

**PENGEMBANGAN SISTEM INFORMASI LAYANAN
PEMERIKSAAN KESEHATAN TENAGA KERJA DENGAN
METODE RATIONAL UNIFIED PROCESS
(STUDI KASUS KLINIK ARGARAYA MEDIKA MALANG)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Alfan Babtista Yordan
145150400111041



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

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI LAYANAN PEMERIKSAAN KESEHATAN
TENAGA KERJA DENGAN METODE RATIONAL UNIFIED PROCESS
(STUDI KASUS KLINIK ARGARAYA MEDIKA MALANG)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Alfan Babtista Yordan
145150400111041

Skripsi ini telah diuji dan dinyatakan lulus pada
3 Januari 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Yusi Tyroni Mursityo S.Kom., M.A.B.
NIP: 19800228 200604 1 001

Dosen Pembimbing II



Djoko Pramono, S.T., M.Kom.
NIP: 19780108 200501 1 002

Mengetahui
Ketua Jurusan Sistem Informasi



Dr. Eng. Herman Tolle, S.T., M.T.
NIP: 19740823 200012 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 pustaka.

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

Malang, 3 Januari 2019



Alfan Babtista Jordan

NIM: 145150400111041

KATA PENGANTAR

Puji dan syukur kehadiran Tuhan Yang Maha Esa atas berkat rahmat serta kasih-Nya sehingga penulis dapat menyelesaikan skripsi. Naskah skripsi ini diajukan untuk memenuhi salah satu syarat untuk memperoleh gelar sarjana. Judul yang penulis ajukan sebagai skripsi adalah “PENGEMBANGAN SISTEM INFORMASI LAYANAN PEMERIKSAAN KESEHATAN TENAGA KERJA DENGAN METODE RATIONAL UNIFIED PROCESS (STUDI KASUS KLINIK ARGARAYA MEDIKA MALANG)”.

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

1. Bapak Yusi Tyroni Mursityo, S.Kom., M.AB. selaku Dosen pembimbing pertama yang telah membimbing dan memberikan banyak solusi dalam pengerjaan skripsi.
2. Djoko Pramono, S.T., M.Kom. selaku Dosen pembimbing kedua yang telah membimbing dan memberikan banyak masukan selama pengerjaan skripsi.
3. Bapak Mochamad Chandra Saputra , S.Kom., M.Eng. selaku pembimbing terdahulu yang telah meluangkan waktunya untuk memberikan bimbingan dalam pengerjaan skripsi.
4. Bapak Dr. Eng. Herman Tolle, S.T, M.T selaku Ketua Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya Malang.
5. Bapak Ismiarta Aknuranda, S.T, M.Sc, Ph.D selaku sekertaris Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya Malang.
6. Bapak Yusi Tyroni Mursityo, S.Kom., M.AB. selaku Ketua Program Studi Sistem Informasi Universitas Brawijaya Malang.
7. Kedua orang tua saya Sutaji dan Ninil Ambarwati dan adik saya Yonanta Anastasya Sorgawi yang telah memberi segenap dukungan dan sangat memotivasi saya sampai saat ini.
8. Seluruh Dosen dan seluruh civitas akademika Fakultas Ilmu Komputer yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Universitas Brawijaya.
9. Bapak Setyo Budi Parmono selaku narasumber Klinik Argaraya Medika Malang yang bersedia membimbing dan mengizinkan serta membantu penulis untuk melakukan penelitian ini.

10. Seluruh teman-teman yang tidak dapat penulis sebutkan satu per satu yang telah memberikan semangat dan dukungan agar skripsi ini cepat terselesaikan.

Semoga Tuhan Yang Maha Esa melimpahkan rahmat berlipat ganda kepada seluruh pihak yang membantu dan berkontribusi pada proses penelitian ini. Tidak lupa saya sampaikan bahwa penelitian yang telah dilakukan memiliki banyak kekurangan, diskusi mengenai penelitian lanjutan, saran, dan kritik yang membangun merupakan beberapa hal yang saya harapkan dapat disampaikan kepada saya.

Malang, 3 Januari 2019



Alfan Babtista Yordan

babtistayordan07@gmail.com

ABSTRAK

Malang salah satu wilayah di Jawa Timur penyumbang tenaga kerja migran tertinggi di Indonesia dapat menjadi sebuah keuntungan bagi Klinik Argaraya Medika dalam mengembangkan bisnis dan pelayanan pemeriksaan kesehatan tenaga kerja. Namun banyaknya jumlah pasien yang mendaftarkan diri untuk pemeriksaan kesehatan tenaga kerja setiap jam pelayanan membuat petugas cukup kesulitan menangani berkas pemeriksaan dan antrian pasien. Meningkatnya jumlah pasien juga membuat staff pelayanan pemeriksaan kesehatan serta dokter kebingungan ketika akan memberikan pelayanan atau melakukan pemeriksaan kesehatan karena sumber daya yang kurang sehingga terkadang membuat salah satu proses pemeriksaan terlewat. Proses pendaftaran pasien layanan pemeriksaan kesehatan yang dilakukan dua kali pada front office dan poliklinik dirasa kurang efisien dan memperlama proses pendaftaran. Proses perekapan hasil pemeriksaan dilakukan dengan pengumpulan seluruh formulir hasil pemeriksaan kesehatan pada bagian administrasi lalu dilakukan pemasukan data pada MS Excel. Hilangnya salah satu formulir prosedur pemeriksaan pasien menjadi masalah yang sering dihadapi sehingga harus mengulangi prosedur pemeriksaan yang telah dilakukan karena proses perekapan belum dilakukan.

Berdasarkan masalah yang diuraikan sebelumnya, maka dapat diberikan solusi untuk mengembangkan sistem informasi layanan pemeriksaan kesehatan tenaga kerja. Sistem dikembangkan dengan metode *Rational Unified Process* (RUP) dengan tahapan *fase inception, elaboration, construction, dan transition*. Hasil pengujian validasi dengan 8 kasus uji untuk menguji fungsional sistem dengan hasil 100% valid dan *compability testing* untuk menguji kebutuhan non fungsional didapat hasil 2 critical issues pada 2 aplikasi peramban. *User Acceptance Testing* yang dilakukan berdasarkan fungsional masing-masing pengguna. Staf petugas *front office* memperoleh presentase nilai sebesar 82%, petugas poliklinik memperoleh presentase nilai sebesar 74%, petugas laboratorium memperoleh presentase nilai sebesar 77%, dan petugas radiologi memperoleh presentase nilai sebesar 85% menunjukkan sistem informasi layanan pemeriksaan kesehatan tenaga kerja dapat diterima dengan baik oleh pengguna.

Kata kunci: *sistem informasi, rational unified proces, pemeriksaan kesehatan, tenaga kerja, proses bisnis*

ABSTRACT

Malang, one of the regions in East Java that contributes the highest number of migrant workers in Indonesia can be an advantage for the Argaraya Medika Clinic in developing business and labor inspection services. However, the large number of patients who register for the health check-up of workers every hour of service makes it difficult for officers to handle the inspection files and queue of patients. The increasing number of patients also makes the health examination service staff and doctors confused when going to provide services or conduct health checks because of lack of resources so that sometimes one of the inspection processes is missed. The process of registering patients for health screening services conducted twice at the front office and polyclinic is considered inefficient and prolongs the registration process. The process of recapitulation of the results of the examination is carried out by collecting all forms of health examination results in the administration section and then entering data into MS Excel. The loss of one of the patient examination procedure forms is a problem that is often faced so it must repeat the examination procedures that have been carried out because the recording process has not been carried out.

Based on the problems described earlier, a solution can be given to develop an information system for labor inspection health services. The system was developed using the Rational Unified Process (RUP) method with the stages of inception, elaboration, construction, and transition phases. The results of the validation test with 8 test cases to test the functional system with 100% valid results and compability testing to test non-functional requirements obtained from 2 critical issues in 2 browser applications. User Acceptance Testing is carried out based on the functionalities of each user. Front office staff received a percentage value of 82%, polyclinic officers obtained a percentage of 74%, laboratory officers obtained a percentage of 77%, and radiology officers obtained a percentage value of 85% indicating that the health inspection service information system was well received by the user.

Keywords: *information systems, rational unified processes, health checks, labor, business processes*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR LAMPIRAN	xix
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	4
1.3 Tujuan	4
1.4 Manfaat.....	5
1.5 Batasan masalah	5
1.6 Sistematika pembahasan.....	5
BAB 2 LANDASAN KEPUSTAKAAN	7
2.1 Kajian Pustaka	7
2.2 Profile Klinik Argaraya Medika Malang.....	13
2.2.1 Visi dan Misi	14
2.2.2 Struktur Organisasi.....	14
2.3 Sistem Informasi	15
2.4 <i>Software Development Life Cycle (SDLC)</i>	15
2.5 <i>Rational Unified Process (RUP)</i>	15
2.6 <i>Business Process</i>	18
2.6.1 Analisis dan Pemodelan Proses Bisnis <i>As-Is</i>	19
2.6.2 Analisis dan Pemodelan Proses Bisnis <i>To-Be</i>	19
2.7 <i>Business Process Modeling Notation (BPMN)</i>	19
2.7.1 <i>Flow Object</i>	19
2.7.2 <i>Connections</i>	21



2.7.3 Swimlanes	22
2.7.4 Artifacts	22
2.8 Pemodelan <i>Use Case</i>	23
2.8.1 Identifikasi Tipe Pemangku Kepentingan.....	23
2.8.2 Analisis Masalah	24
2.8.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna .	24
2.8.4 Identifikasi Fitur	25
2.8.5 Identifikasi Persyaratan Fungsional dan Persyaratan Nonfungsional	25
2.8.6 Permodelan <i>Use Case</i>	25
2.9 <i>Unified Modeling Language (UML)</i>	26
2.9.1 <i>Use Case Diagram</i>	27
2.9.2 <i>Sequence Diagram</i>	28
2.9.3 <i>Class Diagram</i>	29
2.10 <i>Physical Data Model (PDM)</i>	31
2.11 <i>Black Box Testing</i>	31
2.12 <i>Compatibility Testing</i>	32
2.13 <i>User Acceptance Testing (UAT)</i>	32
2.14 Skala Likert	33
BAB 3 METODOLOGI	35
3.1 Studi Literatur	36
3.2 Pengumpulan Data	36
3.3 Metode <i>Rational Unified Process (RUP)</i>	36
3.3.1 Fase Insepsi	37
3.3.2 Fase Elaboration.....	37
3.3.3 Fase Konstruksi.....	38
3.3.4 Fase Transisi	38
3.4 Kesimpulan dan Saran	38
BAB 4 Analisis kebutuhan	39
4.1 Pemodelan Proses Bisnis	39
4.1.1 Pemodelan Proses Bisnis <i>As-Is</i>	39
4.1.2 Pemodelan Proses Bisnis <i>To-Be</i>	43



4.1.3 Analisis <i>Improvement</i> Waktu Proses Bisnis	50
4.2 Analisis Persyaratan	51
4.2.1 Identifikasi Tipe Pemangku Kepentingan.....	51
4.2.2 Analisis Permasalahan.....	52
4.2.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna .	53
4.2.4 Identifikasi Pengguna	58
4.2.5 Identifikasi Fitur	59
4.2.6 Persyaratan Fungsional	61
4.2.7 Persyaratan Non Fungsional	63
4.3 Pemodelan <i>Use Case</i>	64
4.3.1 <i>Use Case Diagram</i>	65
4.3.2 Identifikasi Aktor	66
4.3.3 Spesifikasi <i>Use Case</i>	67
4.4 Kesimpulan Analisis Kebutuhan Fase Insepsi Metode Rational Unifield Process.....	86
BAB 5 perancangan	87
5.1 Pemodelan Interaksi Objek.....	87
5.1.1 <i>Sequence Diagram</i> Menambah Data Pendaftaran Baru.....	87
5.1.2 <i>Sequence Diagram</i> Menambah Data Pemeriksaan Baru	88
5.1.3 <i>Sequence Diagram</i> Mengubah Data Hasil Akhir Pemeriksaan ...	89
5.1.4 <i>Sequence Diagram</i> Mengisi Data Pemeriksaan Fisik dan Kejiwaan	90
5.2 Pemodelan Objek.....	91
5.2.1 <i>Diagram Class</i> Analisis	91
5.2.2 <i>Diagram Class</i> Perancangan.....	92
5.3 Perancangan <i>Database</i>	94
5.4 Perancangan Antarmuka	95
5.4.1 Perancangan Antarmuka <i>Home</i>	95
5.4.2 Perancangan Antarmuka Daftar Pasien Harian	96
5.4.3 Perancangan Antarmuka Tambah Data Pendaftaran Pasien.....	97
5.4.4 Perancangan Antarmuka Hasil Seluruh Pemeriksaan	98
5.5 Perancangan Algoritme	99

5.5.1	Algoritme Menambah Data Pendaftaran Pasien Baru.....	99
5.5.2	Algoritme Menambah Data Pemeriksaan Baru Pasien.....	101
5.5.3	Algoritme Mengubah Hasil Akhir Pemeriksaan Pasien.....	102
5.6	Kesimpulan Perancangan Fase Elaborasi Metode <i>Rational Unified Process</i>	104
BAB 6	IMPLEMENTASI	105
6.1	Implementasi Algoritme	105
6.1.1	Implementasi Algoritme Menambah Data Pendaftaran Pasien Baru	105
6.1.2	Implementasi Algoritme Menambah Data Pemeriksaan Baru .	106
6.1.3	Implementasi Algoritme Mengubah Hasil Akhir Pemeriksaan Pasien	107
6.2	Implementasi Antarmuka Pengguna	108
6.2.1	Antarmuka Halaman <i>Home</i>	109
6.2.2	Antarmuka Halaman Daftar Pasien Harian	109
6.2.3	Antarmuka Halaman Pendaftaran Pasien	110
6.2.4	Antarmuka Halaman Hasil Pemeriksaan Pasien	111
BAB 7	pengujian	112
7.1	Pengujian <i>Validation Testing</i>	112
7.1.1	Perancangan Pengujian <i>Validation Testing</i>	112
7.1.2	Hasil Pengujian <i>Validation Testing</i>	118
7.1.3	Kesimpulan <i>Validation Testing</i>	120
7.2	Pengujian <i>Compatibility Testing</i>	121
7.2.1	Perancangan Pengujian <i>Compatibility Testing</i>	121
7.2.2	Hasil Pengujian <i>Compatibility Testing</i>	125
7.2.3	Kesimpulan Pengujian <i>Compatibility Testing</i>	126
7.3	Kesimpulan Pengujian Fase Konstruksi Metode <i>Rational Unified Process</i>	126
7.4	Pengujian <i>User Acceptance Testing (UAT)</i>	126
7.4.1	Perancangan Pengujian <i>User Acceptance Testing (UAT)</i>	126
7.4.2	Hasil Pengujian <i>User Acceptance Testing (UAT)</i>	132
7.4.3	Kesimpulan Pengujian <i>User Acceptance Testing (UAT)</i>	142

7.5 Kesimpulan Pengujian Fase Transisi Metode <i>Rational Unifield Process</i>	142
BAB 8 penutup	143
8.1 Kesimpulan.....	143
8.2 Saran	144
DAFTAR PUSTAKA.....	145
LAMPIRAN WAWANCARA	147
LAMPIRAN DOKUMENTASI UAT.....	149



DAFTAR TABEL

Tabel 2.1 Literatur <i>Review</i>	9
Tabel 2.2 Simbol <i>event</i>	20
Tabel 2.3 Simbol <i>Activity</i>	20
Tabel 2.4 Simbol <i>Gateway</i>	21
Tabel 2.5 Simbol <i>Connections</i>	21
Tabel 2.6 Simbol <i>Swimlanes</i>	22
Tabel 2.7 Simbol <i>Artifacts</i>	22
Tabel 2.8 Kerangka Dokumentasi Pernyataan Masalah	24
Tabel 2.9 Identifikasi Fitur	25
Tabel 2.10 Identifikasi Fitur	26
Tabel 2.11 Simbol Use Case Diagram	27
Tabel 2.12 Simbol <i>Sequence Diagram</i>	28
Tabel 2.13 Simbol <i>Class Diagram</i>	30
Tabel 2.14 Bobot Nilai Jawaban	34
Tabel 2.15 Persentase Nilai Jawaban	34
Tabel 4.1 Perubahan Aktivitas Proses Bisnis Pendaftaran Pasien	44
Tabel 4.2 Perubahan Aktivitas Proses Bisnis Pemeriksaan Fisik dan Kejiwaan ...	46
Tabel 4.3 Perubahan Aktivitas Proses Bisnis Pemeriksaan Laboratorium	48
Tabel 4.4 Perubahan Aktivitas Proses Bisnis Pemeriksaan Radiologi	50
Tabel 4.5 Analisis Waktu	51
Tabel 4.6 Tipe Pemangku Kepentingan	51
Tabel 4.7 <i>Problem Statement</i>	52
Tabel 4.8 Identifikasi Kebutuhan Pemangku Kepentingan	54
Tabel 4.9 Perbaikan Identifikasi Kebutuhan Pemangku Kepentingan	57
Tabel 4.10 Hasil Identifikasi Pengguna	58
Tabel 4.11 Identifikasi Fitur	59
Tabel 4.12 Hubungan Kebutuhan Pengguna dengan Fitur	60
Tabel 4.13 Perbaikan Identifikasi Fitur	60
Tabel 4.14 Perbaikan Hubungan Kebutuhan Pengguna dengan Fitur	60
Tabel 4.15 Kebutuhan Fungsional	61

Tabel 4.16	Perbaikan Kebutuhan Fungsional.....	62
Tabel 4.17	Kebutuhan Non Fungsional	63
Tabel 4.18	Hubungan Aktivitas Proses bisnis <i>To-Be</i> dengan <i>Use Case</i>	65
Tabel 4.19	Hubungan <i>Use Case</i> dengan Fitur	66
Tabel 4.20	Identifikasi Aktor	67
Tabel 4.21	<i>Spesifikasi Use Case Login</i>	68
Tabel 4.22	<i>Spesifikasi Use Case</i> Melihat Data Pendaftaran Pasien	68
Tabel 4.23	<i>Usecase Scenario</i> Menambah Data Pendaftaran Baru	69
Tabel 4.24	<i>Usecase Scenario</i> Mengubah Data Diri Pasien	70
Tabel 4.25	<i>Usecase Scenario</i> Menambah Data Pemeriksaan Baru.....	71
Tabel 4.26	<i>Usecase Scenario</i> Mengubah Data Pemeriksaan Pasien.....	72
Tabel 4.27	<i>Usecase Scenario</i> Melihat Data Pemeriksaan Pasien.....	73
Tabel 4.28	<i>Usecase Scenario</i> Mengisi Data Hasil Pemeriksaan Fisik	74
Tabel 4.29	<i>Usecase Scenario</i> Melihat Data Hasil Pemeriksaan Fisik.....	74
Tabel 4.30	<i>Usecase Scenario</i> Mengubah Data Hasil Pemeriksaan Fisik.....	75
Tabel 4.31	<i>Usecase Scenario</i> Mengisi Data Hasil Pemeriksaan Laboratorium	76
Tabel 4.32	<i>Usecase Scenario</i> Melihat Data Hasil Pemeriksaan Laboratorium	77
Tabel 4.33	<i>Usecase Scenario</i> Mengubah Data Hasil Pemeriksaan Laboratorium	78
Tabel 4.34	<i>Usecase Scenario</i> Mengisi Data Hasil Pemeriksaan Radiologi	79
Tabel 4.35	<i>Usecase Scenario</i> Melihat Data Hasil Pemeriksaan Radiologi.....	79
Tabel 4.36	<i>Usecase Scenario</i> Mengubah Data Hasil Pemeriksaan Radiologi.....	80
Tabel 4.37	<i>Usecase Scenario</i> Menambah Data Pengguna	81
Tabel 4.38	<i>Usecase Scenario</i> Mengubah Data Pengguna	82
Tabel 4.39	<i>Usecase Scenario</i> Menghapus Data Pengguna.....	83
Tabel 4.40	<i>Usecase Scenario</i> Menghapus Data Pemeriksaan.....	83
Tabel 4.41	<i>Usecase Scenario</i> Mengubah Hasil Akhir Pemeriksaan.....	84
Tabel 4.42	<i>Usecase Scenario</i> Melihat Pelaporan Data Pemeriksaan	85
Tabel 5.1	<i>Pseudocode</i> Menambah Data Pendaftaran Pasien Baru.....	100
Tabel 5.2	Algoritma Menambah Data Pemeriksaan Baru Pasien	101
Tabel 5.3	<i>Pseudocode</i> Mengubah Hasil Akhir Pemeriksaan Pasien	104
Tabel 6.1	Implementasi Algoritme Menambah Data Pendaftaran Pasien Baru	105
Tabel 6.2	Implementasi Menambah Data Pemeriksaan Baru	106



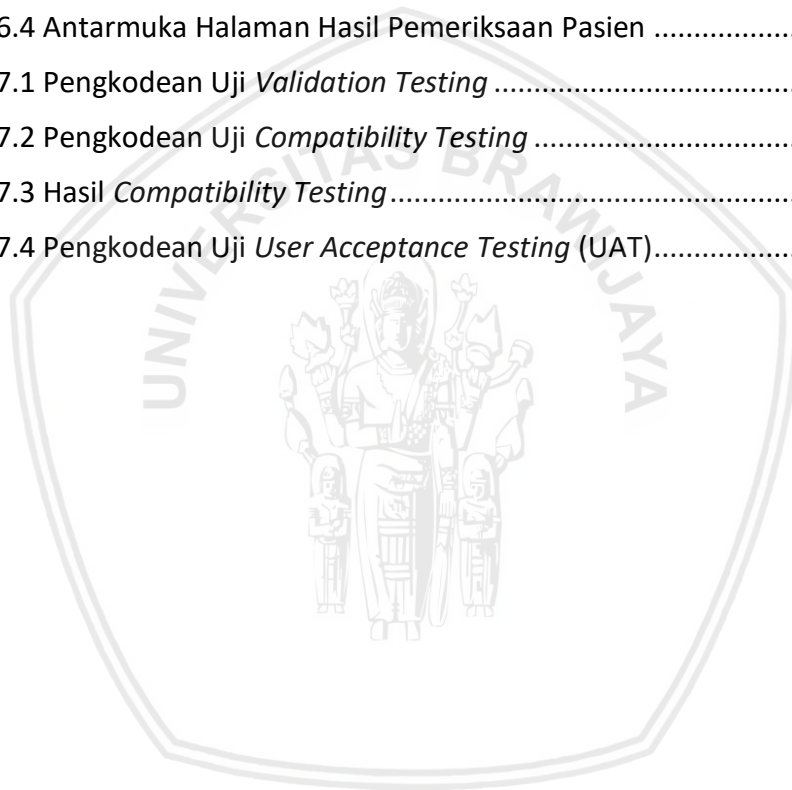
Tabel 6.3 Implementasi Algoritme Mengubah Hasil Akhir Pemeriksaan Pasien	107
Tabel 7.1 Rancangan Tampilan Hasil Pengujian <i>Validation Testing</i>	113
Tabel 7.2 Rancangan Skenario Testing Menambah Data Pendaftaran Baru	113
Tabel 7.3 Rancangan <i>Validation Testing</i> Menambah Data Pendaftaran Baru ..	113
Tabel 7.4 Rancangan <i>Validation Testing</i> Inputan Data Pendaftaran Baru Tidak	114
Tabel 7.5 Rancangan Skenario Testing Menambah Data Pemeriksaan Baru	114
Tabel 7.6 Rancangan <i>Validation Testing</i> Menambah Data Pemeriksaan Baru..	114
Tabel 7.7 Rancangan <i>Validation Testing</i> Inputan Data Pemeriksaan Baru Tidak Sesuai	115
Tabel 7.8 Rancangan Skenario Testing Mengubah Data Hasil Akhir Pemeriksaan	116
Tabel 7.9 Rancangan <i>Validation Testing</i> Mengubah Data Hasil Akhir Pemeriksaan	116
Tabel 7.10 Rancangan <i>Validation Testing</i> Mengubah Data Hasil Akhir Pemeriksaan Belum Lengkap	116
Tabel 7.11 Rancangan Skenario Testing Mengisi Data Pemeriksaan Fisik dan Kejiwaan	117
Tabel 7.12 Rancangan Skenario Testing Mengisi Data Pemeriksaan Fisik dan Kejiwaan	117
Tabel 7.13 Rancangan <i>Validation Testing</i> Inputan Data Pemeriksaan Fisik dan Kejiwaan Tidak Sesuai	118
Tabel 7.14 Hasil Pengujian <i>Validation Testing</i>	118
Tabel 7.15 Kategori Masalah Kompatilitas.....	121
Tabel 7.16 Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>Internet Explorer</i>	122
Tabel 7.17 Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>Microsoft Edge</i>	122
Tabel 7.18 Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>Firefox</i>	122
Tabel 7.19 Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>Safari</i>	123
Tabel 7.20 Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>Opera</i>	123
Tabel 7.21 Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>Chrome</i>	124

Tabel 7.22 Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>iOS</i>	124
Tabel 7.23 Rancangan Pengujian <i>Compatibility</i> Sistem Kasus Uji Peramban <i>Android</i>	124
Tabel 7.24 Tampilan Hasil Pengujian <i>Compatibility</i>	125
Tabel 7.25 Rancangan Kasus Uji UAT untuk Petugas <i>Front Office</i>	127
Tabel 7.26 Rancangan Kasus Uji UAT untuk Petugas Poliklinik.....	128
Tabel 7.27 Rancangan Kasus Uji UAT untuk Petugas Laboratorium.....	129
Tabel 7.28 Rancangan Kasus Uji UAT untuk Petugas Radiologi	130
Tabel 7.29 Rancangan Kasus Uji UAT untuk Petugas Administrasi.....	131
Tabel 7.30 Hasil Pengujian <i>User Acceptance Testing</i> untuk Petugas <i>Front Office</i>	132
Tabel 7.31 Hasil Pengujian <i>User Acceptance Testing</i> untuk Petugas <i>Front Office</i>	134
Tabel 7.32 Hasil Pengujian <i>User Acceptance Testing</i> untuk Petugas Poliklinik .	134
Tabel 7.33 Hasil Pengujian <i>User Acceptance Testing</i> untuk Petugas Poliklinik .	135
Tabel 7.34 Hasil Pengujian <i>User Acceptance Testing</i> untuk Petugas Laboratorium	136
Tabel 7.35 Hasil Pengujian <i>User Acceptance Testing</i> untuk Petugas Laboratorium	137
Tabel 7.36 Hasil Pengujian <i>User Acceptance Testing</i> untuk Petugas Radiologi.	138
Tabel 7.37 Hasil Pengujian <i>User Acceptance Testing</i> untuk Petugas Radiologi.	139
Tabel 7.38 Hasil Pengujian <i>User Acceptance Testing</i> untuk Petugas Administrasi	140
Tabel 7.39 Hasil Pengujian <i>User Acceptance Testing</i> untuk Petugas Administrasi	141

DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi Klink Argaraya Medika Malang	15
Gambar 2.2 Model Pengembangan RUP	16
Gambar 2.3 <i>Milestone</i> pada akhir setiap fase RUP.....	17
Gambar 2.4 Contoh <i>Class Diagram</i>	30
Gambar 3.1 Diagram Alur Metodologi.....	35
Gambar 4.1 Proses Bisnis <i>As-Is</i> Pendaftaran Pasien	40
Gambar 4.2 Proses Bisnis <i>As-Is</i> Pemeriksaan Fisik dan Kejiwaan	41
Gambar 4.3 Proses Bisnis <i>As-Is</i> Pemeriksaan Laboratorium.....	42
Gambar 4.4 Proses Bisnis <i>As-Is</i> Pemeriksaan Radiologi.....	43
Gambar 4.5 Proses Bisnis <i>To-be</i> Pendaftaran Pasien	44
Gambar 4.6 Proses Bisnis <i>To-be</i> Pemeriksaan Fisik dan Kejiwaan.....	46
Gambar 4.7 Proses Bisnis <i>To-be</i> Pemeriksaan Laboratorium	48
Gambar 4.8 Proses Bisnis <i>To-be</i> Pemeriksaan Radiologi	49
Gambar 4.9 Kodifikasi Kebutuhan Pengguna.....	54
Gambar 4.10 Kodifikasi Fitur	59
Gambar 4.11 Kodifikasi Kebutuhan Fungsional	61
Gambar 4.12 Kodifikasi Kebutuhan Non Fungsional	63
Gambar 4.13 Pemodelan <i>UseCase Diagram</i>	64
Gambar 5.1 <i>Sequence Diagram</i> Menambah Data Pendaftaran Baru.....	87
Gambar 5.2 <i>Sequence Diagram</i> Tambah Data Pemeriksaan Baru.....	88
Gambar 5.3 <i>Sequence Diagram</i> Mengubah Data Hasil Akhir Pemeriksaan	89
Gambar 5.4 <i>Sequence Diagram</i> Memperbarui Data Pemeriksaan Fisik dan Kejiwaan.....	90
Gambar 5.5 <i>Class Diagram</i> Analisis.....	92
Gambar 5.6 <i>Class Diagram</i> Perancangan <i>Controllers</i>	93
Gambar 5.7 <i>Class Diagram</i> Perancangan <i>Models</i>	94
Gambar 5.8 PDM Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja	95
Gambar 5.9 Perancangan Antarmuka <i>Home</i>	96
Gambar 5.10 Perancangan Antarmuka Daftar Pasien Harian	97

Gambar 5.11 Perancangan Antarmuka Tambah Data Pendaftaran Pasien.....	98
Gambar 5.12 Perancangan Antarmuka Hasil Seluruh Pemeriksaan	99
Gambar 5.13 <i>Flowchart</i> Menambah Data Pendaftaran Pasien Baru.....	100
Gambar 5.14 <i>Flowchart</i> Menambah Data Pemeriksaan Baru Pasien.....	102
Gambar 5.15 <i>Flowchart</i> Mengubah Hasil Akhir Pemeriksaan Pasien.....	104
Gambar 6.1 Antarmuka Halaman <i>Home</i>	109
Gambar 6.2 Antarmuka Halaman Daftar Pasien Harian	109
Gambar 6.3 Antarmuka Halaman Pendaftaran Pasien	110
Gambar 6.4 Antarmuka Halaman Hasil Pemeriksaan Pasien	111
Gambar 7.1 Pengkodean Uji <i>Validation Testing</i>	112
Gambar 7.2 Pengkodean Uji <i>Compatibility Testing</i>	121
Gambar 7.3 Hasil <i>Compatibility Testing</i>	126
Gambar 7.4 Pengkodean Uji <i>User Acceptance Testing (UAT)</i>	127



DAFTAR LAMPIRAN

LAMPIRAN WAWANCARA	147
A.1 Wawancara pengurus Administrasi & Kesetariatan.....	147
LAMPIRAN DOKUMENTASI UAT	149
B.1 UAT Petugas Poliklinik	149
B.2 UAT Petugas FrontOffice	151
B.3 UAT Petugas Radiologi.....	153
B.4 UAT Petugas Laboratorium.....	155
B.5 UAT Petugas Administrasi.....	157



BAB 1 PENDAHULUAN

1.1 Latar belakang

Dengan perkembangan Teknologi Informasi dan Komunikasi (TIK) yang cepat saat ini organisasi dituntut untuk dapat menerapkan TIK dalam proses bisnisnya. Penerapan TIK dalam proses bisnis organisasi dapat mengoptimalkan kinerja pelayanan serta meningkatkan daya saing instansi. Dalam hal ini Teknologi Informasi dan Komunikasi menjadi sebuah aset yang penting untuk pertumbuhan dan perkembangan organisasi.

Teknologi Informasi dan Komunikasi yang berkembang dengan pesat, juga mendorong berkembangnya sistem informasi. Sistem Informasi, dengan dukungan teknologi informasi, telah menjadi komponen penting dalam organisasi bisnis modern, karena mampu membantu dalam pengembangan bisnis dan mengelola keunggulan kompetitif. Sistem informasi dapat memberikan hasil yang lebih untuk output sebuah sistem, tentunya bila sistem di dalamnya telah berjalan dengan baik.

Argaraya Medika Malang merupakan klinik kesehatan yang memfokuskan pada pelayanan pemeriksaan kesehatan tenaga kerja dan pelayanan kesehatan masyarakat secara umum. Berawal dari Laboratorium Klinik Deksa Media yang berdiri tahun 2004, dan berganti nama menjadi Klinik Argaraya Medika tahun 2013 berperan sebagai penyelenggara pelayanan kesehatan tenaga kerja. Layanan kesehatan yang diberikan Klinik Argaraya Medika antara lain pemeriksaan kesehatan tenaga kerja, rawat jalan, UGD, rawat inap, dan farmasi. Layanan pemeriksaan kesehatan tenaga kerja merupakan pelayanan yang diunggulkan oleh klinik ini. Sehingga diperlukan perhatian khusus untuk menjaga dan meningkatkan kualitas mutu pelayanan pemeriksaan kesehatan tenaga kerja dalam perkembangan waktu. Berdasarkan BNP2TKI yang mempublikasikan Data Penempatan dan Perlindungan TKI Periode 1 JANUARI S.D 30 NOVEMBER 2018, Malang menjadi satu wilayah di Jawa Timur penyumbang tenaga kerja migran tertinggi di Indonesia. Selama periode Januari hingga Oktober 2018, sebanyak 53.525 Pekerja Migran Indonesia (PMI) berasal dari Jawa Timur. Kota Malang sendiri termasuk dalam 20 besar kabupaten dan kota asal PMI (Al Faruq, 2018). Malang salah satu wilayah di Jawa Timur penyumbang tenaga kerja migran tertinggi di Indonesia dapat menjadi sebuah keuntungan bagi Klinik Argaraya Medika dalam mengembangkan bisnis dan pelayanan pemeriksaan kesehatan tenaga kerja. Namun banyaknya jumlah pasien yang mendaftarkan diri untuk pemeriksaan kesehatan tenaga kerja setiap jam pelayanan membuat petugas cukup kesulitan menangani berkas pemeriksaan dan antrian pasien.

Pelayanan yang diberikan oleh Argaraya Medika masih belum menggunakan sistem yang terkomputerisasi, terutama untuk mendukung layanan pemeriksaan kesehatan tenaga kerja. Alur proses layanan dilakukan dimulai dengan pendaftaran pasien pada *front office* dan poliklinik lalu pasien memulai pemeriksaan kesehatan sesuai dengan permintaan pemeriksaan kesehatan yang

dibutuhkan. Pengisian hasil pemeriksaan kesehatan diisikan pada formulir kertas disetiap alur proses pemeriksaan kesehatan mulai dari pemeriksaan fisik dan kejiwaan, pemeriksaan laboratorium, hingga pemeriksaan radiologi. Proses perekapan hasil pemeriksaan dilakukan dengan pengumpulan seluruh formulir hasil pemeriksaan kesehatan pada bagian administrasi lalu dilakukan pemasukan data pada MS Excel. Hilangnya salah satu formulir prosedur pemeriksaan pasien menjadi masalah yang sering dihadapi sehingga harus mengulangi prosedur pemeriksaan yang telah dilakukan karena proses perekapan belum dilakukan. Penggunaan komputer pada saat ini sangat dibutuhkan karena mengingat semakin majunya teknologi dan perkembangan zaman yang menuntut kecepatan dan ketepatan dalam mengolah data serta informasi. Tapi sumber daya manusia (SDM) yang terampil dan bermutu juga sangat menunjang dalam memanfaatkan teknologi komputer. Dengan memanfaatkan teknologi komputer maka proses satu sistem dapat lebih efektif dan efisien atau dengan kata lain suatu sistem yang prosesnya dilakukan secara manual, maka proses sistem tersebut akan berjalan dengan lambat, disamping itu dalam pemrosesan sering terjadi kesalahan (Putro, 2011).

Perekapan data pemeriksaan kesehatan masih dilakukan dengan memasukan data satu persatu pada MS Excel, proses ini dirasa oleh admin kurang efisien waktu, sebelumnya di Argaraya Medika pernah menerapkan Sistem Informasi Rekam Medis namun dalam penggunaannya kurang optimal karena proses alur sistem yang ada di sistem kurang sesuai dengan proses bisnis yang berjalan di Argaraya Medika sehingga sistemnya tidak digunakan kembali. Pengolahan data secara manual, mempunyai banyak kelemahan, selain membutuhkan waktu yang lama, keakuratannya juga kurang dapat diterima, karena kemungkinan kesalahan sangat besar. Dengan dukungan teknologi informasi yang ada sekarang ini, pekerjaan pengolahan data dengan cara manual dapat digantikan dengan sistem informasi komputer (Lidya Andriani, 2009).

Berdasarkan hasil wawancara dengan Klinik Argaraya Medika proses bisnis yang berjalan saat ini dirasa mulai kurang efektif dan efisien. Hal ini disebabkan oleh meningkatnya jumlah pasien yang membutuhkan layanan pemeriksaan kesehatan khususnya tenaga kerja yang akan keluar negeri dan tenaga kerja diperusahaan-perusahaan yang mengharuskan pengecekan kesehatan terhadap karyawanya. Pengisian formulir prosedur pemeriksaan kesehatan dan perekapan data hasil pemeriksaan kesehatan yang masih dilakukan secara manual membutuhkan waktu yang lebih lama sementara jumlah pasien yang meningkat mengakibatkan antrian pasien yang panjang. Proses pendaftaran pasien layanan pemeriksaan kesehatan yang dilakukan dua kali pada *front office* dan poliklinik dirasa kurang efisien dan memperlama proses pendaftaran. Meningkatnya jumlah pasien juga membuat staff pelayanan pemeriksaan kesehatan serta dokter kebingungan ketika akan memberikan pelayanan atau melakukan pemeriksaan kesehatan karena sumber daya yang kurang sehingga terkadang membuat salah satu proses pemeriksaan terlewat. Sehingga saat perekapan data pasien kelengkapan pemeriksaan kesehatan yang dibutuhkan masih kurang. Untuk itu pasien dihubungi kembali untuk melakukan prosedur pemeriksaan kesehatan

yang terlewati. Masalah ini dapat mengakibatkan Klinik Argaraya Medika kehilangan costumernya karena pelayanan yang lama serta kesalahan-kesalahan yang timbul dari proses bisnis yang berjalan saat ini. Dibutuhkan adanya perbaikan serta peningkatan proses bisnis dalam layanan pemeriksaan kesehatan tenaga kerja agar layanan unggulan di Klinik Argaraya Medika ini kualitasnya terjaga dan juga dapat ditingkatkan.

Dari studi literatur yang penulis lakukan, terdapat penelitian yang pernah dikembangkan oleh peneliti sebelumnya, yaitu dilakukan oleh Johni S Pasaribu, Johnson Sihombing (2017) berjudul "PERANCANGAN SISTEM INFORMASI REKAM MEDIS PASIEN RAWAT JALAN BERBASIS WEB DI KLINIK SEHAT MARGASARI BANDUNG". Penelitian ini membahas perancangan sistem informasi rekam medis pasien rawat jalan berbasis web menggunakan metode RUP yang diterapkan di Klinik Sehat Margasari Bandung. Sistem ini dibuat karena dirasa kinerja sistem yang berjalan saat ini kurang efektif dan efisien, seperti pengolahan data pasien dan data rekam medis masih menggunakan media pembukuan atau manual, pengelolaan data pasien di Klinik Sehat Margasari masih belum efektif karena sistem yang digunakan kurang lengkap sehingga pelayanan pasien menjadi lambat dan rekam pasien sering hilang atau tidak ditemukan. Kesimpulan yang diberikan oleh peneliti adalah perancangan sistem informasi pasien rawat jalan ini bisa mempermudah petugas dalam menangani dan mengelola data pasien rawat jalan diharapkan dapat mempermudah dalam mengolah data dokter, pasien, obat, stok obat, transaksi dan mencetak kartu pasien yang berfungsi mempermudah administrasi pendaftaran pasien saat kembali berkunjung dengan menggunakan sistem pada komputer.

Rational Unified Process (RUP) adalah metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan *best practices* yang terdapat dalam pengembangan perangkat lunak. Metode ini memiliki ciri utama yaitu menggunakan *use-case driven* dan pendekatan iteratif. Iteratif disini merupakan proses pengulangan fase pada RUP diantaranya fase *inception*, *elaboration*, *construction* dan *transition*. Iterasi pada setiap tahap dengan tujuan untuk dapat mengakomodasi perubahan kebutuhan perangkat lunak. Pengembangan sistem informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja ini akan dilakukan secara bertahap dan melakukan perubahan atau penambahan fitur sesuai kebutuhan. Ashraf Anwar (2014) menyebutkan bahwa RUP merupakan proses yang dapat dikonfigurasi sehingga dapat cocok untuk pengembangan kecil maupun besar. Dapat bervariasi untuk mengakomodasi situasi yang berbeda serta memberikan dukungan untuk mengonfigurasi proses agar sesuai dengan kebutuhan organisasi tertentu. Salah satu keuntungan dalam penggunaan metode ini adalah pengoptimalan manajemen sehingga pengembangan dapat memenuhi persyaratan pengguna disampaikan secara teratur dan tepat waktu. Untuk itu metode ini dipilih penulis karena metode ini dapat mengakomodasi perubahan kebutuhan perangkat lunak dan setiap fase dalam RUP selalu dilakukan evaluasi agar risiko dan kesalahan sistem dapat diketahui sejak dini serta manajemen waktu yang tepat.

Berdasarkan uraian pernyataan diatas, peneliti tertarik untuk melakukan penelitian tentang pengembangan sistem informasi layanan pemeriksaan tenaga kerja, yang nantinya akan diterapkan di Klinik Argaraya Medika Malang sebagai rekomendasi proses bisnis yang lebih baik dengan penerapan sistem informasi. Judul penelitian yang diambil oleh peneliti yakni "PENGEMBANGAN SISTEM INFORMASI LAYANAN PEMERIKSAAN KESEHATAN TENAGA KERJA DENGAN METODE RATIONAL UNIFIED PROCESS (STUDI KASUS KLINIK ARGARAYA MEDIKA MALANG)". Dengan adanya penelitian ini diharapkan dapat meningkatkan kualitas dari pelayanan pemeriksaan kesehatan tenaga kerja yang ada di Klinik Argaraya Medika.

1.2 Rumusan masalah

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

1. Bagaimana analisis kebutuhan dalam mengembangkan sistem informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja pada fase inepsi?
2. Bagaimana hasil rancangan arsitektur sistem informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dalam fase elaborasi metode *Rational Unfied Process*?
3. Bagaimana implementasi rancangan arsitektur sistem informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dan hasil pengujian sistem menggunakan *validation testing* dan *compability testing* dalam fase konstruksi metode *Rational Unfied Process*?
4. Bagaimanakah hasil pengujian sistem informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja kepada pengguna menggunakan *user acceptance testing* pada fase transisi?

1.3 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini, ialah:

1. Mengetahui analisis kebutuhan dalam mengembangkan sistem informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja pada fase inepsi.
2. Mengetahui rancangan sistem informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja pada fase elaborasi.
3. Mengetahui hasil implementasi sistem informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dan hasil pengujian sistem dengan *validation testing* dan *compatibilty testing* pada fase konstruksi.
4. Mengetahui hasil pengujian sistem informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja kepada pengguna dengan *user accapetance testing* pada fase transisi.

1.4 Manfaat

Manfaat yang diharapkan didapat dari penelitian adalah sebagai berikut:

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

1.5 Batasan masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Perancangan sistem yang dibuat merupakan Sistem Informasi layanan pemeriksaan kesehatan tenaga kerja di Klinik Argaraya Medika Malang.
2. Data yang diperoleh merupakan hasil wawancara langsung dengan pihak Klinik Argaraya Medika Malang.
3. Perancangan sistem yang dilakukan adalah perancangan sistem berbasis website.

Penelitian ini dilakukan mulai analisis kebutuhan, perancangan, implementasi, hingga pengujian sistem.

1.6 Sistematika pembahasan

Sistematika dalam penyusunan skripsi ini adalah sebagai berikut:

1. BAB 1 : PENDAHULUAN

Pada bab ini menjelaskan tentang latar belakang masalah yang ada, rumusan masalah, tujuan, manfaat, batasan masalah serta sistematika penulisan yang di gunakan.

2. BAB 2 : LANDASAN KEPUSTAKAAN

Bab ini menjelaskan mengenai dasar teori yang digunakan dalam mendukung topik penelitian. Pada bab ini juga memuat kajian pustaka yang terdapat referensi skripsi ataupun jurnal dengan penelitian yang hampir sama atau relevan.

3. BAB 3 : METODOLOGI PENELITIAN

Pada bab ini menjelaskan langkah-langkah yang di tempuh atau metode penelitian yang digunakan dalam melakukan penelitian ini. Metode tersebut memuat dalam pengambilan data, dalam pengolahan data serta analisa dan hasil yang atau bahkan rekomendasi yang diberikan dari hasil penelitian ini.

4. BAB 4 : ANALISIS KEBUTUHAN

Bab yang memuat analisa kebutuhan sistem yang akan dibangun. Kebutuhan yang berhubungan baik kebutuhan fungsional maupun kebutuhan non-fungsional berdasarkan metodologi penelitian untuk sistem yang akan dibangun.

5. BAB 5 : PERANCANGAN

bab yang memuat perancangan perancangan hasil analisis kebutuhan yang akan digunakan sebagai dasar implementasi dan pengujian sistem.

6. BAB 6 : IMPLEMENTASI

Bab yang memuat impelementasi perangkat lunak sesuai dengan perancangan yang telah dibuat.

7. BAB 7 : PENGUJIAN

Bab yang memuat hasil pengujian berdasarkan rancangan pengujian sebelumnya. Pengujian *validation testing*, *compatibility testing* dan *user accptance testing*.

8. BAB 8 : PENUTUP

Bab yang membahas tentang kesimpulan dan saran dari proses perancangan dan implementasi sistem informasi yang dibangun.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Sebelum melakukan penelitian ini terdapat beberapa referensi baik dari skripsi sebelumnya maupun jurnal yang ditemukan. Referensi pertama yaitu berasal dari jurnal yang berjudul **“PERANCANGAN SISTEM INFORMASI REKAM MEDIS PASIEN RAWAT JALAN BERBASIS WEB DI KLINIK SEHAT MARGASARI BANDUNG”** Johni S Pasaribu, Johnson Sihombing (2017). Penelitian ini membahas perancangan sistem informasi rekam medis pasien rawat jalan berbasis web menggunakan metode RUP yang diterapkan di Klinik Sehat Margasari Bandung. Metode RUP digunakan mulai pada proses: pemodelan bisnis (perencanaan), pengumpulan kebutuhan, analisis dan desain, implementasi dan pengujian. Fase-fase evaluasi, lingkungan dan pengembangan, konfigurasi dan perubahan manajemen tidak diterapkan karena pembuatan perangkat lunak ini masih merupakan versi awal sehingga pada fase-fase tersebut dilakukan pada pengembangan selanjutnya. Sistem ini dibuat karena dirasa kinerja sistem yang berjalan saat ini kurang efektif dan efisien, seperti pengolahan data pasien dan data rekam medis masih menggunakan media pembukuan atau manual, pengelolaan data pasien di Klinik Sehat Margasari masih belum efektif karena sistem yang digunakan kurang lengkap sehingga pelayanan pasien menjadi lambat dan rekam pasien sering hilang atau tidak ditemukan. Kesimpulan yang diberikan oleh peneliti adalah perancangan sistem informasi pasien rawat jalan ini bisa mempermudah petugas dalam menangani dan mengelola data pasien rawat jalan diharapkan dapat mempermudah dalam mengelola data dokter, pasien, obat, stok obat, transaksi dan mencetak kartu pasien yang berfungsi mempermudah administrasi pendaftaran pasien saat kembali berkunjung dengan menggunakan sistem pada komputer.

Referensi kedua diambil dari jurnal yang berjudul **“The Application of the Unified Modeling Language in Object-Oriented Analysis of Healthcare Information”** Vinod Aggarwal (2002). Dalam penelitian ini menjelaskan bahwa Object-Oriented Analysis (OOA) merupakan teknik yang cocok untuk digunakan dalam mengembangkan sistem informasi kesehatan. Dalam penelitian ini dijabarkan Aktivitas OOA yang dilakukan menghasilkan representasi persyaratan sistem informasi dari berbagai sudut pandang, kemudian representasi persyaratan tersebut dapat disajikan dengan Unified Modeling Language (UML). Penelitian tersebut juga menjelaskan bahwa UML memiliki manfaat dalam menentukan, memvisualisasi, membangun, mendokumentasikan, dan mengkomunikasikan hasil dari aktivitas OOA.

Paper yang dijadikan referensi selanjutnya berjudul **“Moving From Waterfall to Iterative Development – An Empirical Evaluation of Advantages, Disadvantages and Risks of RUP”** Jorge A. Osorio, Michel R.V. Chaudron, Werner Heijstek (2011). Dalam penelitian ini membandingkan proses pengembangan berulang dengan pengembangan *waterfall* dalam studi kasus sebuah organisasi

internasional, metode pengembangan berulang yang dipilih adalah RUP. Penelitian ini menemukan beberapa kategori faktor dimana penggunaan pengembangan berulang lebih baik daripada pengembangan air *waterfall* dalam hal nilai bisnis, namun ada juga beberapa biaya dan batasan yang jelas yang harus dipertimbangkan secara eksplisit oleh perusahaan yang berencana mengenalkan proses pengembangan berulang. Penulis memberikan kesimpulan berupa keuntungan, kerugian, serta resiko penerapan metode RUP.

Penelitian yang dilakukan oleh Ashraf Anwar (2014) dalam paper yang berjudul **“A Review of RUP (Rational Unified Process)”** menjabarkan sejarah, praktik, serta keuntungan dan kekurangan dalam RUP. Dalam paper memberikan kesimpulan RUP semakin sering digunakan dalam industri pengembangan perangkat lunak karena telah teruji dalam berbagai proyek dan domain. Ini adalah platform proses pengembangan perangkat lunak yang dapat dikonfigurasi yang menghadirkan praktik dan arsitektur yang dapat dikonfigurasi. Ini memungkinkan pengembang untuk memilih dan menerapkan hanya komponen proses yang dibutuhkan untuk setiap tahap proyek. Dalam paper ini juga dijelaskan keuntungan dalam penggunaan RUP diantaranya meningkatkan manajemen proyek, memberikan umpan balik secara berkala dan teratur kepada pemangku kepentingan, meningkatkan manajemen resiko, membantu mengimplementasikan kebutuhan yang benar-benar dibutuhkan, memastikan arsitektur sistem bekerja dengan semestinya, serta memfokuskan pengembang pada apa yang sebenarnya penting.

Tabel 2.1 Literatur Review

No	Nama Penulis, jurnal dan tahun	Tujuan Penelitian	Metode penelitian/ pengembangan	Output/hasil penelitian
1	<p><i>“The Application of the Unified Modeling Language in Object-Oriented Analysis of Healthcare Information”</i> Vinod Aggarwal (2002). <i>Journal of Medical Systems</i>, Vol. 26, No. 5, October 2002</p>	<p>Menguraikan proses analisis objectoriented (OOA) menggunakan UML dan menggambarkan hal ini dengan contoh-contoh perawatan kesehatan untuk menunjukkan kepraktisan penerapan UML oleh personel perawatan kesehatan untuk masalah sistem informasi dunia nyata. Serta menjelaskan efek menguntungkan dari Unified Modeling Language (UML) dalam menetapkan, memvisualisasikan, membangun, mendokumentasikan, dan mengkomunikasikan model sistem informasi kesehatan dari perspektif pengguna.</p>	-	<p>UML dapat menjadi dasar untuk alat pemodelan visual atau alat simulasi. Database kesehatan sangat heterogen dan terus berubah untuk akan mendapat manfaat dari spesifikasi metadata. Karena UML adalah standar metadata, pakar basis data mendukung penggunaan UML untuk pemodelan metadata dari basis data rekam medis elektronik. UML juga akan merancang teknologi penggunaan yang lebih canggih dari teknologi orientasi.</p>

No	Nama Penulis, jurnal dan tahun	Tujuan Penelitian	Metode penelitian/ pengembangan	Output/hasil penelitian
2	<p>“PERANCANGAN SISTEM INFORMASI REKAM MEDIS PASIEN RAWAT JALAN BERBASIS WEB DI KLINIK SEHAT MARGASARI BANDUNG” Johni S Pasaribu, Johnson Sihombing (2017). Jurnal Ilmiah Teknologi Informasi Terapan Volume III, No 3, ISSN : 2407 - 3911</p>	<p>Menghasilkan suatu Sistem Informasi Pengelolaan Data Pasien Rawat Jalan yang dibangun melalui tahapan-tahapan RUP identifikasi masalah, analisis kebutuhan, perancangan dan pengembangan aplikasi, implementasi dan pengujian perangkat lunak.</p>	<p>Metode RUP diterapkan mulai pada proses: pemodelan bisnis, pengumpulan kebutuhan, analisis dan desain, implementasi dan pengujian. Fase-fase evaluasi, lingkungan dan pengembangan, konfigurasi dan perubahan manajemen tidak diterapkan karena pembuatan perangkat lunak ini masih merupakan versi awal.</p>	<p>Menghasilkan sistem yang dapat mempermudah dalam mengelola data dokter, pasien, obat, stok obat, transaksi dan mencetak kartu pasien yang berfungsi mempermudah administrasi pendaftaran pasien saat kembali berkunjung dengan menggunakan sistem pada komputer. Perancangan sistem informasi pasien rawat jalan ini bisa mempermudah petugas dalam menangani dan mengelola data pasien rawat jalan. Memberi suatu alternatif pemecahan masalah dalam sistem registrasi dan pengambilan nomor untuk pasien.</p>

No	Nama Penulis, jurnal dan tahun	Tujuan Penelitian	Metode penelitian/ pengembangan	Output/hasil penelitian
3	<p>“Moving From Waterfall to Iterative Development – An Empirical Evaluation of Advantages, Disadvantages and Risks of RUP” Jorge A. Osorio, Michel R.V. Chaudron, Werner Heijstek (2011). <i>Software Engineering and Advanced Applications (SEAA) 2011. IEEE, ISSN: 2376-9505.</i></p>	<p>Mengetahui keuntungan, kerugian, dan resiko dari penggunaan RUP dalam sebuah organisasi serta mengidentifikasi faktor-faktor lain seperti pengaturan control pada projek, flexibility, integerasi stakeholder, manajemen waktu, pendokumentasian, kepuasan pelanggan, dan kualitas produk.</p>	<p>A Penelitian ini mengikuti pendekatan studi kasus eksploratif. Pengumpulan data didasarkan pada wawancara semi-terstruktur dengan personel yang terlibat dengan implementasi RUP dalam suatu organisasi industri.</p>	<p>Keuntungan: iterasi memberikan penjadwalan dan pengendalian aktivitas lebih baik, prediktabilitas pelaksanaan projek mengurangi resiko, membantu tim dengan umpan balik stakeholders untuk penyesuaian. Kekurangan: banyak umpan balik menyebabkan anggaran naik. Resiko: perlunya pelatihan dan pembinaan yang lebih bagi tim, memerlukan ketrampilan dan pengetahuan UML, kurangnya keterlibatan stakeholder.</p>

No	Nama Penulis, jurnal dan tahun	Tujuan Penelitian	Metode penelitian/ pengembangan	Output/hasil penelitian
4	<p>“A Review of RUP (Rational Unified Process)” Ashraf Anwar (2014). International Journal of Software Engineering (IJSE), Volume (5) : Issue (2) : 2014</p>	<p>Memberikan penjabaran sejarah, praktik, serta keuntungan dan kekurangan dalam penggunaan RUP sebagai metode pengembangan perangkat lunak RUPp.</p>	-	<p>RUP memberikan praktik dan arsitektur yang dapat dikonfigurasi. Membuat pengembang bisa memilih dan menerapkan hanya komponen proses yang dibutuhkan untuk setiap tahap proyek. Keuntungan dalam penggunaan RUP diantaranya meningkatkan manajemen proyek, memberikan umpan balik secara berkala dan teratur kepada stakeholders, meningkatkan manajemen resiko, membantu mengimplementasikan kebutuhan yang benar-benar dibutuhkan, memastikan arsitektur sistem bekerja dengan semestinya.</p>

Tabel 2.1 diatas menunjukkan review dari literatur yang telah dikumpulkan oleh peneliti dengan keterangan pada judul jurnal, tahun, tujuan penelitian, metode yang digunakan, dan hasil kesimpulan atau saran. Dalam penelitian **PERANCANGAN SISTEM INFORMASI REKAM MEDIS PASIEN RAWAT JALAN BERBASIS WEB DI KLINIK SEHAT MARGASARI BANDUNG** digunakan dalam penelitian ini sebagai pendukung penerapan sistem informasi dalam dunia kesehatan khususnya penerapan dalam klinik. Penelitian ini juga menggunakan metode pengembangan RUP yang penerapannya hanya fase insepasi, elaborasi, dan konstruksi menunjukkan penerapan metode ini dapat disesuaikan dengan proyek yang sedang dikembangkan oleh tim. Selanjutnya penelitian berjudul **Moving From Waterfall to Iterative Development – An Empirical Evaluation of Advantages, Disadvantages and Risks of RUP** dan **A Review of RUP (Rational Unified Process)** digunakan sebagai dasar penerapan metode RUP dalam penelitian ini. Kelebihan metode RUP dalam menangani perubahan dalam pengembangan, fokus pada arsitektur sistem, manajemen waktu yang tepat, dan fleksibilitas yang dijelaskan dalam kedua penelitian tersebut menjadi dasar penerapan metode ini dalam pengembangan sistem informasi layanan pemeriksaan kesehatan tenaga kerja. Serta dalam penelitian tersebut menunjukkan dan menjelaskan proses penerapan metode RUP yang dapat dijadikan referensi. Penelitian berjudul **The Application of the Unified Modeling Language in Object-Oriented Analysis of Healthcare Information** menjadi dasar pemanfaatan UML sebagai bahasa pemodelan pada proses pengembangan sistem informasi layanan pemeriksaan kesehatan tenaga kerja untuk memodelkan aktivitas, interaksi objek, dan memodelkan objek. Kemudian hasil pemodelan dapat menjadi masukan untuk merancang basis data dan sebagai panduan implementasi.

2.2 Profile Klinik Argaraya Medika Malang

Berawal dari laboratorium Klinik Deksa Medika yang berdiri tahun 2004, dan berganti nama menjadi Klinik Argaraya Medika pada tahun 2013 yang berkiprah sebagai penyelenggara pelayanan kesehatan tenaga kerja merasa terpenggil dan bertanggung jawab untuk memberikan perlindungan maupun peningkatan taraf kesehatan yang diperlukan untuk menjamin terselenggaranya produktifitas tenaga kerja secara menyeluruh, merata, terpadu, dan berkesinambungan.

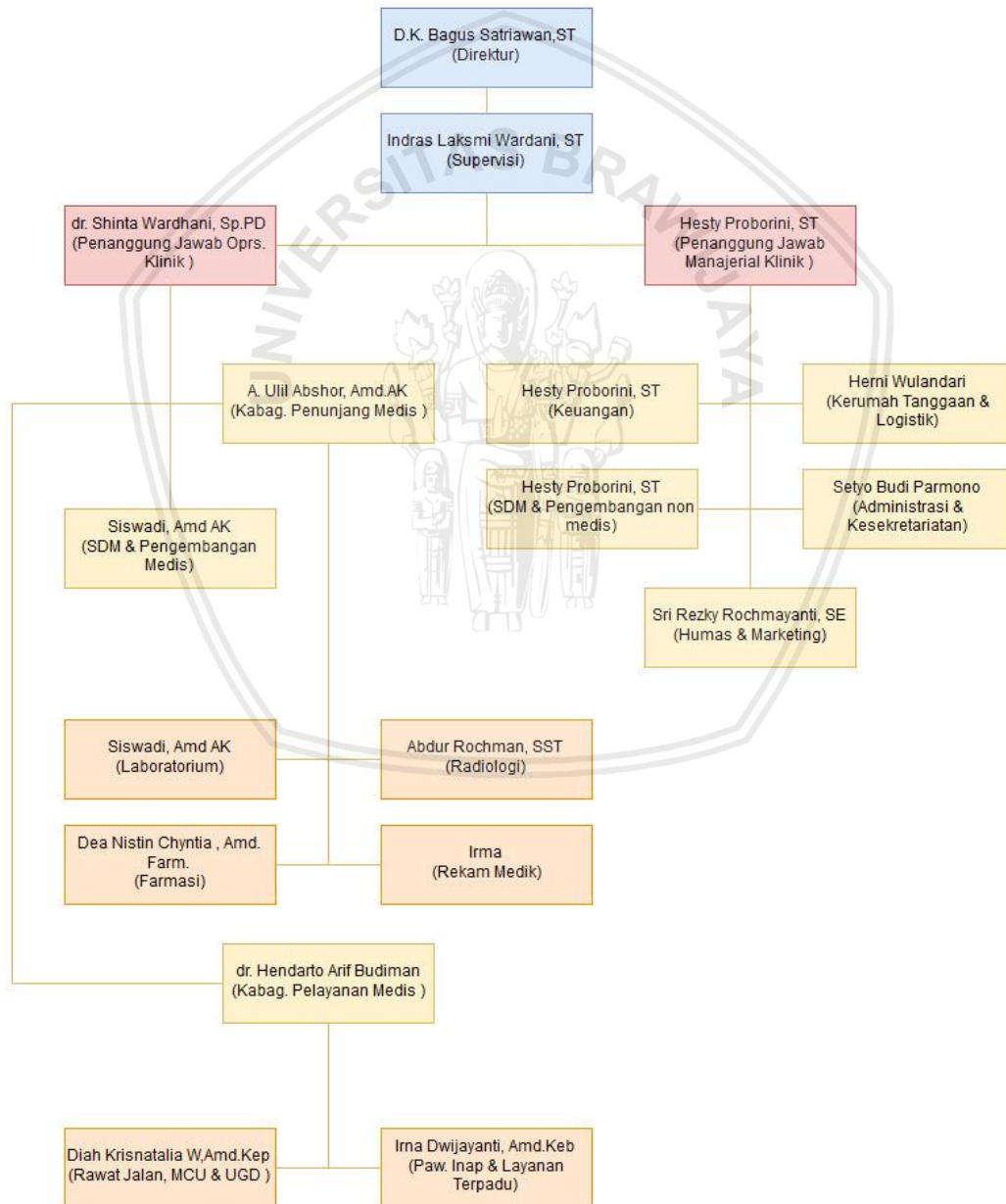
Sejalan dengan kemajuan jaman dan teknologi, kesadaran masyarakat akan arti pentingnya kesehatan juga meningkat. Pelayanan kesehatan yang dimaksud adalah pencegahan, perawatan dan pengobatan rutin yang dilakukan. Sehingga dalam era ini, keinginan masyarakat dalam kesehatan tidak hanya meliputi pengobatan, tetapi juga pencegahan. Perlindungan Rawat Jalan Kesehatan termasuk didalamnya *Medical check-up* saat ini sudah cukup memasyarakat dan sudah biasa dilakukan secara rutin. Manfaatnya selain sebagai usaha preventif bagi individu untuk pemelihara kesehatan, juga sebagai screening bagi perusahaan dalam merekrut karyawan, serta perusahaan asuransi yang memberikan persyaratan check-up kesehatan bagi individu yang ingin mengikuti asuransi dll.

2.2.1 Visi dan Misi

Visi dari Klinik Argaraya Medika Malang adalah “Menjadi Klinik yang terpercaya dengan pelayanan kesehatan profesional yang handal dan mampu berkembang secara berkesinambungan, dikenal nasional maupun internasional”.

Misi dari Klinik Argaraya Medika Malang adalah “Mampu memberikan pelayanan kesehatan yang terbaik kepada pasien Umum, Tenaga Kerja Dalam Negeri (TKDN) dan pasien Tenaga Kerja Luar Negeri (TKLN), Selalu berusaha memuaskan customer, Selalu meningkatkan kinerja, dan Memiliki lingkungan kinerja yang baik karyawan dan customer”.

2.2.2 Struktur Organisasi



Gambar 2.1 Struktur Organisasi Klinik Argaraya Medika Malang

Gambar 2.1 menunjukkan struktur organisasi dari Klinik Argaraya Medika Malang yang dikepalai oleh Direktur dan dibantu Supervisi serta membawahi dua divisi penanggung jawab yaitu Penanggung Jawab Operasional Klinik dan Penanggung Jawab Manajerial Klinik. Penanggung Jawab Manajerial Klinik membawahi keuangan, kerumah tanggaan dan logistik, SDM dan pengembangan non medis, administrasi dan kesekretariatan, dan humas dan marketing. Pada Penanggung Jawab Operasional Klinik membawahi Kabag. Penunjang medis, SDM dan pengembangan medis, dan Kabag. Pelayanan medis.

2.3 Sistem Informasi

Sistem informasi menjadi referensi pada penelitian ini dan akan dibahas pada subbab ini. Sistem merupakan sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai sebuah tujuan. Informasi merupakan data atau fakta yang telah diproses sehingga dapat menjadi sebuah informasi. Dari informasi juga dapat mengurangi ketidakpastian serta mempunyai nilai dalam keputusan karena dengan adanya informasi kita dapat memilih tindakan-tindakan dengan resiko yang paling kecil (Anggadini, 2011).

Sistem informasi memuat berbagai informasi penting mengenai orang, tempat, dan segala sesuatu yang ada di dalam atau di lingkungan sekitar organisasi. Informasi sendiri mengandung suatu arti yaitu data yang telah diolah ke dalam suatu bentuk yang lebih memiliki arti dan dapat digunakan untuk pengambilan keputusan (Anggadini, 2011).

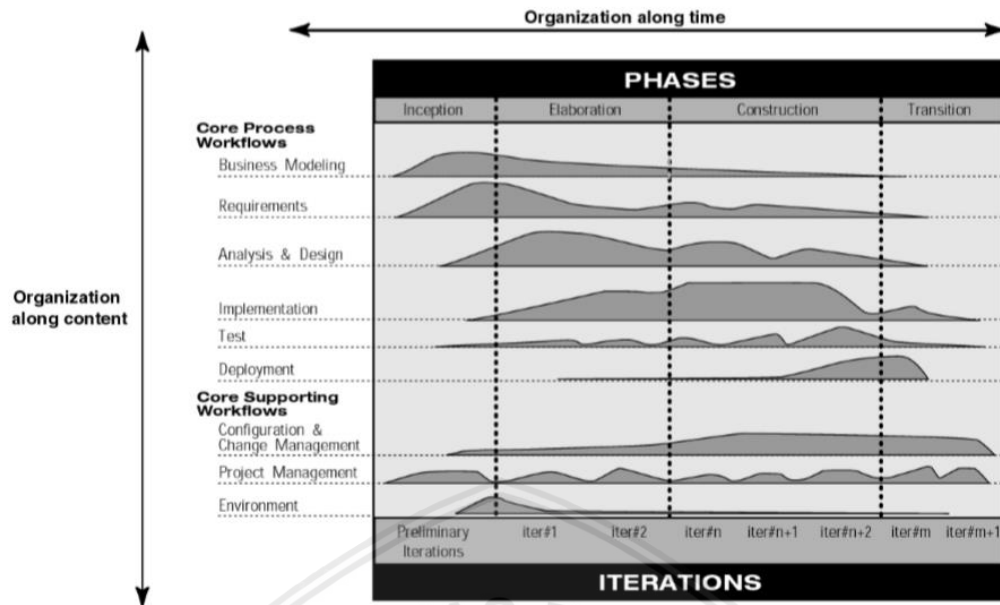
2.4 Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) merupakan sebuah rangkaian tahap-tahap menggunakan metodologi tertentu yang dapat sebagai panduan dalam melakukan suatu proses pengembangan dan perubahan dari suatu sistem informasi. Sedangkan setiap tahapannya merupakan segmen-segmen dari sebuah SDLC yang meliputi beberapa jenis aktifitas (Everett & Mcleod Jr, 2007).

2.5 Rational Unified Process (RUP)

Rational Unified Process (RUP) merupakan pengembangan rekayasa perangkat lunak dengan melakukan pendekatan yang disiplin dalam melakukan setiap tugas dan tanggung jawabnya dalam sebuah organisasi. Tujuan RUP itu sendiri adalah untuk dapat memastikan produksi berkualitas tinggi dan untuk memenuhi semua kebutuhan pemangku kepentingan termasuk waktu dan biaya sesuai dengan rencana yang telah disepakati sebelumnya (IBM, 1998).

RUP di masing-masing notasinya telah terbukti mampu mengelola dan menangkap persyaratan fungsional dan memastikan bahwa penerapan, kontrol desain, dan pengujian perangkat lunak ini dapat memenuhi semua kebutuhan pengguna (IBM, 1998).



Gambar 2.2 Model Pengembangan RUP

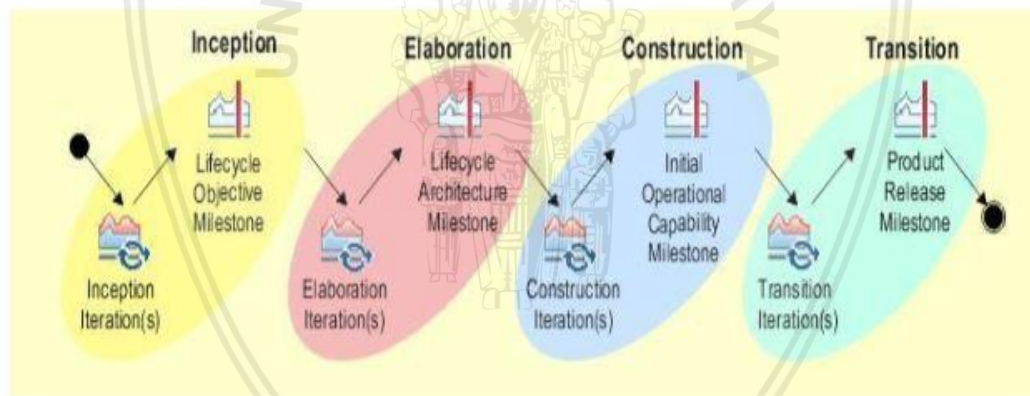
Sumber: IBM (1998)

Pada gambar 2.3 menunjukkan proses dan fase dalam menggunakan kerangka kerja RUP. Proses serta fase dalam RUP dapat dilihat berdasarkan dua jenis sudut pandang, diantaranya secara horizontal dan vertikal. Fase-fase secara horizontal yang dilakukan dijelaskan dibawah ini. 4 fase RUP (IBM, 1998), yaitu:

1. Fase inepsi Dalam fase ini kebutuhan para pemangku kepentingan akan ditentukan serta interaksi antara entitas yang didefinisikan dalam semua entitas. Identifikasi mencakup semua usecases dan deskripsi dari usecase secara signifikan.
2. Fase Elaborasi adalah fase dimana domain dari semua masalah akan dianalisa, menjadi dasar pembuatan arsitektur, membuat rencana proyek, serta elemen dengan resiko tinggi dihilangkan. Tahap ini juga dibuat dengan memahami tentang keseluruhan sistem terkhususnya ruang lingkupnya, fungsi utama dan persyaratan nonfungsional seperti persyaratan kinerja.
3. Fase konstruksi, pada fase ini seluruh dari komponen fitur aplikasi yang telah dirancang mulai dibangun dan diintegrasikan menjadi sebuah produk, dan melakukan uji pada fitur sistem.
4. Fase Transisi merupakan fase terakhir ini sebuah produk yang telah dibangun mulai ditransisikan pada pengguna atau pemangku kepentingan. Fase ini umpan balik dari pengguna banyak didapat sehingga mampu di gunakan untuk acuan untuk rilis mapun perbaikan sebuah perangkat lunak selanjutnya.

Untuk sudut pandang vertical RUP, diantaranya:

1. Pemodelan bisnis bertujuan untuk menggunakan proses bisnis objek yang diteliti sebagai acuan dalam memodelkan proses bisnis untuk membuat pemahaman dari seluruh pemangku kepentingan.
2. Kebutuhan adalah mendeskripsikan apa saja sistem akan lakukan sebagai dasar membuat use case dan deskripsi use case.
3. Analisis dan Desain bertujuan untuk memberikan gambaran bagaimana sistem akan diterapkan pada kegiatan implementasi. Hasil berupa model analisis dan model desain. Model analisis dilakukan kebutuhan untuk mendapatkan elemen-elemen desain dianalisis sedangkan model desain terdiri dari struktur kelas dengan pendefinisian antarmuka dan interaksi antar kelas divisualisasikan.
4. Implementasi yaitu penerapan model dari rancangan atau design yang telah dilakukan menjadi produk jadi berupa sebuah sistem atau perangkat lunak.
5. Pengujian adalah untuk memastikan kinerja dari interaksi antar objek, kesesuaian dengan kebutuhan, dan integrasi antar komponen perangkat lunak.
6. Deployment bertujuan untuk me-realese produk yang sudah dibangun sebelumnya.



Gambar 2.3 Milestone pada akhir setiap fase RUP

Sumber: (Anwar, 1998)

Gambar 2.4 menjelaskan bahwa dalam setiap fase RUP ada *milestone* atau tonggak pencapaian. Setiap fase dalam RUP ada hal-hal yang harus dipenuhi atau tonggak pencapaian harus dicapai sebelum pembangunan dapat dilanjutkan ke fase berikutnya (Anwar,2014). Hal-hal yang dilakukan untuk memenuhi milestone dijelaskan sebagai berikut :

1. Insepsi
 - Para pemangku kepentingan harus setuju, setidaknya untuk sebagian besar, dengan informasi yang disajikan kepada mereka.
 - Persyaratan harus benar-benar dipahami

- Model use case awal (dan sangat dasar) harus disiapkan (rata-rata 10-20% selesai).

Ini adalah tonggak pertama (*milestone*) dalam yang harus dicapai, dan dikenal sebagai "*Lifecycle Objective Milestone*". Jika proyek tidak melewati tonggak itu, proyek itu dapat dibatalkan atau dapat dirancang ulang sehingga dapat memenuhi batasan dan kriteria yang hilang.

2. Elaborasi

- Model *use case* (untuk menunjukkan fungsionalitas sistem). *Use case* dan aktor harus diidentifikasi dan model itu sendiri harus sekitar 80% lengkap.
- Deskripsi arsitektur perangkat lunak.

Ini adalah tonggak kedua yang harus dicapai, dan dikenal sebagai "*Milestone Lifecycle Architecture*". Jika proyek tidak dapat melewati tonggak ini, proyek itu dapat dibatalkan atau dirancang ulang.

3. Konstruksi

- pembuatan sistem operasional dengan menggunakan arsitektur yang dapat dieksekusi yang telah dibuat pada fase sebelumnya.
- Pembuatan program sebagian besar harus sudah dibangun.
- Pengujian sistem dilakukan.

Ini adalah tonggak pencapaian ketiga, dan dikenal sebagai "*initial operational capability*"

4. Transisi

Sistem harus memenuhi persyaratan fungsional dan kebutuhan pengguna atau pemangku kepentingan. Merupakan tonggak terakhir yang harus dicapai atau dikenal sebagai "*Product Release Milestone*". Fase ini mengakhiri siklus pengembangan perangkat lunak RUP.

2.6 Business Process

Business Process menjelaskan sebuah rangkaian aktifitas yang memerlukan sebuah masukan (*input*) dan menghasilkan keluaran (*output*) yang mempunyai nilai bagi pengguna (Monk & Wagner, 2013). Pengguna di sini adalah proses bisnis yang terdiri dari pengguna internal dan pengguna eksternal. pengguna internal adalah seseorang atau sekelompok orang di berbagai departemen dalam perusahaan sementara pengguna eksternal adalah seseorang atau sekelompok orang yang membeli produk. Proses bisnis dapat digunakan untuk memahami cara perusahaan atau organisasi beroperasi. Analisis dan pemodelan *as-is* proses bisnis dan *to-be* proses bisnis adalah salah satu pendekatan yang dapat dilakukan untuk mengetahui organisasi atau perusahaan.

2.6.1 Analisis dan Pemodelan Proses Bisnis *As-Is*

Kegiatan menganalisis dan memodelkan proses bisnis *as-is* merupakan sebuah proses dalam memperoleh dan memodelkan sebuah proses bisnis yang aktual pada suatu perusahaan ataupun organisasi. Tujuan yaitu menyajikan informasi mengenai informasi gambaran aktual pada suatu perusahaan. Sehingga akan mengetahui kekurangan perusahaan dari proses bisnisnya dan mampu memberikan kemungkinan sebuah peningkatan (*improvement*) pada proses bisnis yang terdapat pada suatu perusahaan atau organisasi (Becker, et al., 2013).

Identifikasi proses bisnis saat ini atau *as-is* bisa didapat dengan melakukan kegiatan wawancara kepada pegawai yang terlibat pada eksekusi sebuah proses bisnis dan melakukan sebuah kegiatan observasi kepada pegawai yang terlibat dalam tiap tugas organisasi atau perusahaan. Hasil dari pengidentifikasian ini akan dijadikan dasar dalam membuat skema diagram proses bisnis *as-is*.

2.6.2 Analisis dan Pemodelan Proses Bisnis *To-Be*

Analisis dan pemodelan proses bisnis *to-be* merupakan kegiatan untuk mengidentifikasi dan memodelkan proses bisnis yang telah mengalami peningkatan (*improvement*) berdasarkan analisis terhadap proses bisnis *as-is* (Becker, et al., 2013). Migrasi untuk mengimplementasikan proses bisnis *to-be* dengan mengganti *as-is* proses bisnis dapat dilakukan secara bertahap, sehingga perubahan dalam proses bisnis tidak akan dirasakan secara langsung secara langsung. Becker, dkk. menjelaskan bahwa penerapan proses bisnis yang akan dilakukan dapat dilakukan dalam jangka menengah atau panjang.

2.7 Business Process Modeling Notation (BPMN)




BPMN merupakan notasi standar yang digunakan untuk memberikan solusi pemodelan proses bisnis sehingga proses pertukaran informasi antar pengguna bisnis dapat dilakukan dengan baik. BPMN bertujuan untuk memberikan notasi yang mudah dimengerti untuk semua pengguna bisnis, mulai dari analisis bisnis hingga menyiapkan konsep proses bisnis, kepada pengguna yang memiliki minat dalam mengelola dan memantau bagaimana proses bisnis dijalankan (Object Management Group (OMG), 2011). Di dalam BPMN terdapat empat kategori dari elemen-elemennya yang menunjang pembuatan notasi pemodelan bisnis yaitu *Flow Object*, *Connections*, *Swimlanes*, dan *Artifacts*.

2.7.1 Flow Object

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

1. Event merupakan sesuatu yang terjadi dan memberikan pengaruh dalam proses bisnis yang dapat berasal dari dalam dan luar suatu proses. *Event* terdiri dari *start event*, *intermediate event*, dan *end event*. Dalam suatu proses bisnis akan selalu memiliki *start* dan *end event*. Berikut penjelasan lebih lanjut mengenai simbol *event* yang ditunjukkan pada Tabel 2.2.


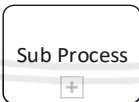
Tabel 2.2 Simbol event

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

Sumber: *Object Management Group* (2011)

2. *Activity* merupakan elemen yang mendeskripsikan aktivitas yang dilakukan dalam suatu proses bisnis. *Activity* terdiri dari *Task* dan *Sub Process*. Berikut penjelasan lebih lanjut mengenai simbol *activity* yang ditunjukkan pada Tabel 2.3.

Tabel 2.3 Simbol Activity





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

Sumber: *Object Management Group* (2011)

3. *Gateway* merupakan elemen yang mengendalikan alur dari sebuah proses di dalam proses bisnis. *Gateway* terdiri dari *exclusive*, *inclusive*, dan *complex*. Berikut penjelasan lebih lanjut mengenai simbol *gateway* yang ditunjukkan pada Tabel 2.4.



Tabel 2.4 Simbol Gateway

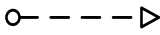
No.	Nama	Simbol	Keterangan
1	<i>Exclusive</i>		Sebagai proses divergen untuk membuat jalur alternative proses yang memungkinkan satu proses yang dipilih dan sebagai proses konvergen untuk menggabungkan jalur alternative.
2	<i>Parallel</i>		Proses yang dijalankan secara bersamaan.
3	<i>Inclusive</i>	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

Sumber: *Object Management Group* (2011)

2.7.2 Connections

Connections merupakan elemen yang menghubungkan antar *flow object* di dalam proses bisnis. *Connections* terdiri dari *sequence flow*, *association*, dan *message flow*. Berikut adalah penjelasan mengenai simbol *connections* yang ditunjukkan pada Tabel 2.5.

Tabel 2.5 Simbol *Connections*

No.	Nama	Simbol	Keterangan
1	<i>Sequence Flow</i>		Menunjukkan urutan aktivitas dalam sebuah proses bisnis.





No.	Nama	Simbol	Keterangan
2	<i>Assosiation</i>	Menunjukkan hubungan antara teks, data, <i>flow object</i> , atau artifak dalam proses bisnis.
3	<i>Message Flow</i>	→	Alur pesan antara aktivitas yang dapat saling ber kirim dan menerima pesan satu sama lain.

Sumber: *Object Management Group* (2011)

2.7.3 Swimlanes

Swimlanes adalah tempat/wadah aktivitas di dalam proses bisnis yang dapat membagi set aktivitas dengan aktivitas lain. *Swimlanes* terdiri dari *pool* dan *lanes*. Berikut penjelasan lebih lanjut mengenai simbol *swimlanes* yang ditunjukkan pada tabel 2.6.

Tabel 2.6 Simbol *Swimlanes*


No.	Nama	Simbol	Keterangan
1	<i>Pool</i>		Wadah yang berisi <i>flow objects</i> , <i>connections</i> , maupun artifak.
2	<i>Lane</i>		Bagian mendetail dari <i>pool</i> .


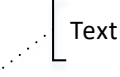

Sumber: *Object Management Group* (2011)

2.7.4 Artifacts

Artifacts menggambarkan sesuatu yang berada di luar sebuah proses. *Artifacts* dapat mendeskripsikan dan mengelola data atau catatan dari sebuah proses. *Artifacts* terdiri dari *data object*, *data store*, *annotation*, dan *group*. Berikut penjelasan lebih lanjut mengenai simbol *artifacts* yang ditunjukkan pada tabel 2.7.

Tabel 2.7 Simbol *Artifacts*

No.	Nama	Simbol	Keterangan
1	<i>Data Object</i>		Data yang dihasilkan dan dibutuhkan oleh sebuah aktivitas di dalam proses bisnis.

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

Sumber: *Object Management Group* (2011)

2.8 Pemodelan *Use Case*

Model *use case* terbukti memiliki nilai yang berharga sebagai bagian dari aktivitas menentukan persyaratan dalam proses perangkat lunak, terutama dalam meningkatkan komunikasi antara tim pengembang perangkat lunak dan pemangku kepentingan dan membuat persyaratan kegiatan penentuan lebih mudah dan lebih tepat (Bittner & Spence, 2002). Bittner dan Spence menambahkan, *use case* menggambarkan bagaimana pengguna menggunakan sistem dan apa yang dilakukan sistem untuk penggunanya.

2.8.1 Identifikasi Tipe Pemangku Kepentingan

Identifikasi tipe stakeholder adalah kegiatan untuk mengenal stakeholder suatu sistem dengan melakukan analisis kebutuhan. Para pemangku kepentingan yang teridentifikasi akan dikelompokkan berdasarkan hubungan dan karakteristik pemangku kepentingan dengan sistem yang sedang dikerjakan (Bittner & Spence, 2002). Ada beberapa kategori pemangku kepentingan diantaranya:

1. *User* (Pengguna) adalah pemangku kepentingan yang terhubung dengan langsung dengan sistem. User adalah individu dalam model *use case* memerankan peran sebagai aktor.
2. *Sponsors* (Sponsor) adalah sebuah investasi yang dilakukan pemangku kepentingan terhadap pengembangan sistem. Umumnya pemangku kepentingan ini merupakan pengguna tidak langsung dari sebuah sistem.
3. *Developers* (Pengembang) adalah pemangku kepentingan terhubung pada suatu aktifitas produksi pada sistem.
4. *Authorities* (Pihak berwenang) adalah pemangku kepentingan mempunyai kemampuan pada memberikan sebuah dukungan pengembangan sistem

didalam bidang teknologi, aturan perundangan, regulasi internal maupun eksternal sebuah perusahaan.

5. Pelanggan (Customers) adalah pemangku kepentingan yang akan membeli sistem yang sudah selesai dibangun.

2.8.2 Analisis Masalah

Analisis masalah adalah proses memahami kebutuhan pengguna dan memahami masalah yang sebenarnya. Dan proses untuk menghasilkan solusi yang diusulkan untuk memenuhi kebutuhan pengguna (Leffingwell-& Widrig, 2003). Tujuan analisis masalah yaitu memperoleh pemahaman lebih baik terkait masalah yang akan diselesaikan sebelum proses pengembangan solusi yaitu sistem dilakukan.

Cara dalam mendokumentasikan analisis masalah dengan baik adalah dengan membuat (*problem statement*) pernyataan masalah (Bittner & Spence, 2002). Pernyataan masalah didapat dengan melakukan wawancara dengan beberapa pemangku kepentingan.

Tabel 2.8 Kerangka Dokumentasi Pernyataan Masalah

<i>The problem of</i>	Deskripsi suatu masalah.
<i>Affects</i>	Masalah yang mempengaruhi pemangku kepentingan.
<i>The impact of which is</i>	Mendeskrepsi dampak pada kegiatan bisnis dan pemangku kepentingan akibat masalah.
<i>A successful solution would</i>	Menyelesaikan masalah dengan menawarkan solusi dan manfaat utama.

Sumber: *Bittner & Spence (2003)*

2.8.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna

Mengidentifikasi kebutuhan pengguna adalah pernyataan masalah yang disampaikan oleh para pemangku kepentingan lebih dalam (Bitner & Spence, 2002). Bittner dan Spence menambahkan dalam memperoleh informasi Pengguna dapat membantu memahami bagaimana dan mengatasi setiap aspek yang berbeda dari apa yang dapat mempengaruhi pemangku kepentingan yang berbeda. (Leffingwell & Widrig, 2003) memberi penejelasan dalam mengidentifikasi identifikasi fitur, kebutuhan pengguna, dan persyaratan pada perangkat lunak bisa dilakukan dengan beberapa teknik:

1. Wawancara
2. *Brainstorming and Idea Reduction*
3. *Requirement workshop*



4. Storyboarding

2.8.4 Identifikasi Fitur

Fitur adalah layanan yang disediakan oleh sistem untuk memenuhi satu atau lebih kebutuhan pemangku kepentingan (Leffingwell & Widrig, 2003). Leffingwell dan Widrig menambahkan, fitur merupakan cara berguna dan mudah untuk mendeskripsikan fungsional sistem tanpa harus terhambat dengan penjelasan fungsional sistem yang terlalu detail. Fitur juga menyediakan ringkasan manfaat yang ditawarkan oleh produk (sistem) yang akan dikembangkan (Bittner & Spence, 2002).

Tabel 2.9 Identifikasi Fitur

Kode Fitur	Deskripsi
Berisi kode identitas fitur.	Berisi penjelasan tentang yang dapat dilakukan oleh sistem.

Sumber: Diadaptasi dari Bittner & Spence (2002)

2.8.5 Identifikasi Persyaratan Fungsional dan Persyaratan Nonfungsional

Bittner dan Spence menyebutkan, Syarat dan ketentuan non-fungsional adalah kategori yang digunakan untuk mengelompokkan perangkat lunak. Fungsi fungsional dan nonfungsional dapat diekspresikan dalam banyak cara, tetapi metode termudah untuk digunakan adalah menggambarkan persyaratan ini dalam kalimat deklaratif.

Persyaratan fungsional adalah layanan yang harus disediakan oleh sistem, persyaratan fungsional juga merupakan pernyataan yang menyatakan bagaimana sistem bereaksi terhadap input tertentu dan bagaimana sistem berjalan ketika dalam situasi tertentu (Sommerville, 2011). Deskripsi persyaratan fungsional harus berisi penjelasan tentang tindakan apa yang dapat dilakukan sistem. Selain itu, ada informasi tentang perilaku sistem ketika menerima input dan output.

Persyaratan non-fungsional merupakan aspek kualitas atau batasan-batasan yang harus dipenuhi oleh sistem (Bittner & Spence, 2002). Leffingwell dan Widrig menjelaskan bahwa persyaratan non-fungsional muncul karena sangat penting bagi analis untuk menentukan aspek-aspek sistem yang berhubungan dengan *usability*, *reliability*, *performance*, dan *supportability* dari sebuah sistem.

2.8.6 Permodelan *Use Case*

Model *use case* digunakan untuk mempresentasikan sistem dalam hal bagaimana penggunaan sistem tersebut (Bittner & Spence, 2002). Bittner dan Spence menambahkan, dalam menentukan *use case*, analis menggunakan persyaratan perangkat lunak sebagai panduan dan seringkali *use case* berisi persyaratan fungsional dari sebuah sistem. Model *use case* berisi dua komponen utama yaitu aktor dan *use case*.

Aktor adalah representasi individu (manusia) atau objek lain yang berinteraksi dengan sistem. Aktor merupakan objek diluar sistem. Dalam model *use case*, aktor memiliki nama dan deskripsi singkat, selain itu, aktor dihubungkan dengan *use case* sesuai dengan interaksi aktor dengan *use case* tertentu.

Use case adalah representasi sebuah nilai yang diberikan oleh sistem kepada aktor. *Use case* tidak sama dengan fitur atau fungsi dari sebuah sistem. *Use case* memiliki nama dan penjelasan singkat. *Use case* juga memiliki spesifikasi yang lebih detail, yang berisi alur ketika aktor dan sistem berinteraksi.

Model *use case* tidak menyediakan deskripsi yang lengkap mengenai persyaratan sebuah sistem, oleh karena itu, model *use case* harus didukung dengan pemodelan-pemodelan persyaratan dan dokumen-dokumen persyaratan lainnya. Setiap *use case* di dalam model *use case* memiliki spesifikasi *use case* yang berisi penjelasan bagaimana aktor dan sistem berkolaborasi untuk memenuhi tujuan yang direpresentasikan oleh *use case* tersebut.

Tabel 2.10 Identifikasi Fitur

Brief Description	Berisi penjelasan singkat sebuah <i>use case</i> .
Actor	Berisi aktor yang berinteraksi dengan <i>use case</i> .
Pre-condition	Berisi satu atau lebih kondisi atau keadaan yang harus dipenuhi oleh sistem sebelum <i>use case</i> dapat dijalankan.
Post-condition	Berisi satu atau lebih kondisi atau keadaan yang dipenuhi oleh sistem ketika <i>use case</i> berakhir.
Extension	Berisi informasi penggunaan <i>use case</i> lain yang bersifat opsional, yang disertakan pada sebuah <i>use case</i> .
Basic-flow	Berisi alur normal yang dilalui ketika <i>use case</i> berjalan.
Alternative Flow	Berisi alur opsional yang akan dilalui ketika <i>use case</i> tidak berjalan sesuai dengan alur normal.
Subflow	Berisi beberapa kelompok alur yang merupakan penyederhanaan alur <i>use case</i> yang terlalu detail.

Sumber: Diadaptasi dari Bittner & Spence (2002)

2.9 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan salah satu dari banyak tools yang berguna dan menajubkan pada dunia pengembangan sistem. UML merupakan Bahasa pemodelan visual yang memungkinkan pengembangan sistem untuk membuat blueprints yang menangkap visi mereka pada suatu standar, mudah untuk dipahami, dan menyediakan mekanisme berbagi secara efektif dan mengkomunikasikan visi mereka dengan pihak lain (Schhmuller,2004).




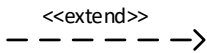
Sedangkan menurut Sukamto, (2013), *Unified Modelling Language* (UML) merupakan salah satu bahasa dalam memodelkan perancangan yang

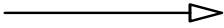
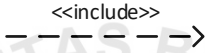
mendefinisikan kebutuhan (*requirement*), analisis desain, dan menggambarkan arsitektur dari pemrograman berorientasi objek. UML diterapkan untuk menganalisa, merancang, dan mengimplementasi sistem yang berbasis perangkat lunak dan pemodelan bisnis (*Object Management Group, 2011*). UML terdiri dari kumpulan diagram yang mempermudah dalam membantu pengembangan suatu sistem atau perangkat lunak yang berorientasi objek. Kumpulan diagram tersebut meliputi *use case diagram, class diagram, sequence diagram, dan activity diagram*. Sekumpulan diagram tersebut menggambarkan sistem yang memudahkan pemahaman untuk membangun sistem.

2.9.1 Use Case Diagram

Use case diagram adalah diagram pemodelan berfungsi untuk memodelkan tingkah laku (*behavior*) suatu sistem atau perangkat lunak yang akan dibangun. Diagram *use case* menampilkan berbagai peran pengguna dan fungsi peran tersebut di dalam sistem (Satzinger, 2010). Diagram *use case* mendeskripsikan fungsionalitas yang terdapat di dalam sistem sebagai aktor yang saling berinteraksi yang dideskripsikan dengan kata kerja. Aktor tersebut adalah orang, proses, atau sistem lain yang saling berinteraksi di dalam sistem yang dibangun. Namun aktor yang digambarkan dengan simbol orang belum tentu merupakan orang, bisa saja aktor yang dimaksud adalah suatu sistem lain (Sukamto, 2013). Berikut daftar simbol yang terdapat dalam *use case diagram* ditunjukkan pada tabel 2.11.

Tabel 2.11 Simbol Use Case Diagram

No.	Nama	Simbol	Keterangan
1	<i>Use Case</i>		Fungsionalitas di dalam sistem yang berinteraksi dengan aktor/unit. <i>Use case</i> biasanya dinyatakan dengan kata kerja.
2	Aktor		Orang, proses, maupun sistem lain yang berinteraksi di dalam sistem yang dibangun.
3	Asosiasi		Komunikasi interaksi antara aktor dengan <i>use case</i> di dalam sistem.
4	<i>Extend</i>		Relasi <i>use case</i> dapat menambahkan <i>behavior/tingkah lakunya</i> kepada <i>use case</i> lain sebagai bentuk spesialisasi. <i>Use case</i> tambahan tersebut dapat berdiri sendiri dan

No.	Nama	Simbol	Keterangan
			biasanya memiliki nama depan yang sama dengan <i>use case</i> yang melakukan <i>extend</i> .
5	<i>Generalization</i>		Relasi generalisasi dan spesialisasi (khusus-umum) antara dua <i>use case</i> , yang salah satunya lebih umum dan lebih khusus.
6	<i>Include</i>		Relasi <i>use case</i> yang menambahkan <i>use case</i> lain sebagai bagian dari proses dalam dirinya yang dibutuhkan sebagai syarat untuk menjalankan fungsi.


Sumber: Diadaptasi dari Object Management Group (2005)






Tabel 2.11 menunjukkan beberapa simbol dalam *use case diagram* dan penjelasan dari setiap simbol. Simbol *use case diagram* yang ditunjukkan adalah *use case*, *actor*, *association*, *extend*, *generalization*, dan *include*.

2.9.2 Sequence Diagram

Sequence diagram adalah diagram sequensial yang menggambarkan tingkah laku objek pada *use case* melalui pendeskripsian waktu hidup objek serta pesan (*message*) yang diterima dan dikirim antara objek satu dengan objek lainnya (Sukanto, 2013). Diagram ini dibuat sesuai dengan pendefinisian *use case* yang mendefinisikan interaksi jalannya pesan di dalam sistem. Di dalam UML objek dalam *sequence diagram* digambarkan dengan segiempat yang memuat nama objek yang memiliki garis ke bawah. Penamaan objek dapat dilakukan dengan tiga cara yaitu nama objek, nama objek dan kelas, atau nama kelas. Berikut simbol-simbol yang terdapat dalam *class diagram* pada tabel 2.12.

Tabel 2.12 Simbol *Sequence Diagram*

No.	Nama	Simbol	Keterangan
1	Aktor		Orang, proses, maupun sistem lain yang berinteraksi di dalam sistem.

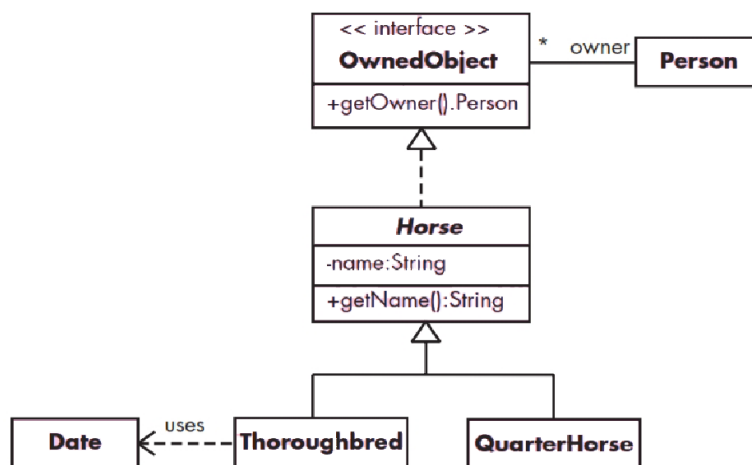
No.	Nama	Simbol	Keterangan
2	<i>Boundary</i>	 : Boundary0	Antarmuka dan interaksi satu aktor atau lebih di dalam sistem.
3	<i>Controller</i>	 : Control1	Mengatur dan mengoordinasikan perilaku dan dinamika suatu sistem dalam menangani tugas utama dan mengontrol alur kerja sistem.
4	<i>Entity</i>	 : Boundary0	Data dan informasi yang harus disimpan oleh sistem.
5	<i>Lifeline</i>		Menyatakan keberadaan suatu objek dalam waktu.
6	<i>Message</i>	 1: Message00	Menggambarkan aliran pesan antar objek yang menunjukkan urutan kejadian di dalam sistem.

Sumber: Diadaptasidari Object Management Group (2005)

Tabel 2.12 menunjukkan beberapa simbol dalam *sequence diagram* dan penjelasan dari setiap simbol. Simbol *sequence diagram* yang ditunjukkan adalah *actor*, *boundary*, *controller*, *entity*, *lifeline*, dan *message*.

2.9.3 Class Diagram

Dalam memodelkan kelas, termasuk atributnya, operasi dan juga hubungan dengan kelas lain, UML menyediakan *class diagram model*. *Class diagram* menyediakan tampilan struktural atau statis dari suatu sistem. *Class diagram* tidak menyediakan sifat dinamis dari komunikasi antara objek kelas dalam diagram. Elemen utama pada *class diagram* adalah simbol kotak, merupaka ikon yang dipakai dalam mewakili antarmuka dan kelas. Pada setiap kotak dipisah menjadi bagian horizontal. Pada bagian tengah terdapat atribut tiap kelas. Setiap atribut mengarah pada sesuatu yang merupakan objek dalam kelas. Pada setiap atribut bisa mempunyai nama, tipe dan simbol. Terdapat simbol yang ditunjukkan dengan # (*protected*), - (*private*), + (*public*), ~ (*package*). Gambar 2.4 menunjukkan contoh *class diagram*.

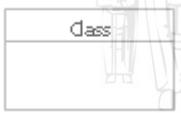






Gambar 2.4 Contoh Class Diagram

Sumber: (Pressman, 2010)

Pada Gambar 2.4 atribut ditunjukkan sebagai *private* (-). Pada setiap operasi bisa ditampilkan dengan *public* (+), ada juga yang diberi parameter, jenis dan tipe pengembalian. Dalam pembuatan struktur *class diagram* harus memperhatikan dan mengerti simbol yang ada dikarenakan apabila ada kesalahan dalam penggambaran simbol maka akan terjadi kesalahpahaman dalam memahaminya dan sulit untuk tahap implementasi selanjutnya. Simbol-simbol yang dapat digunakan dalam pembuatan struktur *class diagram* ditunjukkan pada Tabel 2.13.

Tabel 2.13 Simbol Class Diagram

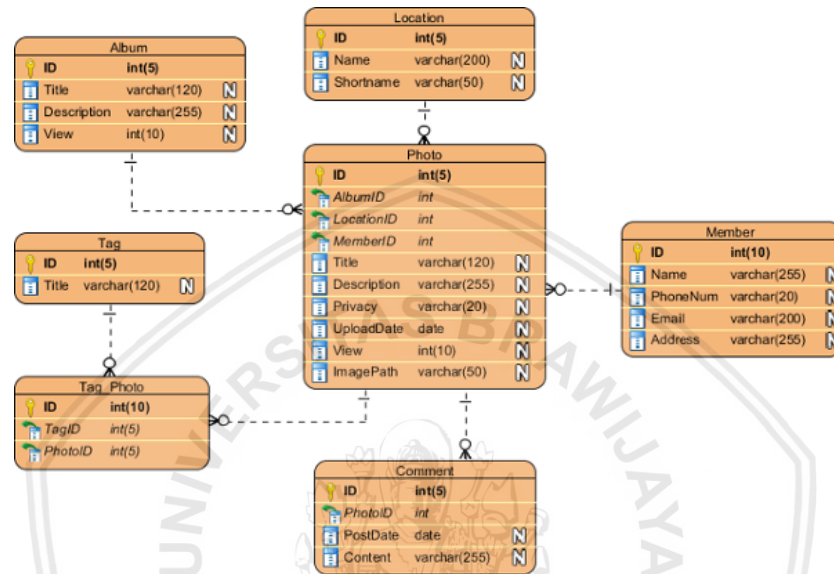
No	Nama	Simbol	Keterangan
1.	Class		Class di struktur sistem. Class mendefinisikan atribut, operasi dan instance.
2.	Association		Relasi yang menunjukkan hubungan antara dua class.
3.	Generalization		Relasi yang menunjukkan pewarisan dari class utama (parent) ke class anak (child)
4.	Dependencies		Relasi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

Sumber: (A.S & Shalahudin, 2015)



2.10 Physical Data Model (PDM)

Physical data model (PDM) mempresentasikan bagaimana model akan dibangun di database. PDM menunjukkan struktur-struktur tabel terdiri dari nama kolom, batasan kolom, tipe data kolom, *primary key*, *foreign key*, dan relasi antar tabel. PDM bertujuan untuk memvisualisasikan dengan model skema internal pada basis data, menggambar tabel dan juga kolom-kolomnya beserta relasi antar tabel.



Gambar 2. 1 Contoh *Relational Model*

Dalam membuat model basis data dapat dilakukan berdasarkan perancangan *class diagram* dengan menggunakan tiga strategi yaitu (Booch, et al., 1999) :

1. Mendefinisikan di setiap kelas yang terdapat pada *class diagram* menjadi tabel. Hal ini merupakan cara yang sederhana, tetapi biasanya mengalami kesulitan jika perancangan *class diagram* terjadi perubahan, yaitu ketika terjadi *subclass* yang ditambah.
2. Mengumpulkan *superclass* dengan *subclass* pada tingkatan hirarki yang sepadan, lalu didefinisikan setiap kelas menjadi tabel. Kelemahannya adalah kapasitas penyimpanan menyebabkan berkurang dengan cepat.
3. Memisahkan *superclass* dengan *subclass* dan mendefinisikan menjadi sebuah tabel. Kelemahannya adalah kebutuhan mekanisme join table dalam proses pengambilan data melalui basis data.

2.11 Black Box Testing

Black Box Testing adalah sebuah cara dalam menguji perangkat lunak berfokus terhadap pengecekan kebutuhan pada fungsional sistem. Pengujian ini bertujuan untuk mengetahui kesalahan yang terjadi pada perangkat lunak seperti kesalahan atau hilangnya fungsi, kesalahan antar muka, kesalahan pada *database*, serta

kesalahan dalam inisialisasi, kesalahan pada kinerja, dan terminasi (Pressman, 2010). Pengujian ini memiliki beberapa keunggulan diantaranya:

1. Pengujian mampu membantu dalam mengungkapkan ketidaksesuaian dalam spesifikasi perangkat lunak karena dilakukan berdasar sudut pandang pengguna.
2. Pengujian bisa dilakukan oleh badan independen dari pengembang, dan memungkinkan mendapatkan objektif perspektif.
3. Tester tidak perlu bagaimana mengimplementasikan perangkat lunak atau mengetahui bahasa pemrograman.
4. Spesifikasi lengkap kasus uji dapat langsung dirancang.

2.12 Compatibility Testing

Compatibility Testing merupakan pengujian perangkat lunak terhadap kompatibilitas suatu perangkat lunak dengan jenis-jenis browser yang digunakan dalam menjalankan sebuah program. Tiap jenis browser menghasilkan hasil yang sedikit berbeda untuk menampilkan sebuah perangkat lunak, terlepas pada standarisasi HTML di WebApp. Mungkin terdapat plug-in yang dibutuhkan atau ketidakterediaan konfigurasi tertentu dalam menjalankan WebApp. Setiap jenis pengaturan komputasi bisa mengakibatkan perbedaan dalam hal resolusi layar sisi klien, kecepatan pemrosesan sisi klien atau kecepatan koneksi. Dalam mengevaluasi kemampuan penerapannya dengan lingkungan komputasi. Komputasi lingkungan mungkin berisi beberapa atau semua elemen yang disebutkan sebagai berikut (Kumar, et al., 2015):

1. *Database* (DB2, Oracle, Sybase, dan lainnya)
2. Sistem perangkat lunak lain (*web server, networking/messaging tool, dan lainnya*).
3. Kompatibilitas *browser* (Chrome, Firefox, Internet Explorer, Safari, Netscape, dan lainnya).

2.13 User Acceptance Testing (UAT)

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

User Acceptance Testing (UAT) merupakan sarana formal dimana perusahaan mengkonfirmasi bahwa sistem yang telah dibangun sudah benar-benar memenuhi persyaratan pengguna. Tahap ini merupakan pengujian akhir yang dilakukan setelah melakukan pengujian fungsional. *User Acceptance Testing (UAT)* dilakukan oleh pengguna dan pengembang perangkat lunak, sebelum perangkat

lunak ini dioperasikan pengujian ini dilakukan sebagai tahap akhir pengujian perangkat lunak. Tujuan utama dari *User Acceptance Testing* (UAT) adalah mengecek perangkat lunak terhadap kebutuhan bisnis, hal ini dilakukan oleh pengguna akhir yang sudah familiar dengan kebutuhan bisnis (Goel & Gupta, 2014).

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

Performance merupakan karakteristik kinerja yang diinginkan dari sistem harus didefinisikan agar data terukur menjadi berguna. Pertanyaan berhubungan dengan spesifikasi dari performa (Naik, 2008).

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

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

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

2.14 Skala Likert

Peneliti menggunakan skala likert untuk menggali respon pengguna saat melakukan *User Acceptance Testing* (UAT). Skala likert akan memudahkan responden dalam menyikapi pertanyaan. Sugiyono (2015) berpendapat bahwa skala likert merupakan skala yang dapat mengukur persepsi, pendapat, dan sikap responden terhadap fenomena sosial. Alternatif jawaban pada skala ini terperinci mulai dari sangat setuju, setuju, netral, tidak setuju dan sangat tidak setuju. Terdapat beberapa pilihan dalam menentukan skala pada skala likert untuk keperluan analisis kuantitatif, salah satunya dengan menggunakan skala 5 kategori dan setiap kategori diberikan skor seperti yang ditunjukkan Tabel 2.14 dan untuk persentase nilai jawaban ditunjukkan pada Tabel 2.15.

Tabel 2.14 Bobot Nilai Jawaban

Pilihan Jawaban	Bobot Nilai
A = Sangat Setuju	5
B = Setuju	4
C = Netral	3
D = Kurang	2
E = Sangat Kurang	1

Sumber: (Sugiyono, 2015)

Tabel 2.15 Persentase Nilai Jawaban

Interval Persentase	Interpretasi Skor
80 sampai 100%	Sangat Setuju
60 sampai 79,99%	Setuju
40 sampai 59,99%	Netral
20 sampai 39,99%	Tidak Setuju
0 sampai 19,99%	Sangat Kurang Setuju

Sumber: (Sugiyono, 2015)

Setelah bobot nilai dan persentase nilai telah ditentukan, selanjutnya adalah mengkalikan jumlah jawaban dengan bobot nilainya. Untuk perhitungannya akan dijelaskan pada Rumus berikut:

1. Jawaban Sangat Setuju (A) = $n \times 5$
2. Jawaban Setuju (B) = $n \times 4$
3. Jawaban Netral (C) = $n \times 3$
4. Jawaban Tidak Setuju (D) = $n \times 2$
5. Jawaban Sangat Tidak Setuju (E) = $n \times 1$

$$\text{Total Nilai} = (n \times 5) + (n \times 4) + (n \times 3) + (n \times 2) + (n \times 1) \quad (2.1)$$

n merupakan jumlah dari responden yang menjawab. Perhitungan selanjutnya adalah mencari nilai tertinggi dari hasil skala likert tersebut. Rumus yang digunakan dijelaskan pada Rumus 2.2, dimana N1 merupakan nilai tertinggi skala likert, n merupakan jumlah responden, dan U merupakan jumlah uji kasus.

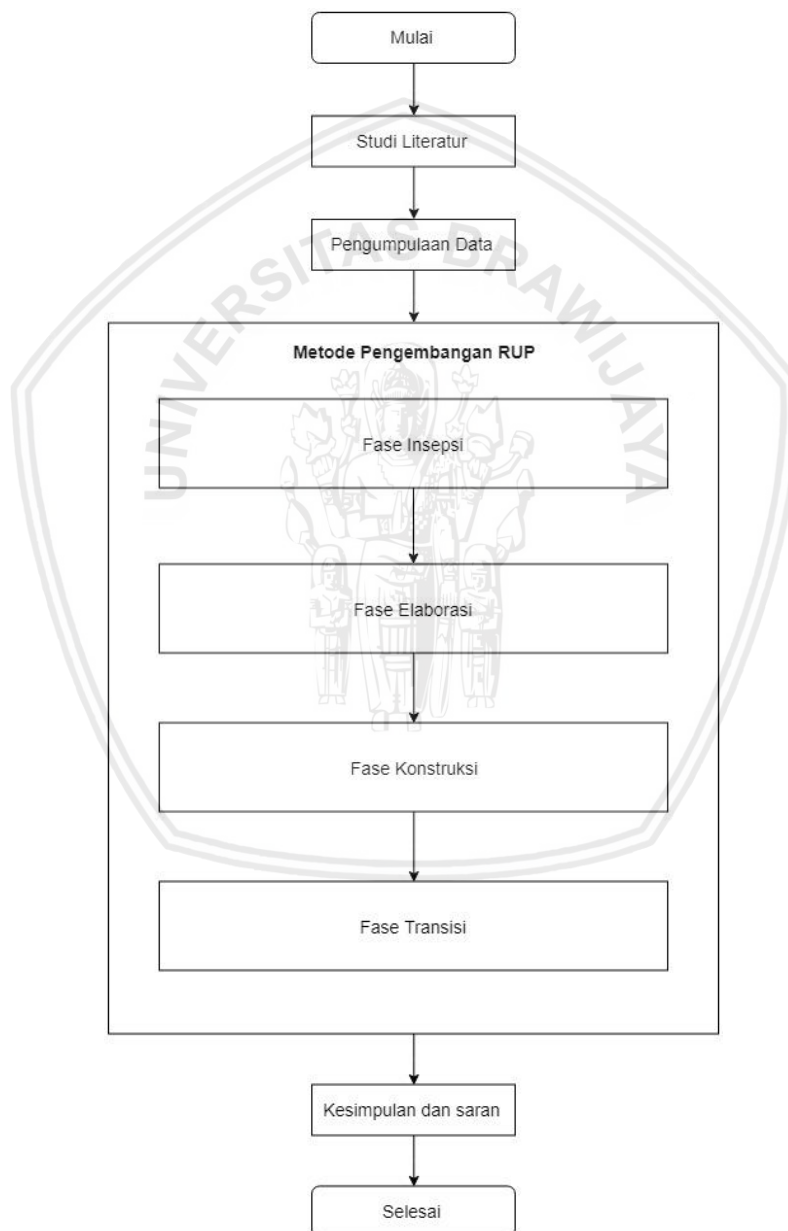
$$Y = N1 \times n \times U \quad (2.2)$$

Sedangkan untuk menghitung hasil dari UAT dengan menggunakan skala likert dijelaskan pada Rumus 2.3 berikut.

$$\text{Rumus Index \%} = (\text{Total Nilai} \div Y) \times 100 \quad (2.3)$$

BAB 3 METODOLOGI

Pada bab ini akan dibahas langkah-langkah yang akan dilakukan dalam mengembangkan Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dengan metode *Rational Unified Process* (RUP). Gambar 3.1 menunjukkan alur atau proses pengembangan Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja Klinik Argaraya Medika. Metodologi penelitian pengembangan sistem informasi dimulai dari studi literatur, pengumpulan data, *inception*, *elaboration*, *construction*, *transition*, kesimpulan dan saran.



Gambar 3.1 Diagram Alur Metodologi

3.1 Studi Literatur

Studi literatur dilakukan untuk menyelesaikan masalah dengan benar melalui pengetahuan pendukung. Sumber referensi diperoleh dari berbagai jurnal, buku, studi kasus, dan artefak di media cetak dan internet. Peneliti melakukan metode studi literatur dengan membaca dan mempelajari buku-buku atau penelitian serupa dan mencari atau menelusuri beberapa situs internet untuk mengumpulkan data dan informasi yang berkaitan dengan masalah yang dibahas dalam pengembangan sistem informasi. Teori dan Pustaka yang terkait dengan penulisan penelitian ini adalah Sistem Informasi, *Software Development Life Cycle*, Metode *Rational Unified Process*, Pemodelan *Use Case*, UML, *Business Process Modeling Nation*, *Physical Data Model*, Pengujian Perangkat Lunak.

3.2 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data untuk mengumpulkan informasi terkait proses bisnis yang berjalan dan informasi kebutuhan sistem dari sudut pandang pengguna. Pengumpulan data ini difokuskan pada layanan pemeriksaan kesehatan tenaga kerja Klinik Argaraya Medika. Pengumpulan data dilakukan dengan Observasi secara langsung di Klinik Argaraya Medika serta Wawancara dengan pihak yang berkepentingan pada Layanan Pemeriksaan Kesehatan Tenaga Kerja Klinik Argaraya Medika.

a. Observasi

Kegiatan ini merupakan metode mengumpulkan data dengan mengamati secara langsung di Klinik Argaraya Medika khususnya pada Layanan Pemeriksaan Kesehatan Tenaga Kerja. Mengamati dan memahami setiap prosedur yang dilakukan pada Layanan Pemeriksaan Kesehatan Tenaga Kerja serta peran dari staff yang menjalankan prosedur layanan. Dari observasi yang dilakukan nantinya akan didapatkan informasi-informasi untuk mendukung penelitian ini.

b. Wawancara

Kegiatan ini merupakan metode mengumpulkan data dengan cara wawancara secara langsung maupun tidak langsung terhadap responden yang dianggap layak untuk memberikan informasi mengenai Layanan Pemeriksaan Kesehatan Tenaga Kerja. Dalam penelitian ini peneliti melakukan wawancara kepada pengurus Administrasi & Kesetiaian sebagai pelaku serta pengurus Layanan Pemeriksaan Kesehatan Tenaga Kerja. Kegiatan ini bertujuan untuk lebih memahami proses bisnis yang sedang berjalan saat ini dan masalah apa saja yang sedang dihadapi. Dalam wawancara ini juga didapatkan analisis kebutuhan dan batasan kegiatan dalam pengembangan sistem.

3.3 Metode *Rational Unified Process* (RUP)

Tahap ini melakukan pengembangan sistem informasi dengan metode yang telah dipilih dan sesuai dengan penelitian ini yaitu *Rational Unified Process* (RUP).

RUP memiliki 4 fase pengulangan dan dalam penelitian ini semua fase RUP digunakan, diantaranya fase inepsi, fase elaborasi, fase konstruksi, dan fase transisi. Dalam penelitian ini setiap fase metode RUP penggunaannya disesuaikan dengan pengembangan sistem informasi layanan pemeriksaan kesehatan tenaga kerja yang dikembangkan.

3.3.1 Fase Inepsi

Fase ini merupakan fase pertama dalam pengembangan menggunakan metode RUP. Proses yang dilakukan dalam tahap ini adalah memodelkan proses bisnis *as-is* dan *to-be* lalu dilakukan analisa kebutuhan dari proses bisnis yang telah dimodelkan serta melakukan pemodelan *use case diagram*. Hal-hal yang akan dilakukan pada fase ini dalam pengembangan sistem informasi layanan pemeriksaan kesehatan tenaga kerja adalah:

1. Mengidentifikasi Proses bisnis *as-is* dari hasil pengumpulan data dengan *Business Process Model Notation*.
2. Menggambarkan proses bisnis *to-be* sebagai solusi terhadap proses bisnis yang sudah dilakukan pada proses bisnis *as-is* dengan *Business Process Model Notation*.
3. Mengidentifikasi semua objek yang akan berhubungan dengan aktor.
4. Memodelkan hasil identifikasi hubungan aktor dengan objek dengan *Unified Model Language* (UML).
5. Dilakukan evaluasi apakah projek sudah memenuhi *Lifecycle Objective Milestone* sehingga dapat melanjutkan pada fase selanjutnya dan dilakukan perbaikan jika tidak memenuhi fase pertama ini.

Dalam fase inepsi nantinya dihasilkan BPMN pemodelan proses bisnis masalah *as-is* saat ini, BPMN pemodelan proses bisnis *to-be* atau rekomendasi solusi terhadap masalah, hasil analisis kebutuhan pemangku kepentingan, analisis identifikasi aktor dan fitur, identifikasi persyaratan fungsional dan non-fungsional sistem, pemodelan *use case diagram*, dan spesifikasi *use case*.

3.3.2 Fase Elaboration

Tujuan dalam fase ini adalah melakukan perancangan atau desain arsitektur dari sistem informasi layanan pemeriksaan kesehatan tenaga kerja. Perancangan sistem akan dibangun sesuai dengan kebutuhan dari kebutuhan yang telah ditentukan pada tahap inepsi. Hal-hal yang akan dilakukan adalah:

1. Mendiskripsikan arsitektur perangkat lunak
2. Perancangan antarmuka dan algoritme sistem yang akan dibuat.
3. Dilakukan evaluasi apakah projek sudah memenuhi *Lifecycle Architecture Milestone* sehingga dapat melanjutkan pada fase selanjutnya dan dilakukan perbaikan perancangan jika tidak memenuhi tahap fase ini.

Hasil dari fase elaboration adalah *Sequence Diagram*, *Class diagram*, *Physical Data Model*, perancangan antarmuka, dan perancangan algoritme sistem informasi layanan pemeriksaan kesehatan tenaga kerja.

3.3.3 Fase Konstruksi

Fase ini melakukan implementasi dari hasil perancangan serta melakukan pengujian terhadap sistem. Dalam fase ini juga dilakukan evaluasi dan perubahan jika dalam tahap sebelumnya terdapat kesalahan. Sebagian besar dari proses *coding* dilakukan dalam fase ini. Hal yang akan dilakukan pada fase ini adalah:

1. Melakukan implementasi sistem dari hasil perancangan sebelumnya.
2. Melakukan pengujian fungsional sistem dengan *validation testing* untuk menguji hasil yang akan dikeluarkan pada tiap fungsionalitas sistem apakah sesuai dengan kebutuhan atau tidak.
3. Melakukan pengujian non-fungsional sistem dengan *compability testing* dengan *SortSite* untuk mengetahui apakah sistem dapat berjalan dengan baik pada *platform* atau perangkat yang berbeda.
4. Dilakukan evaluasi apakah projek sudah memenuhi *Operational Capability Milestone* sehingga dapat melanjutkan pada fase selanjutnya dan dilakukan perbaikan implementasi sistem jika tidak memenuhi tahap fase ini.

Hasil dari fase konstruksi yaitu implemenatasi sistem informasi layanan pemeriksaan kesehatan tenaga kerja berbasis website, hasil pengujian sistem yaitu *validation testing* dan *compability testing*.

3.3.4 Fase Transisi

Fase ini bertujuan untuk memastikan apakah sistem yang dibangun sudah dapat diterima sesuai dengan kebutuhan pengguna. Sistem harus sudah terpenuhi dan pengujian pada pengguna dilakukan langsung untuk didapat *feedback*. Hal yang akan dilakukan pada fase ini adalah:

1. Melakukan pengujian dengan *User Acceptance Testing* (UAT).
2. Dilakukan evaluasi apakah projek sudah memenuhi *Product Release Milestone* serta dilakukan perbaikan pada sistem jika diperlukan.

Hasil dari fase transistion yaitu Hasil pengujian *user acceptance testing* (UAT).

3.4 Kesimpulan dan Saran

Kesimpulan dan saran dilakukan setelah semua fase selesai dilakukan. Hasil akhir dari penelitian ini yang nantinya dapat memberikan kesimpulan sesuai dengan masalah yang diuraikan. Tahap akhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan dalam pembuatan sistem dan untuk penyempurnaan penulisan serta untuk memberikan pertimbangan pengembangan sistem selanjutnya.

BAB 4 ANALISIS KEBUTUHAN

Dalam bab ini berisi pembahasan analisis kebutuhan Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja yang diperoleh dari hasil analisis wawancara dan observasi. Kegiatan yang dilakukan merupakan aktivitas fase inepsi dalam metode RUP. Dalam fase pertama inepsi akan menghasilkan analisa proses bisnis *as-is* dan *to-be*, analisis permasalahan, analisis persyaratan terdiri dari identifikasi pemangku kepentingan, analisis permasalahan, identifikasi kebutuhan pengguna, identifikasi fitur, identifikasi persyaratan fungsional dan nonfungsional, kemudian pemodelan *use case* dan spesifikasi *use case*.

4.1 Pemodelan Proses Bisnis

Pemodelan proses bisnis dilakukan untuk memodelkan serangkaian aktivitas yang dilakukan oleh fungsi-fungsi bisnis untuk memenuhi kebutuhan pengguna. Penggalan informasi mengenai proses bisnis yang akan dimodelkan didapatkan dengan cara melakukan wawancara langsung kepada bagian administrasi Klinik Argaraya Medika Malang dan observasi pada lokasi studi kasus penelitian. Dalam penelitian ini, pemodelan proses bisnis dihasilkan diagram proses bisnis yang saat ini berjalan (*as-is*) dan proses bisnis rekomendasi (*to-be*).

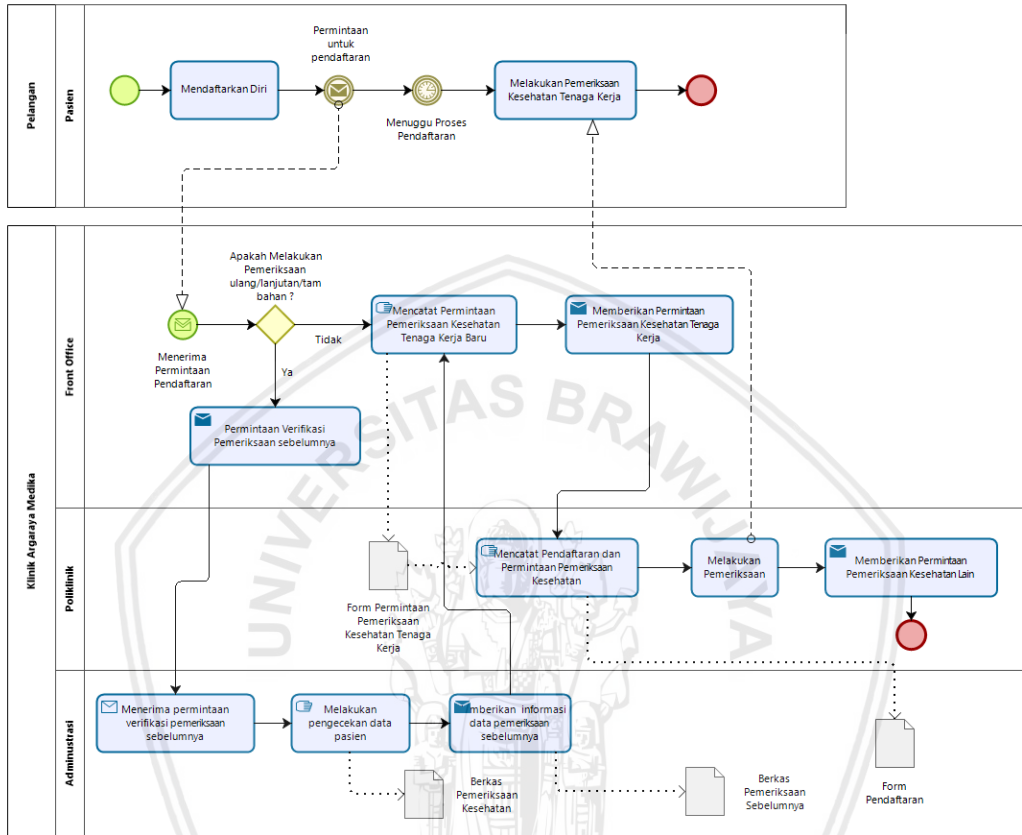
4.1.1 Pemodelan Proses Bisnis As-Is

Proses bisnis *as-is* dilakukan analisis dan pemodelan bertujuan agar peneliti dapat mengidentifikasi kemungkinan perbaikan atau peningkatan pada proses bisnis saat ini dengan solusi yang akan ditawarkan. Proses bisnis *as-is* ini dimodelkan pada keadaan proses bisnis yang sedang berjalan pada Klinik Argaraya Medika Malang, di mana proses bisnis saat ini masih belum ada sistem informasi yang mendukung. Analisis dan permodelan proses bisnis *as-is* dilakukan dengan tujuan menggali permasalahan dan kemungkinan perbaikan. Analisis yang dilakukan dengan melakukan wawancara kepada bagian administrasi Klinik Argaraya Medika Malang dan melakukan observasi di lokasi studi kasus penelitian. Hasil dari analisis tersebut dimodelkan dengan notasi BPMN untuk memudahkan dalam merepresentasikan hasil analisis proses bisnis yang sedang berjalan. Empat proses bisnis dihasilkan berdasarkan analisis proses bisnis *as-is*.

4.1.1.1 Proses Bisnis As-Is Pendaftaran Pasien

Proses dimulai dari pasien datang untuk mendaftarkan diri kepada *front office*. Petugas *Front office* akan melakukan pencatatan pengajuan pemeriksaan kesehatan tenaga kerja pada formulir permintaan pemeriksaan kesehatan tenaga kerja. Setelah itu diberikan kepada petugas poliklinik untuk melakukan pencatatan data diri dan permintaan pemeriksa apa saja yang dibutuhkan pada formulir pendaftaran layanan pemeriksaan kesehatan tenaga kerja. Setelah terdaftar petugas poliklinik akan melakukan pemeriksaan kepada pasien dan membuat permintaan kepada petugas bagian laboratorium atau petugas radiologi jika pasien membutuhkan pemeriksaan. Jika pasien melakukan pemeriksaan ulang,

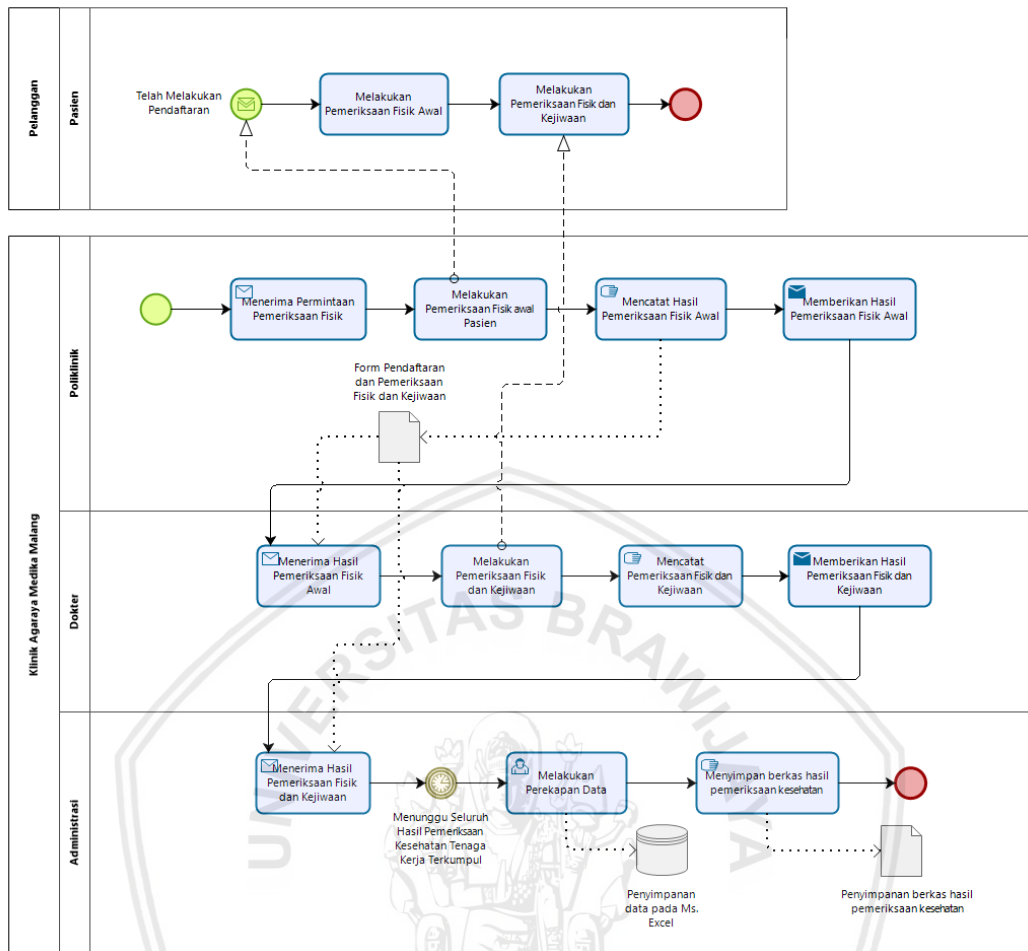
lanjutan, atau tambahan maka akan diverifikasi dahulu oleh petugas *Front office* pada bagian administrasi dan meminta informasi pemeriksaan sebelumnya yang telah dilakukan oleh pasien sebelumnya, setelah itu akan dibuatkan pencatatan formulir permintaan pemeriksaan kesehatan tenaga kerja baru. Pemodelan proses bisnis *as-is* Pendaftaran Pasien terdapat pada Gambar 4.1.



Gambar 4.1 Proses Bisnis As-Is Pendaftaran Pasien

4.1.1.2 Proses Bisnis As-Is Pemeriksaan Fisik dan Kejiwaan

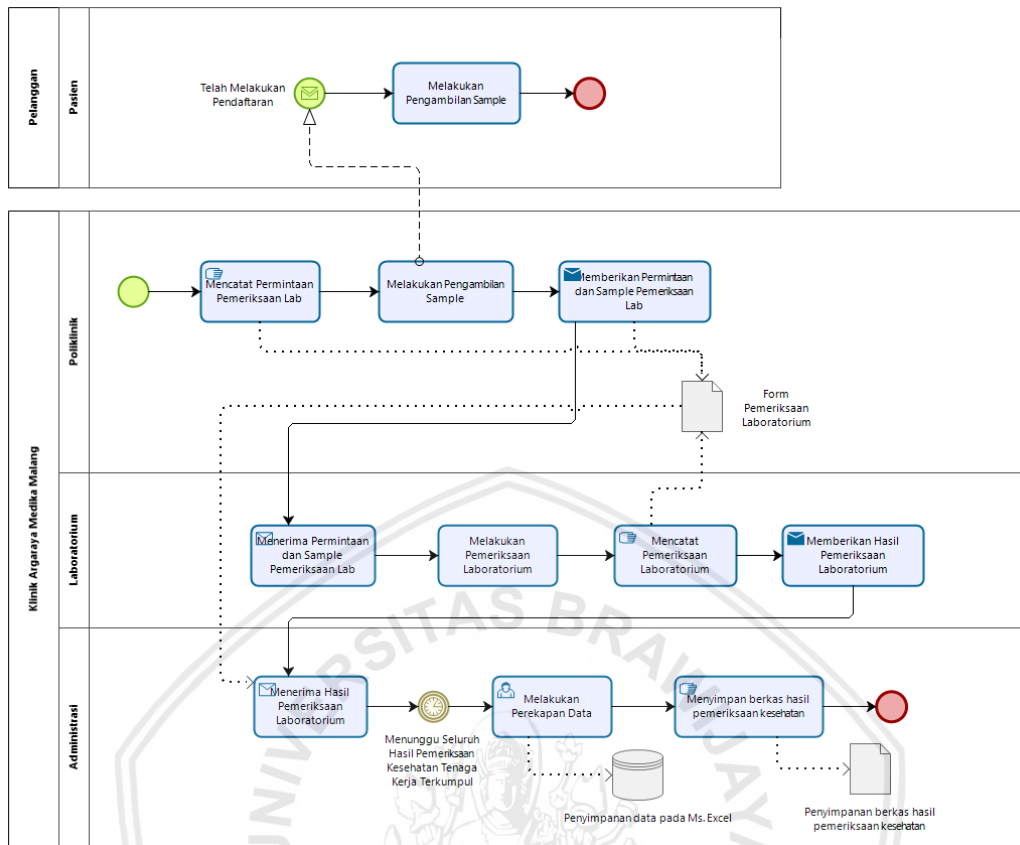
Proses bisnis pemeriksaan fisik dan kejiwaan dilakukan setelah pasien melakukan pendaftaran dan petugas poliklinik akan melakukan pemeriksaan fisik awal dan mencatatnya pada formulir pemeriksaan fisik dan kejiwaan. Selanjutnya jika pemeriksaan fisik awal telah dilakukan dicatat pada formulir pemeriksaan fisik dan kejiwaan lalu akan diberikan kepada dokter untuk melakukan pemeriksaan fisik lanjutan dan kejiwaan. Dokter akan melakukan pemeriksaan fisik dan kejiwaan pada pasien lalu mencatat hasil pemeriksaan pada formulir pemeriksaan fisik dan kejiwaan yang diberikan petugas poliklinik. Saat pencatatan hasil pemeriksaan fisik dan kejiwaan telah selesai maka akan diberikan kepada petugas bagian administrasi untuk dilakukan perekapan. Perekapan data keseluruhan pasien akan dilakukan setelah seluruh data pemeriksaan pasien telah selesai. Keseluruhan data pasien pemeriksaan kesehatan tenaga kerja di masukan ke dalam Ms. Excel dan data berkas akan disimpan dalam penyimpanan berkas hasil pemeriksaan. Pemodelan Proses Bisnis *as-is* Pemeriksaan Fisik dan Kejiwaan terdapat pada Gambar 4.2.



Gambar 4.2 Proses Bisnis As-Is Pemeriksaan Fisik dan Kejiwaan

4.1.1.3 Proses Bisnis As-Is Pemeriksaan Laboratorium

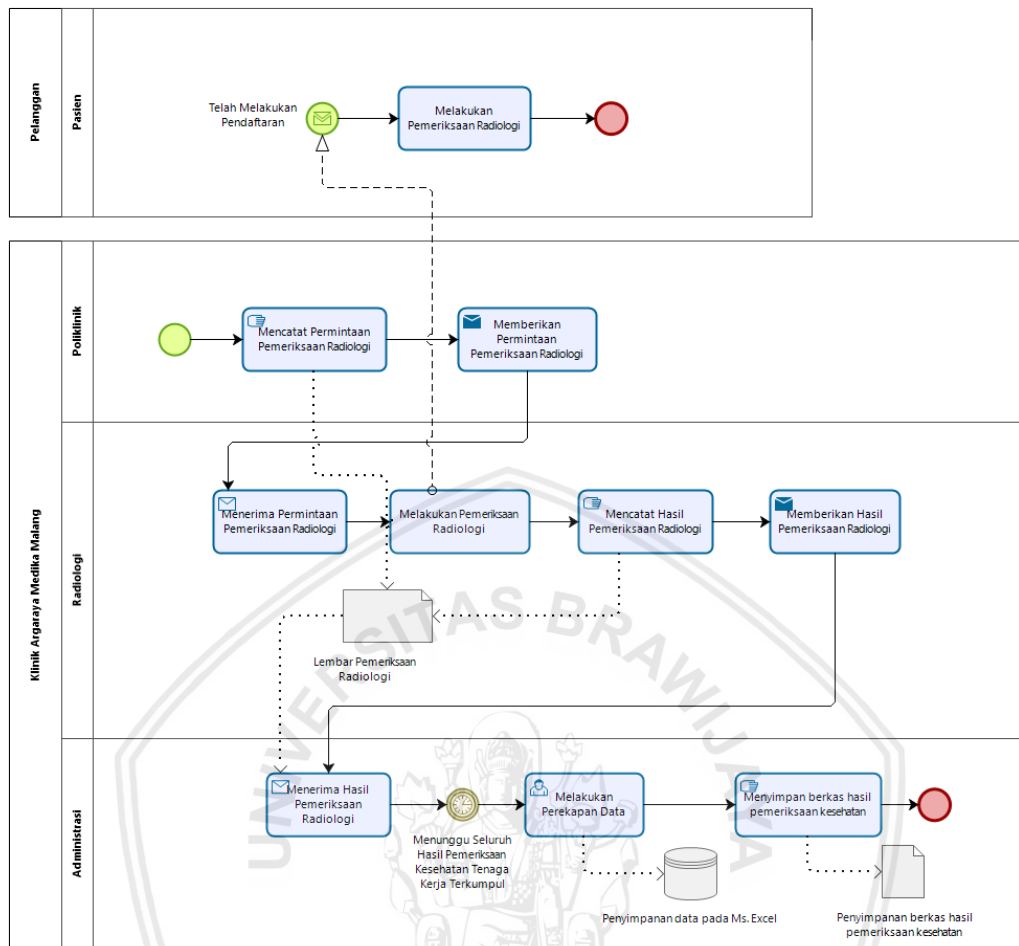
Proses bisnis pemeriksaan laboratorium dilakukan dengan pengambilan *sample* yang dilakukan oleh petugas poliklinik dan pencatatan permintaan pemeriksaan pada formulir pemeriksaan laboratorium, selanjutnya akan diberikan kepada petugas bagian laboratorium untuk dilakukan pemeriksaan. Hasil pemeriksaan akan dicatat oleh petugas laboratorium dan setelah selesai akan diberikan kepada administrasi untuk direkap. Perekapan data hasil pemeriksaan di masukan dalam *Ms. Excel* oleh petugas administrasi. Pemodelan Proses Bisnis *as-is* Pemeriksaan Laboratorium terdapat pada Gambar 4.3.



Gambar 4.3 Proses Bisnis As-Is Pemeriksaan Laboratorium

4.1.1.4 Proses Bisnis As-Is Pemeriksaan Radiologi

Pada proses bisnis pemeriksaan radiologi pemeriksaan dilakukan setelah petugas menerima permintaan pemeriksaan yang diberikan oleh poliklinik, lalu petugas radiologi akan melakukan pemeriksaan yang dibutuhkan dan mencatat hasil pemeriksaan pada lembar pemeriksaan radiologi. Hasil pemeriksaan akan diberikan pada administrasi untuk dilakukan perekapan. Pemodelan Proses Bisnis *as-is* Pemeriksaan Radiologi terdapat pada Gambar 4.4.



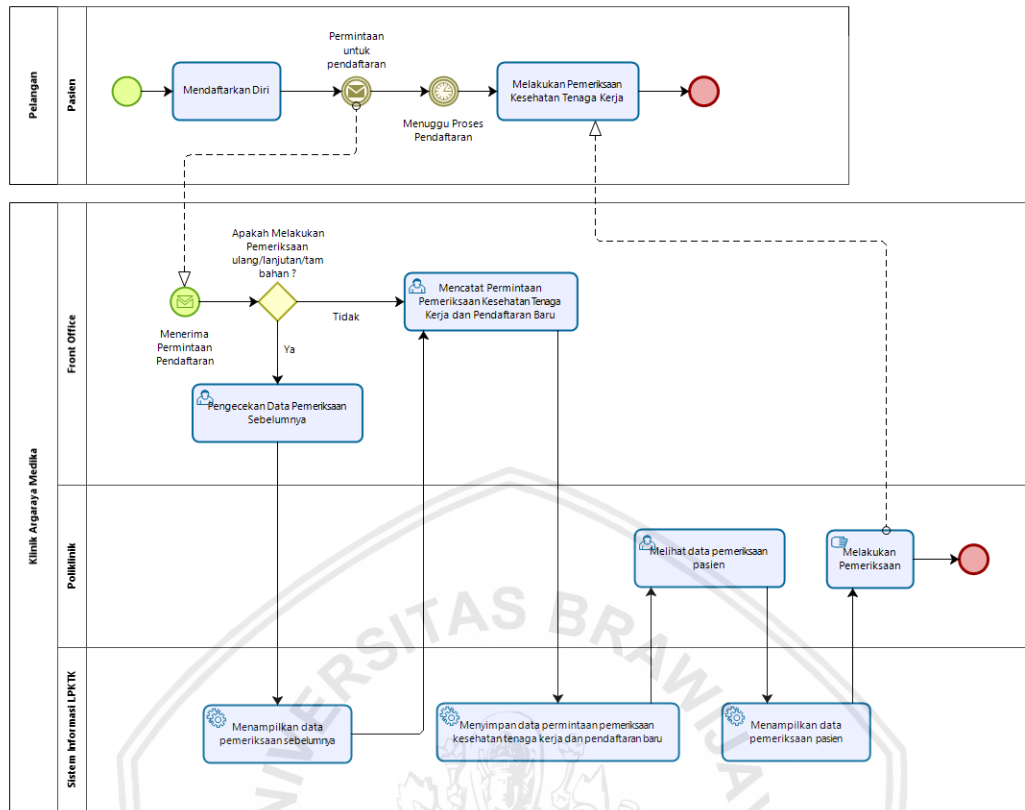
Gambar 4.4 Proses Bisnis As-Is Pemeriksaan Radiologi

4.1.2 Pemodelan Proses Bisnis To-Be

Proses bisnis *to-be* adalah proses bisnis usulan yang ditawarkan kepada pemangku kepentingan dalam penelitian ini. Hasil analisis terhadap proses bisnis *as-is* yang sudah dilakukan digunakan sebagai dasar dalam mengidentifikasi proses bisnis *to-be*. Pemodelan dilakukan untuk menggambarkan perubahan atau penambahan proses bisnis yang diusulkan. Hasil dari pemodelan proses bisnis *to-be* akan digunakan sebagai masukan dalam kegiatan analisis persyaratan.

4.1.2.1 Proses Bisnis To-Be Pendaftaran Pasien

Proses bisnis *to-be* pendaftaran pasien merupakan proses bisnis usulan dengan sistem informasi yang nantinya akan diterapkan di dalam layanan pemeriksaan kesehatan tenaga kerja. Perubahan yang direkomendasikan dari proses bisnis *to-be* dapat dilihat pada Gambar 4.5. Pendaftaran dan pencatatan permintaan pemeriksaan kesehatan tenaga kerja dilakukan keseluruhan oleh petugas *Front Office* di mana sebelumnya dilakukan oleh petugas *Front Office* dan Poliklinik yang melakukan pengisian biodata dua kali pada dua formulir. Poliklinik nantinya akan melakukan pengecekan pendaftaran dan melakukan pemeriksaan.



Gambar 4.5 Proses Bisnis To-be Pendaftaran Pasien

Dalam memodelkan proses bisnis *to-be* selanjutnya dapat kita bandingkan perubahan yang akan dilakukan dengan menganalisis perubahan aktivitas proses bisnis. Perubahan aktivitas proses bisnis yang dilakukan yaitu pengurangan *task*, penambahan *task*, dan pengubahan *task*. Hasil dari perubahan aktivitas proses bisnis dapat dilihat pada Tabel 4.1 berikut. Penambahan *task* pada sistem informasi untuk menunjukkan aktivitas pengguna dalam menggunakan sistem informasi dalam proses bisnis pendaftaran pasien.

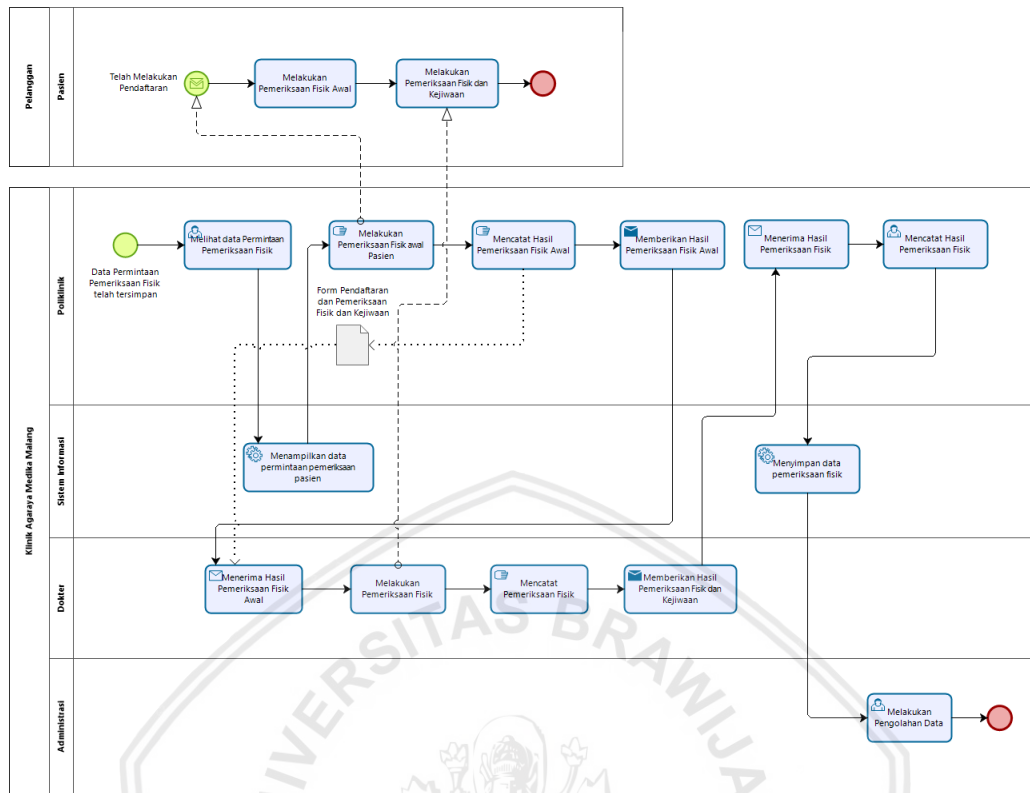
Tabel 4.1 Perubahan Aktivitas Proses Bisnis Pendaftaran Pasien

Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses bisnis <i>to-be</i>	Keterangan
Front Office	Mencatat Permintaan Pemeriksaan Kesehatan Tenaga Kerja Baru	Mencatat permintaan pemeriksaan kesehatan tenaga kerja dan pendaftaran baru	Diubah
Front Office	Memberikan permintaan Pemeriksaan Kesehatan Tenaga Kerja	-	Dieliminasi

Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses bisnis <i>to-be</i>	Keterangan
Front Office	Permintaan verifikasi pemeriksaan sebelumnya	Pengecekan data pemeriksaan sebelumnya	Diubah
Poliklinik	Mencatat pendaftaran dan permintaan pemeriksaan kesehatan	Melihat permintaan pemeriksaan	Diubah
Poliklinik	Memberikan permintaan pemeriksaan kesehatan lain	-	Dieliminasi
Administrasi	Menerima permintaan verifikasi pemeriksaan sebelumnya	-	Dieliminasi
Administrasi	Melakukan pengecekan data pasien	-	Dieliminasi
Administrasi	Memberikan informasi data pemeriksaan sebelumnya	-	Dieliminasi
Sistem Informasi	-	Menampilkan data pemeriksaan sebelumnya	Ditambah
Sistem Informasi	-	Menyimpan data permintaan pemeriksaan dan pendaftaran baru	Ditambah
Sistem Informasi	-	Menampilkan data pemeriksaan pasien	Ditambah

4.1.2.2 Proses Bisnis *To-Be* Pemeriksaan Fisik dan Kejiwaan

Rekomendasi proses bisnis *to-be* pemeriksaan fisik dan kejiwaan dapat dilihat pada Gambar 4.6. Perubahan yang dilakukan petugas poliklinik akan melakukan pengecekan data permintaan pemeriksaan fisik dan kejiwaan melalui sistem dan akan melakukan pemeriksaan fisik awal secara langsung kepada pasien. Data hasil pemeriksaan fisik awal akan dicatat dalam formulir dan diberikan kepada dokter untuk melakukan pemeriksaan fisik dan kejiwaan lanjutan. Setelah selesai maka akan diberikan kepada petugas poliklinik untuk memasukkan data pemeriksaan fisik dan kejiwaan ke dalam sistem. Data hasil pemeriksaan kesehatan nantinya akan diolah oleh petugas administrasi.



Gambar 4.6 Proses Bisnis *To-be* Pemeriksaan Fisik dan Kejiwaan

Dalam memodelkan proses bisnis *to-be* selanjutnya dapat kita bandingkan perubahan yang akan dilakukan dengan menganalisis perubahan aktivitas proses bisnis. Perubahan aktivitas proses bisnis yang dilakukan yaitu pengurangan *task*, penambahan *task*, dan perubahan *task*. Hasil dari perubahan aktivitas proses bisnis dapat dilihat pada Tabel 4.2 berikut. Penambahan *task* pada poliklinik menunjukkan pengisian data dan penggunaan sistem hanya dilakukan oleh poliklinik. *Task* sistem informasi yang ditambahkan bertujuan untuk menunjukkan aktivitas pengguna dalam menggunakan sistem.

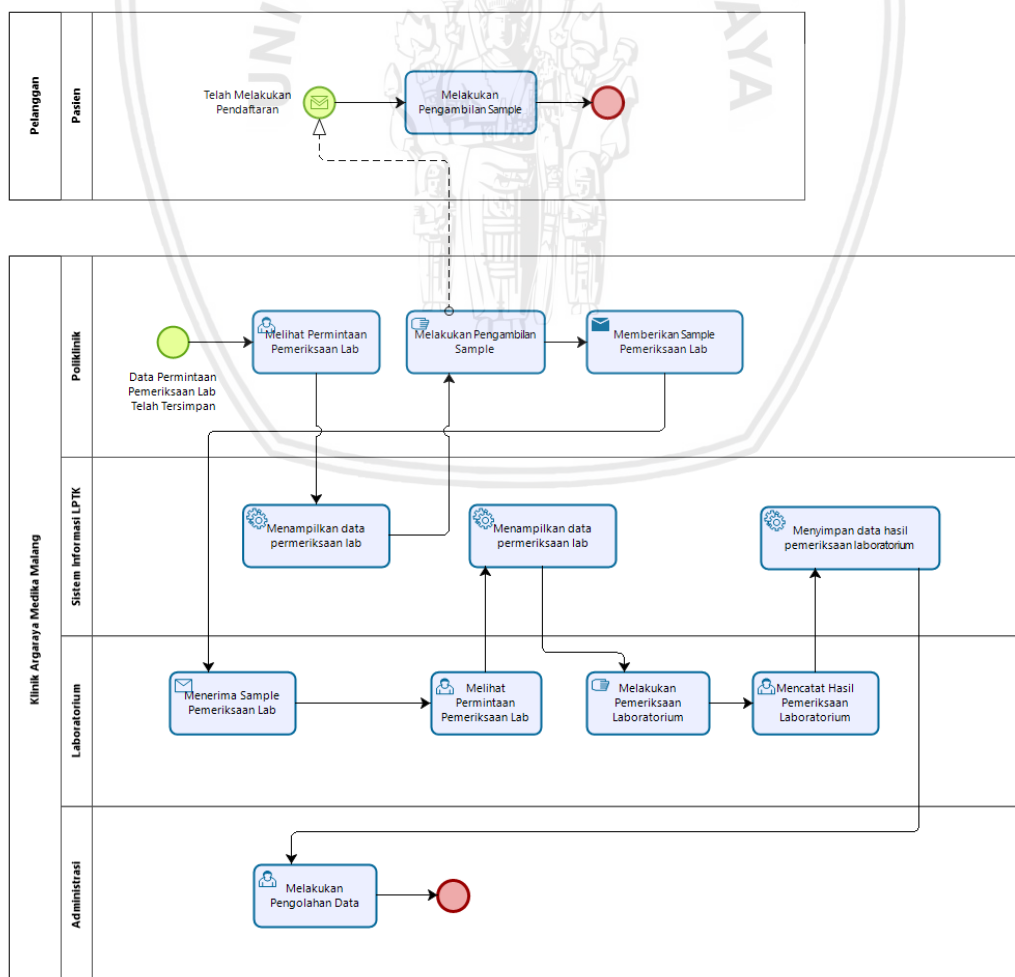
Tabel 4.2 Perubahan Aktivitas Proses Bisnis Pemeriksaan Fisik dan Kejiwaan

Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses bisnis <i>to-be</i>	Keterangan
Poliklinik	Menerima permintaan pemeriksaan fisik	Melihat permintaan pemeriksaan fisik	Diubah
Poliklinik	-	Menerima hasil pemeriksaan fisik	Ditambah
Poliklinik	-	Mencatat hasil pemeriksaan fisik	Ditambah
Administrasi	Menerima hasil pemeriksaan fisik dan kejiwaan	-	Dieliminasi
Administrasi	Melakukan perekapan data	Melakukan pengolahan data	Diubah

Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses bisnis <i>to-be</i>	Keterangan
Administrasi	Menyimpan berkas hasil pemeriksaan kesehatan	-	Dieliminasi
Sistem Informasi	-	Menampilkan data permintaan pemeriksaan pasien	Ditambah
Sistem Informasi	-	Menyimpan data pemeriksaan fisik	Ditambah

4.1.2.3 Proses Bisnis *To-Be* Pemeriksaan Laboratorium

Proses bisnis *to-be* pemeriksaan laboratorium pada petugas laboratorium melakukan pengecekan data pasien yang membutuhkan pemeriksaan laboratorium pada sistem. Petugas poliklinik akan melakukan pengambilan *sample* dan diberikan kepada petugas laboratorium lalu melakukan tes laboratorium. Hasil dari pemeriksaan laboratorium akan di masukan ke dalam sistem dan hasilnya akan diolah oleh petugas administrasi. Pemodelan proses bisnis *to-be* pemeriksaan laboratorium dapat dilihat pada Gambar 4.7.



Gambar 4.7 Proses Bisnis *To-be* Pemeriksaan Laboratorium

Dalam memodelkan proses bisnis *to-be* selanjutnya dapat kita bandingkan perubahan yang akan dilakukan dengan menganalisis perubahan aktivitas proses bisnis. Perubahan aktivitas proses bisnis yang dilakukan yaitu pengurangan *task*, penambahan *task*, dan pengubahan *task*. Hasil dari perubahan aktivitas proses bisnis dapat dilihat pada Tabel 4.3 berikut. Penambahan *task* sistem informasi bertujuan untuk menampilkan aktivitas pengguna dalam menggunakan sistem informasi.

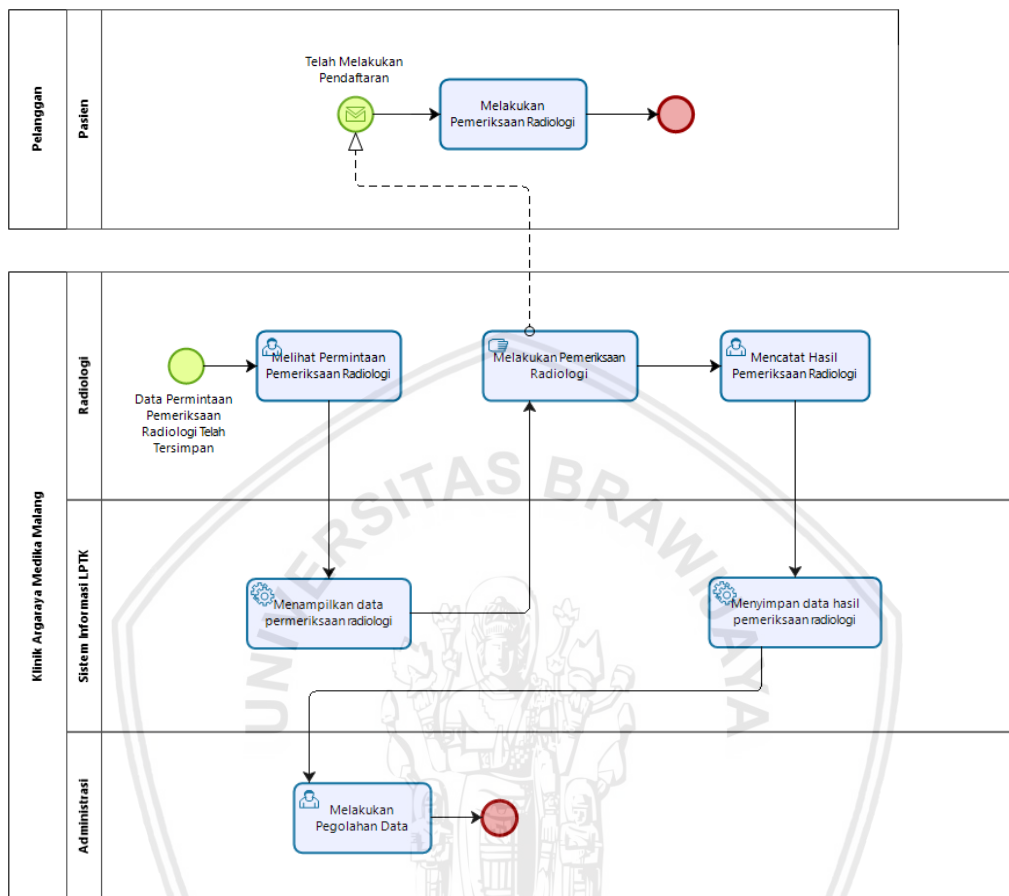
Tabel 4.3 Perubahan Aktivitas Proses Bisnis Pemeriksaan Laboratorium

Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses bisnis <i>to-be</i>	Keterangan
Poliklinik	Memberikan permintaan dan <i>sample</i> pemeriksaan lab	Memberikan <i>sample</i> pemeriksaan lab	Diubah
Poliklinik	Mencatat permintaan pemeriksaan lab	Melihat permintaan pemeriksaan lab	Diubah
Laboratorium	Menerima permintaan dan <i>sample</i> pemeriksaan lab	Menerima <i>sample</i> pemeriksaan lab	Diubah
Laboratorium	Memberikan hasil pemeriksaan laboratorium	Menyimpan data hasil pemeriksaan laboratorium	Diubah
Laboratorium	-	Melihat permintaan pemeriksaan lab	Ditambah
Administrasi	Menerima hasil pemeriksaan laboratorium	-	Dieliminasi
Administrasi	Melakukan perekapan data	Melakukan pengolahan data	Diubah
Administrasi	Menyimpan berkas hasil pemeriksaan kesehatan	-	Dieliminasi
Sistem Informasi	-	Menampilkan data pemeriksaan lab	Ditambah
Sistem Informasi	-	Menyimpan data pemeriksaan lab	Ditambah

4.1.2.4 Proses Bisnis *To-Be* Pemeriksaan Radiologi

Rekomendasi proses bisnis *to-be* pemeriksaan radiologi dapat dilihat pada Gambar 4.8. Perubahan yang dilakukan nantinya petugas radiologi hanya akan

melakukan pengecekan pasien yang membutuhkan pemeriksaan radiologi dan melakukan pengisian hasil dari pemeriksaan radiologi setelah selesai dilakukan. Data hasil pemeriksaan kesehatan nantinya akan diolah oleh petugas administrasi.



Gambar 4.8 Proses Bisnis *To-be* Pemeriksaan Radiologi

Dalam memodelkan proses bisnis *to-be* selanjutnya dapat kita bandingkan perubahan yang akan dilakukan dengan menganalisis perubahan aktivitas proses bisnis. Perubahan aktivitas proses bisnis yang dilakukan yaitu pengurangan *task*, penambahan *task*, dan pengubahan *task*. Hasil dari perubahan aktivitas proses bisnis dapat dilihat pada Tabel 4.4 berikut. Perubahan *task* dalam poliklinik menunjukkan pencatatan permintaan pemeriksaan radiologi untuk diberikan kepada radiologi tidak diperlukan lagi karena permintaan pemeriksaan telah tersimpan dalam sistem yang sebelumnya telah di masukan saat pendaftaran. Penambahan *task* sistem informasi bertujuan untuk menampilkan aktivitas pengguna dalam menggunakan sistem informasi.

Tabel 4.4 Perubahan Aktivitas Proses Bisnis Pemeriksaan Radiologi

Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses bisnis <i>to-be</i>	Keterangan
Poliklinik	Memberikan permintaan pemeriksaan radiologi	-	Dieliminasi
Poliklinik	Mencatat permintaan pemeriksaan radiologi	-	Dieliminasi
Radiologi	Menerima permintaan pemeriksaan radiologi	Melihat permintaan pemeriksaan radiologi	Diubah
Radiologi	Memberikan hasil pemeriksaan radiologi	Menyimpan data hasil pemeriksaan radiologi	Diubah
Administrasi	Menerima hasil pemeriksaan radiologi	-	Dieliminasi
Administrasi	Melakukan perekapan data	Melakukan pengolahan data	Diubah
Administrasi	Menyimpan berkas hasil pemeriksaan kesehatan	-	Dieliminasi
Sistem Informasi	-	Menampilkan data pemeriksaan radiologi	Ditambah
Sistem Informasi	-	Menyimpan data pemeriksaan radiologi	Ditambah

4.1.3 Analisis *Improvement* Waktu Proses Bisnis

Proses bisnis *as-is* yang dilakukan peningkatan dengan memodelkan proses bisnis *to-be* dilakukan analisis waktu. Analisis waktu dilakukan untuk menunjukkan peningkatan proses bisnis setelah dilakukan perubahan dalam hal waktu yang dibutuhkan untuk melakukan proses bisnis. Dalam Tabel 4.5 menunjukkan analisis waktu dari proses bisnis yang dilakukan peningkatan. Analisis waktu diambil dari proses unit proses bisnis yang mengalami peningkatan waktu di antaranya pendaftaran pasien baru, pendaftaran pasien lanjutan / tambahan / ulang, pencatatan data laboratorium, pencatatan data radiologi, dan perekapan data oleh admin.

Tabel 4.5 Analisis Waktu

Proses Bisnis	Waktu proses bisnis <i>as-is</i>	Waktu proses bisnis <i>to-be</i>	Keterangan
Pendaftaran pasien baru	7 menit	3 menit 30 detik	Mempersingkat 3 menit 30 detik
Pendaftaran pasien lanjutan / tambahan / ulang	14 menit	5 menit 30 detik	Mempersingkat 8 menit 30 detik
Pencatatan data pemeriksaan laboratorium	6 menit	5 menit	Mempersingkat 1 menit
Pencatatan data Pemeriksaan Radiologi	6 menit	2 menit 30 detik	Mempersingkat 3 menit 30 detik
Perekapan oleh admin pada setiap pemeriksaan laboratorium, radiologi, fisik dan kejiwaan	15 menit	5 menit	Mempersingkat 10 menit

4.2 Analisis Persyaratan

4.2.1 Identifikasi Tipe Pemangku Kepentingan

Identifikasi tipe pemangku kepentingan dilakukan bertujuan mengelompokkan dan mengetahui pemangku kepentingan sesuai dengan hubungan pemangku kepentingan dengan sistem yang dikembangkan dan karakteristik pemangku kepentingan. Hasil kegiatan ini akan digunakan sebagai informasi dalam melakukan kegiatan menganalisis permasalahan, sehingga peneliti mampu mendapat informasi masalah dari berbagai pemangku kepentingan pada tempat studi kasus. Tipe pemangku kepentingan dijelaskan pada Tabel 4.6.

Tabel 4.6 Tipe Pemangku Kepentingan

Tipe Pemangku Kepentingan	Keterangan	Pemangku Kepentingan
Pengguna	Individu yang melakukan interaksi secara langsung dengan sistem informasi serta berperan sebagai actor pada <i>use case</i> .	<ul style="list-style-type: none"> • Petugas <i>Front Office</i> • Petugas Poliklinik • Petugas Administrasi • Petugas Laboratorium • Petugas Radiologi

Tipe Pemangku Kepentingan	Keterangan	Pemangku Kepentingan
Pengembang	Individu atau organisasi yang melakukan pengembangan sistem informasi.	Peneliti
Pihak yang berwenang	Individu atau organisasi yang dapat memberikan dukungan dengan memberikan informasi yang dibutuhkan terkait regulasi yang berjalan, sehingga solusi dapat diterapkan sesuai aturan yang berlaku.	Klinik Argaraya Medika Malang
Pelanggan	Individu atau organisasi yang akan mendapatkan manfaat dari penerapan sistem informasi.	<ul style="list-style-type: none"> • Klinik Argaraya Medika Malang • Pasien Layanan Pemeriksaan Kesehatan Tenaga Kerja

4.2.2 Analisis Permasalahan

Analisa permasalahan dilakukan dengan melakukan Observasi secara langsung pada lokasi penelitian dan Wawancara dengan pihak yang berkepentingan di Layanan Pemeriksaan Kesehatan Tenaga Kerja Klinik Argaraya Medika serta analisis dari proses bisnis *as-is* yang telah dimodelkan. *Problem statement* yang didapatkan melalui analisis hasil observasi dan wawancara pengurus Administrasi & kesekretariatan sebagai pelaku serta pengurus Layanan Pemeriksaan Kesehatan Tenaga Kerja serta analisis proses bisnis *as-is* terdapat pada tabel 4.7.

Tabel 4.7 Problem Statement

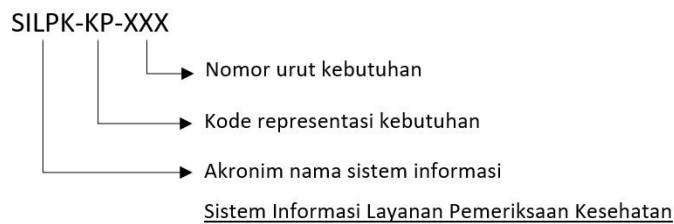
The problem of	<ul style="list-style-type: none"> • Pencatatan permintaan pemeriksaan kesehatan dilakukan dua kali pada <i>front office</i> dan poliklinik membutuhkan waktu yang lebih. • Petugas laboratorium dan radiologi harus menunggu informasi dari poliklinik untuk mengetahui kebutuhan pemeriksaan kesehatan dari pasien melalui formulir. • Perekapan dilakukan diakhir setelah seluruh pasien melakukan pemeriksaan dan dimasukan secara manual dalam Ms. Excel oleh petugas Administrasi.
-----------------------	---

	<ul style="list-style-type: none"> • Pengecekan formuir hasil yang telah terisi hanya dilakukan dengan pemberian paraf pada formulir. • Data diri pasien diisi pada semua formuir pemeriksaan berulang-ulang.
Affects	Pengguna, Pihak yang berwenang, Pelanggan.
The impact of which is	<ul style="list-style-type: none"> • Resiko hilangnya formulir pemeriksaan dan data pemeriksaan belum dilakukan perekapan. • Penyebaran informasi kebutuhan pemeriksaan kesehatan pasien kurang efisien. • Perekapan data oleh petugas Administrasi membutuhkan waktu yang lama karena data pasien dimasukan secara manual ke dalam Ms. Excel. • Memperlambat proses pemeriksaan petugas Laboratorium dan Radiologi harus menunggu petugas Poliklinik memberikan informasi melalui formulir kertas.
A successful solution would	Menyediakan sebuah sistem informasi yang dapat dijadikan sebagai media dalam melakukan pendataan, penyimpanan data maupun penyebaran informasi secara menyeluruh untuk informasi pemeriksaan kesehatan pasien dan dapat diakses bersamaan dengan batasan akses pada pengguna sesuai kebutuhan.

4.2.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna

Identifikasi kebutuhan pengguna didapatkan melalui analisis terhadap *problem statement* dan hasil wawancara dengan pengurus Administrasi & kesekretariatan sebagai pengurus dan pelaku pelayanan pemeriksaan kesehatan tenaga kerja. Hasil identifikasi kebutuhan ini nantinya akan dijadikan dasar identifikasi fitur, yang merupakan tahap untuk identifikasi solusi yang ditawarkan untuk memenuhi kebutuhan pengguna. Dalam identifikasi ini kebutuhan pengguna diberi kode sebagai identitas, keterangan kodifikasi kebutuhan pengguna seperti pada Gambar 4.9.





Gambar 4.9 Kodifikasi Kebutuhan Pengguna

Pada Tabel 4.8 berisi informasi kode identitas untuk masing-masing kebutuhan pengguna, pernyataan kebutuhan pengguna, pemangku kepentingan yang termasuk dalam kategori pengguna dan pengembang, yang memiliki keterkaitan dengan kebutuhan yang telah diidentifikasi, situasi saat ini yang berkaitan dengan kegiatan yang dilakukan untuk memenuhi kebutuhan pengguna, dan solusi yang ditawarkan untuk memenuhi masing-masing kebutuhan yang telah teridentifikasi.

Tabel 4.8 Identifikasi Kebutuhan Pemangku Kepentingan

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi saat ini	Solusi yang ditawarkan
SILPK-KP-001	Sistem harus dapat memberikan layanan pengelola akun pengguna yang berhak melakukan akses terhadap sistem.	Petugas Administrasi.	-	Sistem informasi yang menyediakan layanan pengelola akun pengguna yang berhak melakukan akses terhadap sistem.
SILPK-KP-002	Informasi di dalam sistem harus dapat diakses sesuai dengan identitas.	Petugas <i>Front Office</i> , Petugas Poliklinik, Petugas Administrasi, Petugas Laboratorium, Petugas Radiologi.	-	Sistem informasi yang menyediakan layanan untuk membatasi akses terhadap pengguna dalam menggunakan sistem.
SILPK-KP-003	Sistem harus dapat memberikan	Petugas Poliklinik, Petugas	Informasi data pemeriksaan	Sistem informasi menyediakan layanan

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi saat ini	Solusi yang ditawarkan
	layanan untuk membantu penyebaran informasi data pemeriksaan kesehatan pasien kepada petugas pemeriksa kesehatan lain.	Administrasi, Petugas Laboratorium, Petugas Radiologi.	kesehatan pasien ditulis oleh petugas poliklinik kedalam formulir dan diberikan kepada petugas lain.	informasi data pemeriksaan pasien yang dapat diakses kapan saja kepada seluruh petugas dengan batasan akses informasi sesuai kebutuhan.
SILPK-KP-004	Sistem harus dapat menampilkan informasi hasil pemeriksaan kesehatan pasien TKLN dan TKDN yang dapat diakses kapan saja.	Petugas Poliklinik, Petugas Administrasi, Petugas Laboratorium, Petugas Radiologi.	Informasi data hasil pemeriksaan pasien didapatkan dari melihat formulir hasil pemeriksaan pada setiap tahap pemeriksaan atau melihat melalui berkas pemeriksaan.	Sistem informasi menyediakan layanan menampilkan informasi hasil pemeriksaan kesehatan pasien TKLN dan TKDN yang dapat diakses kapan saja.
SILPK-KP-005	Sistem harus dapat menampilkan informasi semua data pendaftaran pasien TKLN dan TKDN.	Petugas <i>Front Office</i> , Petugas Administrasi.	Informasi data pendaftaran pasien didapatkan dari melihat formulir pendaftaran pasien atau melihat melalui berkas pemeriksaan.	Sistem informasi dapat menyediakan layanan menampilkan informasi data pendaftaran pasien TKLN dan TKDN yang dapat diakses kapan saja.



Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi saat ini	Solusi yang ditawarkan
SILPK-KP-006	Sistem harus dapat menyediakan informasi <i>progress</i> pengerjaan data pemeriksaan kesehatan pasien.	Petugas Administrasi.	Informasi <i>progress</i> pengerjaan didapatkan dari pemberian paraf pada formulir ketika pemeriksaan selesai dan hasil pemeriksaan telah diisi dalam formulir.	Sistem informasi dapat menyediakan informasi <i>progress</i> pengerjaan data pemeriksaan kesehatan pasien.
SILPK-KP-007	Sistem harus menyediakan layanan pengisian dan penyimpanan semua data pendaftaran pasien TKLN dan TKDN.	Front Office, Administrasi.	Data pendaftaran pasien dicatat dalam formulir kertas dan dikumpulkan di petugas administrasi untuk dilakukan perekapan.	Sistem informasi yang menyediakan layanan pengisian dan penyimpanan semua data pendaftaran pasien TKLN dan TKDN.
SILPK-KP-008	Sistem harus menyediakan layanan pengisian dan penyimpanan semua data pemeriksaan kesehatan Fisik dan Kejiwaan pasien TKLN dan TKDN.	Petugas Poliklinik, Petugas Administrasi.	Data pemeriksaan dicatat dalam formulir kertas dan dikumpulkan di petugas administrasi untuk dilakukan perekapan.	Sistem informasi yang menyediakan layanan pengisian dan penyimpanan semua data pemeriksaan kesehatan Fisik dan Kejiwaan pasien TKLN dan TKDN.
SILPK-KP-009	Sistem harus menyediakan	Petugas Laboratorium,	Data pemeriksaan	Sistem informasi yang

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi saat ini	Solusi yang ditawarkan
	layanan pengisian dan penyimpanan semua data pemeriksaan kesehatan Laboratorium pasien TKLN dan TKDN.	Petugas Administrasi.	dicatat dalam formulir kertas dan dikumpulkan di petugas administrasi untuk dilakukan perekapan.	menyediakan layanan pengisian dan penyimpanan semua data pemeriksaan kesehatan Laboratorium pasien TKLN dan TKDN.
SILPK-KP-010	Sistem harus menyediakan layanan pengisian dan penyimpanan semua data pemeriksaan kesehatan Radiologi pasien TKLN dan TKDN.	Petugas Radiologi, Petugas Administrasi.	Data pemeriksaan dicatat dalam formulir kertas dan dikumpulkan di petugas administrasi untuk dilakukan perekapan.	Sistem informasi yang menyediakan layanan pengisian dan penyimpanan semua data pemeriksaan kesehatan Radiologi pasien TKLN dan TKDN.

4.2.3.1 Perbaikan Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna

Perbaikan dilakukan pada Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna pada saat melakukan pengidentifikasian ulang kepada *stakeholder* fase *inception* metode *Rational Unified Process*. Tabel 4.9 menunjukkan perbaikan Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna. Perbaikan dilakukan pada kode kebutuhan SILPK-KP-006 di mana pengguna menambahkan untuk sistem agar dapat menampilkan informasi pengolahan data pemeriksaan. Kebutuhan Pemangku Kepentingan dan Pengguna sebelumnya diperbaiki sesuai dengan Tabel 4.9.

Tabel 4.9 Perbaikan Identifikasi Kebutuhan Pemangku Kepentingan

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi saat ini	Solusi yang ditawarkan
SILPK-KP-006	Sistem harus dapat menyediakan informasi <i>progress</i>	Petugas Administrasi.	Informasi <i>progress</i> pengerjaan didapatkan dari	Sistem informasi dapat menyediakan informasi <i>progress</i>

	pengerjaan data pemeriksaan kesehatan pasien dan menampilkan informasi pengolahan data pemeriksaan.		pemberian paraf pada formulir ketika pemeriksaan selesai dan hasil pemeriksaan telah diisi dalam formulir.	pengerjaan data pemeriksaan kesehatan pasien serta menampilkan informasi pengolahan data pemeriksaan.
--	---	--	--	---

4.2.4 Identifikasi Pengguna

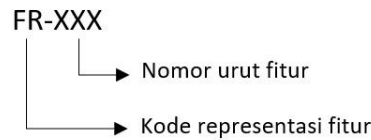
Pada tahap ini juga dilakukan identifikasi pengguna (aktor) yang terlibat langsung dalam penggunaan sistem. Dengan dilakukan identifikasi pengguna (aktor) dapat mengetahui karakter dari masing-masing pengguna (aktor). Penjelasan hasil identifikasi pengguna terdapat pada Tabel 4.10. Informasi mengenai identifikasi pengguna digunakan untuk melakukan identifikasi aktor yang akan dimodelkan ke dalam diagram *use case*.

Tabel 4.10 Hasil Identifikasi Pengguna

Pengguna	Deskripsi
Petugas <i>Front Office</i>	Bertugas memberikan informasi terkait layanan serta mencatat pendaftaran pemeriksaan kesehatan tenaga kerja di Klinik Argaraya Medika Malang.
Petugas Poliklinik	Bertugas untuk melakukan pemeriksaan fisik dan kejiwaan dan melakukan pencatatan hasil pemeriksaan fisik dan kejiwaan.
Petugas Administrasi	Bertanggung jawab untuk melakukan perekapan data keseluruhan hasil pemeriksaan kesehatan pasien.
Petugas Laboratorium	Bertugas melakukan pemeriksaan laboratorium sesuai permintaan pasien serta melakukan pencatatan hasil pemeriksaan.
Petugas Radiologi	Bertugas melakukan pemeriksaan radiologi sesuai permintaan pasien serta melakukan pencatatan hasil pemeriksaan.



4.2.5 Identifikasi Fitur



Gambar 4.10 Kodifikasi Fitur

Identifikasi fitur merupakan deskripsi singkat dari layanan-layanan yang disediakan oleh sistem untuk dapat memenuhi kebutuhan pengguna. Pada setiap fitur diberi kode identitas seperti pada Gambar 4.10. Nantinya identifikasi fitur ini digunakan sebagai informasi identifikasi kebutuhan fungsional dan non-fungsional dari sistem yang akan dikembangkan. Dalam Tabel 4.11 menunjukkan informasi fitur dari Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja pada Klinik Argaraya Medika Malang.

Tabel 4.11 Identifikasi Fitur

Kode Fitur	Deskripsi
FR-01	Sistem dapat mengenali identitas pengguna dan membatasi akses pengguna sesuai hak akses pengguna.
FR-02	Sistem dapat menampilkan data pemeriksaan pasien terkait data diri pasien, kebutuhan pemeriksaan kesehatan, dan hasil pemeriksaan kesehatan.
FR-03	Sistem dapat menampilkan informasi <i>progress</i> pengerjaan pemeriksaan kesehatan pasien.
FR-04	Sistem dapat digunakan untuk mengelola data pendaftaran pasien TKLN dan TKDN.
FR-05	Sistem dapat digunakan untuk mengelola data hasil pemeriksaan fisik dan kejiwaan pada TKLN dan TKDN.
FR-06	Sistem dapat digunakan untuk mengelola data hasil pemeriksaan radiologi pada TKLN dan TKDN.
FR-07	Sistem dapat digunakan untuk mengelola data hasil pemeriksaan fisik dan kejiwaan pada TKLN dan TKDN.
FR-08	Sistem dapat digunakan untuk mengelola akun pengguna.

Masing-masing fitur pada yang sudah sudah diidentifikasi selanjutnya dihubungkan berdasar pada keterkaitan dengan kebutuhan penggunaan yang harus dipenuhi. Penghubungan ini menunjukkan bahwa suatu fitur merupakan solusi dalam memenuhi sebuah kebutuhan tertentu. Pengelompokan dan penghubungan fitur dengan kebutuhan pengguna di Tabel 4.12.

Tabel 4.12 Hubungan Kebutuhan Pengguna dengan Fitur

Kebutuhan Pengguna	Fitur
SILPK-KP-001	FR-08
SILPK-KP-002	FR-01
SILPK-KP-003	FR-02
SILPK-KP-004	
SILPK-KP-005	
SILPK-KP-006	FR-03
SILPK-KP-007	FR-04
SILPK-KP-008	FR-05
SILPK-KP-009	FR-06
SILPK-KP-010	FR-07

4.2.5.1 Perbaikan Identifikasi Fitur

Perbaikan dilakukan pada Identifikasi Fitur disebabkan perbaikan yang dilakukan pada Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna. Tabel 4.13 menunjukkan perbaikan fitur dimana dilakukan penambahan fitur untuk kode fitur FR-09. Fitur selanjutnya diperbaiki sesuai penambahan pada Tabel 4.13 sehingga hubungan kebutuhan pengguna dengan fitur diperbaiki seperti pada Tabel 4.14.

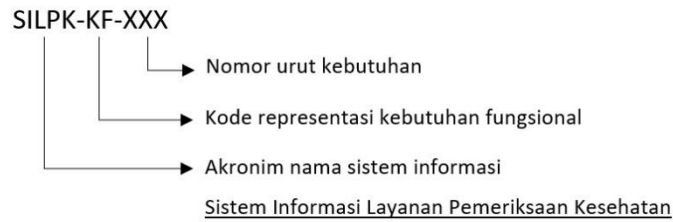
Tabel 4.13 Perbaikan Identifikasi Fitur

Kode Fitur	Deskripsi
FR-09	Sistem dapat menampilkan informasi pelaporan pengolahan data pemeriksaan kepada pengguna.

Tabel 4.14 Perbaikan Hubungan Kebutuhan Pengguna dengan Fitur

Kebutuhan Pengguna	Fitur
SILPK-KP-001	FR-08
SILPK-KP-002	FR-01
SILPK-KP-003	FR-02
SILPK-KP-004	
SILPK-KP-005	
SILPK-KP-006	FR-03
	FR-09
SILPK-KP-007	FR-04
SILPK-KP-008	FR-05
SILPK-KP-009	FR-06
SILPK-KP-010	FR-07

4.2.6 Persyaratan Fungsional



Gambar 4.11 Kodifikasi Kebutuhan Fungsional

Pada tahap ini menjelaskan kebutuhan fungsional dari sistem yang akan dikembangkan. Kebutuhan fungsional merupakan kondisi atau kemampuan yang harus dipenuhi oleh sistem agar fitur sistem dapat berjalan dan memenuhi kebutuhan pengguna. Tabel 4.15 menunjukkan hubungan antara fitur dengan kebutuhan fungsional sistem yang akan dikembangkan. Setiap kebutuhan fungsional diberi kodifikasi seperti pada Gambar 4.11.

Tabel 4.15 Kebutuhan Fungsional

Kode Fitur	Kode Kebutuhan Fungsional	Deskripsi
FR-01	SILPK-KF-01	Sistem dapat menampilkan halaman <i>login</i> pengguna.
	SILPK-KF-02	Sistem dapat mengidentifikasi pengguna berdasarkan data pengguna didalam sistem.
	SILPK-KF-03	Sistem dapat menyediakan fasilitas bagi pengguna untuk keluar dari sistem.
FR-08	SILPK-KF-04	Sistem dapat menambah data pengguna baru.
	SILPK-KF-05	Sistem dapat mengubah data pengguna baru.
	SILPK-KF-06	Sistem dapat menghapus data pengguna.
FR-02	SILPK-KF-07	Sistem dapat menampilkan data informasi pemeriksaan kesehatan pasien TKLN dan TKDN.
	SILPK-KF-08	Sistem dapat menampilkan data informasi data diri pasien TKLN dan TKDN.
	SILPK-KF-09	Sistem dapat menampilkan hasil pemeriksaan fisik pasien.
	SILPK-KF-10	Sistem dapat menampilkan hasil pemeriksaan laboratorium pasien.

Kode Fitur	Kode Kebutuhan Fungsional	Deskripsi
	SILPK-KF-11	Sistem dapat menampilkan hasil pemeriksaan radiologi pasien.
FR-03	SILPK-KF-12	Sistem dapat menampilkan <i>progress</i> pengerjaan data pemeriksaan kesehatan pasien TKLN dan TKDN.
FR-04	SILPK-KF-13	Sistem dapat menambah data pendaftaran pasien baru.
	SILPK-KF-14	Sistem dapat mengubah data diri pasien.
	SILPK-KF-15	Sistem dapat menambah data pemeriksaan pasien.
	SILPK-KF-16	Sistem dapat mengubah data pemeriksaan pasien.
FR-05	SILPK-KF-17	Sistem dapat mengubah data pemeriksaan kesehatan fisik dan kejiwaan.
	SILPK-KF-18	Sistem dapat menyimpan data pemeriksaan kesehatan fisik dan kejiwaan.
FR-06	SILPK-KF-19	Sistem dapat mengubah data pemeriksaan laboratorium.
	SILPK-KF-20	Sistem dapat menyimpan data pemeriksaan laboratorium.
FR-07	SILPK-KF-21	Sistem dapat mengubah data pemeriksaan radiologi.
	SILPK-KF-22	Sistem dapat menyimpan data pemeriksaan radiologi.

4.2.6.1 Perbaikan Persyaratan Fungsional

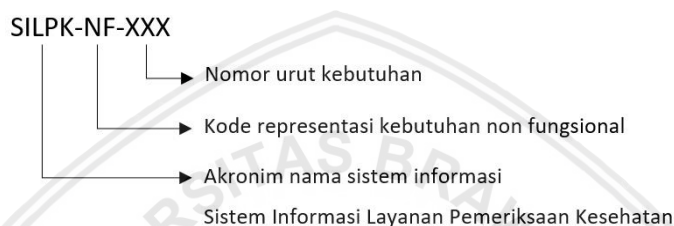
Perbaikan dilakukan pada persyaratan fungsional disebabkan perbaikan yang dilakukan pada Identifikasi Fitur dan saat melakukan identifikasi ulang kepada pemangku kepentingan fase insepisi. Tabel 4.15 menunjukkan perbaikan kebutuhan fungsional dimana dilakukan penambahan kebutuhan untuk kode kebutuhan fungsional SILPK-KF-23, SILPK-KF-24, dan SILPK-KF-25. Kebutuhan fungsional selanjutnya diperbaiki sesuai penambahan pada Tabel 4.15.

Tabel 4.16 Perbaikan Kebutuhan Fungsional

Kode Fitur	Kode Kebutuhan Fungsional	Deskripsi
------------	---------------------------	-----------

FR-04	SILPK-KF-23	Sistem dapat mengubah data hasil akhir pemeriksaan pasien.
FR-09	SILPK-KF-24	Sistem dapat menampilkan data jumlah pemeriksaan hari ini
	SILPK-KF-25	Sistem dapat menampilkan data jumlah pemeriksaan 5 hari sebelum hari ini.

4.2.7 Persyaratan Non Fungsional



Gambar 4.12 Kodifikasi Kebutuhan Non Fungsional

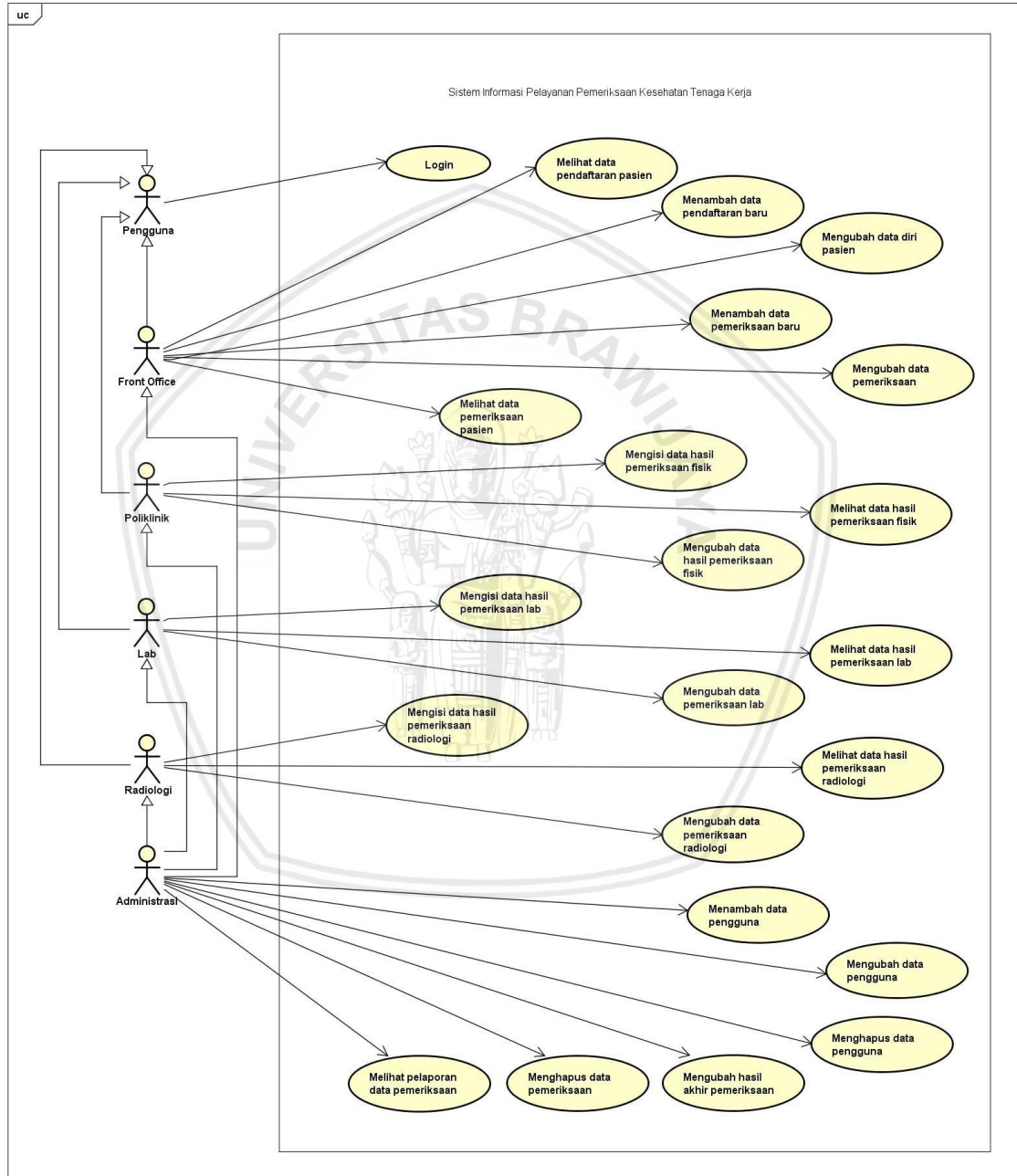
Dalam kebutuhan non fungsional merupakan kebutuhan yang harus dipenuhi oleh sistem agar dapat berjalan seperti seharusnya. Kebutuhan non fungsional ini didapatkan melalui identifikasi aplikasi *browser* serta sistem operasi yang digunakan pada Klinik Argataya Medika Malang. Pada Tabel 4.17 dijelaskan kebutuhan non fungsional dari sistem informasi. Pada kebutuhan non fungsional dilakukan juga kodifikasi seperti pada Gambar 4.12.

Tabel 4.17 Kebutuhan Non Fungsional

Kode Fitur	Kode Kebutuhan Non Fungsional	Deskripsi
FR-01	SILPK-KN-01	Sistem dapat berjalan dengan baik pada aplikasi <i>browser</i> yang berbeda.
FR-02		
FR-03		
FR-04		
FR-05		
FR-06		
FR-07		
FR-08		

4.3 Pemodelan Use Case

Pada tahap ini *Use case* diagram digunakan sebagai pemodelan tingkah laku (*behavior*) dari sistem yang akan dikembangkan. *Use case* Diagram mendeskripsikan fungsionalitas yang terdapat didalam sistem sebagai aktor yang berinteraksi yang dideskripsikan dengan kata kerja. Pemodelan *Use case* diagram ditunjukkan pada Gambar 4.13.



Gambar 4.13 Pemodelan UseCase Diagram

Gambaran sistem pada tahap ini divisualisasikan dalam bentuk *use case* sesuai persyaratan-persyaratan yang sudah dibuat sebelumnya. Pemodelan *use case*

kemudian dilanjutkan dengan identifikasi aktor untuk mengelompokan pengguna sistem berdasar pada karakteristik yang dimiliki, identifikasi *use case* kemudian menentukan tujuan aktor pengguna ketika menggunakan sistem, dan spesifikasi setiap *use case* untuk memberikan penjelasan singkat dari kondisi yang dibutuhkan sebelum melakukan suatu *use case* dan kondisi bagaimana sesudah *use case* dilakukan, serta urutan tahap menyelesaikan *use case* yang harus dilalui.

4.3.1 Use Case Diagram

Pada tahap ini *use case* yang sudah diidentifikasi akan dihubungkan dengan aktivitas proses bisnis *to-be* dan fitur-fitur yang sudah teridentifikasi sebelumnya sebagai solusi yang ditawarkan. *Use case* dengan aktivitas proses bisnis *to-be* dihubungkan untuk memperlihatkan hubungan yang lebih baik untuk penggunaan *use case* dan pemodelan proses bisnis pada proses pengembangan sistem informasi. Hubungan aktivitas proses bisnis *to-be* dengan *use case* terdapat pada Tabel 4.18.

Tabel 4.18 Hubungan Aktivitas Proses bisnis To-Be dengan Use Case

Proses Bisnis	Aktivitas	Use Case
Pendaftaran Pasien Baru	Mencatat permintaan pemeriksaan kesehatan tenaga kerja dan pendaftaran baru	Tambah data pendaftaran baru
		Ubah data diri pasien
		Tambah data pemeriksaan baru
		Ubah data pemeriksaan pasien
Pemeriksaan Fisik	Menampilkan data pemeriksaan pasien	Melihat data pendaftaran pasien
		Melihat data pemeriksaan pasien
Pemeriksaan Fisik	Mencatat hasil pemeriksaan fisik	Mengisi data hasil pemeriksaan fisik
		Mengubah data hasil pemeriksaan fisik
Pemeriksaan Laboratorium	Mencatat hasil pemeriksaan Laboratorium	Mengisi data hasil pemeriksaan Laboratorium
		Mengubah data hasil pemeriksaan Laboratorium
Pemeriksaan Radiologi	Mencatat hasil pemeriksaan Radiologi	Mengisi data hasil pemeriksaan Radiologi
		Mengubah data hasil pemeriksaan Radiologi



Selanjutnya Penghubungan *use case* dengan setiap fitur-fitur yang teridentifikasi. Tujuannya adalah memastikan hasil dari pemodelan *use case* yang telah dilakukan sama dengan atau sesuai hasil analisis persyaratan. Hubungan dari *use case* dengan fitur ditunjukkan Tabel 4.19.

Tabel 4.19 Hubungan Use Case dengan Fitur

Fitur	Usecase
FR-01	Login
FR-02	Melihat data pendaftaran pasien
	Melihat data hasil pemeriksaan fisik pasien
	Melihat data hasil pemeriksaan laboratorium pasien
	Melihat data hasil pemeriksaan radiologi pasien
	Melihat data pemeriksaan pasien
FR-03	Melihat progres pengerjaan pemeriksaan pasien
FR-04	Tambah data pendaftaran baru
	Ubah data diri pasien
	Tambah data pemeriksaan baru
	Ubah data pemeriksaan pasien
	Menghapus data pemeriksaan pasien
	Mengubah hasil akhir pemeriksaan
FR-05	Mengisi data hasil pemeriksaan fisik
	Mengubah data hasil pemeriksaan fisik
FR-06	Mengisi data hasil pemeriksaan Laboratorium
	Mengubah data hasil pemeriksaan Laboratorium
FR-07	Mengisi data hasil pemeriksaan Radiologi
	Mengubah data hasil pemeriksaan Radiologi
FR-08	Tambah data pengguna
	Mengubah data Pengguna
	Menghapus data pengguna
FR-09	Melihat pelaporan data pemeriksaan

4.3.2 Identifikasi Aktor

Identifikasi aktor merupakan penjelasan singkat mengenai tanggung jawab aktor yang telah teridentifikasi. Pada tahap ini juga disajikan penjelasan singkat mengenai tujuan aktor dalam sistem. Pada Tabel 4.20 menunjukkan identifikasi aktor *front office*, poliklinik, laboratorium, radiologi, dan dokter.

Tabel 4.20 Identifikasi Aktor

Nama Aktor	Deskripsi
Pengguna	Aktor pengguna adalah seluruh staff Pelayanan Pemeriksaan Kesehatan Tenaga Kerja pada Klinik Argaraya Medika yang menggunakan sistem secara langsung. Pengguna dapat menggunakan sistem sesuai hak akses yang diberikan oleh sistem.
Petugas <i>Front Office</i>	Aktor <i>Front Office</i> adalah petugas <i>Front Office</i> yang bertugas sebagai penerima pasien. Dalam sistem <i>Front Office</i> menggunakan untuk mengelola data pendaftaran pasien dimana terdapat data diri dan data permintaan pemeriksaan pasien.
Petugas Poliklinik	Aktor petugas poliklinik menggunakan sistem untuk melihat permintaan pemeriksaan fisik dan kejiwaan dan mengelola data pemeriksaan fisik dan kejiwaan.
Petugas Laboratorium	Aktor petugas laboratorium menggunakan sistem untuk melihat permintaan pemeriksaan laboratorium dan mengelola data pemeriksaan laboratorium.
Petugas Radiologi	Aktor petugas radiologi menggunakan sistem untuk melihat permintaan pemeriksaan radiologi dan mengelola data pemeriksaan radiologi.
Petugas Administrasi	Aktor petugas administrasi menggunakan sistem untuk melakukan perekapan data hasil seluruh pemeriksaan kesehatan. Dalam sistem administrasi menggunakan untuk melakukan pengolahan data hasil seluruh pemeriksaan, pengelolaan akun pengguna, dan mengubah hasil akhir pemeriksaan.

4.3.3 Spesifikasi *Use Case*

Spesifikasi *use case* berisi penjelasan singkat mengenai setiap *use case* yang teridentifikasi. Spesifikasi juga memuat informasi mengenai aktor yang menjalankan *use case*, kondisi sebelum dan sesudah *use case* dijalankan, serta tahap-tahap yang dilakukan ketika *use case* dilakukan. Berikut adalah spesifikasi *use case* yang terdapat pada *use case diagram*.

4.3.3.1 Spesifikasi *Use Case Login*

Spesifikasi *Use Case Login* menggambarkan bagaimana aktor masuk kedalam sistem sehingga dapat mengakses informasi sesuai hak akses yang dimiliki. Kondisi akhir dari Spesifikasi *Use Case Login* adalah Aktor masuk kedalam sistem dan di-*redirect* menuju halaman awal. Tabel 4.21 menunjukkan Spesifikasi *Use Case Login*.

Tabel 4.21 Spesifikasi Use Case Login

Login	
Brief Description	Use case ini menggambarkan bagaimana aktor masuk kedalam sistem sehingga dapat mengakses informasi sesuai hak akses yang dimiliki.
Actor	Pengguna, Petugas Poliklinik, Petugas <i>Front Office</i> , Petugas Laboratorium, Petugas Radiologi, Petugas Administrasi.
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah terdaftar dan memiliki akun dalam sistem. • Aktor mengakses halaman <i>login</i>.
Basic flow of Events	<ol style="list-style-type: none"> 1. Aktor mengakses halaman <i>login</i>. 2. Aktor menginputkan <i>username</i> dan <i>password</i>. 3. Aktor menjalankan fungsi login. 4. Sistem akan melakukan otentifikasi dan menampilkan halaman awal.
Alternative Flows	<p>A1. Login gagal Masukan <i>password</i> atau <i>username</i> salah atau data pengguna belum terdaftar didalam sistem, maka sistem akan menotifikasi kesalahan login gagal.</p>
Post-conditions	Aktor masuk kedalam sistem dan di- <i>redirect</i> menuju halaman awal.

4.3.3.2 Spesifikasi Use Case Melihat Data Pendaftaran Pasien

Spesifikasi *Use Case* Melihat Data Pendaftaran Pasien menggambarkan bagaimana aktor melihat data pendaftaran pasien yang terdiri dari data diri dan data pemeriksaan pasien. Kondisi akhir dari Spesifikasi *Use Case* Melihat Data Pendaftaran Pasien adalah Aktor mendapatkan informasi data pendaftaran pasien berupa data diri pasien dan data pemeriksaan yang dilakukan. Tabel 4.22 menunjukkan Spesifikasi *Use Case* Melihat Data Pendaftaran Pasien.

Tabel 4.22 Spesifikasi Use Case Melihat Data Pendaftaran Pasien

Melihat data pendaftaran pasien	
Brief Description	Use case ini menggambarkan bagaimana aktor melihat data pendaftaran pasien yang terdiri dari data diri dan data pemeriksaan pasien.
Actor	Petugas <i>Front Office</i> dan Administrasi.
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data pendaftaran pasien tersedia.
Basic flow	<ol style="list-style-type: none"> 1. Aktor memilih halaman data pasien. 2. Sistem menampilkan halaman data pasien bersama data diri pasien. 3. Aktor memilih detail data.

Melihat data pendaftaran pasien	
	4. Sistem menampilkan halaman pendaftaran pasien bersama data diri pasien dan data pemeriksaan pasien. 5. <i>Usecase</i> selesai.
Alternative Flows	A1. Menangani kondisi data pendaftaran belum tersedia Jika data pendaftaran pasien belum tersedia maka akan menampilkan informasi data tidak tersedia dan <i>usecase</i> selesai.
Post-conditions	Aktor mendapatkan informasi data pendaftaran pasien berupa data diri pasien dan data pemeriksaan yang dilakukan.

4.3.3.3 Spesifikasi Use Case Menambah Data Pendaftaran Baru

Spesifikasi *Use Case* Menambah Data Pendaftaran Baru menggambarkan bagaimana aktor menambah data pendaftaran baru pada pasien TKLN maupun TKDN. Kondisi akhir dari Spesifikasi *Use Case* Menambah Data Pendaftaran Baru adalah Data pendaftaran baru berhasil disimpan pada sistem. Tabel 4.23 menunjukkan Spesifikasi *Use Case* Menambah Data Pendaftaran Baru.

Tabel 4.23 Usecase Scenario Menambah Data Pendaftaran Baru

Menambah Data Pendaftaran Baru	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor menambah data pendaftaran baru pada pasien TKLN maupun TKDN.
Actor	Petugas <i>Front Office</i> dan Administrasi.
Pre-conditions	<ul style="list-style-type: none"> Aktor telah masuk kedalam sistem.
Basic flow of Events	1. Aktor memilih halaman home. 2. Sistem menampilkan halaman home. 3. Aktor memilih menambah pendaftaran baru dengan memilih jenis pemeriksaan. 4. Sistem menampilkan halaman formulir pendaftaran pasien baru. 5. Aktor memasukan data diri dan data pemeriksaan pasien. 6. Aktor memilih menyimpan data pendaftaran pasien. 7. Sistem melakukan proses penyimpanan data. 8. Sistem menampilkan pesan data berhasil disimpan dan mengalihkan pada halaman daftar pasien harian. 9. <i>Usecase</i> selesai.
Alternative Flows	A1. Proses penyimpanan data gagal



Menambah Data Pendaftaran Baru	
	<p>Jika dalam proses penyimpanan data gagal sistem akan menampilkan pesan penambahan data gagal.</p> <p>A2. Masukan data tidak sesuai Jika dalam proses pemasukan data diri dan data pemeriksaan pasien Masukan yang diberikan tidak sesuai dengan yang sudah ditentukan maka akan menampilkan informasi kesalahan inputan.</p>
Post-conditions	Data pendaftaran baru berhasil disimpan pada sistem.

4.3.3.4 Spesifikasi Use case Mengubah Data Diri Pasien

Spesifikasi Use case Mengubah Data Diri Pasien menggambarkan bagaimana aktor mengubah data diri pasien. Kondisi akhir dari Spesifikasi Use case Mengubah Data Diri Pasien adalah Data diri pasien berhasil diubah. Tabel 4.24 menunjukkan Spesifikasi Use case Mengubah Data Diri Pasien.

Tabel 4.24 Usecase Scenario Mengubah Data Diri Pasien

Mengubah Data Diri Pasien	
Brief Description	Use case ini menggambarkan bagaimana aktor mengubah data diri pasien.
Actor	Petugas <i>Front Office</i> dan Administrasi.
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data pendaftaran pasien tersedia.
Basic flow of Events	<ol style="list-style-type: none"> 1. Aktor memilih halaman data pasien. 2. Sistem menampilkan halaman data pasien bersama data diri pasien. 3. Aktor memilih detail data. 4. Sistem menampilkan halaman pendaftaran pasien bersama data diri pasien dan data pemeriksaan pasien. 5. Aktor memilih edit data diri. 6. Sistem menampilkan halaman edit data diri pasien 7. Aktor memasukan data diri pasien yang baru. 8. Aktor memilih simpan data. 9. Sistem memproses penyimpanan data. 10. Sistem menampilkan pesan mengubah data berhasil.
Alternative Flows	<p>A1. Proses penyimpanan data gagal Jika dalam proses penyimpanan data gagal sistem akan menampilkan pesan mengubah data gagal.</p> <p>A2. Masukan data tidak sesuai Jika dalam proses pemasukan data diri pasien masukan yang diberikan tidak sesuai dengan yang sudah ditentukan maka akan menampilkan informasi kesalahan inputan.</p>



Mengubah Data Diri Pasien	
Post-conditions	Data diri pasien berhasil diubah.

4.3.3.5 Spesifikasi Use case Menambah Data Pemeriksaan Baru

Spesifikasi Use case Menambah Data Pemeriksaan Baru menggambarkan bagaimana aktor menambahkan data pemeriksaan TKLN atau TKDN baru pada pasien. Kondisi akhir dari Spesifikasi Use case Tambah Data Pemeriksaan Baru adalah Aktor berhasil menambahkan pemeriksaan baru pada pasien. Tabel 4.25 menunjukkan Spesifikasi Use case Menambah Data Pemeriksaan Baru.

Tabel 4.25 Usecase Scenario Menambah Data Pemeriksaan Baru

Menambah Data Pemeriksaan Baru	
Brief Description	Use case ini menggambarkan bagaimana aktor menambahkan data pemeriksaan TKLN atau TKDN baru pada pasien.
Actor	Petugas <i>Front Office</i> dan Administrasi.
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data pendaftaran pasien tersedia.
Basic flow of Events	<ol style="list-style-type: none"> 1. Aktor memilih halaman data pasien. 2. Sistem menampilkan halaman data pasien bersama data diri pasien. 3. Aktor memilih detail data. 4. Sistem menampilkan halaman pendaftaran pasien bersama data diri pasien dan data pemeriksaan pasien. 5. Aktor memilih tambah data pemeriksaan TKLN atau TKDN baru. 6. Sistem menampilkan halaman formulir tambah pemeriksaan baru 7. Aktor memasukan data pemeriksaan baru pasien. 8. Aktor memilih menyimpan data pemeriksaan baru pasien. 9. Sistem melakukan proses penyimpanan data. 10. Sistem menampilkan pesan data berhasil disimpan dan mengalihkan pada pendaftaran pasien. 11. Usecase selesai.
Alternative Flows	<p>A1. Proses penyimpanan data gagal Jika dalam proses penyimpanan data gagal sistem akan menampilkan pesan penambahan data gagal.</p> <p>A2. Masukan data tidak sesuai Jika dalam proses pemasukan data pemeriksaan pasien masukan yang diberikan tidak sesuai dengan yang sudah ditentukan maka akan menampilkan informasi kesalahan inputan.</p>

Menambah Data Pemeriksaan Baru	
Post-conditions	Aktor berhasil menambahkan pemeriksaan baru pada pasien.

4.3.3.6 Spesifikasi Use case Mengubah Data Pemeriksaan Pasien

Spesifikasi Use case Mengubah Data Pemeriksaan Pasien menggambarkan bagaimana aktor mengubah data pemeriksaan pasien. Kondisi akhir dari Spesifikasi Use case Mengubah Data Pemeriksaan Pasien adalah Aktor berhasil mengubah data pemeriksaan pasien. Tabel 4.26 menunjukkan Spesifikasi Use case Mengubah Data Pemeriksaan Pasien.

Tabel 4.26 Usecase Scenario Mengubah Data Pemeriksaan Pasien

Mengubah Data Pemeriksaan Pasien	
Brief Description	Use case ini menggambarkan bagaimana aktor mengubah data pemeriksaan pasien.
Actor	Petugas Front Office dan Administrasi.
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data pemeriksaan pasien tersedia.
Basic flow of Events	<ol style="list-style-type: none"> 1. Aktor memilih halaman data pasien. 2. Sistem menampilkan halaman data pasien bersama data diri pasien. 3. Aktor memilih detail data. 4. Sistem menampilkan halaman pendaftaran pasien bersama data diri pasien dan data pemeriksaan pasien. 5. Aktor memilih detail data pemeriksaan 6. Sistem menampilkan halaman pemeriksaan pasien. 7. Aktor memilih edit data pemeriksaan. 8. Sistem menampilkan halaman edit data pemeriksaan. 9. Aktor memasukkan data pemeriksaan pasien yang baru. 10. Aktor memilih menyimpan data pemeriksaan pasien. 11. Sistem melakukan proses penyimpanan data. 12. Sistem menampilkan pesan data berhasil disimpan dan mengalihkan pada pendaftaran pasien. 13. Usecase selesai.
Alternative Flows	<p>A1. Proses penyimpanan data gagal Jika dalam proses penyimpanan data gagal sistem akan menampilkan pesan mengubah data gagal.</p> <p>A2. Masukan data tidak sesuai Jika dalam proses pemasukan data pemeriksaan pasien masukan yang diberikan tidak sesuai dengan yang sudah</p>

Mengubah Data Pemeriksaan Pasien	
	ditentukan maka akan menampilkan informasi kesalahan inputan.
Post-conditions	Aktor berhasil mengubah data pemeriksaan pasien.

4.3.3.7 Spesifikasi *Use case* Melihat Data Pemeriksaan Pasien

Spesifikasi *Use case* Melihat Data Pemeriksaan Pasien menggambarkan bagaimana aktor melihat data pemeriksaan TKLN atau TKDN pasien. Kondisi akhir dari Spesifikasi *Use case* Melihat Data Pemeriksaan Pasien adalah Aktor berhasil mendapatkan informasi data pemeriksaan berupa data diri dan data permintaan pemeriksaan. Tabel 4.27 menunjukkan Spesifikasi *Use case* Melihat Data Pemeriksaan Pasien.

Tabel 4.27 Usecase Scenario Melihat Data Pemeriksaan Pasien

Melihat Data Pemeriksaan Pasien	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor melihat data pemeriksaan TKLN atau TKDN pasien.
Actor	Petugas <i>Front Office</i> dan Administrasi.
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data pemeriksaan pasien tersedia.
Basic flow of Events	<ol style="list-style-type: none"> 1. Aktor memilih halaman data pasien. 2. Sistem menampilkan halaman data pasien bersama data diri pasien. 3. Aktor memilih detail data. 4. Sistem menampilkan halaman pendaftaran pasien bersama data diri pasien dan data pemeriksaan pasien. 5. Aktor memilih detail data pemeriksaan 6. Sistem menampilkan halaman pemeriksaan pasien. 7. <i>Usecase</i> selesai.
Alternative Flows	<p>A1. Data pemeriksaan belum tersedia Jika data pemeriksaan pasien belum tersedia maka akan menampilkan informasi data tidak tersedia dan <i>usecase</i> selesai.</p>
Post-conditions	Aktor berhasil mendapatkan informasi data pemeriksaan berupa data diri dan data permintaan pemeriksaan.

4.3.3.8 Spesifikasi *Use case* Mengisi Data Hasil Pemeriksaan Fisik

Spesifikasi *Use case* Mengisi Data Hasil Pemeriksaan Fisik menggambarkan bagaimana aktor mengisi data hasil pemeriksaan fisik sesuai permintaan yang dibuat sebelumnya. Kondisi akhir dari Spesifikasi *Use case* Mengisi Data Hasil Pemeriksaan Fisik adalah Aktor berhasil mengisi data hasil pemeriksaan fisik. Tabel 4.28 menunjukkan Spesifikasi *Use case* Mengisi Data Hasil Pemeriksaan Fisik.

Tabel 4.28 *Usecase Scenario* Mengisi Data Hasil Pemeriksaan Fisik

Mengisi Data Hasil Pemeriksaan Fisik	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor mengisi data hasil pemeriksaan fisik sesuai permintaan yang dibuat sebelumnya.
Actor	Petugas Poliklinik
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data pemeriksaan pasien tersedia.
Basic flow of Events	<ol style="list-style-type: none"> 1. Aktor memilih halaman daftar pasien harian. 2. Sistem menampilkan halaman daftar pasien harian 3. Aktor memilih proses pemeriksaan. 4. Sistem menampilkan halaman hasil pemeriksaan fisik. 5. Aktor memasukan data hasil pemeriksaan fisik. 6. Aktor memilih menyimpan data. 7. Sistem memproses penyimpanan data. 8. Sistem menampilkan pesan data berhasil disimpan dan mengembalikan pada halaman daftar pasien harian 9. <i>Usecase</i> selesai.
Alternative Flows	<p>A1. Proses penyimpanan data gagal Jika dalam proses penyimpanan data gagal sistem akan menampilkan pesan penambahan data gagal.</p> <p>A2. Inputan data tidak sesuai Jika dalam proses pemasukan data pemeriksaan pasien inputan yang diberikan tidak sesuai dengan yang sudah ditentukan maka akan menampilkan informasi kesalahan inputan.</p>
Post-conditions	Aktor berhasil mengisi data hasil pemeriksaan fisik.

4.3.3.9 Spesifikasi *Use case* Melihat Data Hasil Pemeriksaan Fisik

Spesifikasi *Use case* Melihat Data Hasil Pemeriksaan Fisik menggambarkan bagaimana aktor melihat data hasil pemeriksaan fisik yang telah terisi. Kondisi akhir dari Spesifikasi *Use case* Melihat Data Hasil Pemeriksaan Fisik adalah Aktor berhasil mendapatkan informasi data hasil pemeriksaan fisik pasien. Tabel 4.29 menunjukkan Spesifikasi *Use case* Melihat Data Hasil Pemeriksaan Fisik.

Tabel 4.29 *Usecase Scenario* Melihat Data Hasil Pemeriksaan Fisik

Melihat Data Hasil Pemeriksaan Fisik	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor melihat data hasil pemeriksaan fisik yang telah terisi.
Actor	Petugas Poliklinik dan Administrasi
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem.

Melihat Data Hasil Pemeriksaan Fisik	
	<ul style="list-style-type: none"> Data hasil pemeriksaan fisik pasien telah terisi.
Basic flow of Events	<ol style="list-style-type: none"> Aktor memilih halaman daftar pasien harian. Sistem menampilkan halaman daftar pasien harian Aktor memilih detail data. Sistem menampilkan halaman hasil pemeriksaan fisik. <i>Usecase</i> selesai.
Alternative Flows	<p>A1. Data hasil pemeriksaan fisik belum terisi Jika data hasil pemeriksaan fisik belum terisi maka data tidak dapat diakses dan <i>Usecase</i> selesai.</p>
Post-conditions	Aktor berhasil mendapatkan informasi data hasil pemeriksaan fisik pasien.

4.3.3.10 Spesifikasi *Use case* Mengubah Data Hasil Pemeriksaan Fisik

Spesifikasi *Use case* Mengubah Data Hasil Pemeriksaan Fisik menggambarkan bagaimana aktor mengubah data hasil pemeriksaan fisik yang telah terisi. Kondisi akhir dari Spesifikasi *Use case* Mengubah Data Hasil Pemeriksaan Fisik adalah Aktor berhasil mengubah data hasil pemeriksaan fisik pasien. Tabel 4.30 menunjukkan Spesifikasi *Use case* Mengubah Data Hasil Pemeriksaan Fisik.

Tabel 4.30 *Usecase Scenario* Mengubah Data Hasil Pemeriksaan Fisik

Mengubah Data Hasil Pemeriksaan Fisik	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor mengubah data hasil pemeriksaan fisik yang telah terisi.
Actor	Petugas Poliklinik dan Administrasi
Pre-conditions	<ul style="list-style-type: none"> Aktor telah masuk kedalam sistem. Data hasil pemeriksaan fisik pasien telah terisi.
Basic flow of Events	<ol style="list-style-type: none"> Aktor memilih halaman daftar pasien harian. Sistem menampilkan halaman daftar pasien harian Aktor memilih detail data. Sistem menampilkan halaman hasil pemeriksaan fisik. Aktor memilih edit. Sistem menampilkan halaman edit hasil pemeriksaan fisik. Aktor memasukan data hasil pemeriksaan fisik yang baru. Aktor memilih menyimpan data. Sistem memproses penyimpanan data. Sistem menampilkan pesan data berhasil disimpan dan mengembalikan pada halaman daftar pasien harian <i>Usecase</i> selesai.



Mengubah Data Hasil Pemeriksaan Fisik	
Alternative Flows	<p>A1. Proses penyimpanan data gagal Jika dalam proses penyimpanan data gagal sistem akan menampilkan pesan mengubah data gagal.</p> <p>A2. Inputan data tidak sesuai Jika dalam proses pemasukan data pemeriksaan pasien inputan yang diberikan tidak sesuai dengan yang sudah ditentukan maka akan menampilkan informasi kesalahan inputan.</p>
Post-conditions	Aktor berhasil mengubah data hasil pemeriksaan fisik pasien.

4.3.3.11 Spesifikasi Use case Mengisi Data Hasil Pemeriksaan Laboratorium

Spesifikasi Use case Mengisi Data Hasil Pemeriksaan Laboratorium menggambarkan bagaimana aktor mengisi data hasil pemeriksaan Laboratorium sesuai permintaan yang dibuat sebelumnya. Kondisi akhir dari Spesifikasi Use case Mengisi Data Hasil Pemeriksaan Laboratorium adalah Aktor berhasil mengisi data hasil pemeriksaan Laboratorium. Tabel 4.31 menunjukkan Spesifikasi Use case Mengisi Data Hasil Pemeriksaan Laboratorium.

Tabel 4.31 Usecase Scenario Mengisi Data Hasil Pemeriksaan Laboratorium

Mengisi Data Hasil Pemeriksaan Laboratorium	
Brief Description	Use case ini menggambarkan bagaimana aktor mengisi data hasil pemeriksaan Laboratorium sesuai permintaan yang dibuat sebelumnya.
Actor	Petugas Laboratorium dan Administrasi
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data pemeriksaan pasien tersedia.
Basic flow of Events	<ol style="list-style-type: none"> 1. Aktor memilih halaman daftar pasien harian. 2. Sistem menampilkan halaman daftar pasien harian 3. Aktor memilih proses pemeriksaan. 4. Sistem menampilkan halaman hasil pemeriksaan Laboratorium. 5. Aktor memasukan data hasil pemeriksaan Laboratorium. 6. Aktor memilih menyimpan data. 7. Sistem memproses penyimpanan data. 8. Sistem menampilkan pesan data berhasil disimpan dan mengembalikan pada halaman daftar pasien harian 9. Usecase selesai.
Alternative Flows	<p>A1. Proses penyimpanan data gagal Jika dalam proses penyimpanan data gagal sistem akan menampilkan pesan penambahan data gagal.</p>



Mengisi Data Hasil Pemeriksaan Laboratorium	
	<p>A2. Inputan data tidak sesuai Jika dalam proses pemasukan data pemeriksaan pasien inputan yang diberikan tidak sesuai dengan yang sudah ditentukan maka akan menampilkan informasi kesalahan inputan.</p>
Post-conditions	Aktor berhasil mengisi data hasil pemeriksaan Laboratorium.

4.3.3.12 Spesifikasi Use case Melihat Data Hasil Pemeriksaan Laboratorium

Spesifikasi Use case Melihat Data Hasil Pemeriksaan Laboratorium menggambarkan bagaimana aktor melihat data hasil pemeriksaan Laboratorium yang telah terisi. Kondisi akhir dari Spesifikasi Use case Melihat Data Hasil Pemeriksaan Laboratorium adalah Aktor berhasil mendapatkan informasi data hasil pemeriksaan Laboratorium pasien. Tabel 4.32 menunjukkan Spesifikasi Use case Melihat Data Hasil Pemeriksaan Laboratorium.

Tabel 4.32 Usecase Scenario Melihat Data Hasil Pemeriksaan Laboratorium

Melihat Data Hasil Pemeriksaan Laboratorium	
Brief Description	Use case ini menggambarkan bagaimana aktor melihat data hasil pemeriksaan Laboratorium yang telah terisi.
Actor	Petugas Laboratorium dan Administrasi
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data hasil pemeriksaan Laboratorium pasien telah terisi.
Basic flow of Events	<ol style="list-style-type: none"> 1. Aktor memilih halaman daftar pasien harian. 2. Sistem menampilkan halaman daftar pasien harian 3. Aktor memilih detail data. 4. Sistem menampilkan halaman hasil pemeriksaan Laboratorium. 5. Usecase selesai.
Alternative Flows	<p>A1. Data hasil pemeriksaan Laboratorium belum terisi Jika data hasil pemeriksaan Laboratorium belum terisi maka data tidak dapat diakses dan Usecase selesai.</p>
Post-conditions	Aktor berhasil mendapatkan informasi data hasil pemeriksaan Laboratorium pasien.

4.3.3.13 Spesifikasi Use case Mengubah Data Hasil Pemeriksaan Laboratorium

Spesifikasi Use case Mengubah Data Hasil Pemeriksaan Laboratorium menggambarkan bagaimana aktor mengubah data hasil pemeriksaan Laboratorium yang telah terisi. Kondisi akhir dari Spesifikasi Use case Mengubah Data Hasil Pemeriksaan Laboratorium adalah Aktor berhasil mengubah data hasil pemeriksaan Laboratorium pasien. Tabel 4.33 menunjukkan Spesifikasi Use case Mengubah Data Hasil Pemeriksaan Laboratorium.

Tabel 4.33 *Usecase Scenario* Mengubah Data Hasil Pemeriksaan Laboratorium

Mengubah Data Hasil Pemeriksaan Laboratorium	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor mengubah data hasil pemeriksaan Laboratorium yang telah terisi.
Actor	Petugas Laboratorium dan Administrasi
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data hasil pemeriksaan Laboratorium pasien telah terisi.
Basic flow of Events	<ol style="list-style-type: none"> 1. Aktor memilih halaman daftar pasien harian. 2. Sistem menampilkan halaman daftar pasien harian 3. Aktor memilih detail data. 4. Sistem menampilkan halaman hasil pemeriksaan Laboratorium. 5. Aktor memilih edit. 6. Sistem menampilkan halaman edit hasil pemeriksaan Laboratorium. 7. Aktor memasukan data hasil pemeriksaan Laboratorium yang baru. 8. Aktor memilih menyimpan data. 9. Sistem memproses penyimpanan data. 10. Sistem menampilkan pesan data berhasil disimpan dan mengembalikan pada halaman daftar pasien harian 11. <i>Usecase</i> selesai.
Alternative Flows	<p>A1. Proses penyimpanan data gagal Jika dalam proses penyimpanan data gagal sistem akan menampilkan pesan mengubah data gagal.</p> <p>A2. Inputan data tidak sesuai Jika dalam proses pemasukan data pemeriksaan pasien inputan yang diberikan tidak sesuai dengan yang sudah ditentukan maka akan menampilkan informasi kesalahan inputan.</p>
Post-conditions	Aktor berhasil mengubah data hasil pemeriksaan Laboratorium pasien.

4.3.3.14 Spesifikasi *Use case* Mengisi Data Hasil Pemeriksaan Radiologi

Spesifikasi *Use case* Mengisi Data Hasil Pemeriksaan Radiologi menggambarkan bagaimana aktor mengisi data hasil pemeriksaan Radiologi sesuai permintaan yang dibuat sebelumnya. Kondisi akhir dari Spesifikasi *Use case* Mengisi Data Hasil Pemeriksaan Radiologi adalah Aktor berhasil mengisi data hasil pemeriksaan Radiologi. Tabel 4.34 menunjukkan Spesifikasi *Use case* Mengisi Data Hasil Pemeriksaan Radiologi.

Tabel 4.34 Usecase Scenario Mengisi Data Hasil Pemeriksaan Radiologi

Mengisi Data Hasil Pemeriksaan Radiologi	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor mengisi data hasil pemeriksaan Radiologi sesuai permintaan yang dibuat sebelumnya.
Actor	Petugas Radiologi dan Administrasi
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data pemeriksaan pasien tersedia.
Basic flow of Events	<ol style="list-style-type: none"> 1. Aktor memilih halaman daftar pasien harian. 2. Sistem menampilkan halaman daftar pasien harian 3. Aktor memilih proses pemeriksaan. 4. Sistem menampilkan halaman hasil pemeriksaan Radiologi. 5. Aktor memasukan data hasil pemeriksaan Radiologi. 6. Aktor memilih menyimpan data. 7. Sistem memproses penyimpanan data. 8. Sistem menampilkan pesan data berhasil disimpan dan mengembalikan pada halaman daftar pasien harian 9. <i>Usecase</i> selesai.
Alternative Flows	<p>A1. Proses penyimpanan data gagal Jika dalam proses penyimpanan data gagal sistem akan menampilkan pesan penambahan data gagal.</p> <p>A2. Inputan data tidak sesuai Jika dalam proses pemasukan data pemeriksaan pasien inputan yang diberikan tidak sesuai dengan yang sudah ditentukan maka akan menampilkan informasi kesalahan inputan.</p>
Post-conditions	Aktor berhasil mengisi data hasil pemeriksaan Radiologi.

4.3.3.15 Spesifikasi Use case Melihat Data Hasil Pemeriksaan Radiologi

Spesifikasi *Use case* Melihat Data Hasil Pemeriksaan Radiologi menggambarkan bagaimana aktor melihat data hasil pemeriksaan Radiologi yang telah terisi. Kondisi akhir dari Spesifikasi *Use case* Melihat Data Hasil Pemeriksaan Radiologi adalah Aktor berhasil mendapatkan informasi data hasil pemeriksaan Radiologi pasien. Tabel 4.35 menunjukkan Spesifikasi *Use case* Melihat Data Hasil Pemeriksaan Radiologi.

Tabel 4.35 Usecase Scenario Melihat Data Hasil Pemeriksaan Radiologi

Melihat Data Hasil Pemeriksaan Radiologi	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor melihat data hasil pemeriksaan Radiologi yang telah terisi.
Actor	Petugas Radiologi dan Administrasi
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem.

Melihat Data Hasil Pemeriksaan Radiologi	
	<ul style="list-style-type: none"> Data hasil pemeriksaan Radiologi pasien telah terisi.
Basic flow of Events	<ol style="list-style-type: none"> Aktor memilih halaman daftar pasien harian. Sistem menampilkan halaman daftar pasien harian Aktor memilih detail data. Sistem menampilkan halaman hasil pemeriksaan Radiologi. <i>Usecase</i> selesai.
Alternative Flows	<p>A1. Data hasil pemeriksaan Radiologi belum terisi Jika data hasil pemeriksaan Radiologi belum terisi maka data tidak dapat diakses dan <i>Usecase</i> selesai.</p>
Post-conditions	Aktor berhasil mendapatkan informasi data hasil pemeriksaan Radiologi pasien.

4.3.3.16 Spesifikasi *Use case* Mengubah Data Hasil Pemeriksaan Radiologi

Spesifikasi *Use case* Mengubah Data Hasil Pemeriksaan Radiologi menggambarkan bagaimana aktor mengubah data hasil pemeriksaan Radiologi yang telah terisi. Kondisi akhir dari Spesifikasi *Use case* Mengubah Data Hasil Pemeriksaan Radiologi adalah Aktor berhasil mengubah data hasil pemeriksaan Radiologi pasien. Tabel 4.36 menunjukkan Spesifikasi *Use case* Mengubah Data Hasil Pemeriksaan Radiologi.

Tabel 4.36 *Usecase Scenario* Mengubah Data Hasil Pemeriksaan Radiologi

Mengubah Data Hasil Pemeriksaan Radiologi	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor mengubah data hasil pemeriksaan Radiologi yang telah terisi.
Actor	Petugas Radiologi dan Administrasi
Pre-conditions	<ul style="list-style-type: none"> Aktor telah masuk kedalam sistem. Data hasil pemeriksaan Radiologi pasien telah terisi.
Basic flow of Events	<ol style="list-style-type: none"> Aktor memilih halaman daftar pasien harian. Sistem menampilkan halaman daftar pasien harian Aktor memilih detail data. Sistem menampilkan halaman hasil pemeriksaan Radiologi. Aktor memilih edit. Sistem menampilkan halaman edit hasil pemeriksaan Radiologi. Aktor memasukan data hasil pemeriksaan Radiologi yang baru. Aktor memilih menyimpan data. Sistem memproses penyimpanan data.



Mengubah Data Hasil Pemeriksaan Radiologi	
	10. Sistem menampilkan pesan data berhasil disimpan dan mengembalikan pada halaman daftar pasien harian 11. <i>Usecase</i> selesai.
Alternative Flows	A1. Proses penyimpanan data gagal Jika dalam proses penyimpanan data gagal sistem akan menampilkan pesan mengubah data gagal. A2. Inputan data tidak sesuai Jika dalam proses pemasukan data pemeriksaan pasien inputan yang diberikan tidak sesuai dengan yang sudah ditentukan maka akan menampilkan informasi kesalahan inputan.
Post-conditions	Aktor berhasil mengubah data hasil pemeriksaan Radiologi pasien.

4.3.3.17 Spesifikasi *Use case* Menambah Data Pengguna

Spesifikasi *Use case* Menambah Data Pengguna menggambarkan bagaimana aktor menambah data pengguna baru. Kondisi akhir dari Spesifikasi *Use case* Menambah Data Pengguna adalah Data pengguna baru berhasil tersimpan dalam sistem. Tabel 4.37 menunjukkan Spesifikasi *Use case* Menambah Data Pengguna.

Tabel 4.37 *Usecase Scenario* Menambah Data Pengguna

Menambah Data Pengguna	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor menambah data pengguna baru.
Actor	Petugas Administrasi
Pre-conditions	<ul style="list-style-type: none"> Aktor telah masuk kedalam sistem.
Basic flow of Events	1. Aktor memilih halaman data akun pengguna. 2. Sistem menampilkan halaman data akun pengguna. 3. Aktor memilih Menambah pengguna baru 4. Sistem menampilkan halaman Menambah pengguna baru 5. Aktor memasukan data pengguna baru. 6. Aktor memilih simpan data. 7. Sistem memproses penyimpanan data. 8. Sistem menampilkan pesan berhasil dan mengalihkan pada halaman data akun pengguna. 9. <i>Usecase</i> selesai.
Alternative Flows	A1. Proses penyimpanan data gagal Jika dalam proses penyimpanan data gagal sistem akan menampilkan pesan penambahan data gagal. A2. Inputan data tidak sesuai

Menambah Data Pengguna	
	Jika dalam proses pemasukan data pemeriksaan pasien inputan yang diberikan tidak sesuai dengan yang sudah ditentukan maka akan menampilkan informasi kesalahan inputan.
Post-conditions	Data pengguna baru berhasil tersimpan dalam sistem.

4.3.3.18 Spesifikasi *Use case* Mengubah Data Pengguna

Spesifikasi *Use case* Mengubah Data Pengguna menggambarkan bagaimana aktor Mengubah data pengguna. Kondisi akhir dari Spesifikasi *Use case* Mengubah Data Pengguna adalah Aktor berhasil memperbarui data pengguna. Tabel 4.38 menunjukkan Spesifikasi *Use case* Mengubah Data Pengguna.

Tabel 4.38 Usecase Scenario Mengubah Data Pengguna

Mengubah Data Pengguna	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor Mengubah data pengguna.
Actor	Petugas Administrasi
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data pengguna tersedia.
Basic flow of Events	<ol style="list-style-type: none"> 1. Aktor memilih halaman data akun pengguna. 2. Sistem menampilkan halaman data akun pengguna. 3. Aktor memilih edit. 4. Sistem menampilkan halaman edit data pengguna. 5. Aktor memasukan data pengguna yang baru. 6. Aktor memilih simpan data. 7. Sistem memproses penyimpanan data. 8. Sistem menampilkan pesan berhasil dan mengalihkan pada halaman data akun pengguna. 9. <i>Usecase</i> selesai.
Alternative Flows	<p>A1. Proses penyimpanan data gagal Jika dalam proses penyimpanan data gagal sistem akan menampilkan pesan mengubah data gagal.</p> <p>A2. Inputan data tidak sesuai Jika dalam proses pemasukan data pemeriksaan pasien inputan yang diberikan tidak sesuai dengan yang sudah ditentukan maka akan menampilkan informasi kesalahan inputan.</p>
Post-conditions	Aktor berhasil memperbarui data pengguna.

4.3.3.19 Spesifikasi *Use case* Menghapus Data Pengguna

Spesifikasi *Use case* Menghapus Data Pengguna menggambarkan bagaimana aktor Menghapus data pengguna. Kondisi akhir dari Spesifikasi *Use case*

Menghapus Data Pengguna adalah Aktor berhasil menghapus data pengguna. Tabel 4.39 menunjukkan Spesifikasi *Use case* Menghapus Data Pengguna.

Tabel 4.39 Usecase Scenario Menghapus Data Pengguna

Menghapus Data Pengguna	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor Menghapus data pengguna.
Actor	Petugas Administrasi
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data pengguna tersedia.
Basic flow of Events	<ol style="list-style-type: none"> 1. Aktor memilih halaman data akun pengguna. 2. Sistem menampilkan halaman data akun pengguna. 3. Aktor memilih hapus. 4. Sistem menampilkan peringatan penghapusan. 5. Aktor memilih konfirmasi penghapusan. 6. Sistem memproses penghapusan data. 7. Sistem menampilkan pesan berhasil dan mengalihkan pada halaman data akun pengguna. 8. <i>Usecase</i> selesai.
Alternative Flows	<p>A1. Proses penghapusan data gagal Jika dalam proses penghapusan data gagal sistem akan menampilkan pesan penambahan data gagal.</p>
Post-conditions	Aktor berhasil menghapus data pengguna.

4.3.3.20 Spesifikasi *Use case* Menghapus Data Pemeriksaan

Spesifikasi *Use case* Menghapus Data Pemeriksaan menggambarkan bagaimana aktor Menghapus data Pemeriksaan. Kondisi akhir dari Spesifikasi *Use case* Menghapus Data Pemeriksaan adalah Aktor berhasil menghapus data Pemeriksaan. Tabel 4.40 menunjukkan Spesifikasi *Use case* Menghapus Data Pemeriksaan.

Tabel 4.40 Usecase Scenario Menghapus Data Pemeriksaan

Menghapus Data Pemeriksaan	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor Menghapus data Pemeriksaan.
Actor	Petugas Administrasi
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data Pemeriksaan tersedia.
Basic flow of Events	<ol style="list-style-type: none"> 1. Aktor memilih halaman data pasien. 2. Sistem menampilkan halaman data pasien bersama data diri pasien. 3. Aktor memilih detail data.

Menghapus Data Pemeriksaan	
	<ol style="list-style-type: none"> 4. Sistem menampilkan halaman pendaftaran pasien bersama data diri pasien dan data pemeriksaan pasien. 5. Aktor memilih hapus data pemeriksaan. 6. Sistem menampilkan peringatan penghapusan. 7. Aktor memilih konfirmasi penghapusan. 8. Sistem memproses penghapusan data. 9. Sistem menampilkan pesan berhasil dan mengalihkan pada halaman data pendaftaran pasien. 10. <i>Usecase</i> selesai.
Alternative Flows	<p>A1. Proses penghapusan data gagal Jika dalam proses penghapusan data gagal sistem akan menampilkan pesan penambahan data gagal.</p>
Post-conditions	Aktor berhasil menghapus data pemeriksaan.

4.3.3.21 Spesifikasi *Use case* Mengubah Hasil Akhir Pemeriksaan

Spesifikasi *Use case* Mengubah Hasil Akhir Pemeriksaan menggambarkan bagaimana aktor mengubah hasil akhir pemeriksaan menjadi FIT/UNFIT dalam melakukan pekerjaan. Kondisi akhir dari Spesifikasi *Use case* Mengubah Hasil Akhir Pemeriksaan adalah Aktor berhasil mengubah data hasil akhir pemeriksaan. Tabel 4.41 menunjukkan Spesifikasi *Use case* Mengubah Hasil Akhir Pemeriksaan.

Tabel 4.41 Usecase Scenario Mengubah Hasil Akhir Pemeriksaan

Mengubah Hasil Akhir Pemeriksaan	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor mengubah hasil akhir pemeriksaan menjadi FIT/UNFIT dalam melakukan pekerjaan.
Actor	Petugas Administrasi
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data Pemeriksaan tersedia. • Seluruh data pemeriksaan telah terisi.
Basic flow of Events	<ol style="list-style-type: none"> 1. Aktor memilih halaman data pasien. 2. Sistem menampilkan halaman data pasien bersama data diri pasien. 3. Aktor memilih detail data. 4. Sistem menampilkan halaman pendaftaran pasien bersama data diri pasien dan data pemeriksaan pasien. 5. Aktor memilih detail data pemeriksaan. 6. Sistem menampilkan halaman pemeriksaan pasien. 7. Aktor memilih mengubah data akhir pemeriksaan menjadi FIT/UNFIT.



Mengubah Hasil Akhir Pemeriksaan	
	8. Sistem memproses permintaan dan pengecekan data seluruh pemeriksaan. 9. Sistem menampilkan pesan berhasil dan mengalihkan pada halaman pemeriksaan dengan data akhir pemeriksaan FIT/UNFIT 10. <i>Usecase</i> selesai.
Alternative Flows	A1. Proses permintaan dan pengecekan gagal Jika dalam proses permintaan dan pengecekan gagal sistem akan menampilkan pesan mengubah data gagal. A2. Data pemeriksaan belum lengkap terisi Jika dalam proses permintaan dan pengecekan gagal sistem data pemeriksaan belum lengkap terisi maka akan menampilkan pesan data pemeriksaan belum lengkap.
Post-conditions	Aktor berhasil mengubah data hasil akhir pemeriksaan.

4.3.3.2 Spesifikasi *Use case* Melihat Pelaporan Data Pemeriksaan

Spesifikasi *Use case* Melihat Pelaporan Data Pemeriksaan menggambarkan bagaimana aktor melihat pelaporan data pemeriksaan dalam bentuk *chart*. Kondisi akhir dari Spesifikasi *Use case* Melihat Pelaporan Data Pemeriksaan adalah Aktor mendapatkan informasi data pemeriksaan kesehatan tenaga kerja. Tabel 4.42 menunjukkan Spesifikasi *Use case* Melihat Pelaporan Data Pemeriksaan.

Tabel 4.42 *Usecase Scenario* Melihat Pelaporan Data Pemeriksaan

Melihat Pelaporan Data Pemeriksaan	
Brief Description	<i>Use case</i> ini menggambarkan bagaimana aktor melihat pelaporan data pemeriksaan dalam bentuk <i>chart</i>
Actor	Petugas Administrasi
Pre-conditions	<ul style="list-style-type: none"> • Aktor telah masuk kedalam sistem. • Data Pemeriksaan tersedia.
Basic flow of Events	1. Aktor memilih halaman home. 2. Sistem menampilkan halaman home 3. Sistem menampilkan data pemeriksaan dalam <i>chart</i> . 4. <i>Usecase</i> selesai.
Alternative Flows	A1. Data pemeriksaan tidak tersedia Jika data pemeriksaan tidak tersedia maka akan menampilkan <i>chart</i> tanpa data yang ditampilkan dan <i>use case</i> selesai
Post-conditions	Aktor berhasil mendapatkan informasi data pemeriksaan kesehatan tenaga kerja.



4.4 Kesimpulan Analisis Kebutuhan Fase Insepsi Metode Rational Unifield Process

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

Untuk memastikan kebutuhan dari pemangku kepentingan dan pengguna peneliti melakukan pengulangan analisis kebutuhan kepada pemangku kepentingan. Hasil perbaikan identifikasi kebutuhan pengguna pada kode kebutuhan SILPK-KP-006 dimana pengguna menambahkan untuk sistem agar dapat menampilkan informasi pengolahan data pemeriksaan. Dengan adanya perubahan kebutuhan pengguna dilakukan juga perbaikan fitur dengan melakukan penambahan fitur dengan kode FR-09 Sistem dapat menampilkan informasi pelaporan pengolahan data pemeriksaan kepada pengguna. Perbaikan yang dilakukan pada persyaratan fungsional pada kode SILPK-KF-23 Sistem dapat mengubah data hasil akhir pemeriksaan pasien, SILPK-KF-24 Sistem dapat menampilkan data jumlah pemeriksaan hari ini, dan SILPK-KF-25 Sistem dapat menampilkan data jumlah pemeriksaan 5 hari sebelum hari ini.

Hasil akhir analisis persyaratan didapat 9 fitur, 25 persyaratan fungsional dan 1 persyaratan nonfungsional. Kebutuhan analisis persyaratan kemudian telah diterima atau disetujui oleh pemangku kepentingan. *Lifecycle objective milestone* fase insepsi pada RUP maka telah terpenuhi.

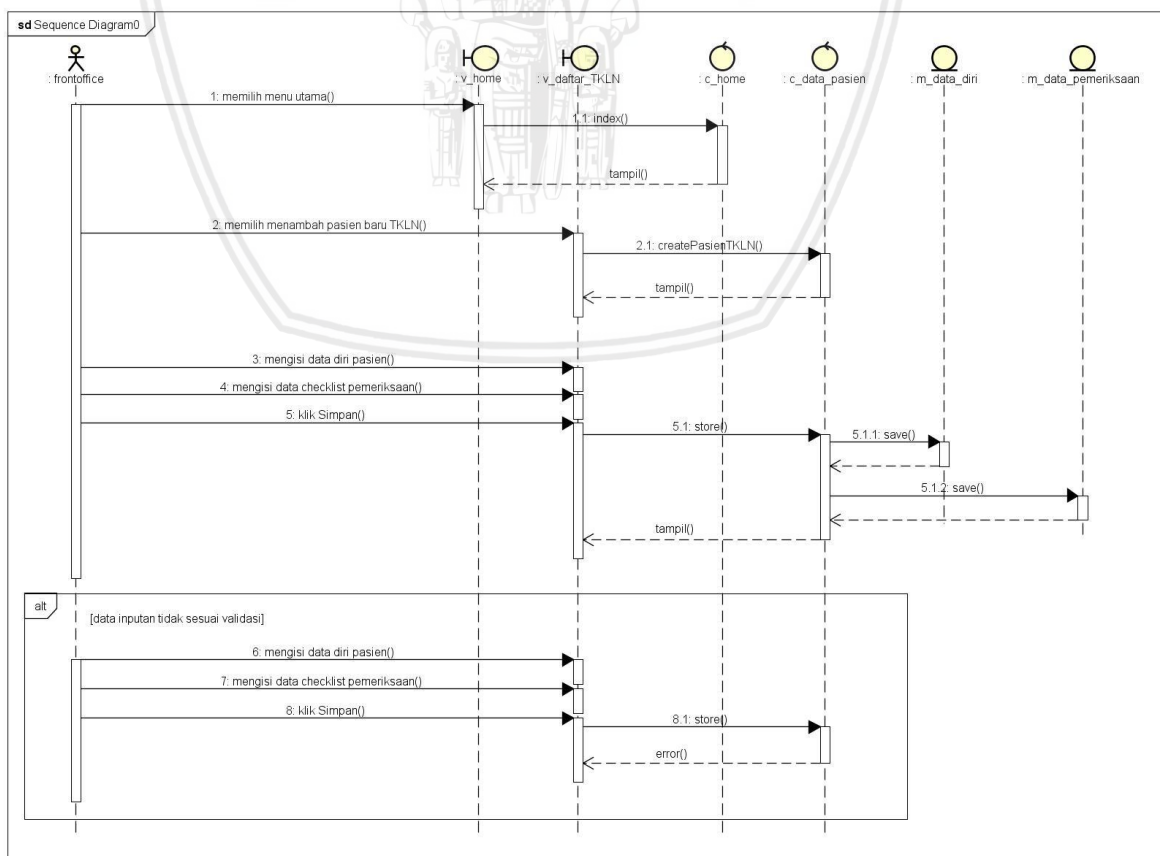
BAB 5 PERANCANGAN

Bab perancangan ini menjelaskan kegiatan dalam metode *Rational Unified Process* (RUP) pada fase elaborasi, dimana dilakukan perancangan sistem yang dibangun sesuai kebutuhan yang telah didefinisikan sebelumnya. Pada fase sebelumnya insepisi evaluasi dilakukan dan terdapat perbaikan pada analisis persyaratan untuk memenuhi *Lifecycle Objective Milestone*. Perancangan sistem yang dilakukan meliputi *sequence diagram*, perancangan basis data, beberapa perancangan antarmuka, dan beberapa perancangan algoritme.

5.1 Pemodelan Interaksi Objek

Sequence diagram merupakan dokumentasi dari pemodelan interaksi objek. Pemodelan interaksi objek dilakukan untuk menggambarkan pertukaran pesan antar entitas sebuah aktor, objek *control*, entitas *boundary*, dan objek *model* yang saling berinteraksi dalam memenuhi kebutuhan pengguna. *Sequence diagram* dibuat berdasarkan pada alur dari beberapa *use case* yang telah didokumentasikan dalam spesifikasi *use case*. Beberapa *Sequence diagram* yang dijelaskan pada bagian ini berupa visualisasi interaksi dalam proses menambah data pendaftaran pasien, menambah data pemeriksaan baru, mengubah data hasil akhir pemeriksaan, dan mengisi data pemeriksaan fisik dan kejiwaan.

5.1.1 *Sequence Diagram* Menambah Data Pendaftaran Baru

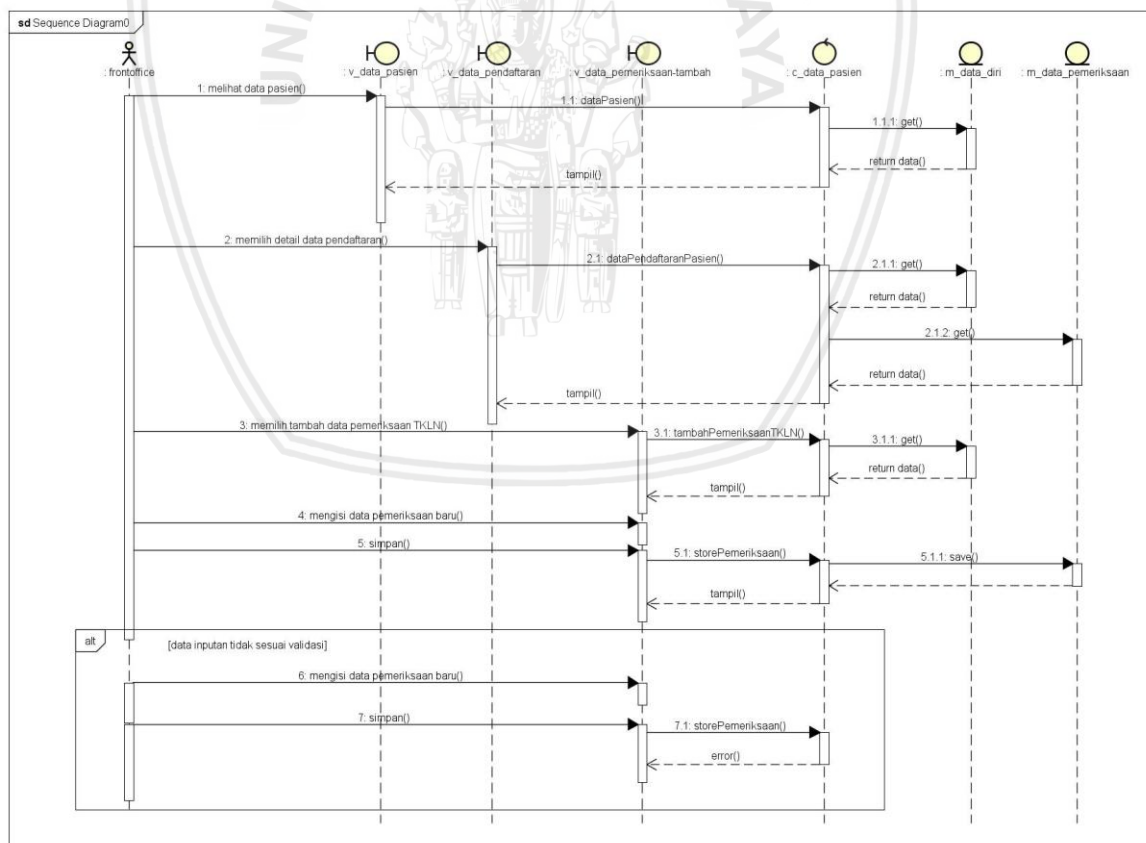


Gambar 5.1 Sequence Diagram Menambah Data Pendaftaran Baru

Sequence diagram Gambar 5.1 merupakan visualisasi interaksi antar objek pada saat aktor yaitu *Front Office* yang menggunakan sistem untuk Menambah Data Pendaftaran Baru. Objek-objek yang terlibat di dalam interaksi di antaranya adalah aktor lalu *v_home* dan *v_daftar_TKLN* sebagai *interface*, kelas *c_data_pasien* dan *c_home* sebagai objek *control*, dan kelas *m_pasien* dan *m_data_pemeriksaan* sebagai objek *model*.

Proses Tambah Data Pendaftaran Baru dimulai saat aktor *frontoffice* mengakses *interface* home dengan memanggil fungsi *index()* pada objek *control* *c_home* dan menampilkan halaman home. Aktor memilih menambah pasien TKLN baru dan memanggil fungsi *createPasienTKLN()* sehingga menampilkan *interface* *v_daftar_pasien* sebagai formulir menambahkan data. Data yang diisi dalam formulir pendaftaran adalah data diri, data pemeriksaan, dan data permintaan pasien. Aktor akan memilih simpan jika data telah lengkap dan memanggil fungsi *store()* pada objek *control* *c_data_pasien*. Objek *control* *c_data_pasien* akan memanggil fungsi *save()* pada objek *model* *m_data_diri* dan *m_data_pemeriksaan* untuk menyimpan inputan data.

5.1.2 Sequence Diagram Menambah Data Pemeriksaan Baru



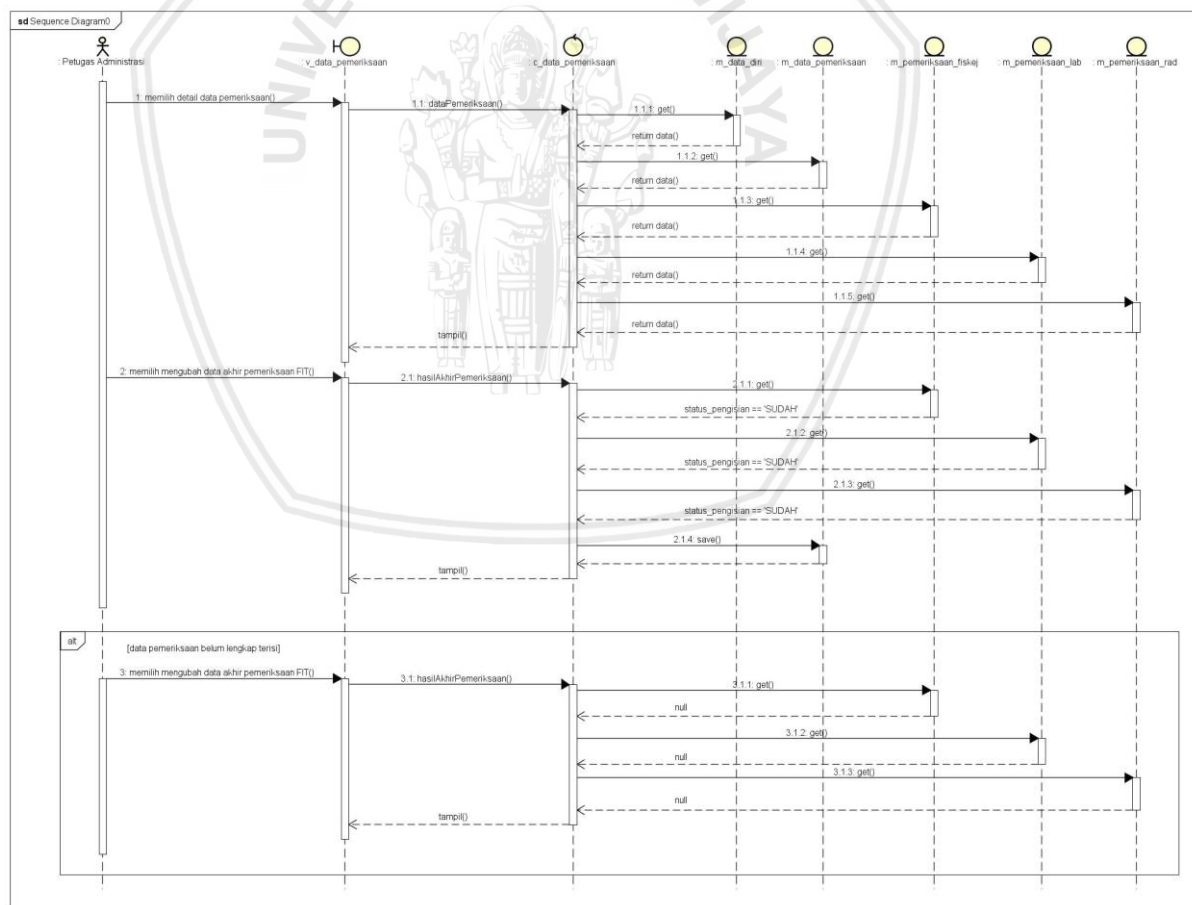
Gambar 5.2 Sequence Diagram Tambah Data Pemeriksaan Baru

Sequence diagram Gambar 5.2 merupakan visualisasi interaksi antar objek pada saat aktor yaitu *Front Office* yang menggunakan sistem untuk Menambah

Data Pemeriksaan Baru. Objek-objek yang terlibat di dalam interaksi di antaranya adalah aktor lalu `v_data_pasien`, `v_data_pendaftaran`, `v_data_pemeriksaan-tambah` sebagai *interface*, kelas `c_data_pasien` sebagai objek *control*, dan kelas `m_data_diri` dan `m_data_pemeriksaan` sebagai objek *model*.

Proses menambah Tambah Data Pemeriksaan Baru dimulai saat aktor *Front Office* mengakses *interface* `v_data_pasien` dengan fungsi `dataPasien()` melalui objek *control* `c_data_pasien` yang mengakses objek *model* `m_data_diri`. Aktor memilih detail data pemeriksaan dengan memanggil fungsi `dataPendaftaranPasien()` melalui objek *control* `c_data_pasien` yang mengakses objek *model* `m_data_diri` dan `m_data_pemeriksaan`. Aktor memilih menambah pemeriksaan dengan memanggil fungsi `tambahPemeriksaanTKLN()` yang mengakses objek *model* `m_data_diri` dan ditampilkan pada *interface* `v_data_pemeriksaan-tambah`. Aktor mengisi data pemeriksaan dan permintaan baru dan memilih simpan untuk memanggil fungsi `storePemeriksaan()` pada objek *control* `c_data_pasien` serta memanggil fungsi `save()` pada objek *model* `m_data_pemeriksaan` untuk menyimpan data inputan.

5.1.3 Sequence Diagram Mengubah Data Hasil Akhir Pemeriksaan



Gambar 5.3 Sequence Diagram Mengubah Data Hasil Akhir Pemeriksaan

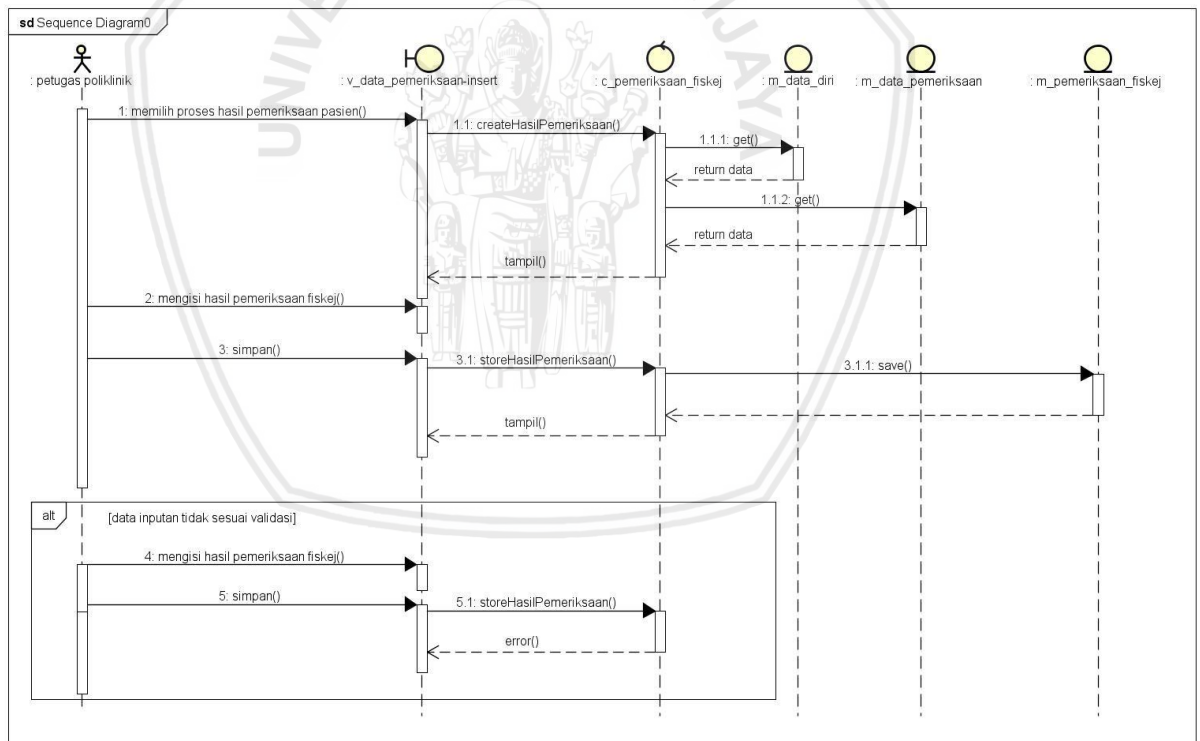
Sequence diagram Gambar 5.3 merupakan visualisasi interaksi antar objek pada saat aktor yaitu Petugas Admin yang menggunakan sistem untuk Mengubah



Data Hasil Akhir Pemeriksaan. Objek-objek yang terlibat di dalam interaksi di antaranya adalah aktor dan *v_data_pemeriksaan interface*, kelas *c_data_pemeriksaan* sebagai objek *control*, dan kelas *m_data_diri*, *m_data_pemeriksaan*, *m_pemeriksaan_fiskej*, *m_pemeriksaan_lab*, dan *m_pemeriksaan_rad* sebagai objek *model*.

Proses ngubah Data Hasil Akhir Pemeriksaan dimulai dengan aktor memilih mengakses halaman pemeriksaan pasien dengan memanggil fungsi *dataPemeriksaan()*. Pada halaman pemeriksaan pasien aktor memilih untuk mengubah data hasil akhir pemeriksaan dengan memanggil fungsi *hasilAkhirPemeriksaan()* melalui objek *control* *c_data_pemeriksaan*. Fungsi *hasilAkhirPemeriksaan()* akan melakukan pengecekan data dengan memanggil objek *model* *m_pemeriksaan_fiskej*, *m_pemeriksaan_lab*, *m_pemeriksaan_rad* dengan kondisi *status_pengisian == 'SUDAH'* maka akan melakukan perubahan data dan menyimpan hasil perubahan data. Pada alternatif saat pengecekan data jika data yang didapat adalah *null* maka akan mengembalikan ke halaman *v_data_pemeriksaan*.

5.1.4 Sequence Diagram Mengisi Data Pemeriksaan Fisik dan Kejiwaan



Gambar 5.4 Sequence Diagram Memperbarui Data Pemeriksaan Fisik dan Kejiwaan

Sequence diagram Gambar 5.4 merupakan visualisasi interaksi antar objek pada saat aktor yaitu Petugas Poliklinik yang menggunakan sistem untuk mengisi data pemeriksaan fisik dan kejiwaan. Objek-objek yang terlibat di dalam interaksi di antaranya adalah aktor dan *v_data_pemeriksaan-insert* sebagai *interface*, kelas

c_pemeriksaan_fiskej sebagai objek *control*, dan kelas m_data_diri, m_data_pemeriksaan, dan m_pemeriksaan_fiskej sebagai objek *model*.

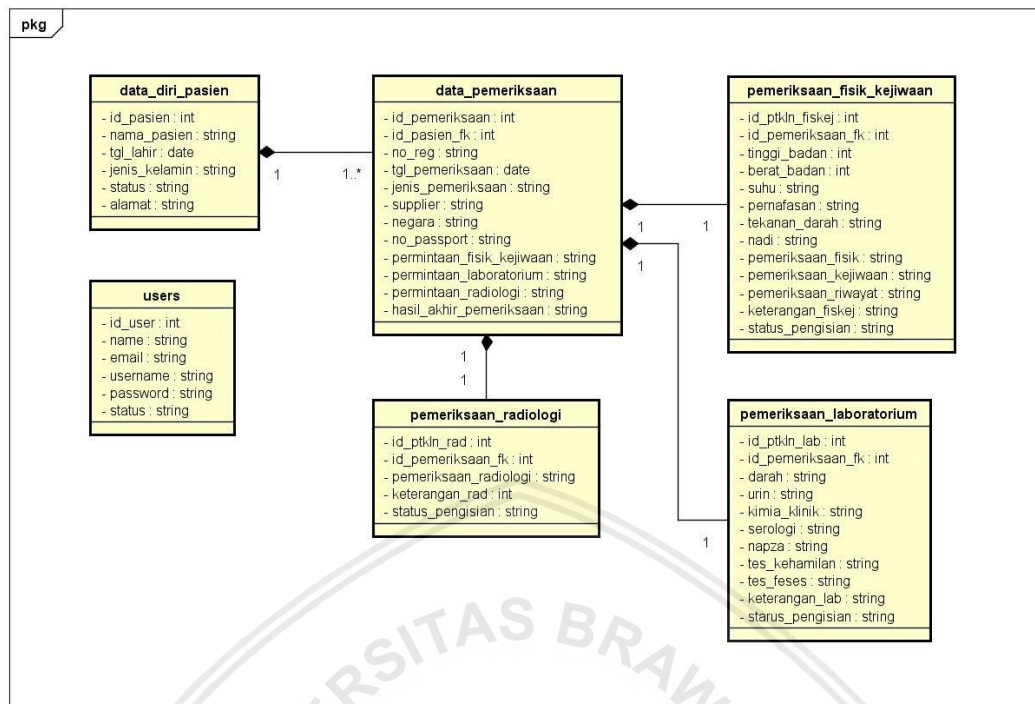
Pada *Sequence diagram* mengisi data pemeriksaan fisik dan kejiwaan dimulai dari aktor memilih memproses hasil pemeriksaan pasien untuk mengakses *interface* v_data_pemeriksaan-insert dengan memanggil fungsi createHasilPemeriksaan() pada objek *control* c_pemeriksaan_fiskej. Interface v_data_pemeriksaan-insert ditampilkan bersama data yang diambil oleh objek *control* c_pemeriksaan_fiskej melalui objek *model* m_data_diri dan m_data_pemeriksaan. Aktor memasukan data hasil pemeriksaan fisik dan kejiwaan dan memilih simpan dengan memanggil fungsi storeHasilPemeriksaan() serta memanggil fungsi save() untuk menyimpan inputan pada objek *model* m_pemeriksaan_fiskej.

5.2 Pemodelan Objek

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

5.2.1 Diagram Class Analisis

Diagram kelas analisis pada Gambar 5.5 menunjukkan visualisasi relasi atau hubungan antar kelas sebagai representasi objek yang menjadi komponen penyusun sistem yang akan dibangun. Kelas-kelas pada diagram diperoleh berdasarkan pemilihan kosakata yang telah diidentifikasi berdasar pada hasil analisis yang telah dilakukan pada bab sebelumnya.



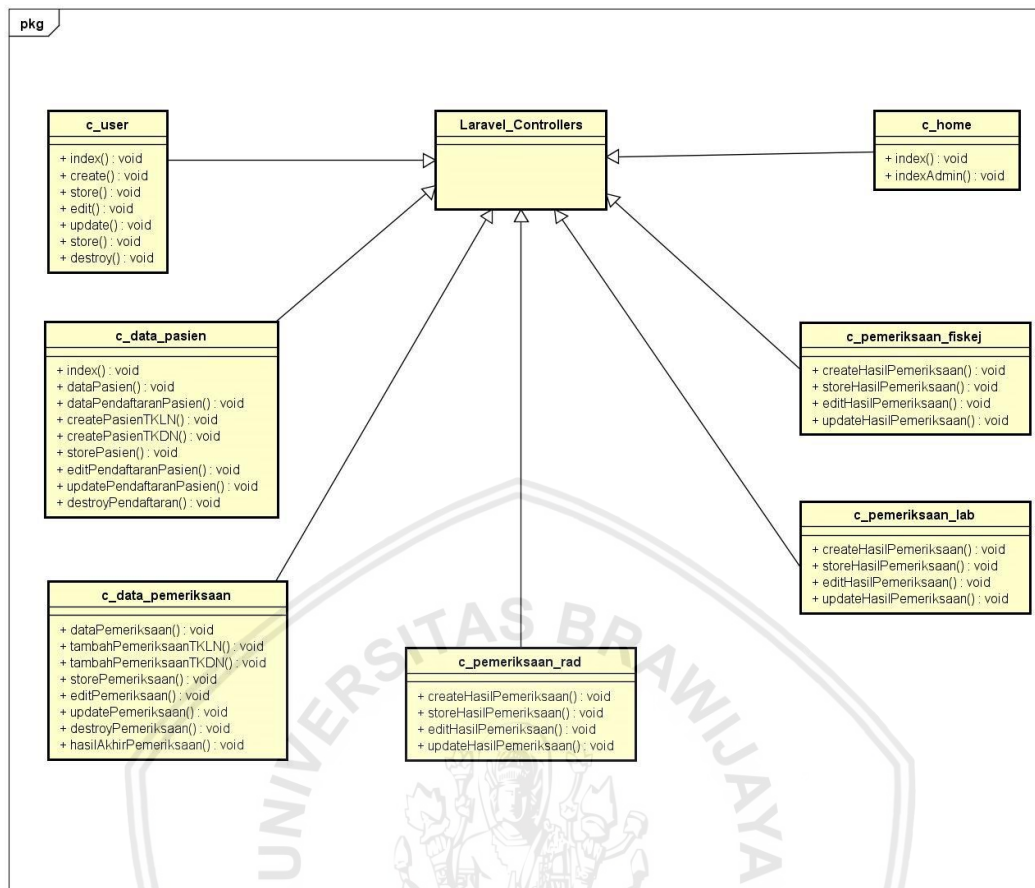
Gambar 5.5 Class Diagram Analisis

Pada gambar diatas dapat kita lihat bahwa kelas data_diri_pasien merupakan representasi objek data diri pasien yang akan melakukan pemeriksaan kesehatan yang memuat atribut identitas pasien dengan id_pasien sebagai identitas objek. Kelas data_pemeriksaan mempresentasikan objek data pemeriksaan yang dilakukan oleh pasien meliputi pemeriksaan TKLN dan TKDN beserta permintaan pemeriksaan dari pasien. Data pemeriksaan fisik dan kejiwaan, laboratorium, dan radiologi direpresentasikan dalam kelas pemeriksaan_fisik_kejiwaan, pemeriksaan_laboratorium, dan pemeriksaan_radiologi. Kelas users mempresentasikan objek data pengguna yang menggunakan sistem dengan hak akses masing-masing.

5.2.2 Diagram Class Perancangan

Diagram kelas perancangan merupakan diagram kelas hasil pengembangan diagram kelas analisis yang telah dibuat, pada dalam Gambar 5.6 dan Gambar 5.7. Diagram kelas perancangan di buat menyesuaikan diagram kelas analisis, identifikasi objek-objek pada *sequence* diagram, dan dengan menyesuaikan *framework* MVC pemrograman PHP *Laravel*.



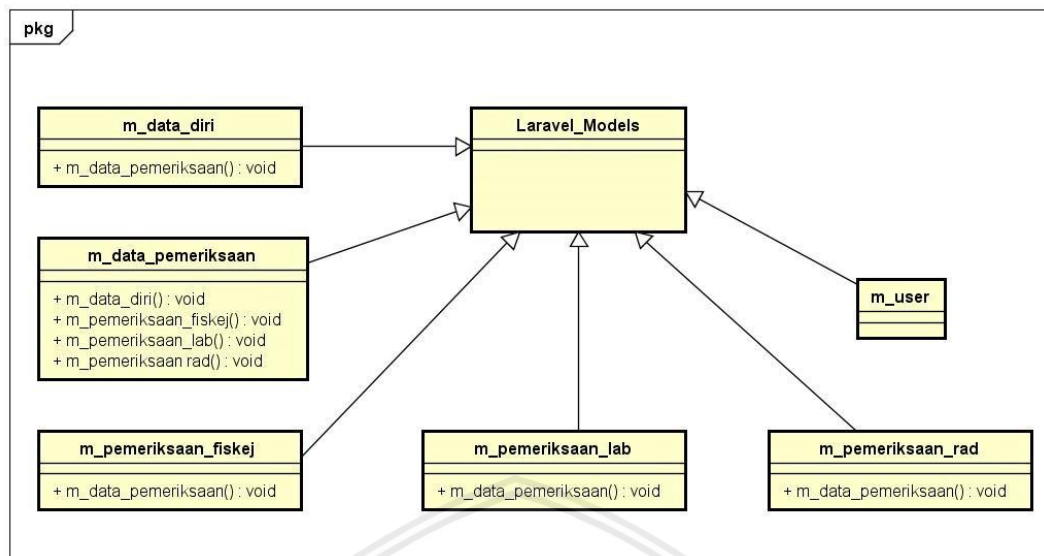


Gambar 5.6 Class Diagram Perancangan Controllers

Gambar 5.6 menjelaskan secara umum tentang fungsi logika yang merepresentasikan kemampuan sistem untuk memenuhi kebutuhan pengguna. Fungsi logika yang merepresentasikan kemampuan sistem untuk melakukan pengolahan data pengguna ditangani oleh fungsi pada subkelas `c_user`. Fungsi logika yang merepresentasikan kemampuan sistem untuk melakukan pengolahan data pendaftaran pasien ditangani oleh fungsi pada subkelas `c_data_pasien`. Subkelas `c_data_pemeriksaan` mempresentasikan kemampuan sistem untuk melakukan pengolahan data pemeriksaan pasien serta hasil akhir pemeriksaan. Subkelas `c_home` mempresentasikan kemampuan sistem untuk memberikan informasi atau pelaporan data pasien yang melakukan pemeriksaan kesehatan.

Subkelas `c_pemeriksaan_fiskej`, `c_pemeriksaan_lab`, `c_pemeriksaan_rad` masing-masing mempresentasikan kemampuan sistem untuk mengelola data hasil pemeriksaan pasien. Pemeriksaan fisik dan kejiwaan dipresentasikan oleh `c_pemeriksaan_fiskej`, pemeriksaan laboratorium dipresentasikan oleh `c_pemeriksaan_lab`, dan pemeriksaan radiologi dipresentasikan oleh `c_pemeriksaan rad`.



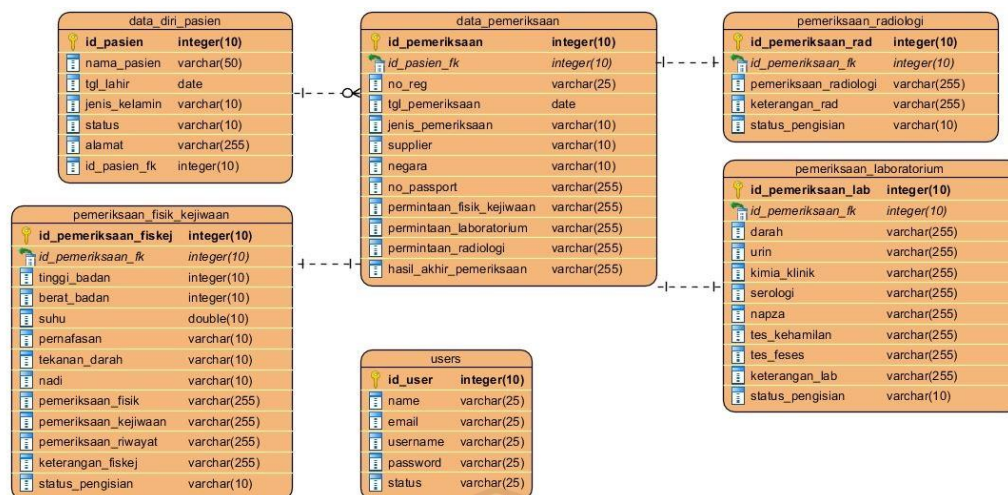


Gambar 5.7 Class Diagram Perancangan Models

Diagram kelas perancangan model Gambar 5.7 secara umum menyediakan fungsi untuk melakukan akses data yang tersimpan pada basis data. Fungsi akses data diri pasien ditangani oleh subkelas model *m_data_diri*. Subkelas *m_data_pemeriksaan* menangani akses data dari data pemeriksaan pasien. Fungsi akses pemeriksaan fisik dan kejiwaan ditangani oleh subkelas *m_pemeriksaan_fiskej*. Fungsi akses pemeriksaan laboratorium ditangani oleh subkelas *m_pemeriksaan_lab*. Fungsi akses pemeriksaan radiologi ditangani oleh subkelas *m_pemeriksaan_rad*. Subkelas *m_user* menangani akses data dari data pengguna.

5.3 Perancangan Database

Rancangan basis data yang ditujukan pada Gambar 5.8 dibuat berdasarkan pemodelan kelas diagram analisis yang terdapat pada Gambar 5.5. Masing-masing kelas yang ada pada diagram kelas analisis direpresentasikan ke tabel yang terdapat pada PDM, sedangkan atribut yang ada di dalam setiap kelas analisis direpresentasikan oleh nama-nama kolom pada setiap tabel pada PDM. Hubungan antar tabel pada PDM dibuat berdasarkan hubungan antar kelas pada diagram kelas analisis.



Gambar 5.8 PDM Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja

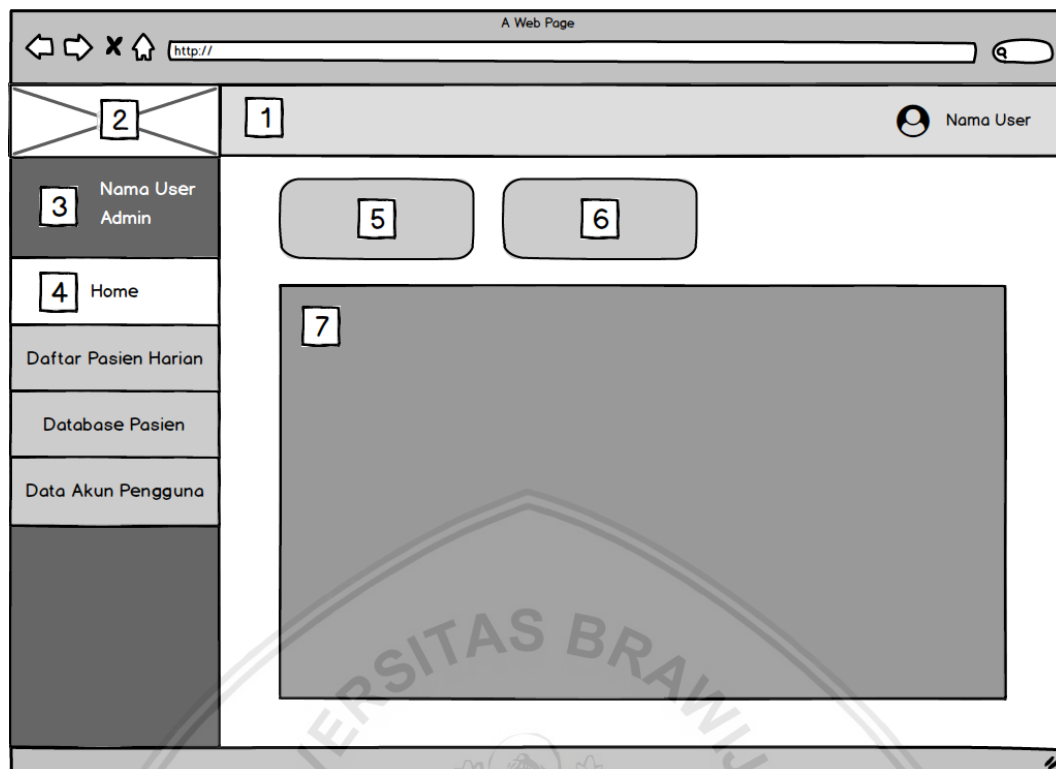
Tabel `data_diri_pasien` menyimpan data diri pasien yang mendaftarkan diri untuk pemeriksaan kesehatan. Data pemeriksaan kesehatan TKLN dan TKDN tersimpan dalam tabel `data_pemeriksaan`. Tabel `data_diri_pasien` memiliki relasi *OnetoMany* dengan tabel `data_pemeriksaan` memungkinkan untuk pasien yang sudah mendaftarkan melakukan beberapa kali pemeriksaan. Data pemeriksaan fisik dan kejiwaan, laboratorium, dan radiologi tersimpan dalam tabel `pemeriksaan_fisik_kejiwaan`, tabel `pemeriksaan_laboratorium`, dan tabel `pemeriksaan_radiologi`. Tabel `pemeriksaan_fisik_kejiwaan`, `pemeriksaan_laboratorium`, dan `pemeriksaan_radiologi` memiliki relasi *OnetoOne* dengan tabel `data_pemeriksaan` sehingga setiap pemeriksaan hanya memiliki satu data pemeriksaan fisik dan kejiwaan, laboratorium, dan radiologi. Data pengguna sistem informasi tersimpan dalam tabel `user`.

5.4 Perancangan Antarmuka

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

5.4.1 Perancangan Antarmuka Home

Pada Gambar 5.9 merupakan tampilan antarmuka ketika pengguna melihat halaman utama. Komponen pada antarmuka ini diantaranya (1) *top-nav*, (2) *logo*, (3) *profile*, (4) *general-menu*. Halaman konten memiliki komponen (5) tambah TKLN, (6) tambah TKDN, (7) *dashboard*.



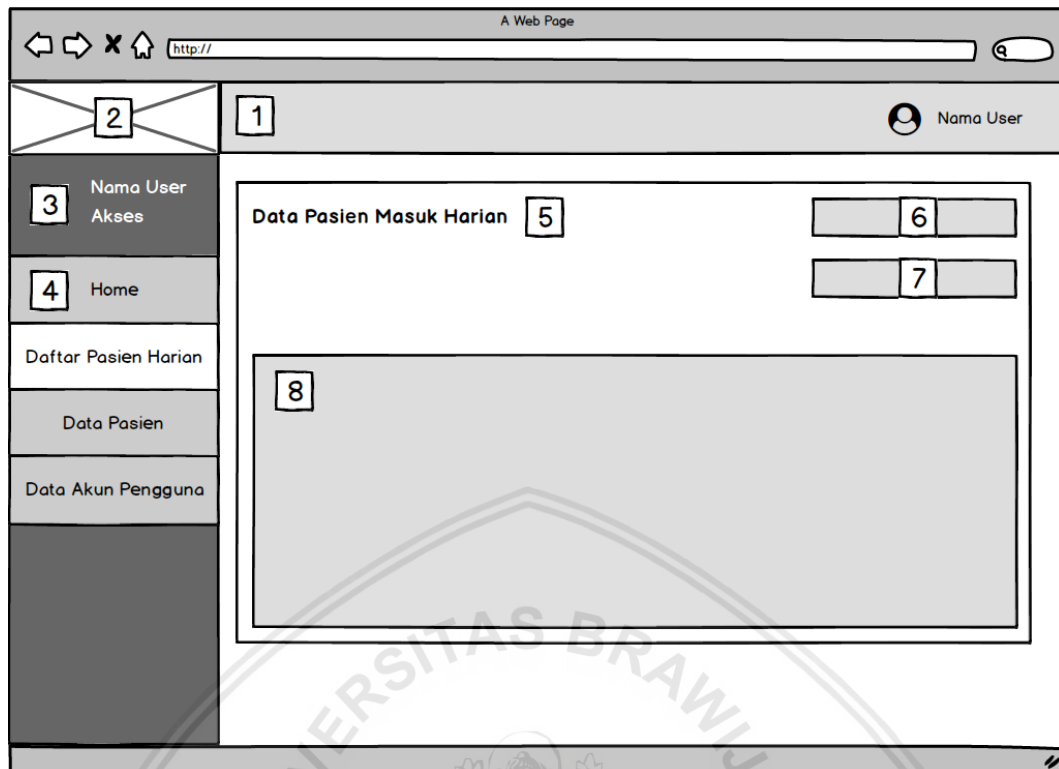
Gambar 5.9 Perancangan Antarmuka *Home*

Pada *top-nav* berisi nama pengguna yang masuk kedalam sistem. *Profile* berisi nama akun beserta hak akses yang dimiliki akun tersebut. *General-menu* berisi menu utama diantaranya *home*, daftar pasien, data pasien, dan data akun pengguna. Pada *dashboard* berisi informasi pengolahan data bagi petugas administrasi.

5.4.2 Perancangan Antarmuka Daftar Pasien Harian

Pada Gambar 5.10 merupakan tampilan antarmuka ketika pengguna ingin melihat daftar pasien harian. Komponen pada antarmuka ini diantaranya (1) *top-nav*, (2) *logo*, (3) *profile*, (4) *general-menu*. Halaman konten memiliki komponen (5) *content*, (6) tanggal hari ini, (7) pencarian data tabel, (8) tabel.

Pada *top-nav* berisi nama pengguna yang masuk kedalam sistem. *Profile* berisi nama akun beserta hak akses yang dimiliki akun tersebut. *General-menu* berisi menu utama diantaranya *home*, daftar pasien, data pasien, dan data akun pengguna. Pada tabel berisi data pemeriksaan yang dibuat pada tanggal sekarang dan informasi progres pengerjaan pemeriksaan.

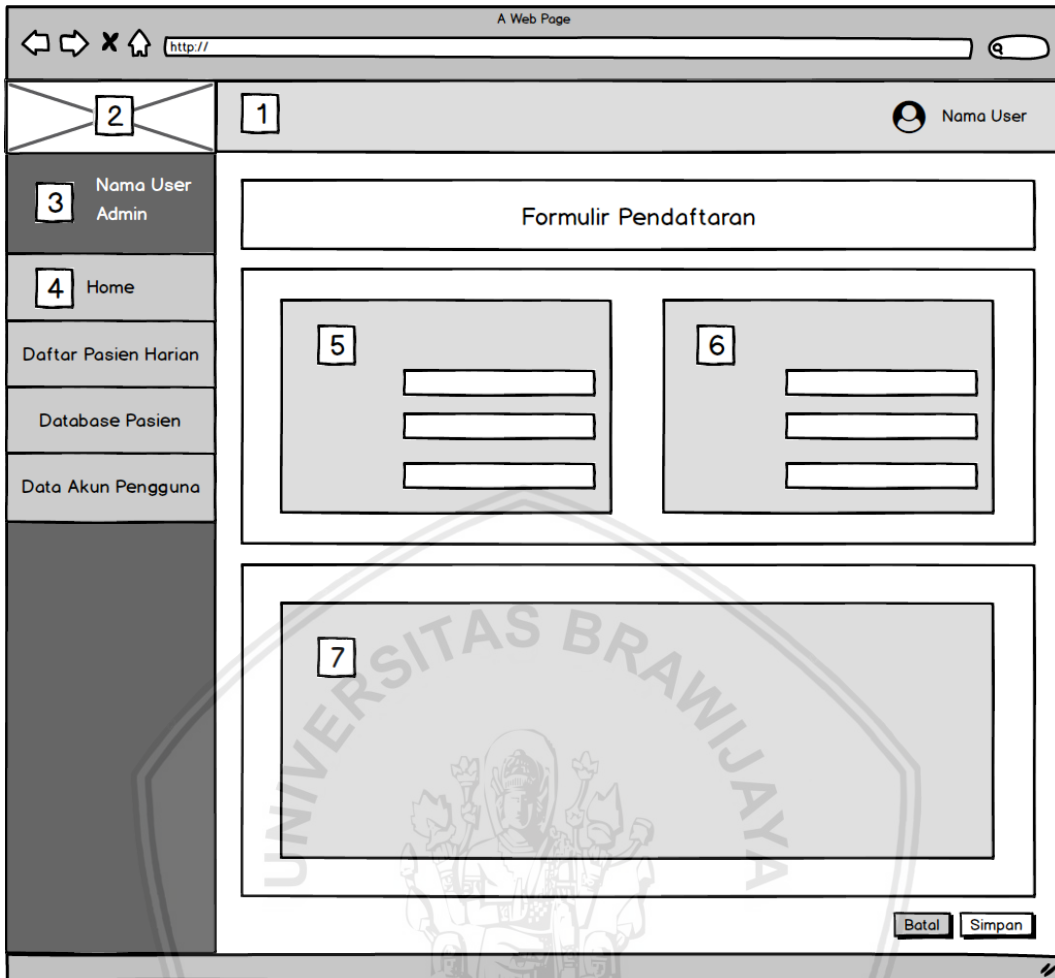


Gambar 5.10 Perancangan Antarmuka Daftar Pasien Harian

5.4.3 Perancangan Antarmuka Tambah Data Pendaftaran Pasien

Pada Gambar 5.10 merupakan tampilan antarmuka ketika pengguna ingin melakukan penambahan pasien atau mendaftarkan pasien baru ke dalam sistem. Komponen pada antarmuka ini diantaranya (1) *top-nav*, (2) *logo*, (3) *profile*, (4) *general-menu*. Halaman konten memiliki komponen (5) *form element* data diri, (6) *form element* data pemeriksaan, (7) *form element* data permintaan pemeriksaan.

Pada *top-nav* berisi nama pengguna yang masuk kedalam sistem. *Profile* berisi nama akun beserta hak akses yang dimiliki akun tersebut. *General-menu* berisi menu utama diantaranya home, daftar pasien, data pasien, dan data akun pengguna. *Form element* data diri merupakan inputan form untuk memasukan data diri pasien yang akan mendaftar. *Form element* data pemeriksaan merupakan inputan form untuk memasukan data pemeriksaan pasien yang akan mendaftar. *Form element* data permintaan pemeriksaan merupakan inputan form untuk memasukan data permintaan pemeriksaan pasien yang akan mendaftar.

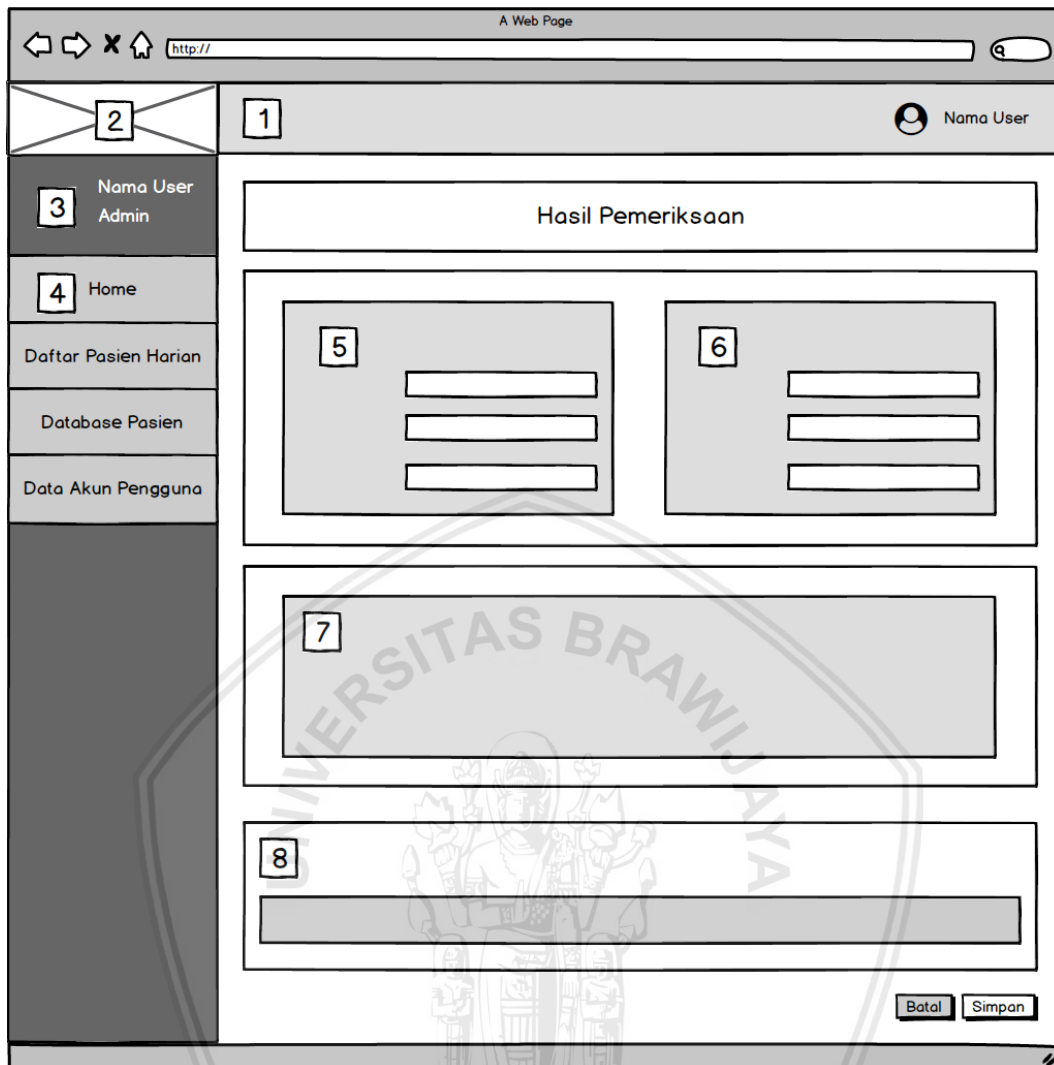


Gambar 5.11 Perancangan Antarmuka Tambah Data Pendaftaran Pasien

5.4.4 Perancangan Antarmuka Hasil Seluruh Pemeriksaan

Pada Gambar 5.11 merupakan tampilan antarmuka ketika pengguna ingin melihat data hasil seluruh pemeriksaan pasien. Komponen pada antarmuka ini diantaranya (1) *top-nav*, (2) *logo*, (3) *profile*, (4) *general-menu*. Halaman konten memiliki komponen (5) *form element* data diri, (6) *form element* data pemeriksaan, (7) *form element* data hasil pemeriksaan, (8) keterangan pemeriksaan.

Pada *top-nav* berisi nama pengguna yang masuk kedalam sistem. *Profile* berisi nama akun beserta hak akses yang dimiliki akun tersebut. *General-menu* berisi menu utama diantaranya home, daftar pasien, data pasien, dan data akun pengguna. *Form element* data diri merupakan inputan form untuk memasukan data diri pasien. *Form element* data pemeriksaan merupakan inputan form untuk memasukan data pemeriksaan pasien. *Form element* data hasil pemeriksaan merupakan inputan form untuk memasukan data hasil pemeriksaan pasien.



Gambar 5.12 Perancangan Antarmuka Hasil Seluruh Pemeriksaan

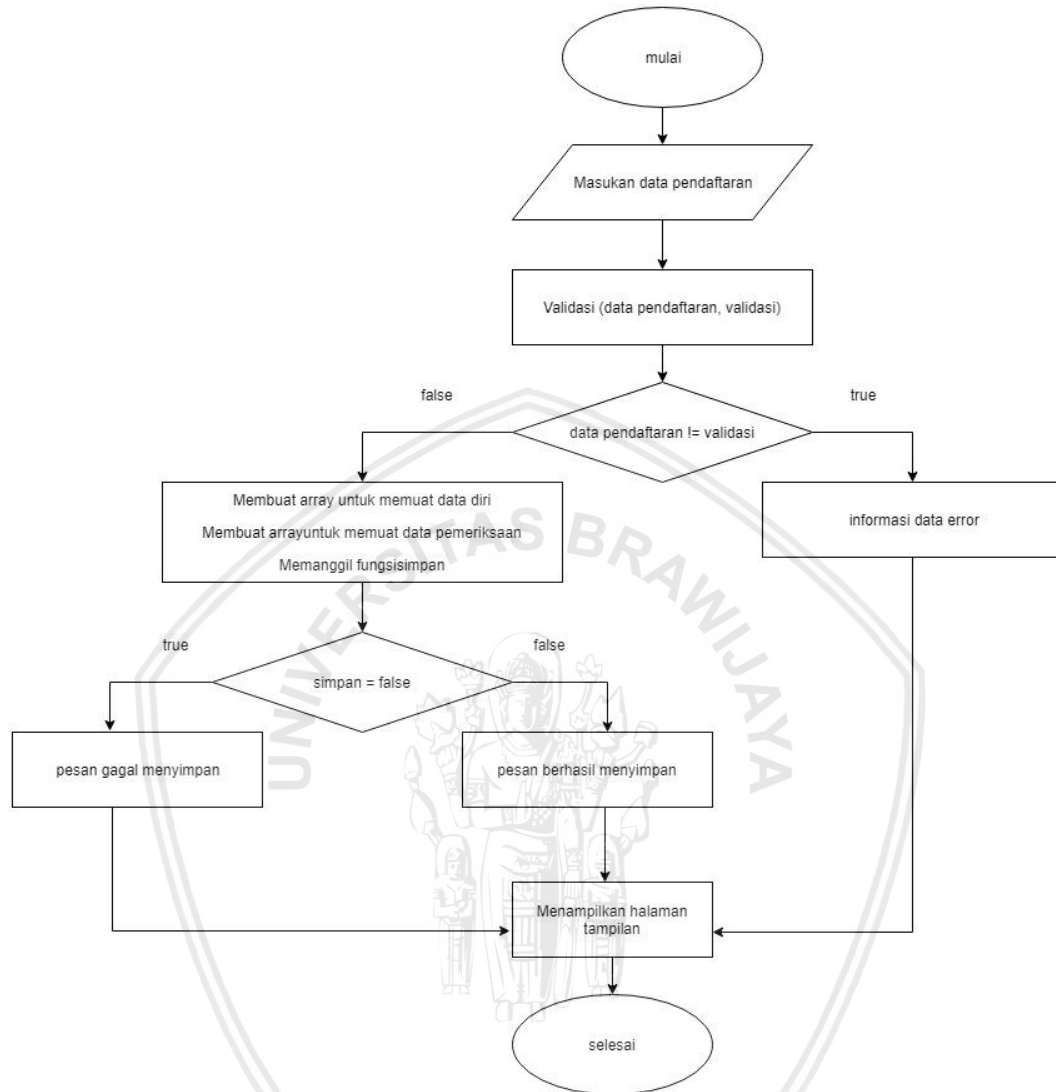
5.5 Perancangan Algoritme

Untuk mengembangkan sistem informasi layanan pemeriksaan kesehatan tenaga kerja dibutuhkan perancangan algoritme. Tujuan perancangan algoritme untuk menentukan serta menyusun operasi logika. Perancangan algoritme didokumentasikan kedalam bentuk *flowchart* dan *pseudocode*. Algoritme yang dibuat pada penelitian ini berasal dari beberapa fungsi pada *sequence diagram*. Hasil dari perancangan algoritme nantinya digunakan sebagai acuan dalam implementasi sistem informasi.

5.5.1 Algoritme Menambah Data Pendaftaran Pasien Baru

Subbab ini menjelaskan algoritma dari sebuah fungsi dari sistem saat pengguna menggunakan sistem untuk menambah data pendaftaran pasien baru. Tujuan akhir dari perancangan algoritma ini adalah tersimpannya data pendaftaran pasien yang terdiri dari data diri dan data pemeriksaan. *Pseudocode* pada Tabel 5.1 merupakan rancangan algoritme fungsi untuk menyimpan data pendaftaran

pasien baru dan Gambar 5.13 merupakan *flowchart* untuk algoritme menambah data pendaftaran baru.



Gambar 5.13 Flowchart Menambah Data Pendaftaran Pasien Baru

Tabel 5.1 Pseudocode Menambah Data Pendaftaran Pasien Baru

Baris	Pseudocode
1	Mulai
2	Masukan data pendaftaran
3	
4	Validasi (data pendaftaran, validasi)
5	If data pendaftaran != validasi then
6	Mengembalikan pada tampilan dengan informasi data error
7	Else
8	Membuat array untuk memuat data diri
9	Membuat array untuk memuat data pemeriksaan
10	Memanggil fungsi simpan
11	If simpan = false then
12	

Baris	Pseudocode
13	Mengembalikan pada tampilan dengan pesan gagal
14	menyimpan
15	Else
16	Mengembalikan pada tampilan dengan pesan berhasil
17	menyimpan
18	End if
	End if

Algoritme dimulai dengan masukan data pendaftaran. Kemudian data tersebut akan dilakukan validasi untuk pengecekan kesesuaian masukan data. Apabila data tidak valid maka akan mengembalikan pada halaman tampilan dengan informasi data yang *error*, namun jika valid maka akan memuat data diri dan data pemeriksaan kedalam *array* untuk dilakukan proses pemasukan data dengan fungsi simpan. Selanjutnya dilakukan pengecekan proses penyimpanan data apabila berhasil maka akan menampilkan pesan berhasil, namun jika tidak maka akan menampilkan pesan gagal pada halaman tampilan.

5.5.2 Algoritme Menambah Data Pemeriksaan Baru Pasien

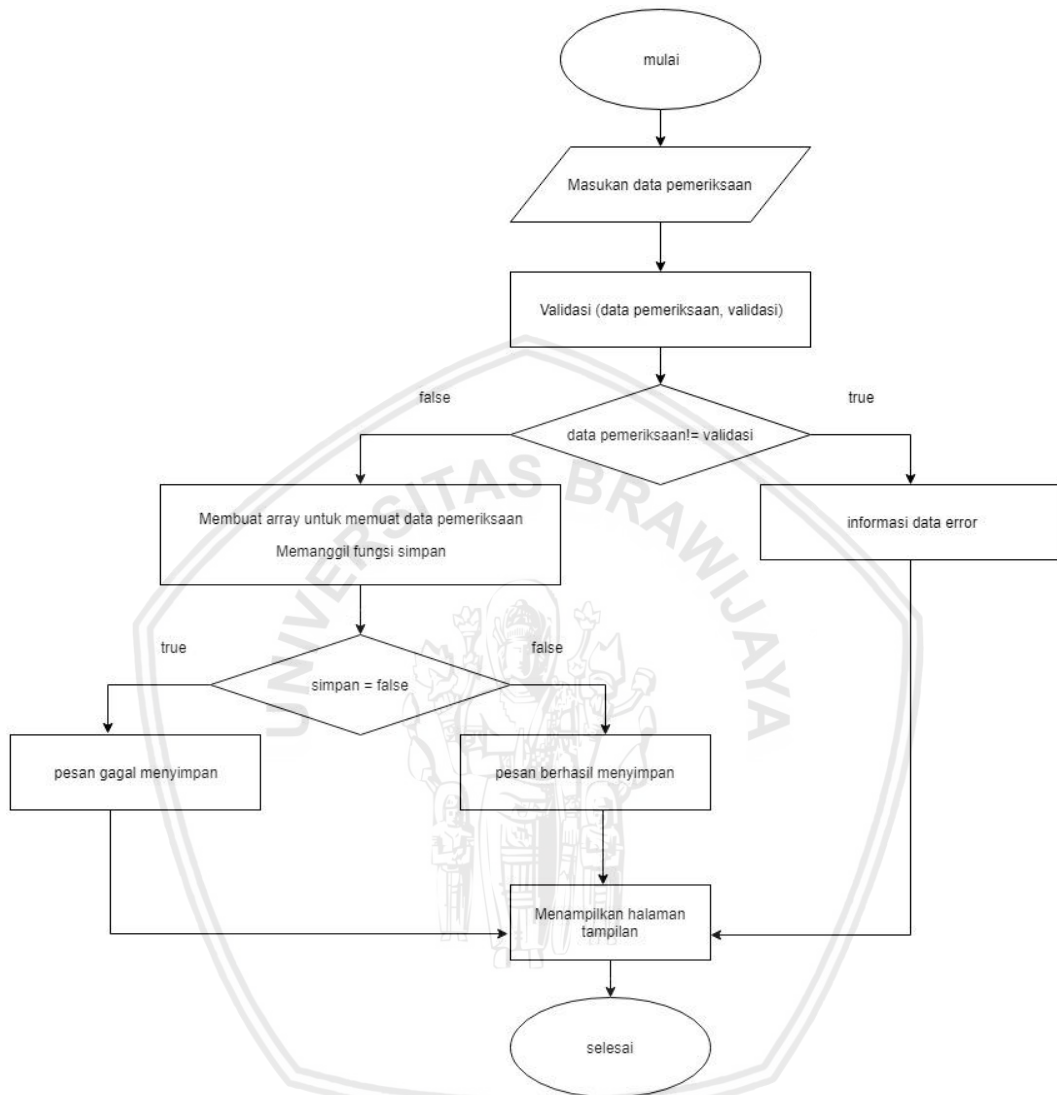
Subbab ini menjelaskan algoritma dari sebuah fungsi dari sistem saat pengguna menggunakan sistem untuk menambah data pemeriksaan pasien. Tujuan dari algoritma ini adalah tersimpahnya data pemeriksaan pasien baru yang diinputkan. *Pseudocode* pada Tabel 5.2 merupakan rancangan algoritme fungsi untuk menyimpan data pemeriksaan pasien baru dan Gambar 5.14 merupakan *flowchart* untuk algoritme menambah data pemeriksaan baru.

Algoritme dimulai dengan masukan data pemeriksaan. Kemudian data tersebut akan dilakukan validasi untuk pengecekan kesesuaian masukan data. Apabila data tidak valid maka akan mengembalikan pada halaman tampilan dengan informasi data yang *error*, namun jika valid maka akan memuat data pemeriksaan kedalam *array* untuk dilakukan proses pemasukan data dengan fungsi simpan. Selanjutnya dilakukan pengecekan proses penyimpanan data apabila berhasil maka akan menampilkan pesan berhasil, namun jika tidak maka akan menampilkan pesan gagal pada halaman tampilan.

Tabel 5.2 Algoritma Menambah Data Pemeriksaan Baru Pasien

Baris	Pseudocode
1	Mulai
2	Masukan data pemeriksaan baru
3	
4	Validasi (data pemeriksaan, validasi)
5	If data pemeriksaan!= validasi then
6	Mengembalikan pada tampilan dengan informasi data error
7	Else
8	Membuat array untuk memuat data pemeriksaan
9	Memanggil fungsi simpan
10	If simpan = false then
11	Mengembalikan pada tampilan dengan pesan gagal
12	Else
13	Mengembalikan pada tampilan dengan pesan berhasil
14	End if

Baris	Pseudocode
15	End if
16	



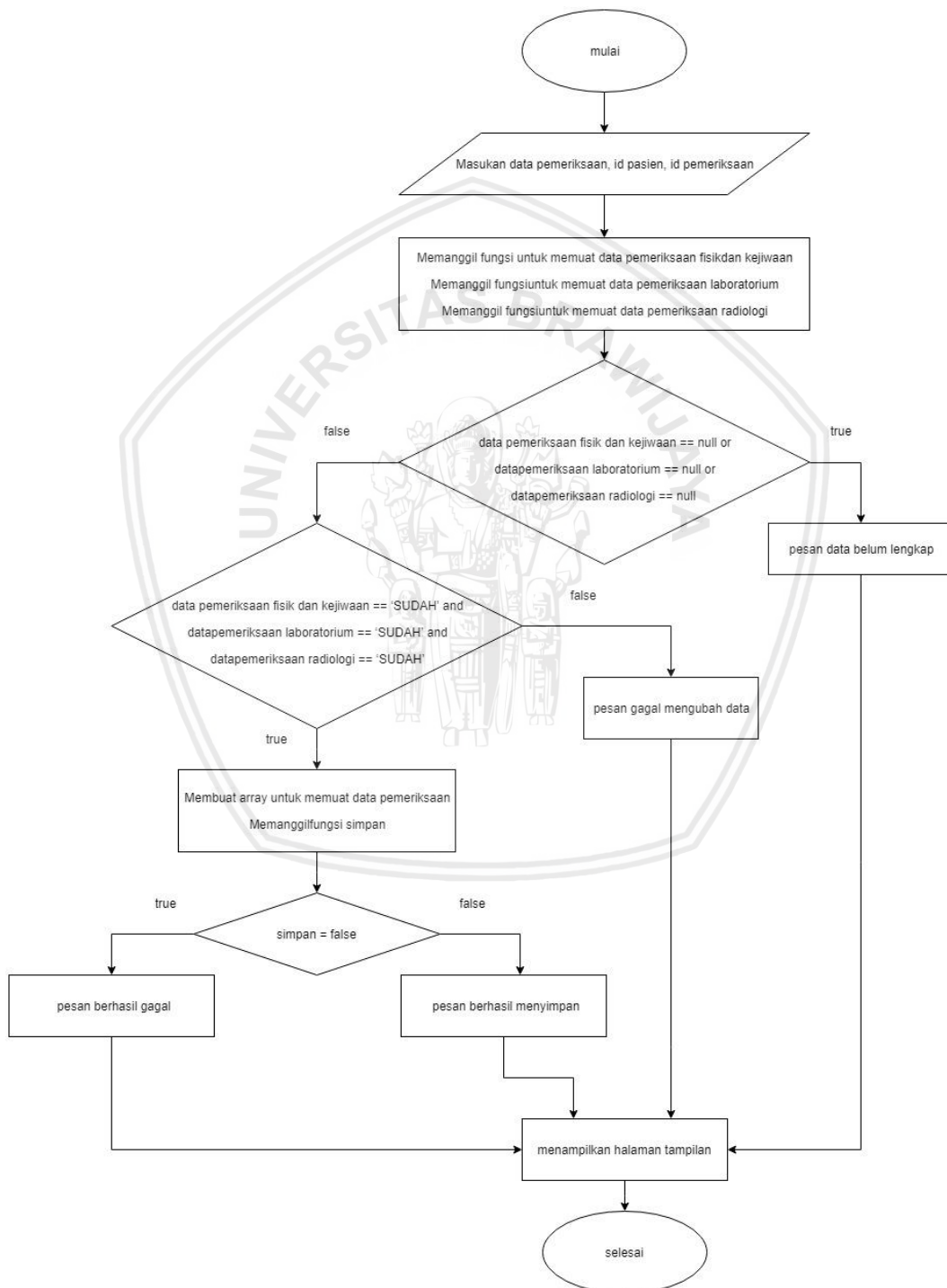
Gambar 5.14 Flowchart Menambah Data Pemeriksaan Baru Pasien

5.5.3 Algoritme Mengubah Hasil Akhir Pemeriksaan Pasien

Subbab ini menjelaskan algoritma dari sebuah fungsi dari sistem saat pengguna menggunakan sistem untuk mengubah data hasil akhir pemeriksaan pasien. Tujuan dari algoritma ini adalah mengubah data hasil pemeriksaan pasien menjadi FIT/UNFIT sesuai kebutuhan pengguna. *Pseudocode* pada Tabel 5.3 merupakan rancangan algoritme fungsi untuk mengubah data hasil akhir pemeriksaan pasien dan Gambar 5.15 merupakan *flowchart* untuk algoritme mengubah data hasil akhir pemeriksaan pasien.

Algoritme dimulai dengan masukan data pemeriksaan, id pemeriksaan, dan id pasien. Kemudian memanggil fungsi untuk memuat semua data pemeriksaan dan dilakukan pengecekan data menggunakan *ifelse*. Pengkondisian pertama dilakukan

untuk pengecekan apakah data setiap pemeriksaan ada yang masih kosong atau tidak jika ada maka akan mengembalikan pada tampilan halaman dengan pesan data belum lengkap. Pada pengkondisian kedua dilakukan untuk pengecekan apabila data semua pemeriksaan == 'SUDAH', jika iya maka akan membuat *array* untuk memuat data pemeriksaan baru. Dalam proses perubahan data, data baru disimpan jika data berhasil disimpan maka akan menampilkan pesan berhasil pada halaman tampilan, namun jika gagal maka akan menampilkan pesan gagal.



Gambar 5.15 Flowchart Mengubah Hasil Akhir Pemeriksaan Pasien

Tabel 5.3 Pseudocode Mengubah Hasil Akhir Pemeriksaan Pasien

Baris	Pseudocode
1	Mulai
2	Masukan id pasien, id pemeriksaan
3	Memanggil fungsi untuk memuat data pemeriksaan fisik dan
4	kejiwaan
5	Memanggil fungsi untuk memuat data pemeriksaan
6	laboratorium
7	Memanggil fungsi untuk memuat data pemeriksaan radiologi
8	
9	If data pemeriksaan fisik dan kejiwaan == null or
10	data pemeriksaan laboratorium == null or
11	data pemeriksaan radiologi == null then
12	Mengembalikan pada tampilan dengan informasi data belum
13	lengkap
14	Elseif data pemeriksaan fisik dan kejiwaan == 'SUDAH' and
15	data pemeriksaan laboratorium == 'SUDAH' and
16	data pemeriksaan radiologi == 'SUDAH' then
17	Membuat array untuk memuat data pemeriksaan
18	Memanggil fungsi simpan
19	If simpan = false then
20	Mengembalikan pada tampilan dengan pesan gagal
21	Else
22	Mengembalikan pada tampilan dengan pesan berhasil
23	End if
24	Else
25	Mengembalikan pada tampilan dengan pesan gagal mengubah
26	data
27	End if
28	

5.6 Kesimpulan Perancangan Fase Elaborasi Metode *Rational Unifield Process*

Perancangan sistem informasi layanan pemeriksaan kesehatan tenaga kerja pada fase elaborasi metode *Rational Unifield Process* telah dilakukan, perancangan dibuat berdasarkan hasil analisis persyaratan yang sudah dilakukan pada fase inepsi. Perancangan yang dilakukan meliputi pemodelan interaksi objek dengan *sequence diagram*, visualisasi model objek dengan *class diagram* dan beberapa rancangan yaitu *physical data model*, algoritme, dan antarmuka sistem sehingga syarat yang harus terpenuhi pada fase elaborasi yaitu deskripsi arsitektur sistem sudah dilakukan, maka *lifecycle architecture milestone* fase elaborasi terpenuhi.

BAB 6 IMPLEMENTASI

Pada bab ini menjelaskan kegiatan dalam metode *Rational Unified Process* (RUP) fase konstruksi tahap implementasi sistem dilakukan, yaitu implementasi dalam membangun sistem informasi berdasarkan hasil perancangan pada bab sebelumnya. Pada fase sebelumnya elaborasi evaluasi dilakukan dan tidak terdapat perubahan atau perbaikan yang dilakukan. Bahasan pada sub bab terdiri dari implementasi beberapa algoritme dan implementasi beberapa antarmuka pengguna sistem.

6.1 Implementasi Algoritme

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

6.1.1 Implementasi Algoritme Menambah Data Pendaftaran Pasien Baru

Kode program Menambah data pendaftaran pasien berfungsi untuk mengarahkan tampilan ke halaman pendaftaran pasien dan menyimpan data inputan. Implementasi algoritme ini dibuat berdasarkan hasil rancangan algoritme menambah data pendaftaran pasien baru pada bab sebelumnya. Implementasi algoritme menambah data pendaftaran pasien baru dapat dilihat pada Tabel 6.1.

Tabel 6.1 Implementasi Algoritme Menambah Data Pendaftaran Pasien Baru

Baris	Kode Program
1	<code>public function storePasien(Request \$request){</code>
2	<code>\$permintaan_fiskej = implode(",", \$request-</code>
3	<code>>get('permintaan_fiskej'));</code>
4	<code>\$permintaan_lab = implode(",", \$request-</code>
5	<code>>get('permintaan_lab'));</code>
6	<code>\$permintaan_rad = implode(",", \$request-</code>
7	<code>>get('permintaan_rad'));</code>
8	
9	<code>\$this->validate(\$request, [</code>
10	<code>'nama_lengkap' =></code>
11	<code>'required unique:data_diri,nama_pasien string',</code>
12	<code>'tgl_lahir' => 'required date(mm-dd-yyy)',</code>
13	<code>'jenis_kelamin' => 'required',</code>
14	<code>'password' => 'required min:6',</code>
15	<code>'status' => 'required',</code>
16	<code>'no_register' =></code>
17	<code>'required unique:data_pemeriksaan,no_reg',</code>
18	<code>'supplier' => 'required',</code>
19	<code>'negara' => 'required',</code>
20	<code>'no_passport' => 'required max:30'</code>
21	<code>]);</code>
22	
23	<code>\$data_diri = new m_data_diri();</code>
24	<code>\$data_diri->nama_pasien = \$request->nama_lengkap;</code>

25	<code>\$data_diri->tgl_lahir = \$request->tgl_lahir;</code>
26	<code>\$data_diri->jenis_kelamin = \$request->jenis_kelamin;</code>
27	<code>\$data_diri->status = \$request->status;</code>
28	<code>\$data_diri->alamat = \$request->alamat;</code>
29	<code>\$data_diri->save();</code>
30	
31	<code>\$data_pemeriksaan = new m_data_pemeriksaan();</code>
32	<code>\$data_pemeriksaan->id_pasien_fk = \$data_diri->id_pasien;</code>
33	<code>\$data_pemeriksaan->no_reg = \$request->no_register;</code>
34	<code>\$data_pemeriksaan->tgl_pemeriksaan = \$request->tgl_masuk;</code>
35	<code>\$data_pemeriksaan->jenis_pemeriksaan = \$request-></code>
36	<code>>pemeriksaan;</code>
37	<code>\$data_pemeriksaan->supplier = \$request->supplier;</code>
38	<code>\$data_pemeriksaan->negara = \$request->negara;</code>
40	<code>\$data_pemeriksaan->no_passport = \$request->no_passport;</code>
41	<code>\$data_pemeriksaan->permintaan_fisik_kejiwaan =</code>
42	<code>\$permintaan_fiskej;</code>
43	<code>\$data_pemeriksaan->permintaan_laboratorium =</code>
44	<code>\$permintaan_lab;</code>
45	<code>\$data_pemeriksaan->permintaan_radiologi =</code>
46	<code>\$permintaan_rad;</code>
47	<code>\$data_pemeriksaan->hasil_akhir_pemeriksaan = \$request-></code>
48	<code>>hasil_akhir;</code>
49	<code>\$data_pemeriksaan->save();</code>
50	<code>return redirect()->route('data_pasien_harian')-</code>
51	<code>>with('alert-success','Data Pasien baru berhasil</code>
52	<code>ditambahkan!');</code>
53	<code>}</code>
54	

6.1.2 Implementasi Algoritme Menambah Data Pemeriksaan Baru

Kode program Menambah Data Pemeriksaan Baru memiliki fungsi untuk mengarahkan tampilan ke halaman data pemeriksaan edit dan melakukan penyimpanan data inputan pemeriksaan. Implementasi algoritme ini dibuat berdasarkan hasil rancangan algoritme menambah data pemeriksaan baru pada bab sebelumnya. Implementasi algoritme menambah data pemeriksaan baru dapat dilihat pada Tabel 6.2.

Tabel 6.2 Implementasi Menambah Data Pemeriksaan Baru

Baris	Kode Program
1	<code>public function storePemeriksaan(Request \$request,</code>
2	<code>\$id_pasien){</code>
3	<code>\$permintaan_fiskej = implode(",", \$request-></code>
4	<code>>get('permintaan_fiskej'));</code>
5	<code>\$permintaan_lab = implode(",", \$request-></code>
6	<code>>get('permintaan_lab'));</code>
7	<code>\$permintaan_rad = implode(",", \$request-></code>
8	<code>>get('permintaan_rad'));</code>
9	
10	<code>\$this->validate(\$request, [</code>
11	<code>'no_register' =></code>
12	<code>'required unique:data_pemeriksaan,no_reg',</code>
13	<code>'supplier' => 'required date(mm-dd-yy) ',</code>



14	'supplier' => 'required',
15	'negara' => 'required',
16	'no_passport' => 'required max:30'
17]);
18	
19	\$data_pemeriksaan_tkln = new m_data_pemeriksaan();
20	\$data_pemeriksaan_tkln->id_pasien_fk = \$id_pasien;
21	\$data_pemeriksaan_tkln->no_reg = \$request->no_register;
22	\$data_pemeriksaan_tkln->tgl_pemeriksaan = \$request->
23	>tgl_masuk;
24	\$data_pemeriksaan_tkln->jenis_pemeriksaan = \$request->
25	>pemeriksaan;
26	\$data_pemeriksaan_tkln->supplier = \$request->supplier;
27	\$data_pemeriksaan_tkln->negara = \$request->negara;
28	\$data_pemeriksaan_tkln->no_passport = \$request->
29	>no_passport;
30	\$data_pemeriksaan_tkln->permintaan_fisik_kejiwaan =
31	\$permintaan_fiskej;
32	\$data_pemeriksaan_tkln->permintaan_laboratorium =
33	\$permintaan_lab;
34	\$data_pemeriksaan_tkln->permintaan_radiologi =
35	\$permintaan_rad;
36	\$data_pemeriksaan_tkln->hasil_akhir_pemeriksaan =
37	\$request->hasil_akhir;
38	\$data_pemeriksaan_tkln->save();
39	return redirect()->route('data_pendaftaran', \$id_pasien)-
40	>with('alert-success','Data Pemeriksaan baru berhasil
41	ditambahkan!');
42	}

6.1.3 Implementasi Algoritme Mengubah Hasil Akhir Pemeriksaan Pasien

Kode program mengubah hasil akhir pemeriksaan pasien memiliki fungsi untuk melakukan perubahan data hasil akhir pemeriksaan menjadi FIT/UNFIT. Perubahan data hasil akhir pemeriksaan ini dilakukan dengan melakukan pengecekan ketersediaan data pada setiap pemeriksaan fiskej, laboratorium, dan radiologi dimana untuk melakukan perubahan data seluruh data pemeriksaan harus sudah terisi. Implementasi algoritme ini dibuat berdasarkan hasil rancangan algoritme mengubah hasil akhir pemeriksaan pasien pada bab sebelumnya. Tabel 6.3 merupakan implementasi algoritme fungsi untuk mengubah hasil akhir pemeriksaan pasien.

Tabel 6.3 Implementasi Algoritme Mengubah Hasil Akhir Pemeriksaan Pasien

Baris	Kode Program
1	public function hasilAkhirPemeriksaan(Request \$request,
2	\$id, \$id_pemeriksaan){
3	\$data_fiskej =
4	m_pemeriksaan_fiskej::where('id_pemeriksaan_tkln_fk',\$id_
5	pemeriksaan)->first();
6	\$data_lab =
7	m_pemeriksaan_lab::where('id_pemeriksaan_tkln_fk',\$id_pem
8	eriksaan)->first();
9	\$data_rad =
10	m_pemeriksaan_rad::where('id_pemeriksaan_tkln_fk',\$id_pem
11	eriksaan)->first();

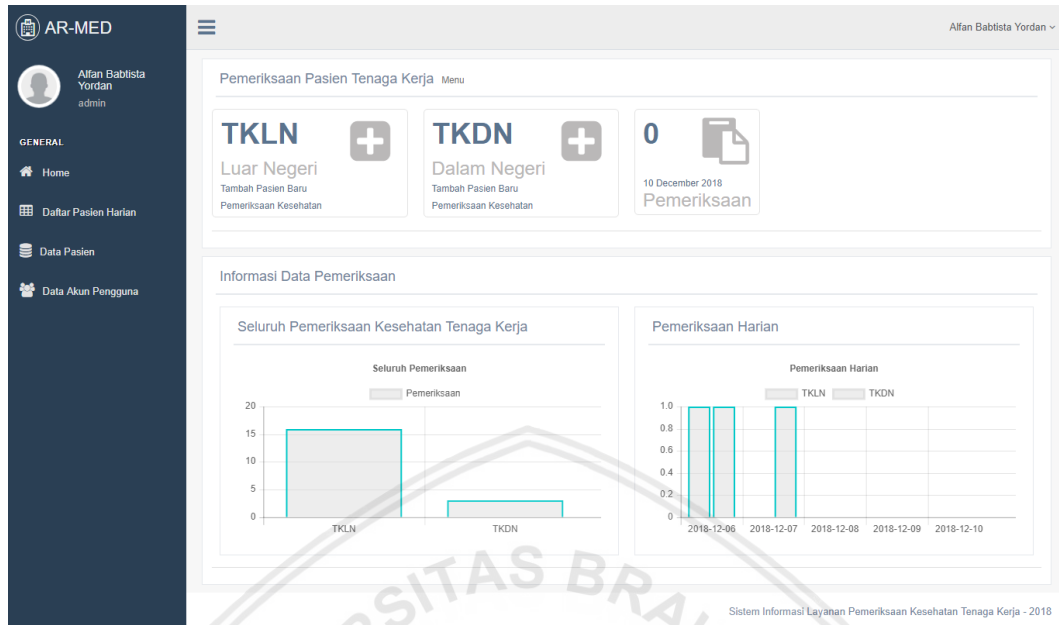


12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42	<pre> if(is_null(\$data_fiskej) or is_null(\$data_lab) or is_null(\$data_rad)){ return redirect()->route('data_pemeriksaan', [\$id, \$id_pemeriksaan])->with('alert-fail','Data hasil pemeriksaan pasien masih belum lengkap!'); }elseif(\$data_fiskej->status_pengisian == 'SUDAH'and \$data_lab->status_pengisian == 'SUDAH' and \$data_rad->status_pengisian == 'SUDAH'){ \$data_pemeriksaan_tkln = m_data_pemeriksaan::where('id_pemeriksaan_tkln',\$id_pemer iksaan)->first(); \$data_pemeriksaan_tkln->hasil_akhir_pemeriksaan = \$request->hasil_akhir; \$data_pemeriksaan_tkln->save(); return redirect()->route('data_pemeriksaan', [\$id, \$id_pemeriksaan])->with('alert-success','Hasil Akhir Pemeriksaan Pasien berhasil diubah!'); } else{ return redirect()->route('home.index')->with('success', true)->with('message','Data Pendaftaran Pasien Berhasil Dibuat!'); } } </pre>
--	--

6.2 Implementasi Antarmuka Pengguna

Hasil tampilan antarmuka pengguna sistem informasi dijelaskan dalam subbab implementasi antarmuka pengguna. Subbab ini hanya menyertakan beberapa hasil implementasi antarmuka pengguna. Implementasi antarmuka pengguna yang ditampilkan adalah antarmuka *home*, antarmuka halaman pasien harian, halaman pendaftaran pasien, dan halaman hasil pemeriksaan pasien.

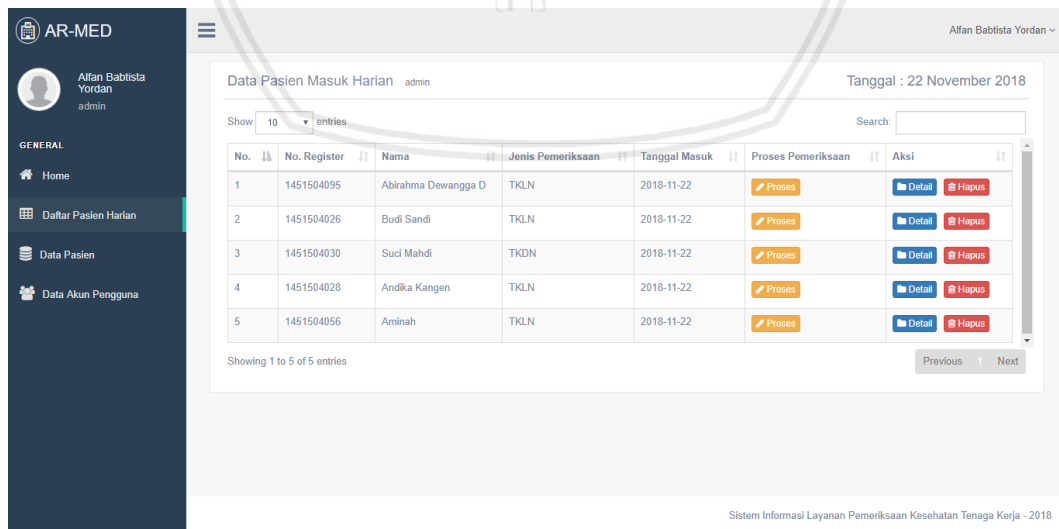
6.2.1 Antarmuka Halaman *Home*



Gambar 6.1 Antarmuka Halaman *Home*

Gambar 6.1 merupakan hasil implementasi dari perancangan halaman *Home* yang telah dibuat pada bab sebelumnya. Halaman ini digunakan pengguna untuk menambahkan data pendaftaran baru, melihat jumlah pemeriksaan hari ini, dan informasi data pemeriksaan. Komponen yang terdapat pada halaman ini adalah penambahan data pendaftaran pasien baru, data jumlah pemeriksaan hari ini, dan grafik *chart* yang menunjukkan informasi seluruh pemeriksaan dan pemeriksaan harian.

6.2.2 Antarmuka Halaman Daftar Pasien Harian



Gambar 6.2 Antarmuka Halaman Daftar Pasien Harian

Gambar 6.2 merupakan hasil implementasi dari perancangan halaman daftar pasien harian yang telah dibuat pada bab sebelumnya. Halaman ini digunakan

repository.ub.ac.id

pengguna untuk melihat pemeriksaan yang sedang dilakukan pada hari ini dan memantau progres pengerjaan pemeriksaan. Komponen yang terdapat pada halaman ini adalah tabel yang menampilkan seluruh data pemeriksaan yang sedang dilakukan hari ini, kolom progres pengerjaan data pemeriksaan, kolom aksi terdapat detail dan hapus data, dan tanggal hari ini.

6.2.3 Antarmuka Halaman Pendaftaran Pasien

AR-MED

Alfan Babbista Yordan
admin

GENERAL

Home

Daftar Pasien Harian

Data Pasien

Data Akun Pengguna

Formulir Pendaftaran

Pasien Pemeriksaan Kesehatan Tenaga Kerja Luar Negeri

Data Diri Pasien

Nama Lengkap

Tanggal Lahir

Jenis Kelamin

Status

Alamat Pasien

No. Register

Tanggal Masuk

Pemeriksaan

Supplier

Negara

Nomor Passport

Permintaan Pemeriksaan

Pemeriksaan Fisik dan Kejiwaan

Pemeriksaan Fisik

Pemeriksaan Kejiwaan

Riwayat Penyakit

Pemeriksaan Laboratorium

Darah Lengkap

Pemeriksaan Urine

Kimia Klinik - SGOT

Kimia Klinik - SGPT

Kimia Klinik - Ureum

Kimia Klinik - Kreatinin

Kimia Klinik - Gula Darah Sesaat

Kimia Klinik - HBsAg

Kimia Klinik - HIV

Tes Kehamilan

Feses

Serologi - TPHA

Serologi - VDRL

Tes Narkoba - Amphetamine

Tes Narkoba - Methamphetamine

Tes Narkoba - THC

Tes Narkoba - Morphine

Tes Narkoba - Benzodiazepine

Tes Narkoba - Cocaine

Pemeriksaan Radiologi

X-ray Thorax

Batal

Simpan

Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja - 2018

Gambar 6.3 Antarmuka Halaman Pendaftaran Pasien

Gambar 6.3 merupakan hasil implementasi dari perancangan halaman pendaftaran pasien yang telah dibuat pada bab sebelumnya. Halaman ini digunakan pengguna untuk menambahkan data pendaftaran pasien pemeriksaan kesehatan baru. Halaman pendaftaran pasien terdapat tiga komponen formulir yang diisi oleh pengguna yaitu data diri pasien, data pemeriksaan, dan data permintaan pemeriksaan pasien.

6.2.4 Antarmuka Halaman Hasil Pemeriksaan Pasien

The screenshot displays the AR-MED interface for viewing patient examination results. The page is titled "Data Hasil Pemeriksaan Kesehatan Tenaga Kerja".

GENERAL

- Home
- Daftar Pasien Harian
- Data Pasien
- Data Akun Pengguna

Data Diri Pasien

Nama Lengkap	Abirahma Dewangga D	No. Register	1451504001
Tanggal Lahir	11/06/2018	Tanggal Masuk	11/06/2018
Jenis Kelamin	<input checked="" type="radio"/> Laki-laki <input type="radio"/> Perempuan	Pemeriksaan	TKLN
Status	Lajang	Pengirim	PT. NAYAKA KALYANA
Alamat Pasien	Malang	Negara	Hongkong
		Nomor Passport	AAA001

Hasil Pemeriksaan Kesehatan

Pemeriksaan Fisik dan Kejiwaan | Pemeriksaan Laboratorium | Pemeriksaan Radiologi

[Edit](#)

Pemeriksaan Fisik

Tinggi Badan	170	cm	Pernafasan	12/44	X/Min
Berat Badan	65	kg	Tekanan Darah	121	mm/Hg
Suhu	27	°C	Nadi	1212	X/Min
Kepala	Normal		Paru	Normal	
Mata	Normal		Jantung	Normal	
Telinga	Normal		Abdomen	Normal	
Hidung	Normal		Anus/Rektum	Normal	
Gigi	Normal		Genitalia Externa	Normal	
Tenggorokan	Normal		Ekstriminas Atas	Normal	
Leher	Normal		Ekstrimitas Bawah	Normal	
Dada	Normal		Kelenjar Getah Bening	Normal	

Gambar 6.4 Antarmuka Halaman Hasil Pemeriksaan Pasien

Gambar 6.4 merupakan hasil implementasi dari perancangan halaman hasil pemeriksaan pasien yang telah dibuat pada bab sebelumnya. Halaman ini digunakan pengguna untuk melihat data hasil pemeriksaan pasien dari permintaan pemeriksaan yang diminta. Halaman hasil pemeriksaan pasien memiliki komponen data diri pasien, data pemeriksaan, dan data hasil pemeriksaan dari fisik dan kejiwaan, laboratorium, dan radiologi.

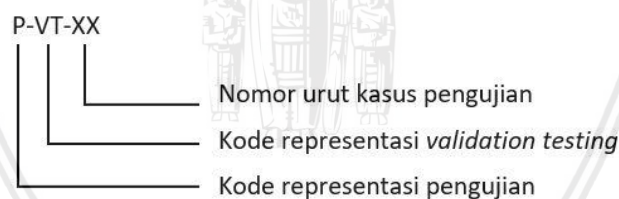
BAB 7 PENGUJIAN

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

7.1 Pengujian *Validation Testing*

7.1.1 Perancangan Pengujian *Validation Testing*

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



Gambar 7.1 Pengkodean Uji *Validation Testing*

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

Tabel 7.1 Rancangan Tampilan Hasil Pengujian *Validation Testing*

No	Kode pengujian	Tujuan Pengujian	Hasil yang didapat	Status
1				
2				

7.1.1.1 Perancangan *Validation Testing* Menambah Data Pendaftaran Baru

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

Tabel 7.2 Rancangan Skenario Testing Menambah Data Pendaftaran Baru

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

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

Tabel 7.3 Rancangan *Validation Testing* Menambah Data Pendaftaran Baru

Kode Pengujian	P-VT-01
Nama Kasus Uji	Kasus uji menambah data pendaftaran baru
Kode Skenario	Skenario 1
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menyimpan data pendaftaran baru.
Test Case	Pengujian dengan memasukan data sesuai format yang ditentukan pada formulir pendaftaran.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengakses halaman pendaftaran baru pasien TKLN atau TKDN. 2. Penguji mengisi formulir pendaftaran sesuai format. 3. Penguji menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan pesan data berhasil ditambahkan dan menampilkan halaman daftar pasien harian bersama data yang baru ditambahkan.

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

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

Kode Pengujian	P-VT-02
Nama Kasus Uji	Kasus uji Inputan Data Pendaftaran Baru Tidak Sesuai
Kode Skenario	Skenario 2
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menampilkan kesalahan inputan data yang tidak sesuai format.
Test Case	Pengujian dengan mengosongkan nama lengkap pada formulir dan memberi inputan <i>passport</i> 100 karakter random.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengakses halaman pendaftaran baru pasien TKLN atau TKDN. 2. Penguji mengisi formulir pendaftaran sesuai format kecuali dengan mengosongkan nama lengkap pada formulir dan memberi inputan <i>passport</i> 100 karakter random. 3. Penguji menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan pesan nama lengkap harus diisi dan maksimal inputan <i>passport</i> 25 karakter.

7.1.1.2 Perancangan *Validation Testing* Menambah Data Pemeriksaan Baru

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

Tabel 7.5 Rancangan Skenario Testing Menambah Data Pemeriksaan Baru

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

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

Tabel 7.6 Rancangan *Validation Testing* Menambah Data Pemeriksaan Baru

Kode Pengujian	P-VT-03
Nama Kasus Uji	Kasus uji menambah data Pemeriksaan baru
Kode Skenario	Skenario 1

Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menyimpan data Pemeriksaan baru.
Test Case	Pengujian dengan memasukan data sesuai format yang ditentukan pada formulir pemeriksaan baru.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengakses halaman pemeriksaan baru pasien TKLN atau TKDN. 2. Penguji mengisi formulir pemeriksaan baru sesuai format. 3. Penguji menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan pesan data berhasil ditambahkan dan menampilkan halaman data pendaftaran bersama data yang baru ditambahkan.

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

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

Kode Pengujian	P-VT-04
Nama Kasus Uji	Kasus uji Inputan Data Pendaftaran Baru Tidak Sesuai
Kode Skenario	Skenario 2
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menampilkan kesalahan inputan data yang tidak sesuai format.
Test Case	Pengujian dengan mengosongkan tanggal pemeriksaan pada formulir dan memberi inputan <i>passport</i> 100 karakter random.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengakses halaman pendaftaran baru pasien TKLN atau TKDN. 2. Penguji mengisi formulir pemeriksaan baru sesuai format kecuali dengan mengosongkan tanggal pemeriksaan pada formulir dan memberi inputan <i>passport</i> 100 karakter random. 3. Penguji menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan pesan tanggal pemeriksaan harus diisi dan maksimal inputan <i>passport</i> 25 karakter.

7.1.1.3 Perancangan *Validation Testing* Mengubah Data Hasil Akhir Pemeriksaan

Perancangan *Validation Testing* ini menjelaskan rancangan pengujian pada beberapa fungsi yang dapat digunakan pengguna untuk Mengubah Data Hasil Akhir Pemeriksaan melalui sistem. Tabel 7.8 menunjukkan beberapa skenario yang

dimiliki alur *use case* Mengubah Data Hasil Akhir Pemeriksaan. Skenario yang ditunjukkan pada Tabel 7.8 akan digunakan untuk mengidentifikasi kasus uji.

Tabel 7.8 Rancangan Skenario Testing Mengubah Data Hasil Akhir Pemeriksaan

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

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

Tabel 7.9 Rancangan *Validation Testing* Mengubah Data Hasil Akhir Pemeriksaan

Kode Pengujian	P-VT-05
Nama Kasus Uji	Kasus uji Mengubah Data Hasil Akhir Pemeriksaan
Kode Skenario	Skenario 1
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat mengubah data hasil akhir pemeriksaan FIT/UNFIT.
Test Case	Pengujian dengan menekan tombol FIT/UNFIT pada pemeriksaan pasien yang seluruh data hasil pemeriksaannya telah terisi.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengakses halaman pemeriksaan pasien TKLN atau TKDN. 2. Penguji menekan tombol FIT/UNFIT.
Hasil yang diharapkan	Sistem menampilkan pesan data berhasil diperbarui dan menampilkan data hasil pilihan FIT/UNIFT.

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

Tabel 7.10 Rancangan *Validation Testing* Mengubah Data Hasil Akhir Pemeriksaan Belum Lengkap

Kode Pengujian	P-VT-06
Nama Kasus Uji	Kasus uji Mengubah Data Hasil Akhir Pemeriksaan Belum Lengkap
Kode Skenario	Skenario 2
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menampilkan pesan bahwa hasil pemeriksaan belum lengkap terisi.
Test Case	Pengujian dengan menekan tombol FIT/UNFIT pada pemeriksaan pasien yang data hasil pemeriksaannya terisi sebagian.

Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengakses halaman pemeriksaan pasien TKLN atau TKDN. 2. Penguji menekan tombol FIT/UNFIT.
Hasil yang diharapkan	Sistem menampilkan pesan data hasil pemeriksaan belum lengkap.

7.1.1.4 Perancangan *Validation Testing* Mengisi Data Pemeriksaan Fisik dan Kejiwaan

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

Tabel 7.11 Rancangan Skenario Testing Mengisi Data Pemeriksaan Fisik dan Kejiwaan

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

Tabel 7.12 menampilkan rancangan pengujian *Validation Testing* Mengisi Data Pemeriksaan Fisik dan Kejiwaan dengan kode P-VT-07. Kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menyimpan data inputan hasil pemeriksaan pasien.

Tabel 7.12 Rancangan Skenario Testing Mengisi Data Pemeriksaan Fisik dan Kejiwaan

Kode Pengujian	P-VT-07
Nama Kasus Uji	Kasus uji Mengisi Data Pemeriksaan Fisik dan Kejiwaan
Kode Skenario	Skenario 1
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menyimpan data hasil pemeriksaan fisik dan kejiwaan.
Test Case	Pengujian dengan memasukan data sesuai format yang ditentukan pada halaman data hasil pemeriksaan fisik dan kejiwaan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengakses halaman hasil pemeriksaan fisik dan kejiwaan. 2. Penguji mengisi hasil pemeriksaan sesuai format. 3. Penguji menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan pesan data berhasil terisi dan menampilkan halaman daftar pasien harian.

Tabel 7.13 menampilkan rancangan pengujian *Validation Testing* Mengubah Data Hasil Akhir Pemeriksaan Belum Lengkap dengan kode P-VT-06. Kasus uji ini dilakukan untuk memastikan bahwa sistem dapat menampilkan pemberitahuan bahwa pemeriksaan belum lengkap ketika menekan tombol hasil akhir pemeriksaan.

Tabel 7.13 Rancangan *Validation Testing* Inputan Data Pemeriksaan Fisik dan Kejiwaan Tidak Sesuai

Kode Pengujian	P-VT-08
Nama Kasus Uji	Kasus uji Inputan Data Pemeriksaan Fisik dan Kejiwaan Tidak Sesuai
Kode Skenario	Skenario 2
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menampilkan kesalahan inputan data yang tidak sesuai format.
Test Case	Pengujian dengan mengosongkan berat badan pada formulir dan memberi inputan <i>text</i> pada tinggi badan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengakses halaman hasil pemeriksaan fisik dan kejiwaan 2. Penguji mengisi formulir pemeriksaan baru sesuai format kecuali dengan mengosongkan berat badan pada formulir dan memberi inputan <i>text</i> pada tinggi badan. 3. Penguji menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan pesan berat badan harus diisi dan inputan tinggi badan harus berupa angka.

7.1.2 Hasil Pengujian *Validation Testing*

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

Tabel 7.14 Hasil Pengujian *Validation Testing*

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

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

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

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

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

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

7.1.3 Kesimpulan *Validation Testing*

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





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

7.2 Pengujian *Compatibility Testing*

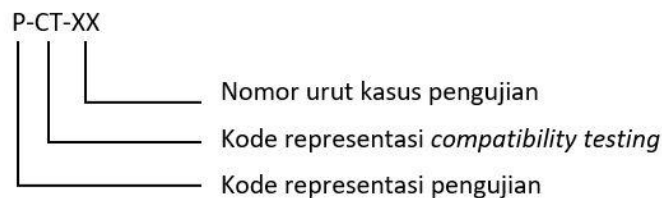
7.2.1 Perancangan Pengujian *Compatibility Testing*

Pengujian *Compatibility Testing* dilakukan untuk mengetahui kesesuaian sistem yang akan dikembangkan dengan beberapa aplikasi peramban. Dalam pengujian ini digunakan aplikasi perangkat lunak Sortsite versi 5.30.822.0 yang akan memeriksa komponen tag HTML, CSS, script, kesalahan dalam pemanggilan halaman, teknologi yang tidak didukung oleh *browser* tertentu. Terdapat beberapa kategori masalah untuk mengetahui tingkat kompatibilitas sistem sesuai dengan yang dijelaskan dalam tabel 7.15.

Tabel 7.15 Kategori Masalah Kompatilitas

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

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



Gambar 7.2 Pengkodean Uji *Compatibility Testing*

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

tiap jenis dan versi peramban untuk melakukan pengujian ini yang ditujukan pada tabel-tabel dibawah ini. Peramban yang dilakukan pengujian adalah *internet explorer, edge, firefox, safari, opera, chrome, iOS, dan Android*.

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

Kode Pengujian	P-CT-01
Nama Kasus Uji	Kasus uji kompabiitas pada peramban <i>internet explorer</i>
Tujuan Pengujian	Pengujian dilakukan untuk mengetahui apakah sistem dapat berjalan baik pada peramban <i>internet explorer</i>
Test Case	Menguji kompabilitas sistem terhadap peramban <i>Internet Explorer</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji menjalankan aplikasi SortSite 2. Penguji memasukkan url sistem pada menu <i>bar address</i> 3. Penguji melakukan checking dengan menekan tombol check 4. Penguji melihat hasil <i>compability testing</i>
Hasil yang diharapkan	Sistem dapat berjalan dengan baik pada peramban <i>Internet Explorer</i>

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

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

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

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

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

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

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

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

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



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

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

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

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

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

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

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

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

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

Tabel 7.24 Tampilan Hasil Pengujian *Compatibility*

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

7.2.2 Hasil Pengujian *Compatibility Testing*

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

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



Gambar 7.3 Hasil *Compatibility Testing*

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

7.2.3 Kesimpulan Pengujian *Compatibility Testing*

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

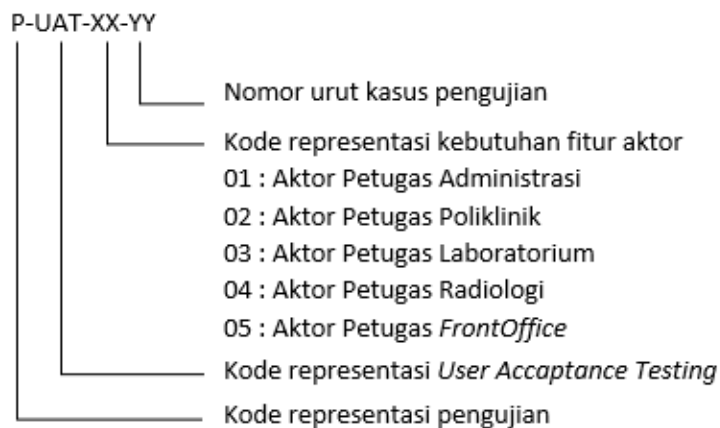
7.3 Kesimpulan Pengujian Fase Konstruksi Metode *Rational Unifeld Process*

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

7.4 Pengujian *User Acceptance Testing (UAT)*

7.4.1 Perancangan Pengujian *User Acceptance Testing (UAT)*

Subbab ini berisi rancangan pengujian user acceptance testing (UAT) sebagai landasan dalam menerapkan pengujian setelah sistem sudah dibangun. *User Acceptance Testing* dilakukan untuk memastikan bahwa sistem yang sudah dikembangkan dapat diterima dengan baik berdasarkan kriteria yang sudah ditentukan pengguna. Dalam melakukan pengujian UAT, terdapat pemberian kode sebagai penanda kasus uji yang dirancang. Ketentuan pemberian kode pengujian ditunjukkan dalam Gambar 7.4.



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

Pengujian yang dilakukan dengan mendemonstrasikan program kepada aktor pengguna sistem. Kemudian penguji menyerahkan kuisioner untuk mengetahui penilaian dari responden terhadap sistem yang dikembangkan. Responden pada pengujian diambil dari petugas administrasi, petugas poliklinik, petugas laboratorium, dan petugas radiologi. Terdapat 4 kriteria pertanyaan yang digunakan dalam menilai sistem yaitu *performance*, *usability*, *functional correctness and completeness*, and *Confidentially and Availability* untuk memastikan bahwa keluaran pada sistem telah sesuai. Berikut adalah uji kasus untuk validasi sistem berdasarkan fungsionalitas yang dimiliki tiap aktor dibedakan menjadi lima tabel yaitu fungsionalitas untuk petugas administrasi, petugas poliklinik, petugas laboratorium, petugas radiologi, dan petugas *frontoffice*. Setiap tabel disajikan informasi kriteria, kode kasus uji, dan daftar pertanyaan.

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

No	Kriteria	Kode Kasus Uji	Daftar Pertanyaan
1	<i>Performance</i>	P-UAT-05-01	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat diakses dengan cepat ?
		P-UAT-05-02	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat mempercepat pendaftaran pasien pemeriksaan kesehatan tenaga kerja ?
		P-UAT-05-03	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat mempercepat pengguna untuk mendapatkan informasi data pendaftaran pasien ?
2	<i>Usability</i>	P-UAT-05-04	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mudah digunakan ?

		P-UAT-05-05	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mempermudah pendaftaran pasien pemeriksaan kesehatan tenaga kerja?
		P-UAT-05-06	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mempermudah pengguna untuk mendapatkan informasi data pendaftaran pasien ?
3	<i>Functional Correctness and Completeness</i>	P-UAT-05-07	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam melihat seluruh data pendaftaran pasien ?
		P-UAT-05-08	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu mencari data pendaftaran pasien tertentu ?
		P-UAT-05-09	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam pendaftaran pasien pemeriksaan kesehatan tenaga kerja ?
4	<i>Confidentially and Availability</i>	P-UAT-05-10	Apakah login pengguna pada Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu keamanan informasi data pendaftaran pasien ?

Tabel 7.26 Rancangan Kasus Uji UAT untuk Petugas Poliklinik

No	Kriteria	Kode Kasus Uji	Daftar Pertanyaan
1	<i>Performance</i>	P-UAT-02-01	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat diakses dengan cepat ?
		P-UAT-02-02	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat mempercepat pengguna untuk mendapatkan informasi pemeriksaan kesehatan fisik dan kejiwaan pasien ?
2	<i>Usability</i>	P-UAT-02-03	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mudah digunakan ?
		P-UAT-02-04	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mempermudah pengelolaan data hasil pemeriksaan fisik dan kejiwaan pasien ?

3	<i>Functional Correctness and Completeness</i>	P-UAT-02-05	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam penyajian informasi permintaan pemeriksaan kesehatan fisik dan kejiwaan pasien ?
		P-UAT-02-06	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam mengelola data hasil pemeriksaan fisik dan kejiwaan pasien ?
4	<i>Confidentially and Availability</i>	P-UAT-02-07	Apakah login pengguna pada Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu keamanan informasi data pemeriksaan fisik dan kejiwaan pasien pemeriksaan kesehatan tenaga kerja ?

Tabel 7.27 Rancangan Kasus Uji UAT untuk Petugas Laboratorium

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

No	Kriteria	Kode Kasus Uji	Daftar Pertanyaan
4	<i>Confidentially and Availability</i>	P-UAT-03-07	Apakah login pengguna pada Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu keamanan informasi data pemeriksaan laboratorium pasien pemeriksaan kesehatan tenaga kerja ?

Tabel 7.28 Rancangan Kasus Uji UAT untuk Petugas Radiologi

No	Kriteria	Kode Kasus Uji	Daftar Pertanyaan
1	<i>Performance</i>	P-UAT-04-01	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat diakses dengan cepat ?
		P-UAT-04-02	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat mempercepat pengguna untuk mendapatkan informasi pemeriksaan kesehatan radiologi pasien ?
2	<i>Usability</i>	P-UAT-04-03	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mudah digunakan ?
		P-UAT-04-04	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mempermudah pengelolaan data hasil pemeriksaan radiologi pasien ?
3	<i>Functional Correctness and Completeness</i>	P-UAT-04-05	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam penyajian informasi permintaan pemeriksaan kesehatan radiologi pasien ?
		P-UAT-04-06	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam mengelola data hasil pemeriksaan radiologi pasien ?
4	<i>Confidentially and Availability</i>	P-UAT-04-07	Apakah login pengguna pada Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu keamanan informasi data pemeriksaan radiologi pasien pemeriksaan kesehatan tenaga kerja ?



Tabel 7.29 Rancangan Kasus Uji UAT untuk Petugas Administrasi

No	Kriteria	Kode Kasus Uji	Daftar Pertanyaan
1	<i>Performance</i>	P-UAT-01-01	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat diakses dengan cepat ?
		P-UAT-01-02	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat mempercepat pengguna untuk mendapatkan informasi pemeriksaan progres pengerjaan data hasil pemeriksaan pasien kesehatan tenaga kerja ?
2	<i>Usability</i>	P-UAT-01-03	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mudah digunakan ?
		P-UAT-01-04	Apakah informasi progres pengerjaan data hasil pemeriksaan pasien mempermudah pengguna mendapatkan informasi progres pengerjaan data hasil pemeriksaan pasien kesehatan tenaga kerja?
		P-UAT-01-05	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mempermudah pengguna mengelola seluruh data pemeriksaan pasien pemeriksaan kesehatan tenaga kerja ?
3	<i>Functional Correctness and Completeness</i>	P-UAT-01-06	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam mengelola data seluruh pemeriksaan pasien pemeriksaan kesehatan tenaga kerja ?
		P-UAT-01-07	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam melihat progress pengerjaan disetiap pemeriksaan kesehatan tenaga kerja ?
4	<i>Confidentially and Availability</i>	P-UAT-01-08	Apakah login pengguna pada Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu keamanan informasi data pemeriksaan pasien pemeriksaan kesehatan tenaga kerja ?

7.4.2 Hasil Pengujian *User Acceptance Testing* (UAT)

User acceptance testing dilakukan berdasarkan perancangan pengujian yang dibuat sebelumnya kepada setiap kasus uji yang dibuat. Terdapat lima rancangan *user acceptance testing* yang sudah didefinisikan untuk lima aktor yang berbeda. Pengujian ini melibatkan tujuh responden yaitu satu orang petugas *front office*, dua orang petugas poliklinik, dua orang petugas laboratorium, satu orang petugas radiologi dan satu orang petugas administrasi. Berikut adalah hasil dari *user acceptance testing* sistem informasi yang dikembangkan.

7.4.2.1 Hasil Pengujian *User Acceptance Testing* untuk Petugas *Front Office*

Subbab ini merupakan hasil kuisioner *User Acceptance Testing* untuk Petugas *Front Office* yang berisi pertanyaan, kode uji, dan jawaban. Responden dalam *User Acceptance Testing* ini adalah satu petugas *front office*. Hasil kuisioner *User Acceptance Testing* untuk Petugas *Front Office* terdapat pada Tabel 7.30.

Tabel 7.30 Hasil Pengujian *User Acceptance Testing* untuk Petugas *Front Office*

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
1	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat diakses dengan cepat ?	P-UAT-05-01	1	0	0	0	0
2	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat mempercepat pendaftaran pasien pemeriksaan kesehatan tenaga kerja ?	P-UAT-05-02	0	1	0	0	0
3	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat mempercepat pengguna untuk mendapatkan informasi data pendaftaran pasien ?	P-UAT-05-03	0	1	0	0	0
4	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mudah digunakan ?	P-UAT-05-04	1	0	0	0	0
5	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mempermudah pendaftaran	P-UAT-05-05	0	1	0	0	0

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
	pasien pemeriksaan kesehatan tenaga kerja?						
6	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mempermudah pengguna untuk mendapatkan informasi data pendaftaran pasien ?	P-UAT-05-06	0	1	0	0	0
7	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam melihat seluruh data pendaftaran pasien ?	P-UAT-05-07	0	1	0	0	0
8	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu mencari data pendaftaran pasien tertentu ?	P-UAT-05-08	0	1	0	0	0
9	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam pendaftaran pasien pemeriksaan kesehatan tenaga kerja ?	P-UAT-05-09	0	1	0	0	0
10	Apakah login pengguna pada Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu keamanan informasi data pendaftaran pasien ?	P-UAT-05-10	0	0	1	0	0

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

- a. Menghitung total nilai berdasarkan jawaban dari satu responden petugas *front office*.

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

Jawaban	Bobot nilai	Jumlah jawaban	Hasil
Sangat Setuju (A)	5	2	10
Setuju (B)	4	7	28
Netral (C)	3	1	3
Tidak Setuju (D)	2	0	0
Sangat Tidak Setuju (E)	1	0	0
Total			41

b. Nilai Y.

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

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

c. Nilai Presentase.

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

$$\text{Rumus Index} = (41 \div 50) \times 100\% = 82\% \text{ (Sangat Setuju)}$$

7.4.2.2 Hasil Pengujian *User Acceptance Testing* untuk Petugas Poliklinik

Subbab ini merupakan hasil kuisioner *User Acceptance Testing* untuk Petugas Poliklinik yang berisi pertanyaan, kode uji, dan jawaban. Responden dalam *User Acceptance Testing* ini adalah dua petugas Poliklinik. Hasil kuisioner *User Acceptance Testing* untuk Petugas Poliklinik terdapat pada Tabel 7.32.

Tabel 7.32 Hasil Pengujian *User Acceptance Testing* untuk Petugas Poliklinik

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
1	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat diakses dengan cepat ?	P-UAT-02-01	0	2	0	0	0
2	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat mempercepat pengguna untuk mendapatkan informasi pemeriksaan kesehatan fisik dan kejiwaan pasien ?	P-UAT-02-02	0	2	0	0	0
3	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mudah digunakan ?	P-UAT-02-03	0	1	1	0	0

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
4	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mempermudah pengelolaan data hasil pemeriksaan fisik dan kejiwaan pasien ?	P-UAT-02-04	0	1	1	0	0
5	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam penyajian informasi permintaan pemeriksaan kesehatan fisik dan kejiwaan pasien ?	P-UAT-02-05	0	2	0	0	0
6	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam mengelola data hasil pemeriksaan fisik dan kejiwaan pasien ?	P-UAT-02-06	0	1	1	0	0
7	Apakah login pengguna pada Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu keamanan informasi data pemeriksaan fisik dan kejiwaan pasien pemeriksaan kesehatan tenaga kerja ?	P-UAT-02-07	0	1	1	0	0

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

- a. Menghitung total nilai berdasarkan jawaban dari dua responden petugas poliklinik.

Tabel 7.33 Hasil Pengujian *User Acceptance Testing* untuk Petugas Poliklinik

Jawaban	Bobot nilai	Jumlah jawaban	Hasil
Sangat Setuju (A)	5	0	0
Setuju (B)	4	10	40

Jawaban	Bobot nilai	Jumlah jawaban	Hasil
Netral (C)	3	4	12
Tidak Setuju (D)	2	0	0
Sangat Tidak Setuju (E)	1	0	0
Total			52

b. Nilai Y.

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

$$Y = 5 \times 2 \times 7 = 70$$

c. Nilai Presentase.

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

$$\text{Rumus Index} = (52 \div 70) \times 100\% = 74\% \text{ (Setuju)}$$

7.4.2.3 Hasil Pengujian *User Acceptance Testing* untuk Petugas Laboratorium

Subbab ini merupakan hasil kuisioner *User Acceptance Testing* untuk Petugas Laboratorium yang berisi pertanyaan, kode uji, dan jawaban. Responden dalam *User Acceptance Testing* ini adalah dua petugas Laboratorium. Hasil kuisioner *User Acceptance Testing* untuk Petugas Laboratorium terdapat pada Tabel 7.34.

Tabel 7.34 Hasil Pengujian *User Acceptance Testing* untuk Petugas Laboratorium

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
1	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat diakses dengan cepat ?	P-UAT-03-01	1	1	0	0	0
2	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat mempercepat pengguna untuk mendapatkan informasi pemeriksaan kesehatan laboratorium pasien ?	P-UAT-03-02	0	2	0	0	0
3	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mudah digunakan ?	P-UAT-03-03	1	0	1	0	0
4	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mempermudah pengelolaan	P-UAT-03-04	1	0	1	0	0



No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
	data hasil pemeriksaan laboratorium pasien ?						
5	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam penyajian informasi permintaan pemeriksaan kesehatan laboratorium pasien ?	P-UAT-03-05	0	1	1	0	0
6	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam mengelola data hasil pemeriksaan laboratorium pasien ?	P-UAT-03-06	0	1	1	0	0
7	Apakah login pengguna pada Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu keamanan informasi data pemeriksaan laboratorium pasien pemeriksaan kesehatan tenaga kerja ?	P-UAT-03-07	0	1	1	0	0

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

- a. Menghitung total nilai berdasarkan jawaban dari dua responden petugas laboratorium.

Tabel 7.35 Hasil Pengujian *User Acceptance Testing* untuk Petugas Laboratorium

Jawaban	Bobot nilai	Jumlah jawaban	Hasil
Sangat Setuju (A)	5	3	15
Setuju (B)	4	6	24
Netral (C)	3	5	15
Tidak Setuju (D)	2	0	0

Jawaban	Bobot nilai	Jumlah jawaban	Hasil
Sangat Tidak Setuju (E)	1	0	0
Total			54

b. Nilai Y.

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

$Y = 5 \times 2 \times 7 = 70$

c. Nilai Presentase.

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

$\text{Rumus Index} = (54 \div 70) \times 100\% = 77\% \text{ (Setuju)}$

7.4.2.4 Hasil Pengujian *User Acceptance Testing* untuk Petugas Radiologi

Subbab ini merupakan hasil kuisioner *User Acceptance Testing* untuk Petugas Radiologi yang berisi pertanyaan, kode uji, dan jawaban. Responden dalam *User Acceptance Testing* ini adalah satu petugas Radiologi. Hasil kuisioner *User Acceptance Testing* untuk Petugas Radiologi terdapat pada Tabel 7.36.

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

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
1	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat diakses dengan cepat ?	P-UAT-04-01	1	0	0	0	0
2	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat mempercepat pengguna untuk mendapatkan informasi pemeriksaan kesehatan radiologi pasien ?	P-UAT-04-02	1	0	0	0	0
3	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mudah digunakan ?	P-UAT-04-03	0	1	0	0	0
4	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mempermudah pengelolaan data hasil pemeriksaan radiologi pasien ?	P-UAT-04-04	0	1	0	0	0



No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
5	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam penyajian informasi permintaan pemeriksaan kesehatan radiologi pasien ?	P-UAT-04-05	0	1	0	0	0
6	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam mengelola data hasil pemeriksaan radiologi pasien ?	P-UAT-04-06	0	1	0	0	0
7	Apakah login pengguna pada Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu keamanan informasi data pemeriksaan radiologi pasien pemeriksaan kesehatan tenaga kerja ?	P-UAT-04-07	0	1	0	0	0

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

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

Tabel 7.37 Hasil Pengujian *User Acceptance Testing* untuk Petugas Radiologi

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

- b. Nilai Y.

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

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

c. Nilai Presentase.

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

Rumus Index = $(30 \div 35) \times 100\% = 85\%$ (Sangat Setuju)

7.4.2.5 Hasil Pengujian *User Acceptance Testing* untuk Petugas Administrasi

Subbab ini merupakan hasil kuisioner *User Acceptance Testing* untuk Petugas Administrasi yang berisi pertanyaan, kode uji, dan jawaban. Responden dalam *User Acceptance Testing* ini adalah satu petugas Administrasi. Hasil kuisioner *User Acceptance Testing* untuk Petugas Administrasi terdapat pada Tabel 7.38.

Tabel 7.38 Hasil Pengujian *User Acceptance Testing* untuk Petugas Administrasi

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
1	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat diakses dengan cepat ?	P-UAT-01-01	1	0	0	0	0
2	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat mempercepat pengguna untuk mendapatkan informasi pemeriksaan progres pengerjaan data hasil pemeriksaan pasien kesehatan tenaga kerja ?	P-UAT-01-02	1	0	0	0	0
3	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja mudah digunakan ?	P-UAT-01-03	0	1	0	0	0
4	Apakah informasi progres pengerjaan data hasil pemeriksaan pasien mempermudah pengguna mendapatkan informasi progres pengerjaan data hasil pemeriksaan pasien kesehatan tenaga kerja?	P-UAT-01-04	0	1	0	0	0
5	Apakah Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja	P-UAT-01-05	0	1	0	0	0

No	Pertanyaan	Kode Uji	Jawaban				
			SS	S	N	TS	STS
	mempermudah pengguna mengelola seluruh data pemeriksaan pasien pemeriksaan kesehatan tenaga kerja ?						
6	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam mengelola data seluruh pemeriksaan pasien pemeriksaan kesehatan tenaga kerja ?	P-UAT-01-06	0	1	0	0	0
7	Apakah dengan adanya Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu dalam melihat progress pengerjaan disetiap pemeriksaan kesehatan tenaga kerja ?	P-UAT-01-07	0	1	0	0	0
8	Apakah login pengguna pada Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja membantu keamanan informasi data pemeriksaan pasien pemeriksaan kesehatan tenaga kerja ?	P-UAT-01-08	0	1	0	0	0

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

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

Tabel 7.39 Hasil Pengujian *User Acceptance Testing* untuk Petugas Administrasi

Jawaban	Bobot nilai	Jumlah jawaban	Hasil
Sangat Setuju (A)	5	2	10
Setuju (B)	4	6	24



Jawaban	Bobot nilai	Jumlah jawaban	Hasil
Netral (C)	3	0	0
Tidak Setuju (D)	2	0	0
Sangat Tidak Setuju (E)	1	0	0
Total			34

b. Nilai Y.

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

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

c. Nilai Presentase.

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

Rumus Index = $(34 \div 40) \times 100\% = 85\%$ (Sangat Setuju)

7.4.3 Kesimpulan Pengujian *User Acceptance Testing* (UAT)

Dari hasil analisis menggunakan skala *likert* maka dapat diambil kesimpulan bahwa untuk pengujian *User Acceptance Testing* untuk staf petugas *front office* memperoleh presentase nilai sebesar 82% yang artinya petugas *front office* sangat setuju. Kemudian untuk pengujian *User Acceptance Testing* untuk petugas poliklinik memperoleh presentase nilai sebesar 74% yang artinya petugas poliklinik setuju. Selanjutnya untuk pengujian *User Acceptance Testing* untuk petugas laboratorium memperoleh presentase nilai sebesar 77% yang artinya petugas laboratorium setuju. Pengujian *User Acceptance Testing* untuk petugas radiologi memperoleh presentase nilai sebesar 85% yang artinya petugas radiologi sangat setuju. Pengujian *User Acceptance Testing* untuk petugas administrasi memperoleh presentase nilai sebesar 85% yang artinya petugas administrasi sangat setuju. Berdasarkan presentase pengujian yang dilakukan dengan *User Acceptance Testing* pada beberapa responden menunjukkan sistem informasi layanan pemeriksaan kesehatan tenaga kerja dapat diterima baik oleh masing-masing pengguna

7.5 Kesimpulan Pengujian Fase Transisi Metode *Rational Unifield Process*

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

BAB 8 PENUTUP

8.1 Kesimpulan

Berdasarkan penelitian yang sudah dilakukan, dapat diambil kesimpulan bahwa Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja dapat dikembangkan melalui beberapa kegiatan pengembangan sistem dengan hasil sebagai berikut :

1. Fase insepisi dilakukan analisis proses bisnis dengan melakukan analisis proses bisnis saat ini (*as-is*) yang menghasilkan proses bisnis perbaikan (*to-be*). Diperoleh empat proses bisnis *to-be* yaitu pendaftaran pasien, pemeriksaan fisik dan kejiwaan, pemeriksaan laboratorium, dan pemeriksaan radiologi. Empat proses bisnis *to-be* tersebut dipermudah dengan adanya bantuan dari sistem yang dikembangkan. Analisis persyaratan kemudian dilakukan sehingga didapat 9 fitur, 25 kebutuhan persyaratan fungsional dan 1 kebutuhan persyaratan nonfungsional. Dari hasil spesifikasi kebutuhan tersebut akhirnya menghasilkan 22 *use case* yang menggambarkan interaksi antara aktor dan sistem. Kebutuhan analisis persyaratan yang dilakukan telah setuju oleh pemangku kepentingan. *Lifecycle objective milestone* fase insepisi pada metode RUP telah terpenuhi.
2. Fase elaborasi rancangan dilakukan berdasarkan hasil analisis persyaratan yang sudah dilakukan pada fase insepisi. Analisis persyaratan menghasilkan model interaksi objek yang divisualisasikan ke dalam *sequence diagram*, model objek divisualisasikan dalam *class diagram* dan beberapa rancangan yaitu *physical data model*, algoritme, dan antarmuka sistem. Sehingga syarat yang harus terpenuhi pada fase elaborasi yaitu deskripsi arsitektur sistem sudah dilakukan, maka *lifecycle architecture milestone* fase elaborasi pada metode RUP telah terpenuhi.
3. Fase konstruksi implementasi sistem dibuat berdasarkan perancangan pada fase elaborasi. Implemententasi dikembangkan dengan kerangka kerja Laravel. Pada fase kontruksi pengujian sistem dilakukan dengan pengujian *validation testing* dengan 8 kasus uji yang sudah didefinisikan berasal dari empat *use case scenario* yang berbeda diantaranya tambah data pendaftaran baru, tambah data pemeriksaan baru, mengubah data hasil akhir pemeriksaan, dan mengisi data hasil pemeriksaan fisik dan kejiwaan menunjukkan hasil status valid dengan presentase 100%. Pengujian nonfungsional dengan *compatibility testing* didapat hasil 2 *critical issues* disebabkan tidak diaktifkannya fungsi SSL (*secure socket layer*). Berdasarkan hasil pengujian menunjukkan bahwa sistem yang diuji telah sesuai dengan kebutuhan persyaratan fungsional dan non fungsional yang telah diidentifikasi. *Initial operational capability milestone* fase konstruksi pada metode RUP telah terpenuhi.

4. Fase transisi pengujian *user acceptance testing* dilakukan kepada 7 responden mewakili tiap aktor yang terdiri dari 1 petugas *front office*, 2 petugas poliklinik, 2 petugas laboratorium, 1 petugas radiologi, dan 1 petugas administrasi. Hasil pengujian *user acceptance testing* didapat presentase sebanyak 82% untuk aktor petugas *front office*, 74% untuk aktor petugas poliklinik, 77% untuk aktor petugas laboratorium, 85% untuk aktor petugas radiologi, dan 85% untuk aktor petugas administrasi. Berdasarkan presentase pengujian yang sudah didapat menunjukkan semua pihak atau aktor yang secara langsung menggunakan sistem informasi layanan pemeriksaan kesehatan tenaga kerja setuju dengan adanya sistem informasi yang sudah dikembangkan maka *release product milestone* fase transisi pada metode RUP telah terpenuhi.

8.2 Saran

Saran yang dapat diberikan sebagai bahan pertimbangan untuk melaksanakan pengembangan lebih lanjut pada Sistem Layanan Pemeriksaan Kesehatan Tenaga Kerja diantaranya adalah :

1. Bagi peneliti selanjutnya dapat melakukan penelitian tingkat kenyamanan dalam penggunaan sistem informasi layanan pemeriksaan kesehatan tenaga kerja khususnya antarmuka pengguna.
2. Perlu dilakukan pengembangan sistem informasi layanan lain yang ada di Klinik Argaraya Medika Malang selain layanan pemeriksaan kesehatan tenaga kerja untuk meningkatkan kualitas layanan secara keseluruhan.

DAFTAR PUSTAKA

- A.S, R. & Shalahudin, M., 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Al Faruq, D. U. (2018). Malang, Kota Penyumbang Terbesar Pekerja Migran. *jatim.metrotvnews.com* 3 Desember 2018. 12 Desember 2018 <http://jatim.metrotvnews.com/peristiwa/Rb13582K-malang-kota-penyumbang-terbesar-pekerja-migran>.
- Aggarwal, V., 2002. The Application of the Unified Modeling Language in ObjectOriented Analysis of Healthcare Information Systems. *Journal of Medical Systems*, 26(5), pp. 383-397.
- Anwar, A. (2014). A Review of RUP (Rational Unified Process). *International Journal of Software Engineering*, 5(2), 8–24. Retrieved from <http://www.cscjournals.org/library/manuscriptinfo.php?mc=IJSE-142>
- Becker, J., Kugeler, M. & Rosemann, M., 2013. *Process Management: A Guide for the Design of Business Processes*. Berlin: Springer-Verlag.
- Bittner, K. & Spence, I., 2002. *Use case Modelling*. San Fransisco, Addison-Wesley Professional.
- Booch, G., Rumbaugh, J. & Jacobson, I., 1999. *The Unified Modeling Language User Guide*. 1st ed. s.l.:Addison Wesley.
- David Kung, J. L., 2016. *An Object-Oriented Analysis and Design Environment*. Dallas, TX, USA, IEEE, pp. 92-100.
- Dwi Puspita Sari, S. J. P. E. R., 2014. *The Development of Project Monitoring Information System (Case Study: PT Tetapundi Prima Kelola)*. South Tangerang, Indonesia, IEEE, pp. 39-43.
- Goel, R. & Gupta, D. N., 2014. Survey on Acceptance Testing Technique. *International Journal of Software and Web Sciences*, VIII(1), pp. 20-23.
- Kumar, M., Singh, S. K. & Dwivedi, D. R. K., 2015. A Comparative Study of Black Box Testing and White Box Testing Techniques. *International Journal of Advance Research in Computer Science and Management Studies*, 3(10), pp. 32-44.
- Leffingwell, D. & Widrig, D., 2003. *Managing Software Requirements: A Use case Approach*. 2nd penyunt. Boston: Addison Wesley.
- Mohamed A. Ben-Zahia, I. J., 2014. *Criteria for Selecting Software Development*. Sousse, IEEE.
- Monk, E. & Wagner, B., 2013. *Concepts in Enterprise Resource Planning*. 4th penyunt. Boston: Course Technology.
- Mulyani, Widiana, dan Bambang Eka P. (2015). *Pembangunan Sistem Informasi Data Balita Pada Posyandu Desa Ploso Kecamatan Punung Kabupaten*

- Pacitan*. Journal Speed – Sentra Penelitian Engineering dan Edukasi. Vol. 7 No. 2.
- Mustaqbal, M. S., Firdaus, R. F. & Rahmadi, H., 2015. PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN). *Jurnal Ilmiah Teknologi Informasi Terapan*, 1(3), pp. 31-36.
- Naik, K., 2008. *SOFTWARE TESTING AND QUALITY ASSURANCE Theory and Practice*. Canada: John Wiley & Sons, Inc.
- Object Management Group, 2011. *The Object Management Group® (OMG®)*. [Online]
Available at: <http://www.omg.org/spec/BPMN/2.0/PDF>
[Accessed 12 12 2018].
- Pressman, R. S., 2010. *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. 7th ed. New York: McGraw-Hill.
- Rosa AS, M. S. 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: INFORMATIKA.
- Sommerville, I., 2011. *SOFTWARE ENGINEERING*. 9 ed. Boston: Pearson Education,.
- Sugiyono, 2015. *Metode Penelitian Pendidikan (Pendekatan Kuantitatif, Kualitatif, dan R&D)*. 21th ed. Bandung: Alfabeta.
- Sulistiyorini, P., 2009. Pemodelan Visual dengan Menggunakan UML dan Rational Rose. *Jurnal Teknologi Informasi DINAMIK*, pp. Volume XIV, No.1, Hal. 23.
- Weske, M., 2012. *Business Process Management : Concepts, Languages, Architectures*. Berlin: Springer.