasarkan *flow graph* fungsi `do_deletariwayatinformcontent` dalam Gambar 6.3 didapatkan nilai *cyclomatic complexity* $V(G)$ yang dihitung pada perhitungan berikut ini.

$$\begin{aligned}
 V(G) &= 1 \\
 V(G) &= E - N + 2 \\
 &= 3 - 4 + 2 \\
 &= 1 \\
 V(G) &= P + 1 \\
 &= 0 + 1 \\
 &= 1
 \end{aligned}$$

Dari perhitungan *cyclomatic complexity* didapatkan 1 jalur independen, yaitu :

- Jalur 1 = 1-2-3-4

Dari jalur independen yang sudah dibuat selanjutnya dilakukan *pengujian test case* atau kasus uji di masing-masing jalurnya. Tabel 6.6 di bawah merupakan *test case* pada tiap-tiap jalur independen fungsi menghapus *informed content*.

Tabel 6.6 Tase Case Fungsi `do_deletariwayatinformcontent()`

N o	Jalu r	Perosedur Uji	Ekspektas i Pengujian	Hasil pengujian	Statu s
1	1-2- 3-4	Menguji method <code>do_deletariwayatinformcontent()</code> dengan memasukkan semua data	Data akan terhapus dalam <i>database</i> dan akan dialihkan	Data terhapus dalam <i>database</i> dan akan dialihkan	valid

			ke halaman riwayat <i>informed content</i>	ke halaman riwayat <i>informed content</i>	
--	--	--	--	--	--

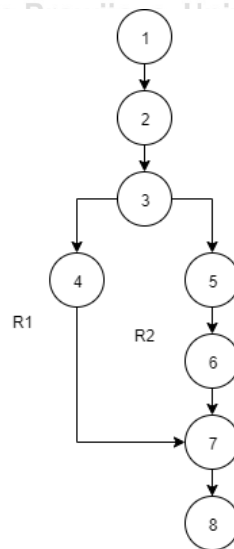
6.1.2 Pengujian Integrasi

Dalam tahap ini pengujian dilakukan dengan tujuan untuk dapat memastikan apakah komponen-komponen yang terdapat dalam sistem dapat berinteraksi sesuai dengan yang diharapkan (Nugroho, 2010). Pada pengujian integrasi ini dilakukan pada method `do_changepassword()` yang akan memanggil method `change_password` pada *class* pegawai seperti yang terlihat pada Tabel 6.7 sampai Tabel 6.9.

Tabel 6.7 Algoritme Fungsi `do_changepassword`

Method <code>do_changepassword()</code>	Node
Mulai method <code>do_changepassword</code>	1
Inisialisasi variable <code>user</code> untuk mendapatkan data <code>user</code> daam row	2
Inisialisasi variable <code>passasli</code> untuk mendapatkan password <code>user</code>	
Inisialisasi variable <code>passlama</code> untuk mendapatkan password lama dari <i>form</i> sebelumnya	
<code>Passlama</code> melakukan hashing	
Inisialisasi variable <code>passbaru1</code> untuk mendapatkan password baru 1 dari <i>form</i>	
<code>Passbaru1</code> melakukan hashing	
Inisialisasi variable <code>passbaru2</code> untuk mendapatkan paword baru 2 dari <i>form</i>	
<code>Passbaru2</code> melakukan hashing	
If <code>passasli == passlama && passbaru1 == passbaru2</code>	3
Memanggil method <code>change_password</code> dengan parameter <code>passbaru1</code>	4
Mengirim pesan mulai session flashdata	
Mengalihkan ke halaman password	
Else	5
Mengirim pesan melalui session flashdata	6
Mengalihkan ke halaman password	
End if	7
selesai	8

Dalam pembentukan node fungsi `do_changepassword` terdapat 8 node yang terbentuk. Setelah menemukan node yang terbentuk, langkah selanjutnya adalah membuat *flow graph* dari fungsi `do_changepassword` berdasarkan node-node yang telah ditentukan. Gambar 6.4 di bawah merupakan *flow graph* dari fungsi `do_changepassword()`.



Gambar 6.4 Flow Graph Fungsi do_changepassword ()

Berdasarkan *flow graph* fungsi `do_changepassword` dalam Gambar 6.4 didapatkan nilai *cyclomatic complexity* $V(G)$ yang dihitung pada perhitungan berikut ini.

$$V(G) = 2$$

$$V(G) = E - N + 2$$

$$= 8 - 8 + 2$$

$$= 2$$

$$V(G) = P + 1$$

$$= 1 + 1$$

$$= 2$$

Dari perhitungan *cyclomatic complexity* dihasilkan 2 jalur independen, diantaranya :

- Jalur 1 = 1-2-3-4-7-8
- Jalur 2 = 1-2-3-5-6-7-8

Tabel 6.8 Algoritme Fungsi change_password

Method change_password()	Node
Memulai method change_password()	1
Memanggil method set dengan parameter password	2
Memanggil method where dengan parameter username	3
Memanggil method update dengan parameter Tabel pegawai	4
selesai	5

Dalam pembentukan node fungsi `change_password` terdapat 8 node yang terbentuk. Setelah menemukan node yang terbentuk, langkah selanjutnya adalah

membikin *flow graph* dari fungsi `do_changepassword` berdasarkan node-node yang telah ditentukan. *Flow graph* dari fungsi `change_password` dapat dilihat pada Gambar 6.5.



Gambar 6.5 Flow Graph Fungsi `change_password` ()

Berdasarkan *flow graph* fungsi `change_password` dalam Gambar 6.2 didapatkan nilai *cyclomatic complexity* $V(G)$ yang dihitung pada perhitungan berikut ini.

$$V(G) = 1$$

$$V(G) = E - N + 2$$

$$= 4 - 5 + 2$$

$$= 1$$

$$V(G) = P + 1$$

$$= 0 + 1$$

$$= 1$$

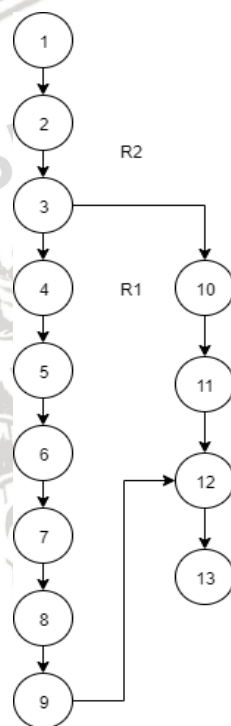
Dari perhitungan *cyclomatic complexity* didapatkan 1 jalur independen, yaitu :

- Jalur 1 = 1-2-3-4-5

Tabel 6.9 Algoritme Gabungan `do_changepassword`

Method <code>do_changepassword()</code>	Node
Mulai method <code>do_changepassword</code>	1
Inisialisasi variable <code>user</code> untuk mendapatkan data <code>user</code> daam row	2
Inisialisasi variable <code>passasli</code> untuk mendapatkan password <code>user</code>	
Inisialisasi variable <code>passlama</code> untuk mendapatkan password lama dari <code>form</code> sebelumnya	
<code>Passlama</code> melakukan hashing	
Inisialisasi variable <code>passbaru1</code> untuk mendapatkan password baru 1 dari <code>form</code>	
<code>Passbaru1</code> melakukan hashing	
Inisialisasi variable <code>passbaru2</code> untuk mendapatkan paword baru 2 dari <code>form</code>	

Passbaru2 melakukan hashing	3
If passasli == passlama && passbaru1 == passbaru2	4
Memulai method change_password()	5
Memanggil method set dengan parameter password	6
Pemanggil method where dengan parameter username	7
Memanggil method update dengan parameter Tabel pegawai	8
Selesai method change_password()	9
Mengirim pesan mulai session flashdata	10
Mengalihkan ke halaman password	11
Else	12
Mengirim pesan melalui session flashdata	13
Mengalihkan ke halaman password	12
End if	13
Selesai	13



Gambar 6.6 Flow Graph Fungsi do_changepassword ()

Berdasarkan *flow graph* fungsi `do_changepassword()` dalam Gambar 6.6 didapatkan nilai *cyclomatic complexity* $V(G)$ yang dihitung pada perhitungan berikut ini.

$$V(G) = 2$$

$$V(G) = E - N + 2$$

$$= 13 - 13 + 2$$

$$= 2$$

$$V(G) = P + 1$$

$$= 1 + 1$$

$$= 2$$

Dari perhitungan *cyclometic complexity* didapatkan 2 jalur independen, yaitu :

- Jalur 1 = 1-2-3-4-5-6-7-8-9-12-13
- Jalur 2 = 1-2-3-10-11-12-13

Dari jalur independen yang sudah dibuat selanjutnya dilakukan pengujian *test case* atau kasus uji di masing-masing jalurnya. Tabel 6.10 di bawah merupakan *test case* pada tiap-tiap jalur independen pada fungsi `do_changepassword()`.

Tabel 6.10 Tase Case Fungsi `do_changepassword()`

No	Jalur	Perosedur Uji	Ekspektasi Pengujian	Hasil pengujian	Status
1	1-2-3-4-5-6-7-8-9-12-13	Menguji method <code>do_changepassword</code> dengan memasukkan semua data	<i>Password</i> akan berubah dan dialihkan ke halaman ganti <i>password</i>	<i>Password</i> berhasil berubah dan dialihkan ke halaman ganti <i>password</i>	Valid
2	1-2-3-10-11-12-13	Menguji method <code>do_changepassword</code> dengan memasukkan data yang salah	<i>Password</i> akan gagal tersimpan	<i>Password</i> gagal tersimpan	Valid

6.1.3 Pengujian Validasi

Pengujian validasi disini bertujuan untuk menemukan error ketika skrip dieksekusi. Tabel 6.11 sampai 6.85 merupakan penjelasan dari seluruh fungsi yang sudah teruji dalam sistem informasi perawatan luka di praktik mandiri pedis care malang.

Tabel 6.11 Kasus Uji *Login*

Nama Kasus Uji	<i>Login</i>
Prosedur	1. Aktor mengisi kolom <i>username</i> = "siska". 2. Aktor mengisi kolom <i>password</i> = "siska". 3. Aktor menekan <i>button login</i>.
Hasil yang diharapkan	Sistem akan masuk ke dalam akun dan menampilkan akan halaman <i>dashboard</i> .
Hasil	Sistem berhasil masuk ke dalam akun dan menampilkan halaman <i>dashboard</i> .
Status	Valid.

Tabel 6.12 Kasus Uji Alternatif Login

Nama Kasus Uji	Login – salah satu kolom tidak terisi
Prosedur	1. Aktor mengisi kolom <i>username</i> = “siska”. 2. Aktor mengisi kolom <i>password</i> = “”. 3. Aktor menekan <i>button</i> login.
Hasil yang diharapkan	Sistem akan menampilkan peringatan <i>please fill out this field</i> .
Hasil	Sistem berhasil menampilkan peringatan <i>please fill out this field</i> .
Status	Valid.

Tabel 6.13 Kasus Uji Alternatif Login

Nama Kasus Uji	Login – masukkan <i>password</i> salah
Prosedur	1. Aktor mengisi kolom <i>username</i> = “siska”. 2. Aktor mengisi kolom <i>password</i> = “siska1”. 3. Aktor menekan <i>button</i> login.
Hasil yang diharapkan	Sistem akan menampilkan peringatan <i>username</i> atau <i>password</i> salah.
Hasil	Sistem berhasil menampilkan peringatan <i>username</i> atau <i>password</i> salah.
Status	Valid.

Tabel 6.14 Kasus Uji Logout

Nama Kasus Uji	Logout
Prosedur	1. Aktor menekan <i>icon</i> foto atau nama pengguna yang berada dipojok kanan atas. 2. Aktor memilih <i>button</i> sign out.
Hasil yang diharapkan	Sistem akan berhasil melakukan <i>logout</i> dan keluar dari sistem.
Hasil	Sistem akan berhasil melakukan <i>logout</i> dan keluar dari sistem.
Status	Valid.

Tabel 6.15 Kasus Uji Daftar

Nama Kasus Uji	Daftar
Prosedur	1. Aktor menekan <i>link</i> daftar di sini. 2. Aktor mengisi <i>field-field</i> yang ada dalam <i>form</i> <i>Username</i> = “eva” <i>Email</i> = eva@gmail.com <i>Password</i> = “eva” <i>Retype New Password</i> = “eva”

	Nama Lengkap dan Gelar = eva@gmail.com Nama Panggilan = "eva" Status Pegawai = "Tetap" Divisi = "PELAYANAN" Dasar Penerimaan "S1" Tanggal Lahir = "12/09/1992" Tanggal Masuk = "12/09/2019" 3. Aktor Menekan <i>button sign up</i> .
Hasil yang diharapkan	Sistem akan menampilkan "Pendaftaran telah diajukan!" pada halaman daftar akun baru.
Hasil	Sistem berhasil menampilkan "Pendaftaran telah diajukan!" pada halaman daftar akun baru.
Status	Valid

Tabel 6.16 Kasus Uji Alternatif Daftar

Nama Kasus Uji	Daftar – salah satu field ada yang tidak terisi
Prosedur	- Aktor menekan <i>link</i> daftar di sini. - Aktor mengosongkan salah satu <i>field</i> yang ada dalam <i>form</i> Username = "eva" Email = eva@gmail.com Password = " " Retype New Password = " " Nama Lengkap dan Gelar = " " Nama Panggilan = " " Status Pegawai = "Tetap" Divisi = "PELAYANAN" Dasar Penerimaan "S1" Tanggal Lahir = "mm/dd/yyyy" Tanggal Masuk = "mm/dd/yyyy" 3. Aktor Menekan <i>button sign up</i>.
Hasil yang diharapkan	Sistem akan menonaktifkan <i>button sign up</i> .
Hasil	Sistem berhasil menonaktifkan <i>button sign up</i> .
Status	Valid.

Tabel 6.17 Kasus Uji Mengubah *Password*

Nama Kasus Uji	Mengubah <i>Password</i>
Prosedur	- Aktor mengisi <i>field password</i> lama = "siska". - Aktor mengisi <i>field new password</i> = "siskaira". - Aktor mengisi <i>field retype new password</i> = "siskaira". - Aktor menekan <i>button save</i>.

Hasil yang diharapkan	Sistem akan mengganti <i>password</i> lama dan data <i>password</i> baru yang sudah diinputkan akan tersimpan di <i>database</i> .
Hasil	Sistem berhasil mengganti <i>password</i> lama dan data <i>password</i> baru yang sudah diinputkan akan tersimpan di <i>database</i> .
Status	Valid.

Tabel 6.18 Kasus Uji Alternatif Mengubah *Password*

Nama Kasus Uji	Mengubah <i>Password</i> – mengisi field <i>password</i> salah
Prosedur	1. Aktor mengisi <i>field password</i> “siskaira”. 2. Aktor mengisi <i>field new password</i> “keong”. 3. Aktor mengisi <i>field retype new password</i> “siskaira”. 4. Aktor menekan <i>button save</i>.
Hasil yang diharapkan	Sistem akan menampilkan pesan <i>error</i> “ubah <i>password</i> gagal” apabila data yang diinputkan tidak sesuai.
Hasil	Sistem berhasil menampilkan pesan <i>error</i> “ubah <i>password</i> gagal” ketika data yang diinputkan tidak sesuai.
Status	Valid

Tabel 6.19 Kasus Uji Alternatif Mengubah *Password*

Nama Kasus Uji	Mengubah <i>Password</i> – data pada field <i>new password</i> kosong
Prosedur	1. Aktor mengisi <i>field password</i> “siskaira”. 2. Aktor mengisi <i>field new password</i> “ ”. 3. Aktor mengisi <i>field retype new password</i> “siskaira”. 4. Aktor menekan <i>button save</i>.
Hasil yang diharapkan	Sistem akan menampilkan pesan <i>error</i> “ <i>please fill out this field</i> ”
Hasil	Sistem berhasil menampilkan pesan <i>error</i> “ <i>please fill out this field</i> ”
Status	Valid.

Tabel 6.20 Kasus Uji Mengedit Biodata Diri

Nama Kasus Uji	Mengedit Biodata Diri
Prosedur	1. menekan <i>button edit</i>. 2. Aktor memilih bagian kolom yang ingin diedit Berat Badan = “45” menjadi

	Berat Badan = "60" 3. Menekan <i>button</i> submit. 4. Data yang lama akan terganti dengan yang baru.
Hasil yang diharapkan	Sistem akan memperbarui biodata dan data baru yang sudah diinputkan akan tersimpan di <i>database</i> .
Hasil	Sistem berhasil memperbarui biodata dan data baru yang sudah diinputkan akan tersimpan di <i>database</i> .
Status	Valid.

Tabel 6.21 Kasus Uji Alternatif Mengedit Biodata Diri

Nama Kasus Uji	Mengedit Biodata Diri – tidak ada simbol "@" waktu mengubah data email
Prosedur	1. menekan <i>button edit</i> . 2. Aktor memilih bagian kolom yang ingin diedit Email = "siska@yahoo.com" menjadi Email = "siskagmail.com" 3. Menekan <i>button submit</i> .
Hasil yang diharapkan	Sistem akan menampilkan notifikasi <i>error</i> "please include an "@" in the email address, 'siskagmail.com' is missing an '@"
Hasil	Sistem berhasil menampilkan notifikasi <i>error</i> "please include an "@" in the email address, 'siskagmail.com' is missing an '@"
Status	Valid.

Tabel 6.22 Kasus Uji Menambah Data Master

Nama Kasus Uji	Menambah Data Master Dressing
Prosedur	1. Aktor memilih <i>dressing</i> . 2. Aktor menekan <i>button</i> tambah data. 3. Aktor mengisi data, Nama <i>dressing</i> = "perban" Harga = "15000" 4. Aktor menekan <i>button</i> simpan.
Hasil yang diharapkan	Sistem akan menambah data <i>master dressing</i> dan akan tersimpan di <i>database</i> .
Hasil	Sistem berhasil menambah data <i>master dressing</i> dan tersimpan di <i>database</i> .
Status	Valid.

Tabel 6.23 Kasus Uji Hapus Data Master

Nama Kasus Uji	Hapus Data Master
-----------------------	--------------------------

Prosedur	1. Aktor memilih <i>master dressing</i> . 2. Aktor menekan <i>button</i> “icon tempat sampah” untuk menghapus data.
Hasil yang diharapkan	Sistem akan menghapus salah satu data <i>master dressing</i> .
Hasil	Sistem berhasil menghapus salah satu data
Status	Valid.

Tabel 6.24 Kasus Uji Merubah Hak Akses Sistem

Nama Kasus Uji	Merubah Hak Akses Sistem
Prosedur	1. Aktor menekan <i>icon</i> gembok. 2. Akor memilih akses untuk pengguna dari = “admin” menjadi = “pelayanan”. 3. Aktor menekan <i>button</i> simpan.
Hasil yang diharapkan	Sistem akan merubah hak akses aktor dari divisi admin menjadi divisi pelayanan.
Hasil	Sistem berhasil merubah hak akses aktor dari divisi admin menjadi divisi pelayanan.
Status	Valid.

Tabel 6.25 Kasus Uji Memvalidasi Registrasi

Nama Kasus Uji	Memvalidasi Registrasi
Prosedur	1. Aktor menekan <i>button</i> aksi terima atau tidak. 2. Aktor menekan <i>combo box</i> akses untuk memilih hak akses. 3. Data pengajuan berhasil divalidasi.
Hasil yang diharapkan	Sistem akan memvalidasi registrasi pegawai baru dengan akun barunya.
Hasil	Sistem berhasil memvalidasi registrasi pegawai baru dengan akun barunya.
Status	Valid

Tabel 6.26 Kasus Uji Menghapus Data Pengguna Sistem

Nama Kasus Uji	Menghapus Data Pengguna Sistem
Prosedur	1. Aktor menekan <i>button</i> “icon tempat sampah”. 2. Muncul notifikasi “apakah yakin ingin menghapus pegawai”. 3. Aktor menekan <i>button</i> “ya”.
Hasil yang diharapkan	Sistem akan menghapus pengguna sistem.
Hasil	Sistem berhasil menghapus pengguna sistem.
Status	Valid.

Tabel 6.27 Kasus Uji Menambah Identitas Klien

Nama Kasus Uji	Menambah Identitas Klien
Prosedur	1. Aktor memilih menu pendaftaran klien. 2. Aktor memilih sub menu identitas klien. 3. Aktor mengisi kolom yang tersedia pada <i>form</i> identitas klien. 4. Aktor menekan <i>button</i> submit untuk menyimpan data klien baru ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menambah data identitas klien dan tersimpan di <i>database</i> .
Hasil	Sistem berhasil menambah data identitas klien dan tersimpan di <i>database</i> .
Status	Valid

Tabel 6.28 Kasus Uji Alternatif Menambah Identitas Klien

Nama Kasus Uji	Menambah Identitas Klien – salah satu kolom tidak terisi
Prosedur	1. Aktor memilih menu pendaftaran klien. 2. Aktor memilih sub menu identitas klien. 3. Aktor mengisi kolom yang tersedia pada <i>form</i> identitas klien. 4. Aktor menekan <i>button</i> submit untuk menyimpan data <i>klien</i> baru ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Hasil	Sistem berhasil menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Status	Valid

Tabel 6.29 Kasus Uji Menghapus Identitas Klien

Nama Kasus Uji	Menghapus Identitas Klien
Prosedur	1. Memilih menu pendaftaran klien. 2. Memilih sub menu riwayat identitas klien. 3. Menekan <i>button</i> "<i>icon</i> tempat sampah". 4. Jika muncul popup pilih <i>button</i> "<i>ya</i>".
Hasil yang diharapkan	Sistem akan menghapus data pendaftaran riwayat klien.
Hasil	Sistem berhasil menghapus data pendaftaran riwayat klien.
Status	Valid

Tabel 6.30 Kasus Uji Mengedit Identitas Klien

Nama Kasus Uji	Menghapus Identitas Klien
Prosedur	1. Memilih menu pendaftaran klien 2. Memilih sub menu riwayat identitas klien 3. Menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Tekan <i>button save</i>.
Hasil yang diharapkan	Sistem akan mengedit data identitas klien dan menyimpannya dalam <i>database</i> .
Hasil	Sistem berhasil mengedit data identitas klien dan menyimpannya dalam <i>database</i> .
Status	Valid.

Tabel 6.31 Kasus Uji Mencetak Identitas Klien

Nama Kasus Uji	Mencetak Identitas Klien
Prosedur	1. Memilih menu pendaftaran klien. 2. Memilih sub menu riwayat identitas klien. 3. Menekan <i>button print</i>. 4. Tekan <i>button keyboard ctrl + p</i> 5. Tekan <i>print</i>.
Hasil yang diharapkan	Sistem akan mencetak <i>form</i> identitas klien.
Hasil	Sistem berhasil mencetak <i>form</i> identitas klien.
Status	Valid

Tabel 6.32 Kasus Uji Menambah *Informed Content*

Nama Kasus Uji	Menambah <i>Informed Content</i>
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu pendaftaran klien. 3. Aktor memilih sub menu <i>informed content</i>. 4. Aktor mengisi kolom yang tersedia pada <i>form informed content</i>. 5. Aktor menekan <i>button submit</i> untuk menyimpan data <i>informed content</i> ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menambah data <i>informed content</i> dan tersimpan di <i>database</i> .
Hasil	Sistem berhasil menambah data <i>informed content</i> dan tersimpan di <i>database</i> .
Status	Valid

Tabel 6.33 Kasus Uji Alternatif Menambah *Informed Content*

Nama Kasus Uji	Menambah <i>Informed Content</i> – salah satu kolom tidak terisi
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem 2. Aktor memilih menu pendaftaran <i>klien</i> 3. Aktor memilih sub menu <i>informed content</i> 4. Aktor mengisi kolom yang tersedia pada <i>form informed content</i> 5. Aktor menekan <i>button</i> submit untuk menyimpan data <i>informed content</i> ke <i>database</i>
Hasil yang diharapkan	Sistem akan menampilkan peringatan “ <i>Please fill out this field</i> ” apabila ada kolom yang tidak terisi.
Hasil	Sistem berhasil menampilkan peringatan “ <i>Please fill out this field</i> ” apabila ada kolom yang tidak terisi.
Status	Valid

Tabel 6.34 Kasus Uji Menghapus *Informed Content*

Nama Kasus Uji	Menghapus <i>Informed Content</i>
Prosedur	1. Aktor memilih menu pendaftaran klien. 2. Aktor memilih sub menu riwayat <i>informed content</i>. 3. Aktor menekan <i>button</i> “<i>icon tempat sampah</i>”. 4. Jika muncul pop up pilih <i>button</i> “<i>ya</i>”.
Hasil yang diharapkan	Sistem akan menghapus data pendaftaran <i>informed content</i> .
Hasil	Sistem berhasil menghapus data pendaftaran <i>informed content</i> .
Status	Valid

Tabel 6.35 Kasus Uji Mengedit *Informed Content*

Nama Kasus Uji	Mengedit <i>Informed Content</i>
Prosedur	1. Aktor memilih menu pendaftaran klien. 2. Aktor memilih sub menu riwayat <i>informed content</i>. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Hasil yang diharapkan	Sistem akan mengedit data <i>informed content</i> dan menyimpannya dalam <i>database</i> .

Hasil	Sistem berhasil mengedit data <i>informed content</i> dan menyimpannya dalam <i>database</i> .
Status	Valid

Tabel 6.36 Kasus Uji Menambah Pengkajian Umum

Nama Kasus Uji	Menambah Pengkajian Umum
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu diagnosa klien. 3. Aktor memilih sub menu pengkajian umum. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> pengkajian umum. 5. Menekan <i>button submit</i> untuk menyimpan data pengkajian umum ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menambah data pengkajian umum dan tersimpan di <i>database</i> .
Hasil	Sistem berhasil menambah data pengkajian umum dan tersimpan di <i>database</i> .
Status	Valid

Tabel 6.37 Kasus Uji Alternatif Menambah Pengkajian Umum

Nama Kasus Uji	Menambah Pengkajian Umum – salah satu kolom tidak terisi
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu diagnosa klien. 3. Aktor memilih sub menu pengkajian umum. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> pengkajian umum. 5. Menekan <i>button submit</i> untuk menyimpan data pengkajian umum ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Hasil	Sistem berhasil menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Status	Valid.

Tabel 6.38 Kasus Uji Menghapus Data Pengkajian Umum

Nama Kasus Uji	Menghapus Data Pengkajian Umum
Prosedur	1. Aktor memilih menu Diagnosa klien. 2. Aktor memilih sub menu riwayat pengkajian umum. 3. Aktor menekan <i>button "icon tempat sampah"</i>.

	4. Jika muncul popup pilih <i>button</i> “ya”.
Hasil yang diharapkan	Sistem akan menghapus data diagnosa pengkajian umum.
Hasil	Sistem berhasil menghapus data diagnosa pengkajian umum.
Status	Valid.

Tabel 6.39 Kasus Uji Mengedit Form Pengkajian Umum

Nama Kasus Uji	Mengedit Form Pengkajian Umum
Prosedur	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu riwayat pengkajian umum. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Hasil yang diharapkan	Sistem akan mengedit <i>form</i> pengkajian umum dan menyimpannya dalam <i>database</i> .
Hasil	Sistem berhasil mengedit <i>form</i> pengkajian umum dan menyimpannya dalam <i>database</i> .
Status	Valid.

Tabel 6.40 Kasus Uji Mencetak Pengkajian Umum

Nama Kasus Uji	Mencetak Pengkajian Umum
Prosedur	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu riwayat pengkajian umum. 3. Aktor menekan <i>button print</i>. 4. Aktor menekan <i>button keyboard</i> ctrl + p. 5. Aktor menekan <i>print</i>.
Hasil yang diharapkan	Sistem akan mencetak <i>form</i> pengkajian umum.
Hasil	Sistem berhasil mencetak <i>form</i> pengkajian umum.
Status	Valid.

Tabel 6.41 Kasus Uji Menambah Pengkajian Luka

Nama Kasus Uji	Menambah Pengkajian Luka
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu diagnosa klien. 3. Aktor memilih sub menu pengkajian luka. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> pengkajian luka.

	5. Aktor menekan <i>button</i> submit untuk menyimpan data pengkajian luka ke <i>database</i> .
Hasil yang diharapkan	Sistem akan menambah data pengkajian luka dan tersimpan di <i>database</i> .
Hasil	Sistem berhasil menambah data pengkajian luka dan tersimpan di <i>database</i> .
Status	Valid.

Tabel 6.42 Kasus Uji Alternatif Menambah Pengkajian Luka

Nama Kasus Uji	Menambah Pengkajian Luka – salah satu kolom tidak terisi
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu diagnosa klien. 3. Aktor memilih sub menu pengkajian luka. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> pengkajian luka. 5. Aktor menekan <i>button</i> submit untuk menyimpan data pengkajian luka ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menampilkan peringatan “ <i>Please fill out this field</i> ” apabila ada kolom yang tidak terisi.
Hasil	Sistem berhasil menampilkan peringatan “ <i>Please fill out this field</i> ” apabila ada kolom yang tidak terisi.
Status	Valid.

Tabel 6.43 Kasus Uji Menghapus Data Pengkajian Luka

Nama Kasus Uji	Menghapus data Pengkajian Luka
Prosedur	1. Aktor memilih menu Diagnosa klien. 2. Aktor memilih sub menu riwayat pengkajian luka. 3. Aktor menekan <i>button</i> “<i>icon</i> tempat sampah”. 4. Jika muncul popup pilih <i>button</i> “<i>ya</i>”.
Hasil yang diharapkan	Sistem akan menghapus data diagnosa pengkajian luka.
Hasil	Sistem berhasil menghapus data diagnosa pengkajian luka.
Status	Valid.

Tabel 6.44 Kasus Uji Mengedit Form Pengkajian Luka

Nama Kasus Uji	Mengedit Form Pengkajian Luka
Prosedur	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu riwayat pengkajian luka.

	3. Aktor menekan <i>button edit</i> . 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i> .
Hasil yang diharapkan	Sistem akan mengedit <i>form</i> pengkajian luka dan menyimpannya dalam <i>database</i> .
Hasil	Sistem berhasil mengedit <i>form</i> pengkajian luka dan menyimpannya dalam <i>database</i> .
Status	Valid.

Tabel 6.45 Kasus Uji Mencetak Pengkajian Luka

Nama Kasus Uji	Mencetak Pengkajian Luka
Prosedur	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu riwayat pengkajian luka. 3. Aktor menekan <i>button print</i> . 4. Aktor menekan <i>button keyboard ctrl + p</i> . 5. Aktor menekan <i>print</i> .
Hasil yang diharapkan	Sistem akan mencetak <i>form</i> pengkajian luka.
Hasil	Sistem berhasil mencetak <i>form</i> pengkajian luka.
Status	Valid.

Tabel 6.46 Kasus Uji Menambah Betes Jensen

Nama Kasus Uji	Menambah Betes Jensen
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu diagnosa klien. 3. Aktor memilih sub menu betes Jensen. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> pengkajian betes Jensen 5. Aktor menekan <i>button submit</i> untuk menyimpan data pengkajian betes Jensen ke <i>database</i> .
Hasil yang diharapkan	Sistem akan menambah data betes Jensen dan tersimpan di <i>database</i> .
Hasil	Sistem berhasil menambah data betes Jensen dan tersimpan di <i>database</i> .
Status	Valid.

Tabel 6.47 Kasus Uji Alternatif Menambah Betes Jensen

Nama Kasus Uji	Menambah Betes Jensen – salah satu kolom tidak terisi
-----------------------	--

Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu diagnosa klien. 3. Aktor memilih sub menu betes Jensen. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> pengkajian betes Jensen. 5. Aktor menekan <i>button submit</i> untuk menyimpan data pengkajian betes Jensen ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Hasil	Sistem berhasil menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Status	Valid

Tabel 6.48 Kasus Uji Menghapus Bete Jensen

Nama Kasus Uji	Menambah Betes Jensen – salah satu kolom tidak terisi
Prosedur	1. Aktor memilih menu Diagnosa klien. 2. Aktor memilih sub menu riwayat betes Jensen. 3. Aktor menekan <i>button "icon tempat sampah"</i>. 4. Jika muncul popup pilih <i>button "ya"</i>.
Hasil yang diharapkan	Sistem akan menghapus data diagnosa pengkajian betes Jensen.
Hasil	Sistem berhasil menghapus data diagnosa pengkajian betes Jensen.
Status	Valid.

Tabel 6.49 Kasus Uji Mengedit Betes Jensen

Nama Kasus Uji	Mengedit Betes Jensen
Prosedur	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu riwayat pengkajian betes Jensen. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Hasil yang diharapkan	Sistem akan mengedit <i>form</i> pengkajian betes Jensen dan menyimpannya dalam <i>database</i> .
Hasil	Sistem berhasil mengedit <i>form</i> pengkajian betes Jensen dan menyimpannya dalam <i>database</i> .
Status	Valid.

Tabel 6.50 Kasus Uji Mencetak Pengkajian Betes Jensen

Nama Kasus Uji	Mencetak Pengkajian Betes Jensen
Prosedur	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu riwayat pengkajian betes Jensen. 3. Aktor menekan <i>button print</i>. 4. Aktor menekan <i>button keyboard</i> ctrl + p. 5. Aktor menekan <i>print</i>.
Hasil yang diharapkan	Sistem akan mencetak <i>form</i> pengkajian betes Jensen.
Hasil	Sistem berhasil mencetak <i>form</i> pengkajian betes Jensen.
Status	Valid.

Tabel 6.51 Kasus Uji Menambah Data Tampilan Klinis Luka

Nama Kasus Uji	Menambah Data Tampilan Klinis Luka
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Memilih menu diagnosa klien. 3. Memilih sub menu tampilan klinis luka. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> tampilan klinis luka. 5. Menekan <i>button</i> submit untuk menyimpan data tampilan klinis luka ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menambah data tampilan klinis luka dan tersimpan di <i>database</i> .
Hasil	Sistem berhasil menambah data tampilan klinis luka dan tersimpan di <i>database</i> .
Status	Valid.

Tabel 6.52 Kasus Uji ALternatif Menambah Data Tampilan Klinis Luka

Nama Kasus Uji	Menambah Data Tampilan Klinis Luka – salah satu kolom tidak terisi
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Memilih menu diagnosa klien. 3. Memilih sub menu tampilan klinis luka. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> tampilan klinis luka. 5. Menekan <i>button</i> submit untuk menyimpan data tampilan klinis luka ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.

Hasil	Sistem berhasil menampilkan peringatan <i>"Please fill out this field"</i> apabila ada kolom yang tidak terisi.
Status	Valid.

Tabel 6.53 Kasus Uji Menghapus Data Tampilan klinis Luka

Nama Kasus Uji	Menghapus Data Tampilan Klinis Luka
Prosedur	1. Aktor memilih menu Diagnosa klien. 2. Aktor memilih sub menu riwayat betes Jensen. 3. Aktor menekan <i>button</i> <i>"icon tempat sampah"</i>. 4. Jika muncul popup pilih <i>button</i> <i>"ya"</i>.
Hasil yang diharapkan	Sistem akan menghapus data diagnosa tampilan klinis luka.
Hasil	Sistem berhasil menghapus data diagnosa tampilan klinis luka.
Status	Valid

Tabel 6.54 Kasus Uji Mengedit Tampilan Klinis Luka

Nama Kasus Uji	Mengedit Tampilan Klinis Luka
Prosedur	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu riwayat tampilan klinis luka. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Hasil yang diharapkan	Sistem akan mengedit <i>form</i> tampilan klinis luka dan menyimpannya dalam <i>database</i> .
Hasil	Sistem berhasil mengedit <i>form</i> tampilan klinis luka dan menyimpannya dalam <i>database</i> .
Status	Valid.

Tabel 6.55 Kasus Uji Mencetak Tampilan Klinis Luka

Nama Kasus Uji	Mencetak Tampilan Klinis Luka
Prosedur	1. Aktor memilih menu diagnosa klien. 2. Aktor memilih sub menu riwayat tampilan klinis luka. 3. Aktor menekan <i>button print</i>. 4. Aktor mekan <i>button keyboard</i> ctrl + p. 5. Aktor menekan <i>print</i>.
Hasil yang diharapkan	Sistem akan mencetak <i>form</i> tampilan klinis luka.
Hasil	Asistem berhasil mencetak <i>form</i> tampilan klinis luka.
Status	Valid.

Tabel 6.56 Kasus Uji Menambah Tindakan

Nama Kasus Uji	Menambah Tindakan
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu tindakan. 3. Aktor memilih sub menu tindakan. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> tindakan. 5. Menekan <i>button submit</i> untuk menyimpan data tindakan ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menambah data tindakan dan tersimpan di <i>database</i> .
Hasil	Sistem berhasil menambah data tindakan dan tersimpan di <i>database</i> .
Status	Valid.

Tabel 6.57 Kasus Uji Alternatif Menambah Tindakan

Nama Kasus Uji	Menambah Tindakan – salah satu kolom tidak terisi
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu tindakan. 3. Aktor memilih sub menu tindakan. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> tindakan. 5. Menekan <i>button submit</i> untuk menyimpan data tindakan ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menampilkan peringatan “ <i>Please fill out this field</i> ” apabila ada kolom yang tidak terisi.
Hasil	Sistem berhasil menampilkan peringatan “ <i>Please fill out this field</i> ” apabila ada kolom yang tidak terisi.
Status	Valid.

Tabel 6.58 Kasus Uji Menghapus Tindakan

Nama Kasus Uji	Menghapus Tindakan
Prosedur	1. Aktor memilih menu tindakan. 2. Aktor memilih sub menu riwayat tindakan. 3. Aktor menekan <i>button “icon tempat sampah”</i>. 4. Jika muncul popup pilih <i>button “ya”</i>.
Hasil yang diharapkan	Sistem akan menghapus data tindakan dalam sistem.
Hasil	Sistem berhasil menghapus data tindakan dalam sistem.

Status	Valid.
--------	--------

Tabel 6.59 Kasus Uji Mencetak Tindakan

Nama Kasus Uji	Mencetak Tindakan
Prosedur	1. Aktor memilih menu tindakan. 2. Aktor memilih sub menu riwayat tindakan. 3. Aktor menekan <i>button</i> print. 4. Aktor menekan <i>button</i> keyboard ctrl + p. 5. Aktor menekan print.
Hasil yang diharapkan	Sistem akan mencetak <i>form</i> tindakan.
Hasil	Sistem berhasil mencetak <i>form</i> tindakan.
Status	Valid.

Tabel 6.60 Kasus Uji Menambah Perkembangan Klien

Nama Kasus Uji	Menambah Perkembangan Klien
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu perkembangan klien. 3. Aktor memilih sub menu perkembangan klien. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> perkembangan klien. 5. Aktor menekan <i>button submit</i> untuk menyimpan data perkembangan klien ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menambah data perkembangan klien dan tersimpan di <i>database</i> .
Hasil	Sistem berhasil menambah data perkembangan klien dan tersimpan di <i>database</i> .
Status	Valid.

Tabel 6.61 Kasus Uji Alternatif Menambah Perkembangan Klien

Nama Kasus Uji	Menambah Perkembangan Klien - salah satu kolom tidak terisi
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu perkembangan klien. 3. Aktor memilih sub menu perkembangan klien. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> perkembangan klien. 5. Aktor menekan <i>button submit</i> untuk menyimpan data perkembangan <i>klien</i> ke <i>database</i>.

Hasil yang diharapkan	Sistem akan menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Hasil	Sistem berhasil menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Status	Valid.

Tabel 6.62 Kasus Uji Menghapus Data Perkembangan Klien

Nama Kasus Uji	Menghapus Data Perkembangan Klien
Prosedur	1. Aktor memilih menu perkembangan klien. 2. Aktor memilih sub menu riwayat perkembangan klien. 3. Aktor menekan <i>button</i> "<i>icon</i> tempat sampah". 4. Jika muncul popup pilih <i>button</i> "<i>ya</i>".
Hasil yang diharapkan	Sistem akan menghapus data perkembangan <i>klien</i> dalam sistem.
Hasil	Sistem berhasil menghapus data perkembangan <i>klien</i> dalam sistem.
Status	Valid

Tabel 6.63 Kasus Uji Mengedit Form Perkembangan Klien

Nama Kasus Uji	Mengedit Form Perkembangan Klien
Prosedur	1. Aktor memilih menu perkembangan klien. 2. Aktor memilih sub menu riwayat perkembangan klien. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Hasil yang diharapkan	Sistem akan mengedit <i>form</i> perkembangan klien dan menyimpannya dalam <i>database</i> .
Hasil	Sistem berhasil mengedit <i>form</i> perkembangan klien dan menyimpannya dalam <i>database</i> .
Status	Valid.

Tabel 6.64 Kasus Uji Mencetak Form Perkembangan Klien

Nama Kasus Uji	Mencetak form Perkembangan Klien
Prosedur	1. Aktor memilih menu perkembangan klien. 2. Aktor memilih sub menu riwayat perkembangan klien. 3. Aktor menekan <i>button print</i>. 4. Aktor menekan <i>button keyboard</i> ctrl + p. 5. Aktor menekan <i>print</i>.

Hasil yang diharapkan	Sistem akan mencetak <i>form</i> perkembangan klien.
Hasil	Sistem berhasil mencetak <i>form</i> perkembangan klien.
Status	Valid.

Tabel 6.65 Kasus Uji Menambah Kunjungan

Nama Kasus Uji	Menambah Kunjungan
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu kunjungan. 3. Aktor memilih sub menu kunjungan. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> kunjungan. 5. Aktor menekan <i>button submit</i> untuk menyimpan data kunjungan ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menambah data kunjungan dan tersimpan di <i>database</i> .
Hasil	Sistem berhasil menambah data kunjungan dan tersimpan di <i>database</i> .
Status	Valid.

Tabel 6.66 Kasus Uji Alternatif Menambah Kunjungan

Nama Kasus Uji	Menambah Kunjungan – salah satu kolom tidak terisi
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu kunjungan. 3. Aktor memilih sub menu kunjungan. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> kunjungan. 5. Aktor menekan <i>button submit</i> untuk menyimpan data kunjungan ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Hasil	Sistem berhasil menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Status	Valid.

Tabel 6.67 Kasus Uji Menghapus Kunjungan

Nama Kasus Uji	Menghapus Kunjungan
Prosedur	1. Aktor memilih menu kunjungan. 2. Aktor memilih sub menu riwayat kunjungan. 3. Aktor menekan <i>button "icon tempat sampah"</i>. 4. Jika muncul popup pilih <i>button "ya"</i>.

Hasil yang diharapkan	Sistem akan menghapus data kunjungan dalam sistem.
Hasil	Sistem berhasil menghapus data kunjungan dalam sistem.
Status	Valid.

Tabel 6.68 Kasus Uji Mengedit Form Kunjungan

Nama Kasus Uji	Mengedit Form Kunjungan
Prosedur	1. Aktor memilih menu kunjungan. 2. Aktor memilih sub menu riwayat kunjungan. 3. Aktor menekan <i>button edit</i>. 4. Aktor muncul <i>form edit</i> dan mengisi bagian yang ingin <i>diedit</i>. 5. Aktor menekan <i>button save</i>.
Hasil yang diharapkan	Sistem akan mengedit <i>form</i> kunjungan dan menyimpannya dalam <i>database</i> .
Hasil	Sistem berhasil mengedit <i>form</i> kunjungan dan menyimpannya dalam <i>database</i> .
Status	Valid.

Tabel 6.69 Kasus Uji Mencetak Kunjungan

Nama Kasus Uji	Mencetak Kunjungan
Prosedur	1. Aktor memilih menu kunjungan. 2. Aktor memilih sub menu riwayat kunjungan. 3. Aktor menekan <i>button print</i>. 4. Aktor menekan <i>button keyboard ctrl + p</i>. 5. Aktor menekan <i>print</i>.
Hasil yang diharapkan	Sistem akan mencetak <i>form</i> kunjungan
Hasil	Sistem berhasil mencetak <i>form</i> kunjungan
Status	Valid.

Tabel 6.70 Kasus Uji Menambah Pembayaran

Nama Kasus Uji	Menambah Pembayaran
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu pembayaran. 3. Aktor memilih sub menu pembayaran. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> pembayaran. 5. Aktor menekan <i>button submit</i> untuk menyimpan data pembayaran ke <i>database</i>.

Hasil yang diharapkan	Sistem akan menambah data pembayaran dan tersimpan di <i>database</i> .
Hasil	Sistem berhasil menambah data pembayaran dan tersimpan di <i>database</i> .
Status	Valid.

Tabel 6.71 Kasus Uji Alternatif Menambah Pembayaran

Nama Kasus Uji	Menambah Pembayaran - salah satu kolom tidak terisi
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu pembayaran. 3. Aktor memilih sub menu pembayaran. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> pembayaran. 5. Aktor menekan <i>button submit</i> untuk menyimpan data pembayaran ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Hasil	Sistem berhasil menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Status	Valid.

Tabel 6.72 Kasus Uji Menghapus *Form* Pembayaran

Nama Kasus Uji	Menghapus <i>Form</i> Pembayaran
Prosedur	1. Memilih menu pembayaran. 2. Memilih sub menu riwayat pembayaran. 3. Menekan <i>button "icon tempat sampah"</i>. 4. Jika muncul pop up pilih <i>button "ya"</i>.
Hasil yang diharapkan	Sistem akan menghapus data pembayaran dalam sistem.
Hasil	Sistem berhasil menghapus data pembayaran dalam sistem.
Status	Valid.

Tabel 6.73 Kasus Uji Mencetak *Form* Pembayaran

Nama Kasus Uji	Mencetak <i>Form</i> Pembayaran
Prosedur	1. Memilih menu pembayaran. 2. Memilih sub menu riwayat pembayaran. 3. Menekan <i>button print</i>. 4. Tekan <i>button keyboard ctrl + p</i>. 5. Tekan <i>print</i>.
Hasil yang diharapkan	Sistem akan mencetak <i>form</i> pembayaran.

Hasil	Sistem berhasil mencetak <i>form</i> pembayaran.
Status	Valid.

Tabel 6.74 Kasus Uji Menambah Penolakan Tindakan

Nama Kasus Uji	Menambah Penolakan Tindakan
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu dokumen. 3. Aktor memilih sub menu penolakan tindakan. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> penolakan tindakan. 5. Menekan <i>button submit</i> untuk menyimpan data penolakan tindakan ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menambah data penolakan tindakan dan tersimpan di <i>database</i> .
Hasil	Sistem berhasil menambah data penolakan tindakan dan tersimpan di <i>database</i> .
Status	Valid.

Tabel 6.75 Kasus Uji Alternatif Menambah Penolakan Tindakan

Nama Kasus Uji	Menambah Penolakan Tindakan – salah satu kolom tidak terisi
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu dokumen. 3. Aktor memilih sub menu penolakan tindakan. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> penolakan tindakan. 5. Menekan <i>button submit</i> untuk menyimpan data penolakan tindakan ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Hasil	Sistem berhasil menampilkan peringatan " <i>Please fill out this field</i> " apabila ada kolom yang tidak terisi.
Status	Valid.

Tabel 6.76 Kasus Uji Menghapus Penolakan Tindakan

Nama Kasus Uji	Menghapus Penolakan Tindakan
Prosedur	1. Aktor memilih menu dokumen. 2. Aktor memilih sub menu riwayat penolakan tindakan. 3. Aktr menekan <i>button</i> "<i>icon tempat sampah</i>". 4. Jika muncul popup pilih <i>button</i> "<i>ya</i>".

Hasil yang diharapkan	Sistem akan menghapus data penolakan tindakan dalam sistem.
Hasil	Sistem berhasil menghapus data penolakan tindakan dalam sistem.
Status	Valid.

Tabel 6.77 Kasus Uji Mengedit *Form* Penolakan Tindakan

Nama Kasus Uji	Mengedit <i>Form</i> Penolakan Tindakan
Prosedur	1. Aktor memilih menu dokumen. 2. Aktor memilih sub menu riwayat penolakan tindakan. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit. 5. Aktor menekan <i>button save</i>.
Hasil yang diharapkan	Sistem akan mengedit <i>form</i> penolakan tindakan dan menyimpannya dalam <i>database</i> .
Hasil	Sistem berhasil mengedit <i>form</i> penolakan tindakan dan menyimpannya dalam <i>database</i> .
Status	Valid.

Tabel 6.78 Kasus Uji Mencetak penolakan Tindakan

Nama Kasus Uji	Mencetak Penolakan tindakan
Prosedur	1. Aktor memilih menu dokumen. 2. Aktor memilih sub menu riwayat penolakan tindakan. 3. Aktor menekan <i>button print</i>. 4. Aktor menekan <i>button keyboard ctrl + p</i>. 5. Aktor menekan <i>print</i>.
Hasil yang diharapkan	Sistem berhasil mencetak <i>form</i> penolakan tindakan.
Hasil	Sistem berhasil mencetak <i>form</i> penolakan tindakan.
Status	Valid.

Tabel 6.79 Kasus Uji Menambah Data Rujukan

Nama Kasus Uji	Menambah Data Rujukan
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu dokumen. 3. Aktor memilih sub menu rujukan. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> rujukan.

	5. Aktor menekan <i>button submit</i> untuk menyimpan data rujukan ke <i>database</i> .
Hasil yang diharapkan	Sistem akan menambah data rujukan dan tersimpan di <i>database</i> .
Hasil	Sistem berhasil menambah data rujukan dan tersimpan di <i>database</i> .
Status	Valid.

Tabel 6.80 Kasus Uji Alternatif Menambah Data Rujukan

Nama Kasus Uji	Menambah Data Rujukan – salah satu kolom tidak terisi
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Aktor memilih menu dokumen. 3. Aktor memilih sub menu rujukan. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> rujukan. 5. Aktor menekan <i>button submit</i> untuk menyimpan data rujukan ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menampilkan peringatan “ <i>Please fill out this field</i> ” apabila ada kolom yang tidak terisi.
Hasil	Sistem berhasil menampilkan peringatan “ <i>Please fill out this field</i> ” apabila ada kolom yang tidak terisi.
Status	Valid.

Tabel 6.81 Kasus Uji Menghapus Data Rujukan

Nama Kasus Uji	Menghapus Data Rujukan
Prosedur	1. Aktor memilih menu dokumen. 2. Aktor memilih sub menu riwayat rujukan. 3. Aktor menekan <i>button “icon tempat sampah”</i>. 4. Jika muncul popup pilih <i>button “ya”</i>.
Hasil yang diharapkan	Sistem akan menghapus data rujukan dalam sistem.
Hasil	Sistem berhasil menghapus data rujukan dalam sistem.
Status	Valid.

Tabel 6.82 Kasus Uji Mengedit Form Rujukan

Nama Kasus Uji	Mengedit Form Rujukan
Prosedur	1. Aktor memilih menu dokumen. 2. Aktor memilih sub menu riwayat rujukan. 3. Aktor menekan <i>button edit</i>. 4. Muncul <i>form edit</i> dan mengisi bagian yang ingin diedit.

	5. Aktor menekan <i>button save</i> .
Hasil yang diharapkan	Sistem akan mengedit <i>form</i> rujukan dan menyimpannya dalam <i>database</i> .
Hasil	Sistem berhasil mengedit <i>form</i> rujukan dan menyimpannya dalam <i>database</i> .
Status	Valid.

Tabel 6.83 Kasus Uji Mencetak *Form* Rujukan

Nama Kasus Uji	Mencetak <i>Form</i> Rujukan
Prosedur	1. Memilih menu dokumen. 2. Memilih sub menu riwayat rujukan. 3. Menekan <i>button print</i>. 4. Tekan <i>button keyboard</i> ctrl + p. 5. Tekan <i>print</i>.
Hasil yang diharapkan	Sistem akanl mencetak <i>form</i> rujukan.
Hasil	Sistem berhasil mencetak <i>form</i> rujukan.
Status	Valid.

Tabel 6.84 Kasus Uji Tambah Jadwal

Nama Kasus Uji	Tambah Jadwal
Prosedur	1. Aktor telah <i>login</i> dan berhasil masuk ke dalam sistem. 2. Memilih menu penjadwalan klien. 3. Memilih sub menu lihat jadwal. 4. Aktor mengisi kolom yang tersedia pada <i>form</i> lihat jadwal. 5. Menekan <i>button submit</i> untuk menyimpan data rujukan ke <i>database</i>.
Hasil yang diharapkan	Sistem akan menambah data jadwal dan tersimpan di <i>database</i> .
Hasil	Sistem berhasil menambah data jadwal dan tersimpan di <i>database</i> .
Status	Valid.

Tabel 6.85 Kasus Uji Hapus Jadwal

Nama Kasus Uji	Hapus Jadwal
Prosedur	1. Aktor memilih menu penjadwalan klien. 2. Aktor memilih sub menu riwayat penjadwalan. 3. Aktor menekan <i>button "icon sampah"</i>. 4. Jika muncul notifikasi "Anda yakin ingin menghapus riwayat data ini?" pilih <i>button "ya"</i>.

Hasil yang diharapkan	Sistem akan menghapus data penjadwalan pada sistem.
Hasil	Sistem berhasil menghapus data penjadwalan pada sistem.
Status	Valid.

6.1.4 Pengujian Compatibility

Penelitian ini menggunakan *tool* SortSite dalam pegujian *compatibility*. Pengujian digunakan untuk menguji kompatibilitas sistem supaya bisa diakses pada beberapa *browser* seperti yang dijelaskan pada Tabel 6.86 cara penggunaan *tool SortSite*.

Tabel 6.86 Pengujian Compatibility

No	Pengujian
1	Menggunakan <i>tool</i> SortSite
2	Menggunakan alamat penyimpanan file dalam komputer untuk melakukan pengecekan yaitu file://C:/xampp/htdocs/project7
3	Klik “Check”
4	Muncul hasil pengujian kompatibilitas

Summary Issues file:///C:/xampp/htdocs/project7

Errors Accessibility Compatibility Search Standards Usability

This tab shows pages that exhibit browser-specific behavior, or trigger browser bugs.

Browser	IE	Edge	Firefox	Safari	Opera	Chrome	iOS	Android
Version	11	18	69	13	60	77	≤ 11 12 13	≤ 3 4*
Critical Issues	OK	OK	OK	OK	OK	OK	OK	OK
Major Issues	OK	OK	OK	OK	OK	OK	OK	OK
Minor Issues	OK	OK	OK	OK	OK	OK	OK	OK

Key

- Missing content or functionality
- Major layout or performance problems
- Minor layout or performance problems

* Most Android devices from 4.4 onwards use Chrome as the default browser, older versions use the original Android stock browser

Priority Description and URL Guideline and Line# Count

Expand all 0 issues

Gambar 6.7 Hasil Pengujian Compatibility Menggunakan Tool SortSite

Pada Gambar 6.7 menampilkan hasil pengujian yang dilakukan menggunakan *tool* SortSite. Dari hasil tersebut didapatkan hasil uji bahwa sistem dapat bekerja pada 8 *browser* meliputi IE, Edge, Firefox, Safari, Chrome, Opera, Ios, Android dan tidak ada problem didalamnya.

BAB 7 PENUTUP

7.1 Kesimpulan

Dalam pengembangan sistem informasi perawatan luka praktik mandiri pedis care malang berbasis web diperoleh kesimpulan yang akan dijabarkan di bawah ini, diantaranya:

1. Didasarkan pada hasil analisis kebutuhan yang dibutuhkan dalam penelitian pengembangan sistem informasi perawatan luka praktik mandiri pedis care malang berbasis web diperoleh 59 kebutuhan fungsional dan 1 kebutuhan non fungsional. Pada kebutuhan fungsional yang sudah dijelaskan, akan dimodelkan dalam analisis *use case diagram* yang menghasilkan 59 *use case* yaitu alur aktor ketika berinteraksi dengan sistem di tiap-tiap fungsi akan dijelaskan dalam *use case scenario*.
2. Didasarkan hasil perancangan dalam penelitian didapatkan 4 perancangan, yaitu terdapat perancangan arsitektur, data, komponen dan antarmuka sistem. Perancangan arsitektur akan merancang rancangan *sequence diagram* dan *class diagram*. Sedangkan perancangan data dibuat dengan gambaran *Entity Relationship Diagram* (ERD) yang menghasilkan 33 entitas kemudian diilustrasikan ke dalam bentuk *Physical Data Model* (PDM). Pada perancangan komponen dihasilkan *pseudocode* yaitu perancangan yang fungsi untuk menentukan alur kode program. Pada perancangan antarmuka terdapat 3 sampel yang menggambarkan rancangan antarmuka pada sistem perawatan luka praktik mandiri pedis care malang berbasis web yaitu antarmuka halaman betes jensen, antarmuka halaman ganti *password* dan antarmuka halaman *delete informed content*.
3. Didasarkan pada hasil implementasi, pertama diperoleh spesifikasi sistem yang di dalamnya memuat, spesifikasi perangkat keras, spesifikasi perangkat lunak dan spesifikasi OS. Kedua yaitu implementasi data terdapat *physical data model* sebagai hasil desain implementasi sistem. ketiga yaitu implementasi kode program didapatkan berdasarkan hasil dari perancangan komponen. Keempat yaitu implementasi antarmuka berdasarkan rancangan antarmuka yang sudah dibuat.
4. Didasarkan pada hasil pengujian yang sudah dilakukan, pengujian unit dilakukan dengan metode *white box testing* yang nantinya akan dihasilkan nilai yang valid terhadap keseluruhan jalur uji yang ada. Pengujian integrasi yang dilakukan dengan metode *white box testing* juga berhasil mengintegrasikan method-method yang telah diuji. Pada pengujian validasi menggunakan metode *black box testing* terhadap 59 kebutuhan beserta kondisi alternatifnya yang menghasilkan nilai valid di semua *test case* sehingga sistem informasi perawatan luka praktik mandiri pedis care malang dapat berjalan dengan baik. Sedangkan pada *compatibility testing*

yang dikerjakan tidak terdapat problem didalamnya. Sistem dijalankan di semua jenis *browser* termasuk android dan ios.

7.2 Saran

Dari kesimpulan diatas, saran yang dapat diberikan pada penelitian selanjutnya, terkait Pengembangan Sistem Informasi Perawatan Luka Praktik Mandiri Pedis Care Malang Berbasis Web yaitu sistem ini dapat dikembangkan dengan *platform mobile* seperti android atau iOS. Diharapkan sistem dapat menyediakan fitur-fitur yang lebih lengkap dan dikembangkan menggunakan layout tampilan design yang lebih tertata.



DAFTAR PUSTAKA

A.S, R. and Shalanudin, M., 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: BI-Obses.

Afdaliyah, N., 2019. Pengembangan Aplikasi Web Rekam Medis Studi Kasus RSIA. Prof. Dr. H. 3(1), pp.267–273.

Almunawar, M.N. and Anshari, M., 2015. Health Information Systems (HIS): Concept and Technology. [online] (March). Available at: <<http://arxiv.org/abs/1203.3923>>.

Bisono, T.A., Wijanarko, T. and Putra, A., 2018. Perancangan Sistem Pendaftaran Klien Rawat Jalan Rumah Sakit Melalui Short Message Service Gateway (Studi Kasus : RS. Panti Wilasa “ Dr. Cipto ” Semarang). 16(1), pp.53–66.

Gunadarma, U., 2015. *Rapid Application Development*.

Kendall, K.E., and Kendall, J.E., 2011. *Systems Analysis and Design*. 8th Edition. New York: Pearson Education.

Kristiawan, A., 2012. Sistem Informasi Rekam Medis Pada Klinik Mitra Medika Semarang.

Kristiawan, A., 2017. Jurnal sistem informasi rekam medis pada klinik paramitha medika.

Marchewka, J. T., 2015. *Information Technology Project Management*. 5th ed. Hoboken: John Wiley.

Nugroho, A., 2010. *Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP*. Yogyakarta: Andi.

Pemenkes no 269, 2008. *Peraturan menteri kesehatan Republik Indonesia nomor 269/MENKES/PER/III/2008*.

Pressman, R.s., 2010. *Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku I*. Yogyakarta: Andi

Rosa, A.S., & Shalahuddin. M., 2014. *Rekayasa Perangkat lunak (Terstruktur dan Berorientasi Objek)*. Bandung : Informatika.

Satzinger, J. W., Jackson, R. B. & Burd, S. D., 2012. *System Analysis And Design In A Changing World*. 6th ed. Boston: Joe Sabatino.

shalahuddin, M., et al., 2013. *Rekayasa Perangkat Lunak : Terstruktur dan Berorientasi Objek*. Bandung: Informatika.

Susanto, E. and Sugiharto, 2017. *Manajemen informasi kesehatan IV : Etika Profesi dan Hukum Kesehatan*.

Yakub. 2012. *Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu

LAMPIRAN A FORM PENDUKUNG

A.1 Roadmap Pengerjaan Skripsi Awal

Nama : Siska Irawati

Judul : Pengembangan Sistem Informasi Perawatan Luka Di Praktik

Mandiri Pedis care Malang Berbasis Web

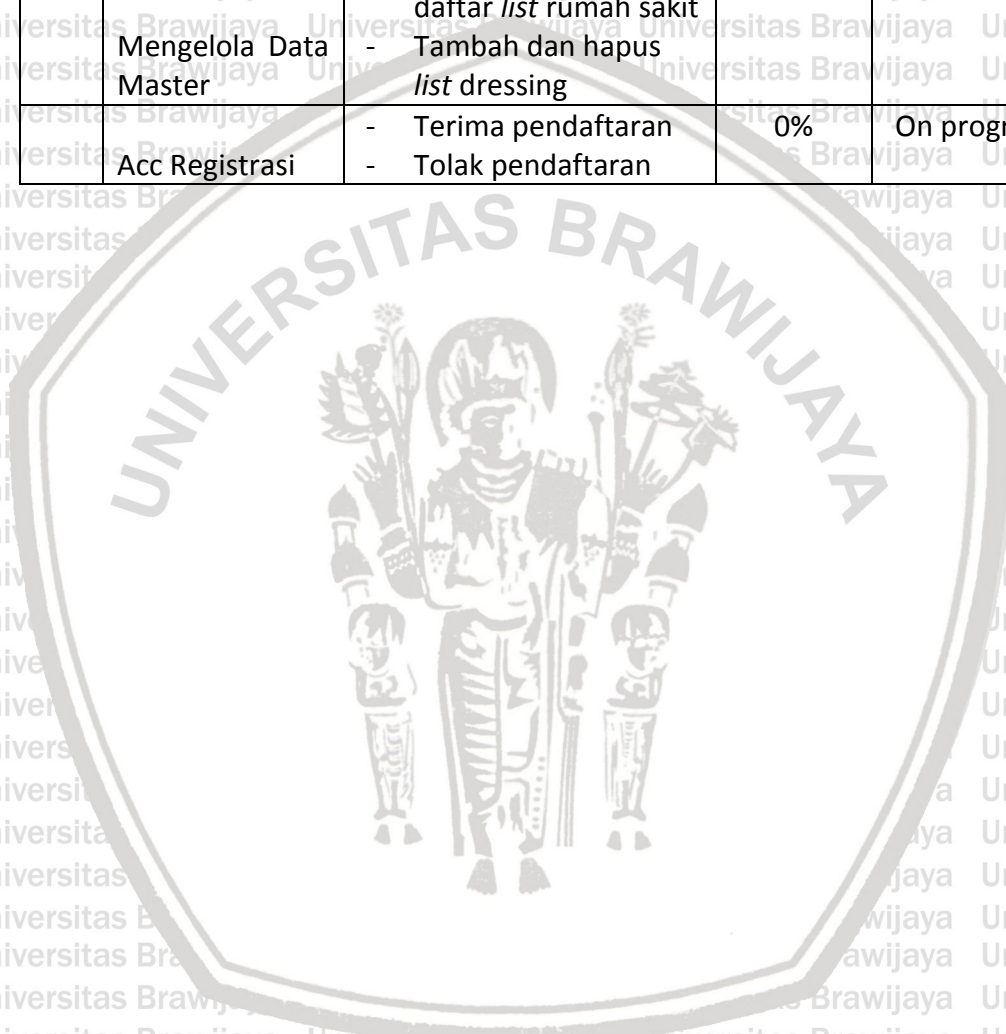
Status Proses Skripsi : p0/p1/p2

Tanggal : 01/02/2019

Status : Progress Awal

No	Deskripsi Task	Milestone Progress	Progress	Status
1	Intro			
	Register	- Mendaftar akun pegawai baru	0%	On progress
	Login	-	0%	On progress
	Profile	- Melihat profile akun	0%	On progress
	Ganti Password	- Mengubah Password	0%	On progress
2	Rekam Medis			
	Pendaftaran klien	- Tambah, hapus, ubah identitas klien - Tambah, hapus, ubah informed content	0%	On progress
	Diagnosa	- Tambah, hapus, ubah Pengkajian umum - Tambah, hapus, ubah pengkajian luka - Tambah, hapus, ubah tampilan klinis luka - Tambah, hapus, ubah betes jensen	0%	On progress
	Tindakan	- Tambah, hapus, ubah form tindakan	0%	On progress
	Kunjungan	- Tambah, hapus, ubah Form Kunjungan	0%	On progress
3	Transaksi			
	Pembayaran	- Tambah dan hapus pembayaran	0%	On progress
4	Administrator			
	Mengelola Data Pegawai	- Tambah sanksi - Reset password - Ubah hak akses - Hapus akun pegawai	0%	On progress

Mengelola Data Master	- Tambah dan hapus daftar status	0%	On progress
	- Tambah dan hapus daftar bidang		
	- Tambah dan hapus daftar pendidikan		
	- Tambah dan hapus daftar agama		
	- Tambah dan hapus daftar <i>list</i> rumah sakit		
Acc Registrasi	- Tambah dan hapus <i>list</i> dressing		
	- Terima pendaftaran	0%	On progress
	- Tolak pendaftaran		



A.2 Roadmap Pengerjaan Skripsi Iterasi 1

Nama : Siska Irawati

Judul : Pengembangan Sistem Informasi Perawatan Luka Di Praktik

Mandiri Pedis care Malang Berbasis Web

Status Proses Skripsi : p0/p1/p2

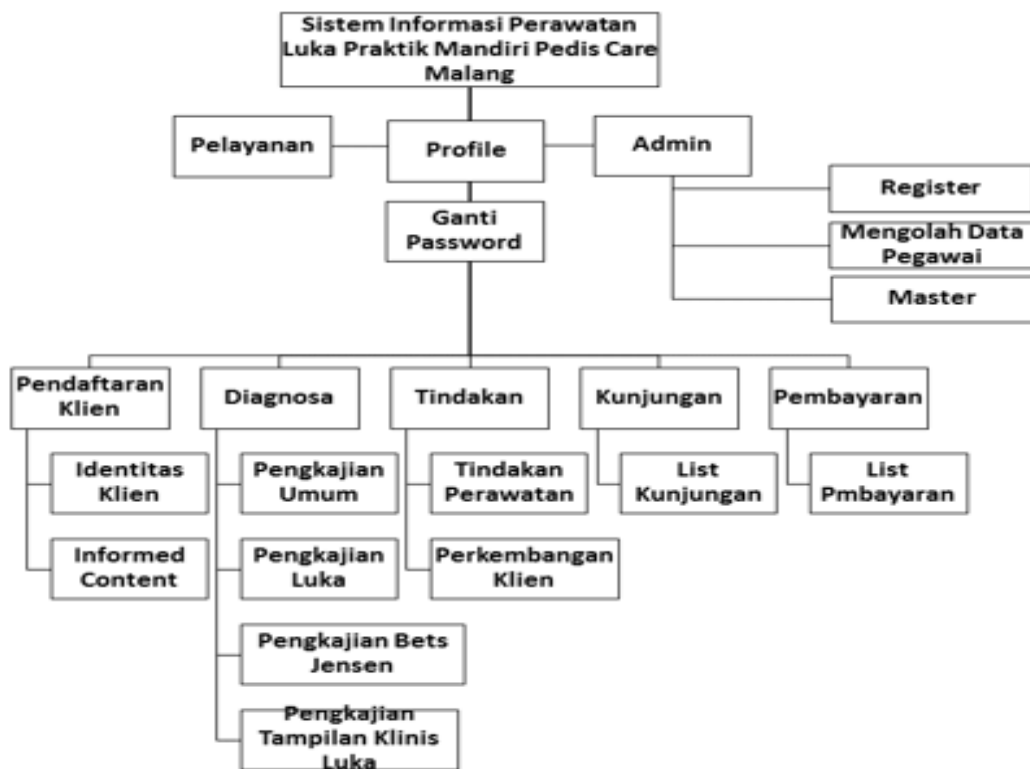
Tanggal : 20/03/2019

Status : Iterasi 1

No	Deskripsi Task	Milestone Progress	Progress	Status
1	Intro			
	Register	- Mendaftar akun pegawai baru	100%	Finish
	Login	-	100%	Finish
	Profile	- Melihat profile akun	100%	Finish
	Ganti Password	- Mengubah Password	100%	Finish
2	Rekam Medis			
		- Tambah, hapus, ubah identitas klien - Tambah, hapus, ubah informed content		
		Request Pengguna : - Field pada halaman informed content harusnya tidak perlu mengisi lagi nama klien - Pada form informed content sediakan upload berkas foto		
	Pendaftaran klien		70%	On Progress
		- Tambah, hapus, ubah Pengkajian umum - Tambah, hapus, ubah pengkajian luka - Tambah, hapus, ubah tampilan klinis luka - Tambah, hapus, ubah betes jensen		On Progress
	Diagnosa		60%	
		Request Pengguna : Tambah form perkembangan klien		On Progress
		- Tambah, hapus, ubah form tindakan		
	Tindakan	- Tambah, hapus, ubah form perkembangan klien	70%	
	Kunjungan	- Tambah, hapus, ubah form kunjungan	100%	Finish
3	Transaksi			
	Pembayaran	- Tambah dan hapus pembayaran	50%	On Progress

4	Administrator			
	Mengelola Data Pegawai	- Tambah sanksi - Reset password - Ubah hak akses - Hapus akun pegawai	100%	Finish
	Mengelola Data Master	- Tambah dan hapus daftar status - Tambah dan hapus daftar bidang - Tambah dan hapus daftar pendidikan - Tambah dan hapus daftar agama - Tambah dan hapus daftar <i>list</i> rumah sakit - Tambah dan hapus <i>list</i> dressing	100%	Finish
	Acc Registrasi	- Terima pendaftaran - Tolak pendaftaran	100%	Finish

A.3 Perancangan Work Breakdown Structure (WBS) Iterasi 1



Gambar Work Breakdown Structure Iterasi ke 1

A.4 Roadmap Pengerjaan Skripsi Iterasi 2

Nama : Siska Irawati

Judul : Pengembangan Sistem Informasi Perawatan Luka Di Praktik

Mandiri Pedis care Malang Berbasis Web

Status Proses Skripsi : p0/p1/p2

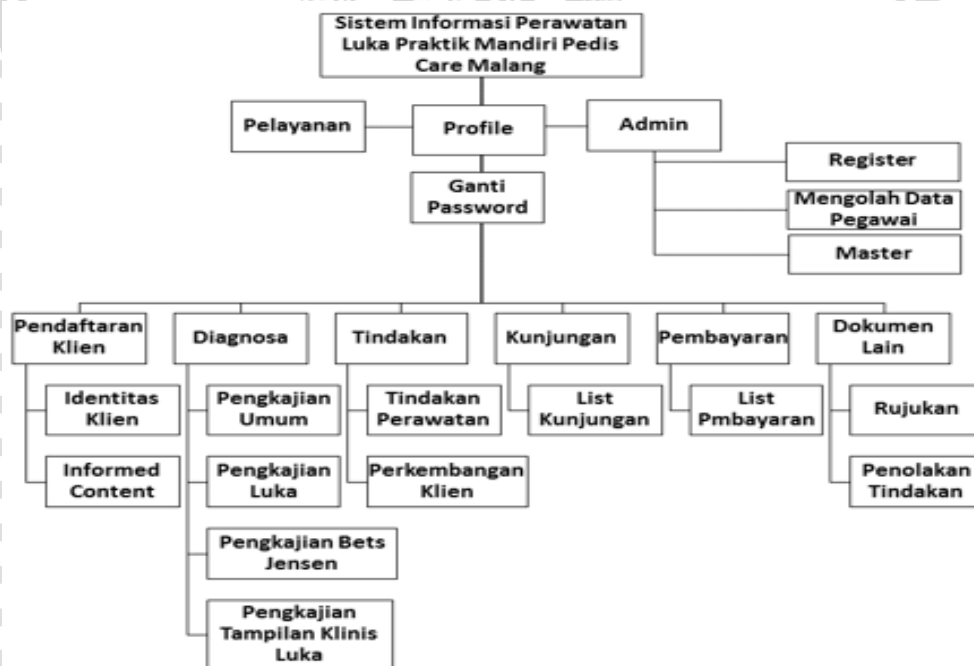
Tanggal : 02/04/2019

Status : Iterasi 2

No	Deskripsi Task	Milestone Progress	Progress	Status
1	Intro			
	Register	- Mendaftar akun pegawai baru	100%	Finish
	Login	-	100%	Finish
	Profile	- Melihat profile akun	100%	Finish
	Ganti Password	- Mengubah Password	100%	Finish
2	Rekam Medis			
		- Tambah, hapus, ubah identitas klien - Tambah, hapus, ubah informed content		
		Request Pengguna :		
		- Field pada halaman informed content harusnya tidak perlu mengisi lagi nama klien - Pada form informed content sediakan upload berkas foto	100%	Finish
	Pendaftaran klien			
		- Tambah, hapus, ubah Pengkajian umum - Tambah, hapus, ubah pengkajian luka - Tambah, hapus, ubah tampilan klinis luka - Tambah, hapus, ubah betes jensen	100%	Finish
	Diagnosa			
		Request Pengguna : Tambah form perkembangan klien		On Progress
		- Tambah, hapus, ubah form tindakan - Tambah, hapus, ubah form perkembangan klien	70%	
	Tindakan			
		- Tambah, hapus, ubah form kunjungan	100%	Finish
	Kunjungan			
		Request Pengguna :		On Progress
		- Tambah form penolakan tindakan dan rujukan	80%	
	Dokumen Urgency			

		- Tambah, hapus, ubah penolakan tindakan - Tambah, hapus, ubah rujukan		
3	Transaksi			
	Pembayaran	- Tambah dan hapus pembayaran	100%	Finish
4	Administrator			
	Mengelola Data Pegawai	- Tambah sanksi - Reset password - Ubah hak akses - Hapus akun pegawai	100%	Finish
	Mengelola Data Master	- Tambah dan hapus daftar status - Tambah dan hapus daftar bidang - Tambah dan hapus daftar pendidikan - Tambah dan hapus daftar agama - Tambah dan hapus daftar <i>list</i> rumah sakit - Tambah dan hapus <i>list</i> dressing	100%	Finish
	Acc Registrasi	- Terima pendaftaran - Tolak pendaftaran	100%	Finish

A.5 Perancangan Work Breakdown Structure (WBS) Iterasi 2



Gambar Work Breakdown Structure Iterasi ke 2

A.6 Roadmap Pengerjaan Skripsi Iterasi 3

Nama : Siska Irawati

Judul : Pengembangan Sistem Informasi Perawatan Luka Di Praktik

Mandiri Pedis care Malang Berbasis Web

Status Proses Skripsi : p0/p1/p2

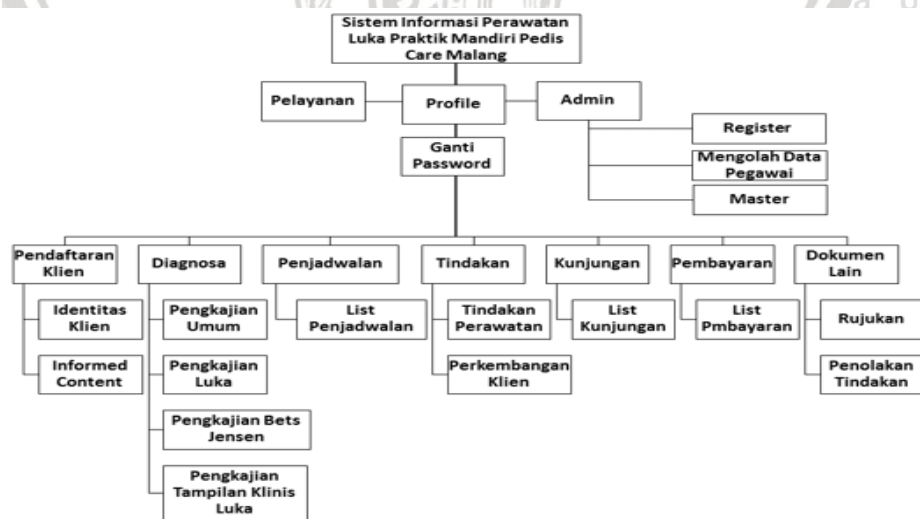
Tanggal : 15/10/2019

Status : Iterasi 3

No	Deskripsi Task	Milestone Progress	Progress	Status
1	Intro			
	Register	- Mendaftar akun pegawai baru	100%	Finish
	Login	-	100%	Finish
	Profile	- Melihat profile akun	100%	Finish
	Ganti Password	- Mengubah Password	100%	Finish
2	Rekam Medis			
		- Tambah, hapus, ubah identitas klien - Tambah, hapus, ubah informed content		
		Request Pengguna :		
		- Field pada halaman informed content harusnya tidak perlu mengisi lagi nama klien - Pada form informed content sediakan upload berkas foto	100%	Finish
	Pendaftaran klien			
		- Tambah, hapus, ubah Pengkajian umum - Tambah, hapus, ubah pengkajian luka - Tambah, hapus, ubah tampilan klinis luka - Tambah, hapus, ubah betes jensen	100%	Finish
	Diagnosa			
		Request Pengguna : Tambah form perkembangan klien		Finish
		- Tambah, hapus, ubah form tindakan - Tambah, hapus, ubah form perkembangan klien	100%	Finish
	Tindakan			
		- Tambah, hapus, ubah form kunjungan	100%	Finish
	Kunjungan			
		Request Pengguna :		Finish
		- Tambah form penolakan tindakan dan rujukan	100%	
	Dokumen Urgency			

		- Tambah, hapus, ubah penolakan tindakan - Tambah, hapus, ubah rujukan		
		Request Pengguna :		On Progress
		- Tambah Menu penjadwalan - Tambah & hapus schedule penjadwalan berobat	5%	
3	Transaksi			
	Pembayaran	- Tambah dan hapus pembayaran	100%	Finish
4	Administrator			
	Mengelola Data Pegawai	- Tambah sanksi - Reset password - Ubah hak akses - Hapus akun pegawai	100%	Finish
	Mengelola Data Master	- Tambah dan hapus daftar status - Tambah dan hapus daftar bidang - Tambah dan hapus daftar pendidikan - Tambah dan hapus daftar agama - Tambah dan hapus daftar <i>list</i> rumah sakit - Tambah dan hapus <i>list</i> dressing	100%	Finish
	Acc Registrasi	- Terima pendaftaran - Tolak pendaftaran	100%	Finish

A.7 Perancangan Work Breakdown Structure (WBS) Iterasi 3



Gambar Work Breakdown Structure Iterasi ke 2

A.8 Roadmap Pengerjaan Skripsi Hasil Akhir

Nama : Siska Irawati

Judul : Pengembangan Sistem Informasi Perawatan Luka Di Praktik

Mandiri Pedis care Malang Berbasis Web

Status Proses Skripsi : p0/p1/p2

Tanggal : 22/10/2019

Status : Hasil Akhir

No	Deskripsi Task	Milestone Progress	Progress	Status
1	Intro			
	Register	- Mendaftar akun pegawai baru	100%	Finish
	Login	-	100%	Finish
	Profile	- Melihat profile akun	100%	Finish
	Ganti Password	- Mengubah Password	100%	Finish
2	Rekam Medis			
		- Tambah, hapus, ubah identitas klien - Tambah, hapus, ubah informed content		
		Request Pengguna :		
		- Field pada halaman informed content harusnya tidak perlu mengisi lagi nama klien - Pada form informed content sediakan upload berkas foto	100%	Finish
	Pendaftaran klien			
		- Tambah, hapus, ubah Pengkajian umum - Tambah, hapus, ubah pengkajian luka - Tambah, hapus, ubah tampilan klinis luka - Tambah, hapus, ubah betes jensen	100%	Finish
	Diagnosa			
		Request Pengguna : Tambah form perkembangan klien		Finish
		- Tambah, hapus, ubah form tindakan - Tambah, hapus, ubah form perkembangan klien	100%	Finish
	Tindakan			
		- Tambah, hapus, ubah form kunjungan	100%	Finish
	Kunjungan			
		Request Pengguna :		Finish
		- Tambah form penolakan tindakan dan rujukan	100%	Finish
	Dokumen Urgency			

		- Tambah, hapus, ubah penolakan tindakan - Tambah, hapus, ubah rujukan		
		Request Pengguna :		On Progress
		Tambah Menu penjadwalan		
	Penjadwalan	Tambah & hapus schedule penjadwalan berobat	100%	
3	Transaksi			
	Pembayaran	- Tambah dan hapus pembayaran	100%	Finish
4	Administrator			
	Mengelola Data Pegawai	- Tambah sanksi - Reset password - Ubah hak akses - Hapus akun pegawai	100%	Finish
	Mengelola Data Master	- Tambah dan hapus daftar status - Tambah dan hapus daftar bidang - Tambah dan hapus daftar pendidikan - Tambah dan hapus daftar agama - Tambah dan hapus daftar <i>list</i> rumah sakit - Tambah dan hapus <i>list</i> dressing	100%	Finish
	Acc Registrasi	- Terima pendaftaran - Tolak pendaftaran	100%	Finish

A.9 Transkrip Wawancara 1

Nama Narasumber : Tubagus
Jabatan Narasumber : Koordinator Divisi Pelayanan
Tanggal Wawancara : 28 Januari 2019
Waktu Wawancara : 10.00 WIB
Lokasi Wawancara : Praktik Mandiri Pedis Care Jalan Mayjend Panjaitan
no. 68 A Malang jawa timur
Topik Wawancara : Identifikasi permasalahan tempat praktik

1. Pedis care itu tempat praktik untuk penanganan apa ya Pak?
 - Pedis care ini tempat praktik mandiri yang menangani perawatan luka yang kronis untuk memperoleh kesehatan yang lebih baik.
2. Perawatan luka apa saja yang biasanya ditangani dalam praktik mandiri pedis care?
 - Banyak, perawatan luka diabetes, perawatan luka bakar, perawatan luka kanker, perawatan luka infeksi, perawatan luka tekan (decubitus), dan disini juga terdapat perawatan kaki dan food SPA, cek gula, kolesterol, asam urat, tensi.
3. Disini ditempat praktik mandiri pedis care ada apa saja kah Pak?
 - Disini divisi ada 4 yaitu divisi pelayanan, divisi marketing, rumah tangga, dan farmasi mas yang dippimpin seorang CEO dan wakilnya manajer Operasional.
4. Apa kendala yang biasanya terjadi dalam tempat praktik pedis care ini Pak?
 - Kita kesulitan dalam penanganan klien dalam dokumentasinya yang masih manual. Selama ini kita menginginkan data tiap klien dapat terintegrasi satu tempat dan sampai sekarang masih belum bisa terlaksana. Dan mekanisme penulisan ulang yang memberatkan biasanya sampai merasakan capek yang terlalu, soalnya penanganan satu klien disini saja sudah 2-3 jam an lebih ditambah harus menulis data klinis klien pada kertas nanti perminggunya direkap lagi ke google form.
 - Sulitnya berbagi informasi data, kadang ditemukan beberapa staff pegawai yang memerlukan data klien terburu-buru untuk melihat riwayat pemeriksaan sebelumnya biasanya menghubungi saya dulu tapi ya gitu dikontaknya lewat chat atau telpon untuk minta soft *file* nya. Kadang saya kan tidak memegang hp ketika melakukan perawatan atau sedang libur jadi agak susah aja bagi saya yang harus buka datanya lagi di laptop.
5. Bagaimana alur pemeriksaan pasien baru sampai akhirnya dirawat di tempat ini Pak?
 - pasien datang menemui bagian rumah tangga yang jaga di meja depan dekat pintu masuk, kemudian dari bagian rumah tangga

disambungkan ke bagian pelayanan untuk ditangani lebih lanjut. Dan pasien mengisi beberapa dokumen seperti dokumen informed content, informasi data diri pasien, diagnosa, dan mengisi tindakan yang diberikan pada klien hingga akhir proses pembayaran.

6. kira-kira dalam hal perbaikan sistem, hal yang ingin dirubah itu apa saja pak?

- Kami membutuhkan sistem yang bisa kelola inventaris seperti barang-barang klinik dan kelola keuangan.

7. Kalau dalam hal rekam medis itu tidak perlu dirubah biar penanganan pasien lebih fleksibel dan efisien lagi pak?

- Sejauh ini memang perlu tapi saya belum tau bentuknya seperti apa itu sistem-sistem yang seperti itu jadi kita ya selama ini masih catatat manual dikertas.

8. Sejarah berdirinya tempat pedis care ini seperti apa ya Pak hingga bisa berdiri seperti sekarang?

- Pedis care itu berdiri pada tahun 2015. Awalnya CEO kami mendirikan tempat ini karena sepengetahuannya di malang masih belum ada tempat praktek/ klinik yang menangani luka. Pasien pada tahun awal dibuka masih dua orang dan pegawainya dua orang dengan semua tugas yang bersangkutan dikerjakan oleh 2 orang tersebut. Sekarag tahun 2019 jumlah pegawai adalah 10, bagian pelayanannya 4, marketing 2, rumah tangga 1, dan freeland 3. Disini selain melakukan perawatan klien juga adaagenda diluar seperti seminar untuk update ilmu dan pelatihan. Pada tahun 2016 pasien berkembang menjadi 20 orang dan sekarang 38 orang. Praktik mandiri pedis care berbeda dengan praktek klinik dokter umum. Kalau praktek dokter umum biasanya pasien datang dipriksa, diberi resep langsung pulang. Kalau di tempat kami tidak, dalam merawat kien ditangani mulai dri cucui lukanya, dibersihkan bagian yang kotor sampai estimasi waktu penanganan klien paling lama tiga jam. Dan perawatan bisa dilakukan dirumah pasien.

A.10 Transkrip Wawancara 2

Nama Narasumber : Tubagus
 Jabatan Narasumber : Koordinator Divisi Pelayanan
 Tanggal Wawancara : 28 November 2019
 Waktu Wawancara : 13.00 WIB
 Lokasi Wawancara : Praktik Mandiri Pedis Care Jalan Mayjend Panjaitan no. 68 A Malang Jawa timur
 Topik Wawancara : Identifikasi *form* yang dikembangkan

1. Apa perbedaan antara *form* Pengkajian umum, pengkajian luka dan tampilan klinis luka serta *form* betes jensen?

- Pengkajian umum : Pengkajian secara umum, jadi klien datang itu kita gatau ya dia membuat riwayat apa pokoknya pastinya datang kesini dia datang dengan membawa riwayat luka dan tapi sebelum luka pastinya terdapat komplikasi-komplikasi lain yang kita gatau makanya harus ada pengkajian umum.
- Pengkjaian luka :pengkajian luka ini lebih mengerucut ke fokus lukanya
- Pengkajian betes jensen : Pengkajian betes jensen berfungsi untuk memprediksi lama penyembuhan, ini secara umumpengertian yang lain yaitu alat ukur untuk prediksi kira-kira penyembuhannya kira-kira berapa bulan.
- Tampilan klinis luka : lebih merujuk ke lukanya contoh dari panjangnya luka, lebar luka dan dasar luka jadi lebih ke lukanya

2. Penjelasan ulang mengenai prosedur penanganan klien ?

- Untuk pasien baru Biasanya menghubungi, kita biasanya tanya terlebih dahulu dapat informasi darimana apakah dari instagram atau medsos lain atau kemungkinan orangnya sering lewat ditempat di sekitar tempat praktik kita. Atau bisa jadi taunya di siaran TV di batu TV trus mungkin juga pas kita pengabdian masyarakat itu pasien bisa tau makanyamau ke klinik kita. Pengabdian masyarakat biasanya kita mengadakan pemeriksaan gratis seperti pemeriksaan gula darah, asam urat, kolesterol dll tujuannya adalah untuk marketing juga biar dikenal banyak orang. Jadi balik lagi ke prosesnya tadi yaitu, pasien datang ke pedis care sebelum datang biasanya menghubungi kita CP yang dihubungi jadi masing-masing tim pelayanan disini mempunyai CP jadi nanti dihubungkannya ke CP siapa maka itu yang akan berhubungan dengan klien yang baru. CP tadi melaporkan bahwa ada pasien baru dan biasanya kalau ada pasien baru dilaporkan dulu ke saya sebagai koordinator divisi pelayanan untuk mengshcedule jadwal besoknya untuk dilakukan perawatan atau minta hari ini bisa, tapi yang peting

lapor dulu ke bagian koordinator divisi pelayanan supaya pelayananyang dilakukan bisa berjalan mulusdan tidak terjadi bentrok jadwal.

3. Mengenai bentrok jadwal misalnya ada klien baru yang mau berobat, sedangkan disini sudah ada klien lama yang masih ditangani, kira-kira gimana prosedur pelayanannya lagi kalau terjadi hal seperti itu pak?

- Jadi tetep, karena luka sifatnya bukan emergency sebenarnya ya,kalau emergency kan mengancam nyawa kalau luka biasanya jarang mengancam nyawa. Kalau mengancam nyawa pasti saya rujukkan ke rumah sakit jadi luka itu jarang ada yang mengancam nyawa, tapi tetap akan kita tangani dengan prosedural selesainya pasien lama yang harus ditangani duluan. Antisipasi yang terjadi bila terjadi pasien baru yang tiba-tiba datang ke tempat sedangkan kamar masih full digunakan pasien lama yang masih berada dalam proses penanganan,pasien barunya kita beri edukasi untuk penjadwalan.

4. Prosedur penanganan pasien lama seperti apa?

- Pasien lama melakukan penjadwalan dulu ke kita baru bisa berobat dihari yang sudah kita tentukan untuk menghindari bentrok jadwal,kalau misal minta sekarang juga bisa tapi kita juga melihat SDM yang ada di tempat praktik, jika SDM nya mampu maka kita akan mengiyakan permintaan klien untuk bisa berobat langsung atau home care.

5. Berdasarkan form pengkajian umum, pengkajian luka,pengkajian betes jensen dan pengkajian tampilan klinis luka apakah form itu semua hanya diisi sekali saja sebagai riwayat form pertama pendaftaran? Untuk agenda berobat selanjutnya apakah form tersebut masih diisi atau sudah ganti ke form yang lain?

- Gausah, jadi *form-form* itu tadi adalah *form* rekam medis awal klien. Untuk laporan kelanjutan progres luka selama perawatan dengan kita itu kita ngisi ke *form* yang ada di *google form* yaitu *form* perkembangan klien.

6. Berdasarkan *form* yang saya lihat salah satunya terdapat *form* tindakan, itu kapan diisinya, trus apa bedanya dengan *form* perkembangan klien?

- Jadi untuk *form* tindakan, ketika klien datang (klien lama),kalau klien lama kan berarti kita sudah tau perkembangan luka sebelumnya dan terapi apa yang sudah kita berikan ke klien jadi untuk yang perawatan hari ini kita tulis sendiri di *form* perkembangan klien.
- Jadi klien lama itu diisinya di *googe form* yang diisi itu perkembanganluka klien termasuk obat yang akan kita berikan. Sedangkan klien baru tetap diisi di *form hard copy* di *form* tindakan

sebagai tindakan awal yang kita berikan setelah dilakukan pengkajian.

7. Cara pengisian form gimana pas sedang dilakukannya penanganan klien?

- Kalau SDM nya mencukupi ada asisten yang bagian menulis hasil perawatan kalau kurang mencukupi biasanya ditulis sendiri sebelum tindakan contoh yang bagian diisi itu seperti form identitas klien, form informed content, dan pengkajian umum, tapi kalau yang pengkajian luka, tampilan klinis luka dan betes jensen ngisinya harus buka luka klien terlebih dahulu untuk tau dalamnya dan ngisinya itu di sela-selama membalut atau selesai membalut.



A.11 Halaman Sampul



Kode :

NAMA PASIEN

No. Registrasi

A.12 Halaman Informed Content

PEDIS CARE

Pusat Perawatan Luka, Diabetes, dan Stoma

Sebagai dan Divisi: Kardiologi, Nefrologi

Jl. Mayjen Panjaitan no. 68A Malang, no. telp : (0341) 567022, email : pediscaremalang@gmail.com



CP. 2a

INFORMED CONCENT

Surat Pernyataan Persetujuan

Yang bertanda tangan di bawah ini,

Nama Lengkap :
 Jenis Kelamin :
 Tanggal Lahir / Usia :
 Alamat :
 No. Telp / HP :

Bertindak sebagai wakil / wali keluarga klien, dengan identitas klien sebagai berikut,

Nama Klien :
 Jenis kelamin :
 Tanggal Lahir / Usia :
 Diagnosa :

Menyatakan SETUJU untuk dilakukan tindakan Atas klien tersebut di atas,
 dengan telah diinformasikannya terlebih dahulu tentang kondisi pasien tersebut dan saya mengerti serta
 memahami konsekuensi dari tindakan tersebut, sesuai dengan penjelasan yang telah di berikan tenaga
 kesehatan yang melakukan perawatannya.

Demikianlah surat ini saya buat dengan penuh kesadaran dan tanpa tekanan dari pihak manapun.

Malang,

A.13 Halaman Identitas Klien



PEDIS CARE
 Pusat Perawatan Luka, Diabetes, dan Stoma
Sebagai dan Dokter Tera Sertifikasi
 Jl. Mayjen Panjaitan no. 69A Malang, no. telp : (0341) 567822, email : pediscaremalang@gmail.com

GP. 3

IDENTITAS KLIEN

No. RM
 1. DATA KLIEN
 Nama Klien :
 Tanggal Lahir / Usia :
 Alamat :
 No. Telp / No. Hp :
 No. Fax / Email :
 Agama :
 Status :
 Jenis Kelamin :
 Kebangsaan :
 Penjamin : () Pribadi () Jaminan, jika Ya Jenis :

2. DATA PENANGGUNGJAWAB KLIEN
 Nama Penjamin :
 Tanggal Lahir / Usia :
 Alamat :
 No. Telp / No. Hp :
 No. Fax / Email :
 Hub. Dengan Klien :

3. DATA RUJUKAN
 Rujukan : Sendiri / Brosur / Saudara / Rekan (Nama Saudara/ Rekan)
 Rujukan Dari : Dokter / RS / Lain2 :

RENCANA PROGRAM () Perawatan Luka (PL) () Lain-lain :
 Malang,

(.....) (.....)
 Nama Petugas Nama Klien/Keluarga

A.15 Halaman Pengkajian Umum

CP. 5a

PEDIS CARE
 Pusat Perawatan Luka, Diabetes, dan Stoma
Specialist dan Dokter Rawat Nadi Akut
 Jl. Mayjen Porjitan no. 68A Malang, no. telp : (0341) 587622, email : pediscaremalang@gmail.com

PENGKAJIAN UMUM

1. Informasi Klien		Tgl lahir/umur	Jenis Kelamin
Nama :		Pekerjaan	Agama
Alamat :			
Telepon :		Sumber	Suku
Penanggung Jawab :		Pembiayaan	
Alamat :			
Dokter yang merawat atau merujuk :		Diagnosa Medis :	
Telepon / Hg :		Tanggal dirawat :	
Dokter Konsultan :		Advice Terapi :	
.....			

2. Riwayat Kesehatan Masa Lalu Pernah dirawat ☐ Ya ☐ Tidak Jelaskan Riwayat Operasi ☐ Ya ☐ Tidak Jelaskan Riwayat ☐ Ya ☐ Tidak Jelaskan	5. Sistem Kardiovaskuler ☐ Takikardi ☐ Bradikardi ☐ Disfonasia ☐ Palpitasi ☐ Edema ☐ Akut hangat ☐ CRT = 2 sek ☐ Nyeri dada ☐ Pucat ☐ Mammur ☐ Lainnya Jelaskan :
3. Riwayat Penyakit Keluarga ☐ Hipertensi ☐ Asma ☐ Diabetes ☐ Lainnya Jelaskan	6. Sistem Respirasi ☐ Sesak ☐ Batuk ☐ Sirosis ☐ Wheezing ☐ Ronchi ☐ Vesikuler ☐ Cuping Hidung ☐ Respirasi otot bantu nafas ☐ Lainnya Jelaskan :
4. Status Kesehatan Saat Ini Keluhan Utama Keluhan Umum Keadaan : Tanda Vital : TD mmHg, N x/mn RR x/mn, S °C Status Gizi : TB cm, BB kg	7. Sistem Pencernaan ☐ Mual ☐ Muntah ☐ Stomatitis ☐ Asidosis ☐ Hepatomegali ☐ Distensi Abdomen ☐ Diare ☐ Konstipasi ☐ Nyeri Abdomen ☐ Anoreksia ☐ Peristaltik usus x / m

A.16 Halaman Pengkajian Luka

PEDIS CARE
Pusat Perawatan Luka, Diabetes, dan Stoma
Sangat dan Didukung Tim Stoma Ahli
Jl. Mayjen Panjaitan no. 58A Malang, no. telp : (0341) 597622, email : pediscaremalang@gmail.com



CP. 5c

PENGKAJIAN LUKA (Tampilan Klinis Luka)
(Lengkapi pengkajian awal dan uji kontrol jika warna dasar luka berubah atau interval 7-10 hari, gunakan
skore yang tertera untuk setiap luka individu)

Nomor Luka												
Tanggal Pengkajian												
Analgetik (pre dressing)	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N
Ukuran luka												
Panjang (cm/mm)												
Lebar (cm/mm)												
Dalam (cm/mm)												
Determining / lubang / goa												
Tracking												
Tepi luka												
Warna Dasar luka (masukkan %)												
Nekrosis (hitam)												
Slough (kuning)												
Granulasi (Merah)												
Epitelisasi (Merah Muda)												
Hypergranulasi (Merah merentang)												
Infeksi (Kuning Hijau)												
Lain Lain (Sebutkan) =												
Tepi dan Jenis eksudat (tandai kotak yang sesuai)												
Serius												
Blood												
Haemorrhagic												
Purulent												
Haemopurulent												
Odour / Bau												
Lain - Lain (sebutkan)												
Jumlah Eksudat : Banyak (B) Sedang (S)												
Sedikit (S) Tidak Ada (T)												
Kulit Sekitar Luka (Tandai kotak yang sesuai)												
Normal												
Kering												
Berkilap												
Odematous												
Macerasi												
Kallus												
Erythema												
Rasa Sakit												
Tidak Ada (T), Sedikit (S), Sedang (S), Sangat												
Sakit (SS)												
Frakturasi :												
Constant (C), Terbatas (T)												
Stadium Luka (1,2,3,4)												

A.17 Halaman Betes Jensen

PEDIS CARE
Pusat Perawatan Luka, Diabetes, dan Stroke
Smangrai dan Dindia; Kawi Nalati - Adu
Jl. Mayjen Panjaitan no. 68A Malang, no. telp : (0341) 587022, email : pediscaremalang@gmail.com


Good Wound Care, Better Outcome

CP. 66

PENSKAJIAN BETES-JENSEN
(Isi for pedis awal pengkajian atau rangkai luka baru)

Bates	Pengkajian	Tgl :			
		No. Luka :	No. Luka :	No. Luka :	No. Luka :
1. Ukuran Luka	1 = P x L < 4 cm 2 = P x L 4-16 cm 3 = P x L 16-36 cm 4 = P x L 36-80 cm 5 = P x L > 80 cm				
2. Kedalaman	1 = Tidak ada erythema pada kulit sehat 2 = Kehilangan kulit "partial thickness" melibatkan epidermis dan atau dermis 3 = Kehilangan kulit "full thickness" melibatkan kerusakan / nekrosis 4 = Tidak jelas akibat nekrosis 5 = Kehilangan kulit "full thickness" dengan kerusakan yang luas, jaringan nekrosis atau kerusakan otot, tulang atau struktur yang mendukung				
3. Tepi Luka	1 = samar, tidak terlihat jelas 2 = batas tepi terlihat, menyatu dengan dasar luka 3 = jelas, tidak menyatu dengan dasar luka 4 = jelas, tidak menyatu dengan dasar luka, tebal 5 = Jelas, fibrotik, perit tebal hiperkeratosis				
4. GDA	1 = tidak ada 2 = gos < 2 cm di area manapun 3 = gos 2 - 4 cm < 50 % pinggir luka 4 = gos 2 - 4 cm > 50 % pinggir luka 5 = gos > 4 cm di area manapun				
5. Tipe Eksudat	1 = Tidak Ada 2 = Bloody 3 = Serosanguineous 4 = Serous 5 = Purulent				
6. Jumlah Eksudat	1 = Kering 2 = Moist				

A.18 Halaman Tindakan

[illegible]