

**PENGEMBANGAN SISTEM INFORMASI MANAJEMEN
PERPUSTAKAAN PADA RUANG BACA FAKULTAS ILMU
KOMPUTER UNIVERSITAS BRAWIJAYA**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

I Gde Yuda Pratama

NIM: 135150407111032



**PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI MANAJEMEN PERPUSTAKAAN PADA
RUANG BACA FAKULTAS ILMU KOMPUTER UNIVERSITAS BRAWIJAYA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
I Gde Yuda Pratama
NIM: 135150407111032

Skripsi ini telah diuji dan dinyatakan lulus pada
31 Juli 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Satrio Agung Wicaksono, S.Kom.,

Moch. Chandra Saputra, S.Kom, M.T.,

M.Kom.

M.Eng.

NIP: 19860521 201212 1 001

NIK: 201609 860106 1 001

Mengetahui

Ketua Jurusan Sistem Informasi



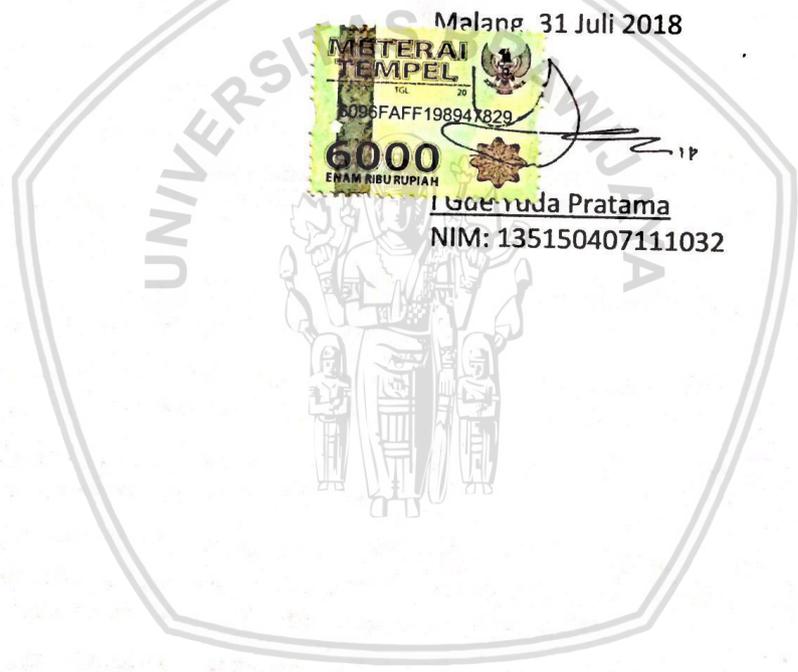
Dr. Eng. Herman Tolle, S.T, M.T.

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 unsurunsur 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).



KATA PENGANTAR

Segala puji bagi Allah SWT atas limpahan rahmat yang diberikan sehingga penulis dapat menyelesaikan skripsi berjudul “PENGEMBANGAN SISTEM INFORMASI MANAJEMEN PERPUSTAKAAN PADA RUANG BACA FAKULTAS ILMU KOMPUTER UNIVERSITAS BRAWIJAYA”.

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 telah membantu dalam penyelesaian skripsi ini diantaranya:

1. Bapak Satrio Agung Wicaksono, S.Kom., M.Kom. selaku Dosen pembimbing pertama yang telah membimbing, memberikan banyak solusi dalam pengerjaan skripsi, serta menjadi pembimbing yang sabar, solutif, dan memberikan contoh sebagai pribadi yang profesional dalam memberikan bimbingan kepada mahasiswa.
2. Bapak Mochamad Chandra Saputra, S.Kom, M.Eng. selaku Dosen pembimbing kedua yang telah membimbing dan mengajarkan kedisiplinan dalam menyelesaikan kewajiban. Memberikan contoh sebagai pribadi yang sabar, solutif, dan profesional dalam memberikan bimbingan kepada mahasiswa.
3. Bapak Suprpto, S.T., M.T. selaku Ketua Program Studi Sistem Informasi, Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Herman Tolle, Dr. Eng., S.T, M.T selaku Ketua Jurusan Sistem Informasi, Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Wayan Firdaus Mahmudy , S.Si, M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
6. Ayah dan Ibu saya yang selalu memberikan dukungan doa, moril, materil, motivasi, kedisiplinan , dan nasihat untuk menjadi pribadi yang sabar dan pekerja keras. Serta adik-adik saya yang saya sayangi.
7. Bapak Prasetyo Iskandar, S.T. selaku Kemahasiswaan yang telah memberikan banyak nasihat, dan motivasi dalam penyelesaian skripsi ini.
8. Bapak Syaifudin, Drs., M.hum. selaku petugas ruang baca yang telah memberi kesempatan dan kemudahan dalam melakukan penelitian sehingga sangat membantu dalam penyelesaian skripsi ini.
9. Ni Putu Yunita Sri Lestari yang telah memberikan semangat, motivasi, dan menemani dalam penyelesaian skripsi ini.
10. Teman-teman Jurusan Sistem Informasi, Fakultas Ilmu Komputer, yang tidak dapat saya sebutkan namanya satu per satu, yang telah memberikan

saya dukungan moril, menjadi teman dan sahabat yang baik dan menyenangkan selama masa perkuliahan.

11. Semua pihak yang telah banyak membantu untuk berbagi ilmu yang tidak bisa penulis sebutkan semuanya

Semoga Allah SWT 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, 31 Juli 2018

Penulis

pratama_yudaa@yahoo.com



ABSTRAK

Pada Fakultas Ilmu Komputer Universitas Brawijaya terdapat perpustakaan yang disebut ruang baca. Ruang baca menyediakan koleksi buku dan berbagai laporan penelitian yang digunakan oleh mahasiswa untuk belajar, mencari informasi dan referensi terkait dengan matakuliah, tugas akhir skripsi, dan beberapa riset lainnya. Saat ini, ruang baca telah menggunakan sistem informasi dengan memanfaatkan sistem informasi berbasis *desktop* untuk memberikan informasi koleksi buku yang ada di ruang baca. Dalam mengakses sistem informasi dan layanan perpustakaan masih bersifat *offline*, sehingga dalam proses peminjaman buku, mahasiswa datang ke ruang baca, belum adanya informasi terkait jumlah ketersediaan pada setiap buku, belum adanya pengelolaan denda, sehingga petugas ruang baca mengalami kesulitan dalam mengelola peminjaman dan pengembalian buku. Maka penelitian ini dilakukan dengan melakukan pengembangan sistem informasi manajemen perpustakaan yang dapat mengatasi masalah peminjaman, pengembalian, pengelolaan denda, serta untuk mencari informasi secara *online*. Pada penelitian ini menggunakan metode *waterfall model* dalam pengembangan sistem informasi yang akan dibangun dan menggunakan permodelan *Unified Modeling Language* (UML) untuk mempermudah memahami perancangan terhadap perangkat lunak yang dibangun. Berdasarkan analisis masalah menghasilkan 13 fitur sistem, 28 kebutuhan fungsional, dan 1 kebutuhan non-fungsional memberikan gambaran umum kemampuan yang dapat dilakukan oleh sistem informasi manajemen perpustakaan. Hasil dari pengujian validasi, sistem dapat berjalan dengan baik dan dinyatakan valid. Hasil UAT menunjukkan bahwa sistem 100% dapat diterima dari sisi petugas ruang baca dan mahasiswa serta hasil pengujian komparabilitas, terdapat *critical issue* yang muncul akibat tidak diaktifkannya fitur *ssl* atau *https*.

Kata kunci: Sistem informasi, perpustakaan, *UML*, *use case*, *waterfall model*.

ABSTRACT

Computer Science Faculty of Brawijaya University has a reading room that provides a collection of books and various research reports used by students to learn, looking for information and references related to subjects, thesis final assignment, and some other research. The reading room has been using information systems by utilizing desktop-based information systems to provide information on the collection of books in the reading room. Currently to access the information system and library services is still offline, so in the process of borrowing books, students must come to the reading room, there is no information related to the amount of availability in each book, the absence of management of fines, so that the reading room staff have difficulty in managing borrowing and return of books. So this research is done by developing a library management information system that can overcome the problem of borrowing, returns, management of fines, and to seek information online. This research using waterfall model method in information system development that will be built and using Unified Modeling Language (UML) to make it easier to understand the design of the built software. Based on problem analysis resulting on 13 features system, 28 functional requirements, and 1 non-functional requirement provides an overview of the capabilities that can be done by the library management information system. The result of validation testing, the system can run well and declared valid. The results of UAT show that 100% system can be received from the side of the reading room staff and students and the results of compatibility testing, there is a critical issue that appear due to ssl or https feature is not enabled.

Keywords: Information sistem, library, UML, use case, waterfall model.

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	4
1.5 Batasan Masalah	4
1.6 Sistematika Pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Tinjauan Pustaka	6
2.2 Perpustakaan	7
2.3 Sistem Informasi	7
2.4 Sistem Informasi Manajemen.....	8
2.5 Proses Bisnis.....	8
2.6 <i>Bussines Proses Modeling and Notation (BPMN)</i>	9
2.6.1 <i>Flow Object</i>	9
2.6.2 <i>Connections</i>	11
2.6.3 <i>Swimlanes</i>	11
2.6.4 <i>Artifacts</i>	12
2.7 Software Development Life Cycle (SDLC)	13
2.8 <i>Waterfall Model</i>	13
2.9 Tahap Analisis Kebutuhan.....	14
2.9.1 Analisis Proses Bisnis.....	15



2.9.2 Identifikasi Tipe Pemangku Kepentingan.....	16
2.9.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna.....	17
2.9.4 Identifikasi Pengguna.....	17
2.9.5 Identifikasi Fitur	17
2.9.6 Identifikasi Persyaratan Fungsional	18
2.9.7 Identifikasi Persyaratan Non-Fungsional	18
2.10 Tahap Desain Perangkat Lunak dan Sistem	18
2.10.1 <i>Unified Modeling Language</i> (UML).....	18
2.10.2 <i>Physical Data Model</i>	25
2.11 Tahap Implementasi dan Pengujian Unit	26
2.11.1 Framework CodeIgniter	26
2.11.2 Web Service.....	28
2.11.3 JSON	29
2.12 Tahap Pengujian Sistem dan Integrasi.....	30
2.12.1 Pengujian Black-Box.....	30
BAB 3 METODOLOGI PENELITIAN	33
3.1 Identifikasi Masalah	34
3.2 Studi Literatur	34
3.3 Pengumpulan Data	35
3.4 Analisis Kebutuhan	35
3.5 Perancangan Sistem.....	36
3.6 Implementasi	36
3.7 Pengujian	36
3.8 Penarikan Kesimpulan dan Saran	36
BAB 4 Analisis Kebutuhan	37
4.1 Analisis Proses Bisnis	37
4.1.1 Analisis Proses Bisnis <i>As-Is</i>	37
4.1.2 Analisis Permasalahan.....	39
4.1.3 Permodelan Proses Bisnis <i>To-Be</i>	40
4.2 Pemetaan Proses Bisnis	45
4.3 Identifikasi Tipe Pemangku Kepentingan	46

4.4	Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna	47
4.5	Identifikasi Pengguna.....	49
4.6	Identifikasi Fitur	50
4.7	Persyaratan Fungsional.....	52
4.8	Persyaratan Non-Fungsional.....	54
4.9	Pemodelan <i>Use case</i> Diagram	55
4.9.1	Spesifikasi <i>Use case</i>	56
4.10	Permodelan Activity Diagram	70
4.10.1	<i>Activity diagram</i> mengelola peminjaman buku.....	70
4.10.2	<i>Activity diagram</i> mengelola pengembalian buku	71
4.10.3	<i>Activity diagram</i> melakukan pencarian buku	72
4.10.4	<i>Activity diagram</i> mengelola data denda.....	73
BAB 5	PERANCANGAN.....	75
5.1	Perancangan Arsitektur Sistem.....	75
5.2	Perancangan <i>Sequence Diagram</i>	76
5.2.1	<i>Sequence Diagram</i> Mengelola Peminjaman Buku.....	76
5.2.2	<i>Sequence Diagram</i> Mengelola Pengembalian Buku	78
5.2.3	<i>Sequence Diagram</i> Melakukan Pencarian Buku	79
5.2.4	<i>Sequence Diagram</i> Mengelola Data Denda	80
5.3	Perancangan Kelas Diagram	81
5.3.1	Diagram Kelas Analisis.....	81
5.3.2	Diagram Kelas Perancangan.....	82
5.4	Perancangan Basis Data.....	85
5.5	Perancangan Antarmuka Pengguna	86
5.5.1	Antarmuka Login	86
5.5.2	Antarmuka Melihat Detail Buku.....	87
5.5.3	Antarmuka Menambahkan Peminjaman Buku	88
5.5.4	Antarmuka Melihat Informasi Buku yang dipinjam	88
5.5.5	Antarmuka Melihat Riwayat Peminjaman dan Denda.....	89
5.6	Perancangan Algoritme	90
5.6.1	Mengelola Peminjaman Buku	90
5.6.2	Mengelola Pengembalian Buku	91



5.6.3 Melakukan Pencarian Buku.....	91
5.6.4 Mengelola Data Denda	92
5.7 Perancangan <i>Web Service</i>	92
5.7.1 <i>Service</i> Menampilkan Data Denda Mahasiswa	92
BAB 6 IMPLEMENTASI	94
6.1 Spesifikasi Lingkungan Implementasi	94
6.2 Implementasi Algoritme	95
6.2.1 Mengelola Peminjaman Buku	95
6.2.2 Mengelola Pengembalian Buku	96
6.2.3 Melakukan Pencarian Buku.....	96
6.2.4 Mengelola Data Denda	96
6.3 Implementasi Antarmuka Pengguna	97
6.3.1 Antarmuka <i>Login</i>	97
6.3.2 Antarmuka Melihat Detail Buku.....	98
6.3.3 Antarmuka Menambahkan Peminjaman Buku	99
6.3.4 Antarmuka Melihat Informasi Buku yang dipinjam	100
6.3.5 Antarmuka Melihat Riwayat Peminjaman dan Denda.....	102
6.4 Implementasi Web Service	103
6.4.1 Implementasi Web Service Menampilkan Data Denda Mahasiswa.....	103
BAB 7 PENGUJIAN DAN Analisis pengujian	105
7.1 Pengujian <i>Black-Box</i>	105
7.1.1 Pengujian Validasi	105
7.1.2 Pengujian Validasi Mengelola Peminjaman Buku	105
7.1.3 Pengujian <i>Validasi</i> Mengelola Pengembalian Buku.....	108
7.1.4 Pengujian <i>Validasi</i> Mengelola Data Denda.....	109
7.2 <i>User Acceptance Testing</i>	110
7.3 <i>Compatibility Testing</i>	114
7.4 Analisis Hasil.....	114
BAB 8 PENUTUP	116
8.1 Kesimpulan.....	116
8.2 Saran	117

DAFTAR PUSTAKA..... 118
LAMPIRAN A HASIL WAWANCARA..... 120
LAMPIRAN B HASIL USER ACCEPTANCE TESTING (UAT) 126



DAFTAR TABEL

Tabel 2.1 Perbandingan Pengembangan Sistem	7
Tabel 2.2 Tipe <i>Event</i>	9
Tabel 2.3 Tipe <i>Activity</i>	10
Tabel 2.4 Tipe <i>Gateway</i>	10
Tabel 2.5 <i>Tipe Connections</i>	11
Tabel 2.6 Tipe <i>Swinlanes</i>	11
Tabel 2.7 Tipe <i>Artifacts</i>	12
Tabel 2.8 Analisis Permasalahan	15
Tabel 2.9 Analisis Tipe Pemangku Kepentingan	16
Tabel 2.10 Fitur Sistem.....	17
Tabel 2.11 Simbol-Simbol <i>Use Case Diagram</i>	19
Tabel 2.12 Format Spesifikasi <i>Use Case</i>	20
Tabel 2.13 Simbol-Simbol <i>Activity Diagram</i>	21
Tabel 2.14 Simbol-Simbol <i>Sequence Diagram</i>	22
Tabel 2.15 Simbol-Simbol <i>Class Diagram</i>	24
Tabel 2.16 Kriteria UAT	31
Tabel 4.1 Analisis Permasalahan	39
Tabel 4.2 Daftar Proses Bisnis Usulan (<i>To-Be</i>)	40
Tabel 4.3 Perubahan Aktivitas Proses Bisnis Peminjaman Buku Berserta.....	42
Tabel 4.4 Perubahan Aktivitas Proses Bisnis Pengembalian Buku Berserta	44
Tabel 4.6 Datar Aktivitas Proses Bisnis	46
Tabel 4.7 Tipe Pemangku Kepentingan.....	47
Tabel 4.8 Kebutuhan Pemangku Kepentingan dan Pengguna	48
Tabel 4.9 Identifikasi Pengguna	50
Tabel 4.10 Fitur Sistem.....	50
Tabel 4.11 Persyaratan Fungsional	52
Tabel 4.12 Persyaratan Non-Fungsional	54
Tabel 4.13 Spesifikasi <i>Use case</i> mengelola peminjaman buku	56
Tabel 4.14 Spesifikasi <i>Use case</i> mengelola pengembalian buku	57
Tabel 4.15 Spesifikasi <i>Use case</i> melakukan pencarian informasi buku	58



Tabel 4.16 Spesifikasi <i>Use case</i> mengelola data buku	59
Tabel 4.17 Spesifikasi <i>Use case</i> mengelola data mahasiswa	61
Tabel 4.18 Spesifikasi <i>Use case</i> mengelola data denda	63
Tabel 4.19 Spesifikasi <i>Use case</i> melihat informasi buku yang dipinjam	64
Tabel 4.20 Spesifikasi <i>Use case</i> melihat riwayat transaksi	64
Tabel 4.21 Spesifikasi <i>Use case</i> melakukan perubahan <i>password</i>	65
Tabel 4.22 Spesifikasi <i>Use case</i> melihat riwayat peminjaman dan denda	66
Tabel 4.23 Spesifikasi <i>Use case</i> mengelola data rak buku	67
Tabel 4.24 Spesifikasi <i>Use case</i> melakukan proses otentikasi	68
Tabel 4.25 Spesifikasi <i>Use case</i> melakukan proses keluar dari sistem	69
Tabel 5.1 <i>Psdeudocode</i> Menambahkan Peminjaman Buku	90
Tabel 5.2 <i>Psdeudocode</i> Mengelola Pengembalian Buku	91
Tabel 5.3 <i>Psdeudocode</i> Melakukan Pencarian Buku	91
Tabel 5.4 <i>Psdeudocode</i> Melihat Data Denda	92
Tabel 5.5 <i>Service</i> Untuk Menampilkan Data Denda	92
Tabel 6.1 Spesifikasi Perangkat Keras	94
Tabel 6.2 Spesifikasi Perangkat Lunak	94
Tabel 6.3 Implementasi Algoritme Menambahkan Peminjaman Buku	95
Tabel 6.4 Implementasi Algoritme Mengelola Pengembalian Buku	96
Tabel 6.5 Implementasi Algoritme Melihat Data Denda	96
Tabel 6.6 Implementasi Algoritme Melakukan Pencarian Buku	97
Tabel 6.7 Implementasi <i>Controller</i> Denda	103
Tabel 6.8 Implementasi <i>Model</i> M_ServiceDenda	104
Tabel 7.1 Matriks Skenario Pengujian Validasi Mengelola Peminjaman Buku ...	105
Tabel 7.2 Rencana Pengujian dan Kasus Uji Validasi Fungsi Menampilkan Peminjaman Buku	106
Tabel 7.3 Rencana Pengujian dan Kasus Uji Validasi Fungsi Menambahkan Peminjaman Buku	107
Tabel 7.4 Rencana Pengujian dan Kasus Uji Validasi Fungsi Menambahkan Peminjaman Buku	107
Tabel 7.5 Rencana Pengujian dan Kasus Uji Validasi Fungsi Mengelola Pengembalian Buku	108
Tabel 7.6 Matriks Skenario Pengujian Validasi Mengelola Data Denda	109

Tabel 7.7 Rencana Pengujian dan Kasus Uji Validasi Fungsi Menampilkan Denda 109

Tabel 7.8 Rencana Pengujian dan Kasus Uji Validasi Fungsi Menampilkan Denda 110

Tabel 7.9 Task yang perlu dikerjakan tester 111

Tabel 7.10 Hasil Pengujian UAT 113



DAFTAR GAMBAR

Gambar 2.1 Contoh BPMN.....	13
Gambar 2.2 <i>Waterfall Model</i>	13
Gambar 2.3 Contoh <i>Use Case Diagram</i>	20
Gambar 2.4 Contoh <i>Activity Diagram</i>	22
Gambar 2.5 Contoh <i>Activity Diagram</i>	23
Gambar 2.6 Contoh <i>Class Diagram</i>	25
Gambar 2.7 Contoh <i>Physical Data Model</i>	26
Gambar 2.8 Struktur Kerja Code Igniter	27
Gambar 2.9 Permodelan Web Service	28
Gambar 2.10 Contoh Representasi Objek JSON	30
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	33
Gambar 4.1 Proses Bisnis <i>As-Is</i> Peminjaman Buku	38
Gambar 4.2 Proses Bisnis <i>As-Is</i> Pengembalian Buku	39
Gambar 4.3 Kodifikasi Proses Bisnis.....	40
Gambar 4.4 Proses Bisnis <i>To-Be</i> Peminjaman Buku	42
Gambar 4.5 Proses Bisnis <i>To-Be</i> Pengembalian Buku.....	44
Gambar 4.6 Proses Bisnis <i>To-Be</i> Pengelolaan Denda	45
Gambar 4.7 Kodifikasi Aktivitas Proses Bisnis.....	45
Gambar 4.8 Kodifikasi Kebutuhan Pengguna.....	47
Gambar 4.9 Kodifikasi Fitur	50
Gambar 4.10 Kodifikasi Persyaratan Fungsional.....	52
Gambar 4.11 Use Case Diagram.....	55
Gambar 4.12 <i>Activity Diagram</i> Mengelola Peminjaman Buku	71
Gambar 4.13 <i>Activity Diagram</i> Mengelola Pengembalian Buku	72
Gambar 4.14 <i>Activity Diagram</i> Melakukan Pencarian Buku.....	73
Gambar 4.15 <i>Activity Diagram</i> Mengelola Data Denda.....	74
Gambar 5.1 Arsitektur Sistem	75
Gambar 5.2 <i>Sequence Diagram</i> Menambahkan Peminjaman Buku	77
Gambar 5.3 <i>Sequence Diagram</i> Menghapus Peminjaman Buku	78
Gambar 5.4 <i>Sequence Diagram</i> Melihat Peminjaman Buku	78



Gambar 5.5 <i>Sequence</i> Diagram Pengembalian Buku	79
Gambar 5.6 <i>Sequence</i> Diagram Melakukan Pencarian Buku	80
Gambar 5.7 <i>Sequence</i> Diagram Mengelola Data Denda	81
Gambar 5.8 Diagram Kelas Analisis.....	82
Gambar 5.9 Diagram Kelas Perancangan <i>Controller</i>	83
Gambar 5.10 Diagram Kelas Perancangan <i>Model</i>	84
Gambar 5.11 Diagram Kelas Perancangan Relasi <i>Controller</i> dan <i>Model</i>	85
Gambar 5.12 PDM Sistem Informasi Manajemen Perpustakaan	86
Gambar 5.13 Antarmuka Login	87
Gambar 5.14 Antarmuka Melihat Detail Buku.....	87
Gambar 5.15 Antarmuka Menambahkan Peminjaman Buku.....	88
Gambar 5.16 Antarmuka Melihat Informasi Buku yang dipinjam	89
Gambar 5.17 Antarmuka Melihat Riwayat Peminjaman dan Denda.....	90
Gambar 6.1 Antarmuka <i>Login</i>	97
Gambar 6.2 Antarmuka Melihat Detail Buku.....	99
Gambar 6.3 Antarmuka Menambahkan Peminjaman Buku.....	100
Gambar 6.4 Antarmuka Melihat Informasi Buku yang dipinjam	101
Gambar 6.5 Antarmuka Melihat Riwayat Peminjaman dan Denda.....	102
Gambar 6.6 Hasil Implementasi Web Service Melihat Data Denda Mahasiswa	104
Gambar 7.1 Kodifikasi Pengujian Validasi.....	105
Gambar 7.2 Hasil pengujian kompatibilitas pada SortSite	114



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Kebutuhan informasi dalam arus perkembangan teknologi yang terus menerus meningkat menjadi hal yang sangat penting. Adanya teknologi yang terus berkembang, maka penyebaran data dan informasi sangatlah mudah dan cepat. Salah satu teknologi yang makin berkembang saat ini adalah adanya sistem informasi yang dapat berfungsi untuk mengumpulkan data, memprosesnya menjadi informasi dan mendistribusikannya kepada para pengguna (Hall, 2008).

Sistem informasi diartikan secara teknis sebagai seperangkat komponen yang saling terkait dalam pengumpulan, memproses, menyimpan, dan mendistribusikan suatu informasi agar mendukung dalam pengambilan keputusan dan pengendalian di dalam suatu organisasi (Laudon, 2014). Sehingga bisa dibilang bahwa sistem informasi merupakan suatu sistem yang bisa mengolah data dengan baik mulai dari yang sederhana hingga yang kompleks dan dijadikan sebuah informasi agar bisa dimanfaatkan dalam pengambilan keputusan dalam suatu organisasi.

Perpustakaan merupakan salah satu pusat informasi (Suwarno, 2016). Dalam pasal 1 Undang-undang Perpustakaan No. 43 tahun 2007, Perpustakaan adalah institusi pengelola koleksi karya tulis, karya cetak, dan karya rekam secara profesional dengan sistem yang baku guna memenuhi kebutuhan pendidikan, penelitian, pelestarian, informasi, dan rekreasi para pemustaka. Keberadaan perpustakaan berperan penting dalam meningkatkan pengetahuan dan wawasan, maka pengelolaan dan fasilitas sangat penting dalam menunjang mutu pelayanan perpustakaan menjadi lebih baik.

Pada Fakultas Ilmu Komputer Universitas Brawijaya terdapat perpustakaan yang biasa disebut ruang baca. Ruang Baca menyediakan koleksi buku dan berbagai laporan penelitian. Ruang baca digunakan oleh mahasiswa untuk belajar, mencari informasi dan referensi terkait dengan mata kuliah, tugas akhir skripsi, dan beberapa riset lainnya. Saat ini, ruang baca telah menggunakan sistem informasi dengan memanfaatkan sistem informasi berbasis *desktop* dan memiliki tiga orang petugas yang untuk melayani aktivitas pelayanan yang berjalan pada ruang baca. Dalam mengakses informasi mengenai koleksi buku yang tersedia dan layanan perpustakaan masih bersifat *offline* (tidak terhubung dengan internet). Selain itu, ruang baca tersebut juga menyediakan layanan peminjaman buku dan pengembalian buku.

Berdasarkan hasil wawancara yang penulis lakukan dengan bapak Syaifuddin selaku petugas ruang baca, mereka mengalami kesulitan pada saat mahasiswa dalam melakukan peminjaman buku dan pengembalian buku. Dalam proses peminjaman buku, mahasiswa mengisikan *form* peminjaman buku pada selembur kertas dan menyerahkan KTM (Kartu Tanda Mahasiswa) sebagai jaminan peminjaman buku kepada petugas ruang baca. Dalam proses pengembalian buku, mahasiswa menyerahkan buku yang dipinjam kepada petugas ruang baca dan

petugas ruang baca melakukan pencarian *form* peminjaman buku serta KTM mahasiswa secara satu persatu pada tempat penyimpanan, sehingga membutuhkan waktu yang lama dan beresiko hilang atau rusak jika terjadi bencana yang tidak bisa diperkirakan. Selain itu dilakukan wawancara kepada beberapa mahasiswa, terdapat beberapa permasalahan yang peneliti temukan berkaitan dengan sistem yang saat ini diterapkan di ruang baca. Permasalahan tersebut meliputi (1) sistem informasi yang diterapkan saat ini berbasis desktop yang dapat memberikan informasi kepada mahasiswa mengenai koleksi buku hanya tersedia di ruang baca, sehingga mahasiswa harus datang ke ruang baca jika ingin menggunakan sistem pada waktu operasional ruang baca yaitu setiap hari senin hingga jumat mulai pukul 08.00 hingga pukul 16.00; (2) belum adanya informasi terkait jumlah ketersediaan pada setiap buku yang ada dan penataan susunan buku, sehingga mahasiswa harus mencari buku pada setiap rak yang ada, dan membutuhkan waktu yang lama; (3) belum adanya pengelolaan denda terkait peminjaman buku, sehingga adanya mahasiswa yang mengembalikan buku melebihi batas waktu yang telah ditentukan.

Berdasarkan masalah diatas, dibutuhkan sebuah aplikasi sistem informasi manajemen perpustakaan berbasis web sehingga sistem dapat diakses 7 x 24 jam selama terhubung dengan internet. Sistem tersebut diharapkan dapat membantu mahasiswa dalam memberikan informasi mengenai koleksi buku, ketersediaan buku, dan penataan susunan buku. Selain itu, sistem juga diharapkan membantu petugas ruang baca dalam melakukan proses peminjaman buku, pengembalian buku, dan mengelola denda terkait peminjaman buku. Dalam mengelola denda, sistem informasi manajemen perpustakaan menyediakan *web service* yang diberikan kepada sistem informasi akademik mahasiswa (SIAM UB). Dengan adanya *web service*, maka tagihan denda mahasiswa akan dimasukkan kedalam tagihan UKT mahasiswa pada semester selanjutnya. Untuk mengetahui sistem yang dikembangkan sesuai dengan kebutuhan petugas ruang baca dan mahasiswa, maka perlu dilakukan pengujian validasi. Selain itu, perlu dilakukannya pengujian UAT (*User Acceptance Testing*) dan pengujian kompatibilitas. Pengujian UAT dilakukan untuk mengetahui apakah suatu sistem telah dikembangkan memenuhi kriteria untuk diterima oleh petugas ruang baca dan mahasiswa, sedangkan pengujian kompatibilitas dilakukan karena sistem dikembangkan berbasis web, sehingga sistem dapat diakses pengguna dengan menggunakan *browser* dan sistem operasi yang berbeda. Tujuan dari pengujian kompatibilitas untuk memastikan apakah sistem tidak memiliki masalah ketika dijalankan pada komputer yang berbeda, tampilan perangkat yang berbeda, sistem operasi, *browser*, dan kecepatan jaringan (Pressman, 2010).

SDLC (*Software Development Life Cycle*) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem perangkat lunak (A.S. & Shalahuddin, 2018). Salah satu model SDLC yang diterapkan dalam pengembangan sistem informasi manajemen perpustakaan adalah *waterfall* model. Tahap-tahap utama dari *waterfall* model adalah analisis kebutuhan, perancangan sistem, implementasi sistem, dan pengujian sistem

(Sommerville, 2011). Pada analisis kebutuhan dilakukan pengumpulan data untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami sesuai dengan kebutuhan pengguna. Setelah melakukan analisis kebutuhan, maka dilanjutkan pada tahap perancangan sistem. Pada tahap perancangan sistem, dilakukan permodelan untuk pembangunan perangkat lunak dengan menggunakan UML (*Unified Modeling Language*). UML merupakan bahasa visual untuk permodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan tesks-teks pendukung (A.S. & Shalahuddin, 2018). Setelah perancangan sistem dibuat, maka dilanjutkan pada tahap implementasi berdasarkan perancangan sistem yang telah dibuat sebelumnya dan dilakukan pengujian. Tujuan dilakukannya pengujian sistem adalah menemukan kesalahan-kesalahan yang mungkin terjadi dalam perangkat lunak yang diuji.

Dari permasalahan yang sudah dijelaskan sebelumnya, disarankan agar membangun sistem informasi yang akan dilakukan pada skripsi ini dengan judul "PENGEMBANGAN SISTEM INFORMASI MANAJEMEN PERPUSTAKAAN PADA RUANG BACA FAKULTAS ILMU KOMPUTER UNIVERSITAS BRAWIJAYA". Adanya penelitian ini diharapkan akan mempermudah pertugas ruang baca serta mahasiswa terkait memberikan pelayanan di Ruang Baca Fakultas Ilmu Komputer.

1.2 Rumusan Masalah

Dari penjelasan latar belakang yang sudah dijelaskan sebelumnya, maka penulis dapat merumuskan masalah sebagai berikut:

1. Bagaimana hasil analisis proses bisnis dan kebutuhan sistem informasi manajemen perpustakaan pada Ruang Baca Fakultas Ilmu Komputer?
2. Bagaimana hasil perancangan sistem informasi manajemen perpustakaan pada Ruang Baca Fakultas Ilmu Komputer?
3. Bagaimana hasil implementasi sistem informasi manajemen perpustakaan pada Ruang Baca Fakultas Ilmu Komputer?
4. Bagaimana hasil pengujian sistem informasi manajemen perpustakaan pada Ruang Baca Fakultas Ilmu Komputer?

1.3 Tujuan

Dari rumusan masalah yang sudah dijelaskan diatas terdapat tujuan yang ingin dicapai dalam penelitian sebagai berikut :

1. Melakukan analisis proses bisnis dan kebutuhan sistem dari sistem informasi manajemen perpustakaan pada Ruang Baca Fakultas Ilmu Komputer.
2. Membuat perancangan sistem informasi manajemen perpustakaan pada Ruang Baca Fakultas Ilmu Komputer.
3. Membuat sistem informasi manajemen perpustakaan pada Ruang Baca Fakultas Ilmu Komputer sesuai dengan perancangan yang sudah dibuat.

4. Menguji hasil sistem informasi manajemen perpustakaan pada Ruang Baca Fakultas Ilmu Komputer dengan *black-box testing* berupa pengujian validasi, pengujian UAT (*User Acceptance Testing*), dan pengujian kompatibilitas.

1.4 Manfaat

Manfaat dari penelitian ini adalah dapat mempermudah proses bisnis dalam penggunaan sistem informasi manajemen perpustakaan pada ruang baca fakultas ilmu komputer universitas brawijaya dengan pengembangan yang telah dilakukan sehingga meningkatkan kenyamanan bagi mahasiswa dan petugas ruang baca.

1.5 Batasan Masalah

Agar pembahasan tidak melebar ke arah yang lebih luas, maka penelitian ditekankan pada hal-hal berikut:

1. Penelitian ini hanya dilakukan pada ruang baca Fakultas Ilmu Komputer Universitas Brawijaya.
2. Sistem informasi manajemen perpustakaan ini akan dibangun berbasis web menggunakan bahasa pemrograman PHP dan basis data MySQL.
3. Pengembangan sistem informasi manajemen perpustakaan ini tidak mencakup digitalisasi koleksi perpustakaan.
4. Pengembangan sistem informasi manajemen perpustakaan ini menggunakan pendekatan OOP (*Object-Oriented Programming*) dengan *framework code igniter*.

1.6 Sistematika Pembahasan

Sistematikan penulisan dari skripsi ini terdiri dari beberapa bagian utama sebagai berikut:

BAB I Pendahuluan

Pada bab ini memuat latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat, metodologi pembahasan dan sistematikan penulisan skripsi.

BAB II Landasan Kepustakaan

Pada bab ini menguraikan tentang kajian pustakan dan dasar teori yang mendasari serta mendukung pengembangan sistem informasi manajemen perpustakaan.

BAB III Metodologi Penelitian

Pada bab ini membahas dan menguraikan metode yang akan digunakan dalam penelitian terdiri dari analisis masalah, studi literatur, pengumpulan data, analisis kebutuhan, perancangan, implementasi, pengujian serta penarikan kesimpulan dan saran.

BAB IV Analisis Kebutuhan

Pada bab ini membahas gambaran dari kondisi saat ini (*as-is*) serta proses bisnis (*to-be*) pada ruang baca serta analisis kebutuhan sistem yang kemudian akan menjadi landasan dalam perancangan sistem.

BAB V Perancangan Sistem

Pada bab ini membahas dari sistem yang akan dibangun dan hasil implementasi sistem informasi manajemen perpustakaan pada Ruang Baca yaitu perancangan arsitektur, perancangan *sequence diagram*, perancangan kelas diagram, perancangan basis data, perancangan antarmuka, perancangan algoritme, dan perancangan *web service*.

BAB VI Implementasi Sistem

Membahas mengenai proses implementasi dalam pengembangan sistem informasi manajemen perpustakaan pada Ruang Baca yang sesuai dengan perancangan yang telah dibuat.

BAB VII Pengujian Sistem

Memuat proses hasil pengujian dan analisis sistem yang telah diimplementasikan dan membandingkan dengan rancangan dan tujuan awal sistem.

BAB VIII Kesimpulan dan Saran

Memuat kesimpulan dan saran yang diperoleh dari pembuatan dan pengujian sistem serta saran untuk pengembangan lebih lanjut dari sistem informasi manajemen perpustakaan pada ruang baca.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Tinjauan pustakan pada penelitian ini membahas mengenai penelitian terdahulu yang sudah pernah dilakukan. Penelitian pertama dilakukan oleh Razvan Daniel Zota dan Liviu Ciovisa (2015) dengan judul "*Designing Software Solutions Using Business Processes*". Penelitian tersebut membahas mengenai desain terkait solusi perangkat lunak dengan menggunakan proses bisnis dalam memberikan solusi dalam menyediakan solusi perangkat lunak yang lebih baik. Dalam mengimplementasikan solusi perangkat lunak tersebut dengan tepat, sesuai dengan waktu dan anggaran yang sudah disepakati, maka pertama kali harus mengidentifikasi dan membangun semua proses bisnis internal yang diperlukan oleh perusahaan yang nantinya ditransformasikan menjadi kebutuhan *fungsional* dan *non-fungsional*, atribut kualitas perangkat lunak, dan proses yang digunakan dalam merancang perangkat lunak yang benar.

Penelitian kedua dilakukan oleh Adam Przybylek (2011) dengan judul "*Bridging The Gap Between Proses Models and Use-Case Models*". Penelitian tersebut menjelaskan teknik untuk melewati hambatan dalam menyelaraskan bisnis proses dan sistem informasi dalam mendukung proses bisnis tersebut. Penyelarasan tersebut dilakukan dengan menerapkan teknik untuk menjamin *traceability* pada saat proses permodelan proses bisnis ke tahap permodelan use case. Teknik tersebut memiliki manfaat untuk meningkatkan keterkaitan terhadap proses *business engineering* dengan *requirement engineering*. Hasil penelitian tersebut menjadi dasar untuk mengadaptasi teknik yang dapat menjamin *traceability* terhadap proses permodelan proses bisnis dengan permodelan *use case* yang dilakukan pada penelitian ini. Dokumentasi mengenai *traceability* kedua proses tersebut dilakukan dengan menambahkan tabulasi yang menunjukkan hubungan aktivitas pada proses bisnis usulan dengan *use case* yang teridentifikasi dalam penelitian ini.

Penelitian ketiga dilakukan oleh Faishal Faris Shodiqin (2016) dengan judul "*Rancang Bangun Sistem Informasi Manajemen Perpustakaan Pada Ruang Baca Fakultas Ilmu Komputer Universitas Brawijaya*". Penelitian tersebut menjelaskan sistem informasi perpustakaan berbasis *desktop* yang menyediakan fasilitas bagi petugas untuk mengelola data koleksi buku dan pelayanan sirkulasi perpustakaan berupa peminjaman dan pengembalian buku. Akan tetapi, sistem informasi tersebut hanya memberikan kemudahan bagi petugas saja sedangkan mahasiswa atau pihak lain yang ingin mengakses informasi berupa koleksi buku dan pelayanan perpustakaan masih belum tersedia secara *online* sehingga mahasiswa harus datang ke ruang baca untuk mencari informasi dan pengecekan ketersediaan buku pada rak. Dengan dibuatnya sistem informasi manajemen perpustakaan yang menyediakan informasi berupa pelayanan *online* meliputi pengelusran koleksi buku, pemesanan, pengembalian, dan informasi mengenai daftar buku yang ada di Perpustakaan.

Berdasarkan pada penelitian sebelumnya, maka peneliti melakukan penelitian yang berjudul “Pengembangan Sistem Informasi Manajemen Perpustakaan pada Ruang Baca Fakultas Ilmu Komputer”. Terdapat beberapa pengembangan sistem yang akan dilakukan dari pengembangan sebelumnya akan dijabarkan pada tabel 2.1 dibawah ini:

Tabel 2.1 Perbandingan Pengembangan Sistem

Fitur sistem sebelumnya	Fitur sistem yang akan dikembangkan
<ul style="list-style-type: none"> - Pengeloaan buku - Pemesanan buku - Pengembalian buku - Pengelolaan mahasiswa - Melihat informasi transaksi buku - Melihat history transaksi 	<ul style="list-style-type: none"> - Pengelolaan buku - Peminjaman buku - Pengembalian buku - Pengelolaan mahasiswa - Melihat detail informasi informasi beserta ketersediaan - Melihat informasi peminjaman buku yang dipinjam - Melihat riwayat peminjaman dan denda - Melihat riwayat transaksi - Pengelolaan denda dengan menggunakan <i>web service</i>

2.2 Perpustakaan

Berdasarkan Undang-undang Perpustakaan No. 43 tahun 2007, Perpustakaan adalah institusi pengelola koleksi karya tulis, karya cetak, dan karya rekam secara profesional dengan sistem yang baku guna memenuhi kebutuhan pendidikan, penelitian, pelestarian, informasi, dan rekreasi para pemustaka. Sedangkan menurut Wiji Sumano (2016), perpustakaan merupakan salah satu pusat informasi dalam menghimpun, mengelola, menyimpan, melestarikan, menyajikan, serta memberdayakan informasi. Agar informasi yang dikelola mempunyai manfaat yang produktif bagi masyarakat, informasi tersebut harus memenuhi kriteria benar, tepat, cepat, dikemas dengan menarik dan siap saji.

2.3 Sistem Informasi

Sistem informasi bisa didefinisikan sebagai gabungan dari orang, perangkat lunak, perangkat keras, sumber data, jaringan komunikasi, prosedur dan kebijakan yang digunakan untuk menyimpan, mengambil, mengubah serta menyebarkan suatu informasi kedalam organisasi (O'Brien, 2007).

Sistem informasi didefinisikan juga sebagai kombinasi dari perangkat lunak, perangkat keras dan juga jaringan telekomunikasi yang dibangun lalu dipakai oleh orang untuk pengumpulan, pembuatan dan pendistribusian data yang berguna

dalam suatu pengaturan organisasi. Perangkat lunak mengacu pada rangkaian program yang dapat memberitahu komputer dalam melakukan tugas tertentu. Perangkat lunak mengacu pada alat-alat pada komputer fisik, misalnya keyboard dan monitor.

Sehingga dari pengertian tersebut bisa disimpulkan bahwa sistem informasi merupakan suatu sistem yang bisa mengolah sebuah *input* data dan menghasilkan output berupa informasi yang bisa digunakan dalam membantu organisasi dalam mencapai tujuannya.

2.4 Sistem Informasi Manajemen

Sistem informasi manajemen merupakan salah satu jenis sistem informasi yang ditujukan untuk menghasilkan informasi bagi pihak manajemen dan untuk pengambilan keputusan. Pada umumnya sistem informasi manajemen menghasilkan informasi untuk memantau kinerja, memelihara koordinasi dan menyediakan sistem informasi untuk operasi organisasi.

Dalam mengembangkan sebuah sistem informasi manajemen, hal terpenting yang perlu dipahami dari sebuah perusahaan dimana sistem informasi manajemen akan digunakan adalah proses-proses bisnis yang sedang berlangsung dalam pelaksanaan tugas. Hal ini dikarenakan sistem informasi manajemen dibangun dalam rangka meningkatkan kinerja perusahaan, sedangkan kinerja perusahaan ditentukan oleh seberapa bagus proses-proses bisnis tersebut dirancang, dikelola dan dikoordinasikan. Dalam mengembangkan sebuah sistem informasi yang mampu untuk memenuhi kebutuhan pengguna dan mencapai tujuan tertentu dari sebuah organisasi, bagi pengembang dari sistem tersebut diperlukan pengetahuan mengenai orang-orang yang akan terlibat dalam penggunaan sistem serta pengetahuan mengenai ruang lingkup sistem. Selain itu, dibutuhkan juga pengetahuan mengenai proses daur hidup dari pengembangan sistem yang bertujuan untuk menguraikan, menyusun, dan memantau perkembangan dari pengembangan sistem disepanjang tahapan proses tersebut. Proses daur hidup yang dibutuhkan untuk pengembangan sistem tersebut, dapat disebut sebagai proses *Software Development Life Cycle*.

2.5 Proses Bisnis

Proses bisnis adalah serangkaian aktivitas yang membutuhkan sebuah (atau lebih) *input* dan menghasilkan *output* yang memiliki nilai bagi *customer* (Monk & Wagner, 2013). Monk dan Wagner menambahkan, *customer* sebuah proses bisnis terdiri dari *customer* eksternal dan *customer* internal. *Customer* eksternal merupakan seseorang atau sekelompok orang yang membeli produk, sedangkan *customer* internal merupakan seseorang atau sekelompok orang dalam berbagai macam departemen dalam perusahaan. Salah satu cara dalam memodelkan proses bisnis dengan menggunakan BPMN (*Business Proses Modeling and Notation*).

2.6 Business Proses Modeling and Notation (BPMN)

Object Management Group (OMG) menjelaskan bahwa BPMN merupakan notasi-notasi standar yang digunakan memberikan solusi permodelan proses bisnis sehingga proses pertukaran informasi antar pengguna bisnis dapat dilakukan dengan baik. BPMN memiliki tujuan untuk menyediakan notasi yang mudah dipahami oleh seluruh pengguna bisnis, mulai dari analisis bisnis untuk mempersiapkan konsep proses bisnis, hingga pengguna memiliki kepentingan untuk mengelola dan mengawasi bagaimana proses bisnis dijalankan.

BPMN dirancang untuk memberikan mekanisme permodelan proses bisnis yang sederhana dan mudah dipahami, namun disaat yang bersamaan, BPMN juga mampu menangani visualisasi kompleksitas proses bisnis. BPMN memiliki empat kategori elemen sebagai penunjang dalam pembuatan BPMN (*Object Management Group*, 2011). Empat kategori elemen tersebut adalah :

2.6.1 Flow Object

Flow Object dibagi menjadi 3, yaitu *event*, *activity*, dan *gateway*. Berikut penjelasannya :

1. Event

Event merupakan sesuatu yang terjadi dan memiliki dampak dalam proses bisnis. Suatu *event* dapat berasal dari internal dan eksternal suatu proses. *Event* dibagi menjadi tiga yaitu *start event*, *intermediate event*, dan *end event*. Setiap proses selalu memiliki sebuah *start event* untuk menunjukkan awal dari proses bisnis. Tabel 2.2 merupakan penjelasan dari tipe *event*.

Tabel 2.2 Tipe Event

No	Tipe	Deskripsi	Simbol
1	<i>Start</i>	Mendeskripsikan dimana suatu proses dimulai	
2	<i>Intermediate</i>	Mendeskripsikan dimana suatu terjadi diantara awal dan akhir proses. Akan mempengaruhi alur dari proses, tapi tidak akan memulai atau memberhentikan proses.	
3	<i>End</i>	Mendeskripsikan suatu proses berakhir.	

Sumber : *Object Management Group* (2011)



2. *Activity*

Activity merupakan tugas yang dilakukan dalam sebuah proses bisnis. *Activity* ditunjukkan dengan kotak dengan ujung bulat dengan nama yang menjelaskan aktivitas yang dilakukan. Terdapat dua macam *activity* yaitu *task* dan *sub process*. Tabel 2.3 Merupakan penjelasan dari tipe *activity*.

Tabel 2.3 Tipe Activity

No	Tipe	Deskripsi	Simbol
1	<i>Task</i>	Merupakan aktivitas yang dilakukan pada alur proses	
2	<i>Sub Process</i>	Merupakan sebuah aktivitas majemuk yang dimasukkan dalam proses. Aktivitas majemuk tersebut dapat dijelaskan dengan lebih detail.	

Sumber : Object Management Group (2011)

3. *Gateway*

Gateway bertanggung jawab mengontrol bagaimana alur dari sebuah proses bisnis. Tabel 2.4 merupakan penjelasan dari tipe *gateway*.

Tabel 2.4 Tipe Gateway

No	Tipe	Deskripsi	Simbol
1	<i>Exclusive</i>	Sebagai <i>divergence</i> : digunakan untuk membuat jalur alternatif dalam sebuah proses, tapi hanya satu yang dipilih. Sebagai <i>convergence</i> : digunakan untuk menggabungkan jalur alternatif.	
2	<i>Parallel</i>	Mendeskripsikan proses yang dijalankan secara bersamaan	
3	<i>Inclusive</i>	Mendeskripsikan sebuah proses yang dipecah menjadi beberapa jalur.	



Tabel 2.4 Tipe Gateway (Lanjutan)

No	Tipe	Deskripsi	Simbol
4	<i>Complex</i>	Mendeskripsikan alur yang kompleks pada sebuah proses bisnis	

Sumber : Object Management Group (2011)

2.6.2 Connections

Connections adalah elemen yang menghubungkan *flow objects*. Tabel 2.5 merupakan penjelasan dari tipe *connections*.

Tabel 2.5 Tipe Connections

No	Tipe	Deskripsi	Simbol
1	<i>Sequence Flow</i>	Menunjukkan urutan aktivitas yang dilakukan pada sebuah proses	
2	<i>Association</i>	Menunjukkan hubungan antara data, teks, artifak lain, dan <i>flow object</i> pada sebuah proses	
3	<i>Message Flow</i>	Menunjukkan alur pesan antara dua partisipan yang mampu mengirim dan menerima pesan	

Sumber : Object Management Group (2011)

2.6.3 Swimlanes

Swimlanes merupakan wadah grafis yang membagi suatu set aktivitas dengan aktivitas lain. Tabel 2.6 merupakan penjelasan tipe *swimlanes*.

Tabel 2.6 Tipe Swinlanes

No	Tipe	Deskripsi	Simbol
1	<i>Pool</i>	Merupakan wadah yang berisi satu proses dan <i>sequence flow</i> yang menghubungkan aktivitas	



Tabel 2.6 Tipe *Swinlanes* (Lanjutan)

No	Tipe	Deskripsi	Simbol	
2	<i>Lane</i>	Digunakan untuk mempresentasikan tanggungjawab aktivitas pada sebuah proses.	Pool	

Sumber : Object Management Group (2011)

2.6.4 Artifacts

Artifacts mempresentasikan sebuah objek diluar sebuah proses. *Artifact* dapat mempresentasikan data atau catatan yang menjelaskan sebuah proses atau dapat digunakan untuk mengelola tugas atau proses. Tabel 2.7 merupakan penjelasan tipe *artifacts*.

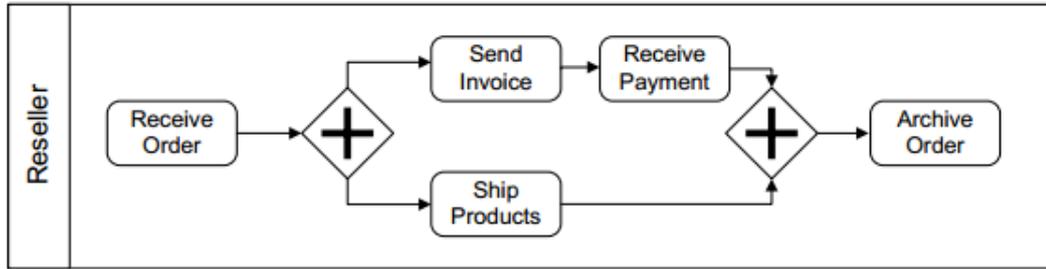
Tabel 2.7 Tipe *Artifacts*

No	Tipe	Deskripsi	Simbol
1	<i>Data Object</i>	Mempresentasikan informasi yang mengalir pada sebuah proses seperti dokumen bisnis, surat, <i>email</i> dan lain-lain.	
2	<i>Data Store</i>	Tempat dimana proses dapat membaca atau menulis data	
3	<i>Annotation</i>	Menunjukkan informasi tambahan kepada pembaca sebuah diagram BPMN.	
4	<i>Group</i>	Memungkinkan untuk mengelompokkan elemen secara informal	

Sumber : Object Management Group (2011)



Pada gambar 2.1 dapat dilihat contoh permodelan proses bisnis dengan menggunakan BPMN.



Gambar 2.1 Contoh BPMN

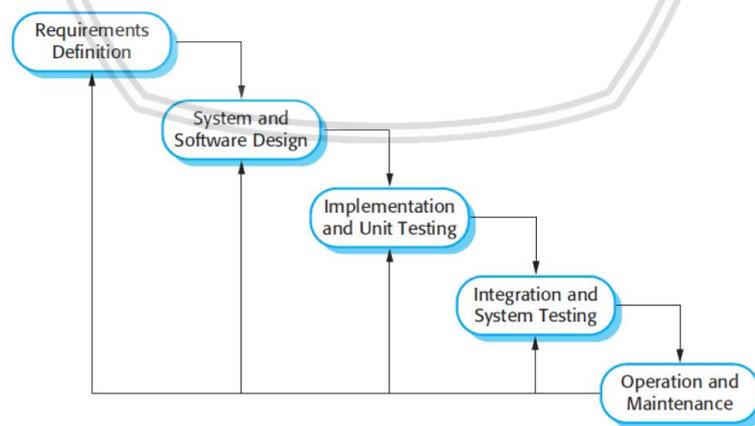
Sumber : (Weske, 2007)

2.7 Software Development Life Cycle (SDLC)

SDLC (*Software Development Life Cycle*) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem perangkat lunak. Dalam SDLC sendiri terdapat beberapa teknik pengembangan sistem yang umum digunakan, sedangkan untuk penelitian ini akan menggunakan teknik *Waterfall Model*.

2.8 Waterfall Model

Menurut Sommerville, tahapan utama dari *waterfall model* langsung mencerminkan aktifitas pengembangan dasar. Terdapat 5 tahapan pada *waterfall model*, yaitu *requirements definition, system and software design, implementation and unit testing, integration and system testing*, dan *operation and maintenance* (Sommerville, 2011).



Gambar 2.2 Waterfall Model

Sumber : Sommerville (2011)



Pada gambar 2.2 merupakan *waterfall model* dimana terdapat beberapa tahapan-tahapan yang harus dilakukan secara berurutan. Berikut ini adalah penjelasan dari tahapan-tahapan tersebut:

1. Analisis Kebutuhan

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan, kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Penentuan fitur, kendala dan tujuan dihasilkan melalui konsultasi dengan pengguna sistem. Semua hal tersebut akan ditetapkan secara rinci dan berfungsi sebagai spesifikasi sistem. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang baik.

2. Desain Perangkat Lunak dan Sistem

Proses desain sistem membagi kebutuhan-kebutuhan menjadi sistem perangkat lunak atau perangkat keras. Dalam tahapan ini akan dibentuk suatu arsitektur sistem secara keseluruhan. Desain perangkat lunak termasuk menghasilkan fungsi sistem perangkat lunak dalam bentuk yang mungkin ditransformasikan ke dalam satu atau lebih program yang dapat dijalankan.

3. Implementasi dan Pengujian unit

Dalam tahapan ini, hasil dari desain perangkat lunak akan di realisasikan sebagai satu set program atau unit program. Setiap unit akan diuji apakah sudah memenuhi spesifikasinya.

4. Pengujian Sistem dan Intergrasi

Dalam tahapan ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh (*system testing*) untuk memastikan sistem sudah memenuhi persyaratan yang ada. Setelah itu sistem akan dikirim ke pengguna sistem.

5. *Maintenance*

Dalam tahapan ini, sistem diinstal dan mulai digunakan. Selain itu juga memperbaiki *error* yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

2.9 Tahap Analisis Kebutuhan

Analisis kebutuhan merupakan proses mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan, kebutuhan yang harus dipenuhi oleh program yang akan dikembangkan. Pada tahap ini melibatkan analisis kebutuhan bisnis dan kebutuhan sistem untuk menentukan persyaratan fungsional dan non-fungsional.

2.9.1 Analisis Proses Bisnis

Pada tahap analisis proses bisnis akan dilakukan analisis proses proses bisnis yang saat ini sedang berjalan (*as-is*) dan analisis proses bisnis usulan (*to-be*) yang akan dijadikan sebagai perbaikan terhadap permasalahan yang ada. Dalam melakukan analisis proses bisnis *as-is*, langkah awal yang yang dibutuhkan dengan mengumpulkan Informasi mengenai pemangku kepentingan yang terlibat dalam suatu proses. Setelah itu dilakukan pengumpulan informasi mengenai kondisi proses bisnis yang berjalan saat dengan menggunakan teknik wawancara dan observasi secara langsung. Kemudian hasil dari informasi mengenai proses bisnis dan pihak yang terlibat dalam proses akan dimodelkan dengan menggunakan *Business Process Model and Notation*. Setelah dimodelkan, informasi yang telah terkumpulkan dapat didokumentasikan ke dalam tabel analisis permasalahan sehingga dapat dilakukan analisis mengenai solusi dan perbaikan terhadap proses bisnis *as-is*.

Dalam melakukan analisis proses bisnis *as-is*, pertama-tama dibutuhkan pengumpulan Informasi mengenai pemangku kepentingan yang terlibat dalam suatu proses. Setelah Informasi tersebut telah diketahui, selanjutnya dapat dilakukan pengumpulan Informasi mengenai kondisi proses bisnis yang berjalan saat ini. Teknik wawancara dan observasi merupakan teknik yang paling tepat untuk memahami kondisi proses bisnis yang tengah berjalan saat ini (Brandenburg, 2017). Kemudian Informasi mengenai proses bisnis yang ada dan pihak yang terlibat terhadap proses bisnis tersebut dapat digambarkan dan dimodelkan menggunakan *Business Process Model and Notation*. Setelah itu, Informasi yang telah dikumpulkan dapat didokumentasikan ke dalam tabel analisis permasalahan sehingga dapat dilakukan analisis mengenai solusi dan perbaikan terhadap proses bisnis *as-is*. Pada Tabel 2.8 berikut merupakan contoh dari cara dalam melakukan analisis permasalahan.

Tabel 2.8 Analisis Permasalahan

Masalah	[Deskripsi masalah]
Mempengaruhi	[Pemangku kepentingan yang terpengaruh oleh masalah]
Dampak	[Dampak dari masalah]
Solusi	[Solusi beserta manfaatnya]

Sumber : Bittner & Spence (2002)

Setelah melakukan analisis permasalahan dan merumuskan solusinya, maka dimodelkan terhadap proses bisnis *to-be*. Tujuan dari analisis proses bisnis *to-be* adalah menjelaskan dan memberi gambaran kepada pemangku kepentingan organisasi terkait mengenai solusi dan perbaikan terhadap proses bisnis yang sedang berjalan, apakah telah memenuhi kebutuhan dari pemangku kepentingan yang terlibat dalam proses bisnis tersebut (Brandenburg, 2017).

2.9.2 Identifikasi Tipe Pemangku Kepentingan

Pemangku kepentingan adalah seseorang yang secara materi terkena dampak dari hasil sebuah sistem atau proyek yang memproduksi sebuah sistem (Bittner & Spence, 2002). Identifikasi tipe pemangku kepentingan dilakukan setelah melakukan analisis proses bisnis pada tahap sebelumnya. Identifikasi tipe pemangku kepentingan merupakan aktivitas analisis persyaratan yang digunakan untuk mengetahui siapa saja pemangku kepentingan dari sebuah sistem atau proyek, kemudian pemangku kepentingan tersebut akan dikelompokkan berdasarkan karakteristik dan hubungan pemangku kepentingan dengan sistem atau proyek yang sedang dikerjakan. Bittner dan Spence menjelaskan bahwa terdapat beberapa kategori pemangku kepentingan yang terdiri dari:

1. Pengguna (*User*), merupakan pemangku kepentingan yang berinteraksi secara langsung dengan sistem. Pengguna merupakan individu-individu yang akan memerankan peran sebagai aktor yang ada pada model *use-case*.
2. Sponsor (*Sponsors*), merupakan pemangku kepentingan yang menanam investasi kedalam pengembangan sistem atau proyek. Pemangku kepentingan ini pada umumnya adalah pengguna tidak langsung dari sistem.
3. Pengembang (*Developers*), merupakan pemangku kepentingan yang terlibat dalam kegiatan produksi dan mendukung pengembangan sistem.
4. Pihak yang berwenang (*Authorities*), merupakan pemangku kepentingan yang memiliki keahlian khusus dalam memberikan dukungan dalam pengembangan sistem dalam bidang regulasi internal atau eksternal sebuah perusahaan, teknologi, aturan perundangan dan standar organisasi.
5. Pelanggan (*Customers*), merupakan pemangku kepentingan yang memberi sistem yang telah dikembangkan.

Namun Bittner dan Spence tidak menekankan bahwa kategori atau tipe pemangku kepentingan tidak terbatas pada kategori-kategori yang sudah dijelaskan sebelumnya. Tipe pemangku kepentingan bisa saja dikembangkan bergantung pada jenis organisasi atau perusahaan, macam-macam calon pengguna sistem, dan sebagainya. Pada Tabel 2.9 merupakan cara dalam melakukan analisis tipe pemangku kepentingan.

Tabel 2.9 Analisis Tipe Pemangku Kepentingan

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
[Tipe pemangku kepentingan]	[Menjelaskan tipe pemangku kepentingan]	[Menyebutkan pemangku kepentingan yang termasuk dalam tipe pemangku kepentingan]

Sumber: Bittner & Spence (2002)

2.9.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna

Kebutuhan pemangku kepentingan dan pengguna menggambarkan masalah dalam proses bisnis atau pribadi dari pemangku kepentingan yang perlu dilakukan analisis bagaimana solusi yang dibutuhkan untuk mendukung tujuan bisnis dan menyelesaikan masalah yang ada. Identifikasi kebutuhan pemangku kepentingan dan pengguna dapat memberikan sebuah pandangan terhadap masalah utama dan solusi yang dapat menyelesaikan masalah tersebut (Bittner & Spence, 2002).

Identifikasi kebutuhan dari pemangku kepentingan dilakukan untuk mengetahui bagaimana hubungan antara aktor dengan masalah-masalah yang ada sebelum diterapkannya aplikasi atau sistem. Dengan mengetahui siapa saja pemangku kepentingan yang akan menggunakan sistem tersebut, pengembang aplikasi bisa menentukan fungsi dan fitur apa saja yang ada dalam aplikasi dengan menyesuaikan karakteristik dari pemangku kepentingan tersebut. Leffingwell dan Widrig menjelaskan, identifikasi kebutuhan dan pengguna, identifikasi fitur, dan persyaratan perangkat lunak dapat dilakukan dengan beberapa teknik antara lain wawancara, *requirement workshop*, *brainstorming and idea reduction*, dan *storyboarding*.

2.9.4 Identifikasi Pengguna

Identifikasi pengguna merupakan tahap yang dilakukan untuk mengetahui individu yang akan menggunakan sistem informasi secara langsung. Pengguna sistem merupakan bagian dari kelompok kategori pengguna dan pengembang dalam tipe pemangku kepentingan. Hasil identifikasi pengguna diperoleh berdasarkan pemangku kepentingan yang tergolong didalam kategori pengguna dan pengembang dalam tipe pemangku kepentingan.

2.9.5 Identifikasi Fitur

Fitur adalah layanan yang disediakan oleh sistem untuk memenuhi satu atau lebih kebutuhan pemangku kepentingan dan pengguna (Leffingwell & Widrig, 2003). Fitur diperoleh dari hasil analisis proses bisnis, identifikasi tipe dan kebutuhan pemangku kepentingan dan pengguna pada tahap sebelumnya dalam memenuhi kebutuhannya. Leffingwell dan Widrig juga menambahkan, fitur merupakan cara berguna dan mudah untuk mendeskripsikan fungsional sistem tanpa harus terhambat dengan penjelasan fungsional sistem yang terlalu detail. Suatu fitur dapat memiliki beberapa kebutuhan fungsional (Bittner & Spence, 2002). Tabel 2.10 merupakan cara dalam menjelaskan fitur perangkat lunak yang telah ditentukan.

Tabel 2.10 Fitur Sistem

Kode Fitur	Fitur	Deskripsi
[Kode suatu fitur, sebagai tiap fitur]	[Fitur sistem]	[Deskripsi dari fitur]

Sumber : Bittner & Spence (2002)

2.9.6 Identifikasi Persyaratan Fungsional

Persyaratan fungsional dan persyaratan non-fungsional merupakan kategori yang umum digunakan untuk mengelompokkan persyaratan perangkat lunak (Bittner & Spence, 2002). Persyaratan fungsional dan persyaratan non-fungsional didapatkan dari hasil identifikasi fitur yang akan dikembangkan. Persyaratan fungsional adalah layanan yang sebaiknya dapat disediakan oleh sistem, persyaratan fungsional juga merupakan pernyataan yang menyatakan bagaimana sistem bereaksi terhadap input tertentu, dan bagaimana sistem berjalan ketika berapa pada situasi tertentu (Sommerville, 2011). Deskripsi dari persyaratan fungsional sebaiknya berisi penjelasan mengenai apa tindakan yang dapat dilakukan oleh sebuah sistem. Selain itu didalamnya terdapat informasi perilaku sistem ketika menerima input dan output.

2.9.7 Identifikasi Persyaratan Non-Fungsional

Persyaratan non-fungsional merupakan aspek kualitas atau batasan-batasan yang harus dipenuhi oleh sistem (Bittner & Spence, 2002). Batasan-batasan tersebut dapat berupa batasan waktu, batasan mengenai proses pengembangan, batasan mengenai keamanan, serta batasan yang diberlakukan sebagai ketentuan. Kebutuhan non-fungsional sering kali tidak hanya berlaku untuk masing-masing fitur dari sistem, melainkan berlaku untuk sistem secara keseluruhan (Sommerville, 2011).

2.10 Tahap Desain Perangkat Lunak dan Sistem

Desain sistem merupakan proses yang melibatkan pengembang perangkat lunak dan perancang untuk menentukan rencana dari solusi yang mencakup perancangan arsitektur perangkat lunak, skema konseptual basis data, definisi struktur data serta desain antarmuka. Dalam melakukan desain sistem, peneliti akan menggunakan *Unified Modelling Language*.

2.10.1 *Unified Modeling Language* (UML)

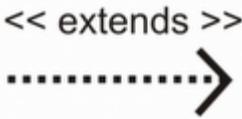
Unified Modelling Language (UML) adalah sebuah bahasa standar untuk memodelkan perangkat lunak. UML dapat digunakan untuk menggambarkan, membangun, dan mendokumentasikan artifak yang diperlukan dalam proses pengembangan sistem (Pressman, 2010). Tujuan dari UML adalah untuk menggambarkan sebuah mekanisme pemodelan yang mudah dipahami dalam pembangunan suatu sistem. UML memiliki berbagai jenis diagram diantaranya *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

2.10.1.1 *Use Case Diagram*

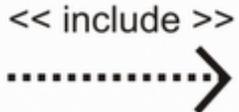
Use case diagram adalah sebuah diagram untuk menggambarkan tingkah laku sebuah sistem yang akan dibangun beserta fungsionalitas yang ada didalam sistem. *Use case diagram* juga digunakan untuk menggambarkan interaksi antara pengguna dengan sistem yang memiliki nilai. (Booch, et al., 2005).

Keuntungan menggunakan *use case diagram* adalah mudah dipahami oleh pihak pemangku kepentingan, menunjukkan tujuan yang ingin dicapai melalui serta alat komunikasi kebutuhan yang diinginkan pengguna dan pengembangan untuk memastikan sistem yang dibuat dengan yang pelanggan ingin. Tabel 2.11 merupakan penjelasan dari simbol *use case diagram*.

Tabel 2.11 Simbol-Simbol *Use Case Diagram*

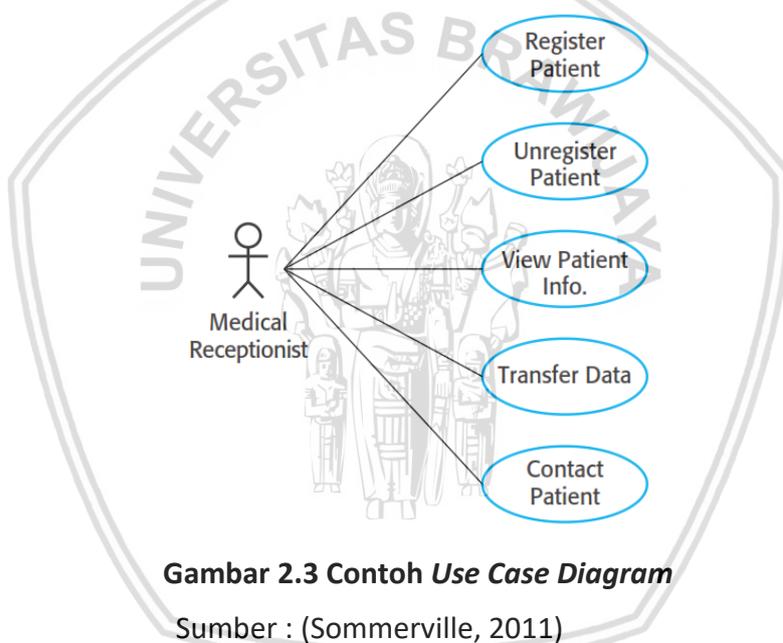
No	Nama	Gambar	Fungsi
1	<i>Uce Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
2	<i>Actor</i>		Orang, proses, atau sistem lain yang dapat berinteraksi dengan sistem yang akan dibuat. Walaupun simbol dari aktor adalah gambar orang akan tetapi belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3	<i>Association</i>		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	<i>Extend</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
5	<i>Generalization</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah

Tabel 2.11 Simbol-Simbol Use Case Diagram(Lanjutan)

No	Nama	Gambar	Fungsi
			fungsi yang lebih umum dari lainnya
6	<i>Include</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

Sumber: A.S & Shalahuddin (2013)

Contoh *use case* diagram akan ditunjukkan pada gambar 2.3.



Gambar 2.3 Contoh Use Case Diagram

Sumber : (Sommerville, 2011)

Setiap *use case* didalam *use case* diagram memiliki spesifikasi *use case* yang berisi penjelasan bagaimana aktor dan sistem berkolaborasi untuk memenuhi tujuan yang di representasikan oleh *use case* tersebut. Format penulisan spesifikasi *use case* terdapat pada tabel 2.12.

Tabel 2.12 Format Spesifikasi Use Case

Brief Description	Berisi penjelasan singkat sebuah <i>uce case</i> .
Actor	Berisi aktor yang berinteraksi dengan <i>use case</i> ,
Pre-Condition	Berisi satu atau lebih kondisi yang harus dipenuhi oleh sistem sebelum <i>use case</i> dijalankan.

Tabel 2.12 Format Spesifikasi Use Case(Lanjutan)

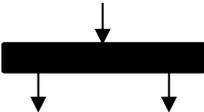
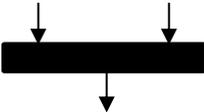
Basic Flow	Berisi alur normal yang dilalui ketika <i>use case</i> berjalan.
Alternative Flow	Berisi alur opsional yang akan dilalui ketika <i>use case</i> tidak berjalan sesuai dengan alur normal.
Subflow	Berisi beberapa kelompok alur yang merupakan penyederhanaan alur <i>use case</i> yang terlalu detail.
Post-Codition	Berisi satu atau lebih kondisi yang dipenuhi oleh sistem ketika <i>use case</i> berakhir.

Sumber: Bittner & Spence (2002)

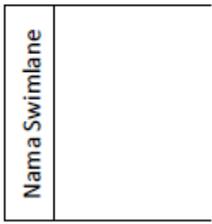
2.10.1.2 Activity Diagram

Activity diagram merupakan sebuah diagram visual yang menggambarkan alur dari sebuah aktivitas seperti sistem, bisnis, alur kerja, atau proses yang lain. *Activity Diagram* berfokus pada aktivitas yang dilakukan dan siapa yang bertanggung jawab pada proses tersebut (Booch, et al., 2005). Pada tabel 2.13 menggambarkan simbol-simbol yang ada pada *activity diagram*.

Tabel 2.13 Simbol-Simbol Activity Diagram

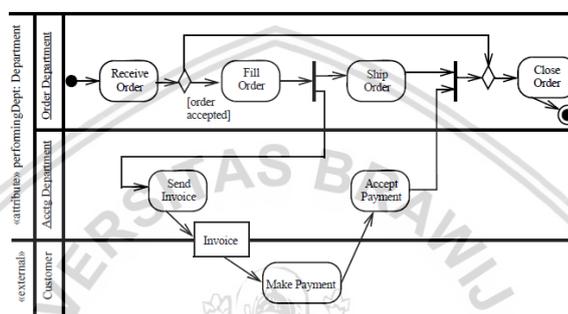
No	Nama	Gambar	Fungsi
1	Status Awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal
2	Aktifitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3	<i>Decision</i>		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4	<i>Fork</i> (Percabangan)		Asosiasi <i>fork</i> dimana pemisah beberapa aliran ko nkuren dari satu aliran tunggal
5	<i>Join</i> (Penggabungan)		Asosiasi <i>join</i> dimana lebih dari satu aktivitas digabungkan menjadi satu
6	Status Akhir		Status Akhir yang dilakukan sistem, sebuah diagram memiliki sebuah status akhir

Tabel 2.13 Simbol-Simbol Activity Diagram (Lanjutan)

No	Nama	Gambar	Fungsi
7	Swimlane		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: A.S & Shalahuddin (2018)

Contoh activity diagram akan ditunjukkan pada gambar 2.4.



Gambar 2.4 Contoh Activity Diagram

Sumber: Object Management Group (2011)

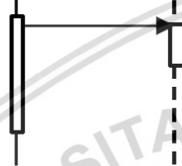
2.10.1.3 Sequence Diagram

Sequence Diagram menggambarkan urutan interaksi antara aktor dan objek di dalam sebuah sistem dan interaksi antara objek. Kelebihan dari *sequence diagram* adalah menunjukan urutan fungsi dari sebuah objek ke objek lain. *Sequence diagram* membantu dalam representasi detail dari sebuah *use case* dan digunakan untuk menunjukan interaksi pada suatu *use case* pada suatu skenario dalam perangkat lunak (Sommerville, 2011). Pada table 2.14 merupakan simbol-simbol pada *sequence diagram*.

Tabel 2.14 Simbol-Simbol Sequence Diagram

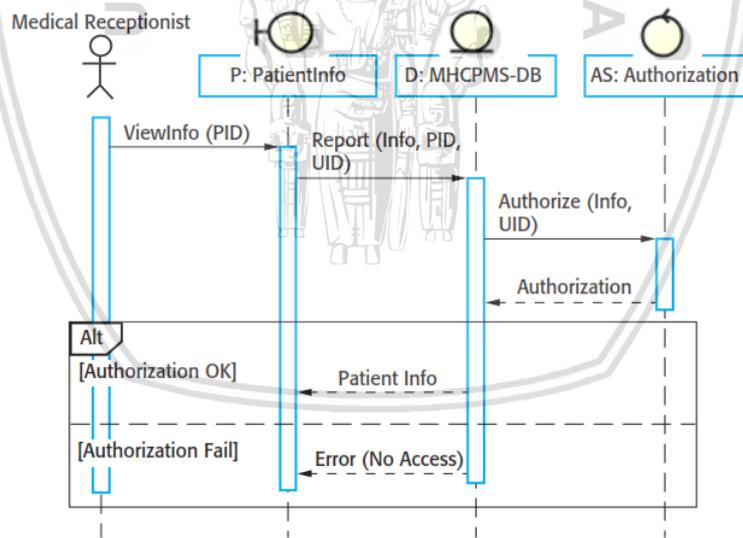
No	Nama	Gambar	Fungsi
1	Actor		Menggambarkan orang yang berinteraksi dengan sistem
2	Boundary		Menggambarkan antarmuka dan interaksi satu atau lebih aktor dengan sistem
3	Controller		Menggambarkan perilaku mengatur, mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem,

Tabel 2.14 Simbol-Simbol *Sequence Diagram* (Lanjutan)

No	Nama	Gambar	Fungsi
			menangani tugas utama dan mengontrol alur kerja suatu sistem
4	<i>Entity</i>		Menggambarkan informasi atau data yang harus disimpan oleh sistem
5	<i>Lifeline</i>		Mengindikasikan keberadaan sebuah obyek dalam basis waktu
6	<i>Message</i>		Menggambarkan pesan / hubungan antar objek, yang menunjukkan urutan kejadian yang terjadi

Sumber: A.S & Shalahuddin (2018)

Contoh *sequence diagram* akan ditunjukkan pada gambar 2.5.



Gambar 2.5 Contoh Activity Diagram

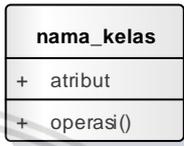
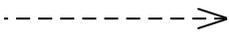
Sumber: (Sommerville, 2011)



2.10.1.4 Class Diagram

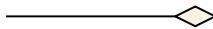
Class Diagram merupakan suatu spesifikasi, apabila spesifikasi tersebut diinstansiasi akan menghasilkan objek yang merupakan inti dari pengembangan berorientasi objek. *Class diagram* menggambarkan (atribut/properti) dari perangkat lunak, sekaligus berisi layanan untuk memanipulasi keadaan tersebut (metode/fungsi).

Tabel 2.15 Simbol-Simbol *Class Diagram*

No	Nama	Gambar	Deskripsi
1	<i>Class</i> / kelas		Notasi yang menggambarkan kelas pada struktur sistem.
2	<i>Interface</i> / antarmuka		Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3	<i>Association</i> / asosiasi		Notasi yang menggambarkan relasi antar kelas dengan makna umum. Asosiasi biasanya disertai dengan <i>multiplicity</i> .
4	<i>Directed association</i> / asosiasi berarah		Notasi yang menggambarkan relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain.
5	Generalisasi		Notasi yang menggambarkan relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
6	<i>Dependency</i> / kebergantungan		Notasi yang menggambarkan relasi antar kelas dengan makna kebergantungan antar kelas.

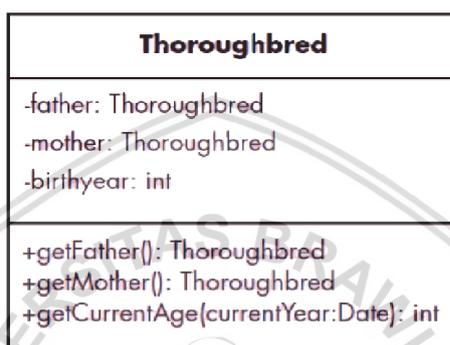


Tabel 2.15 Simbol-Simbol *Class Diagram* (Lanjutan)

No	Nama	Gambar	Deskripsi
7	<i>Aggregation</i> / agregasi		Notasi yang menggambarkan relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

Sumber: A.S & Shalahuddin (2018)

Contoh *class diagram* akan ditunjukkan pada gambar 2.6.



Gambar 2.6 Contoh *Class Diagram*

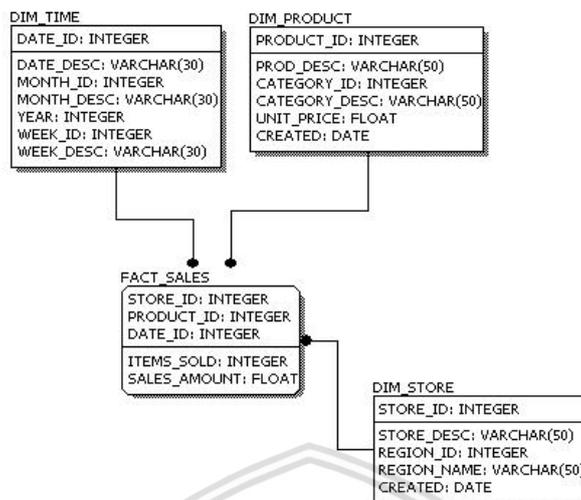
Sumber: (Pressman, 2010)

2.10.2 *Physical Data Model*

Physical data model merepresentasikan bagaimana model akan dibangun di *database*. Sebuah *physical data model* menampilkan struktur tabel yang terdiri dari nama kolom, tipe data kolom, *constraints* kolom, *primary key*, *foreign key*, dan relasi antara tabel (1keydata, 2017). Terdapat beberapa tahap dalam pembuatan *physical data model* yaitu:

1. Merubah entitas menjadi tabel.
2. Merubah relasi menjadi *foreign key*.
3. Merubah atribut menjadi kolom.
4. Memodifikasi *physical data model* sesuai dengan kebutuhan.

Pada Gambar 2.7 dapat dilihat contoh *physical data model*.



Gambar 2.7 Contoh *Physical Data Model*

Sumber : 1keydata(2017)

Dalam menerapkan *physical data model*, peneliti menggunakan pendekatan perancangan berorientasi objek. Pertama-tama, *class* dan hubungan antar *class* dipetakan kedalam tabel-tabel dirancangan basis data relational. Dalam penerapan perancangan menggunakan pendekatan ini, *persistent class* dan atributnya dipetakan menjadi entitas. Hubungan asosiasi-asosiasi dipetakan menjadi relasi non-spesifik di dalam entity relationship diagram (ERD). Jenis atribut kelas dipetakan ke tipe elemen data yang sesuai pada ERD. Dengan menggunakan ERD memungkinkan perancang basis data untuk melakukan pemeriksaan normalisasi terhadap skema basis data serta mengimplementasikan *foreign key*. Dari ERD yang telah dibuat, perancang dapat membuat *physical data model* untuk berbagai sistem manajemen basis data relational yang diinginkan (IBM, 2018).

2.11 Tahap Implementasi dan Pengujian Unit

Tahap selanjutnya adalah Tahap Implementasi dimana pada tahap implementasi ini dilakukan penerjemahan desain sistem yang telah didefinisikan sebelumnya pada tahap analisis kebutuhan sistem dan juga telah dilakukan pemodelan pada tahap desain sistem.

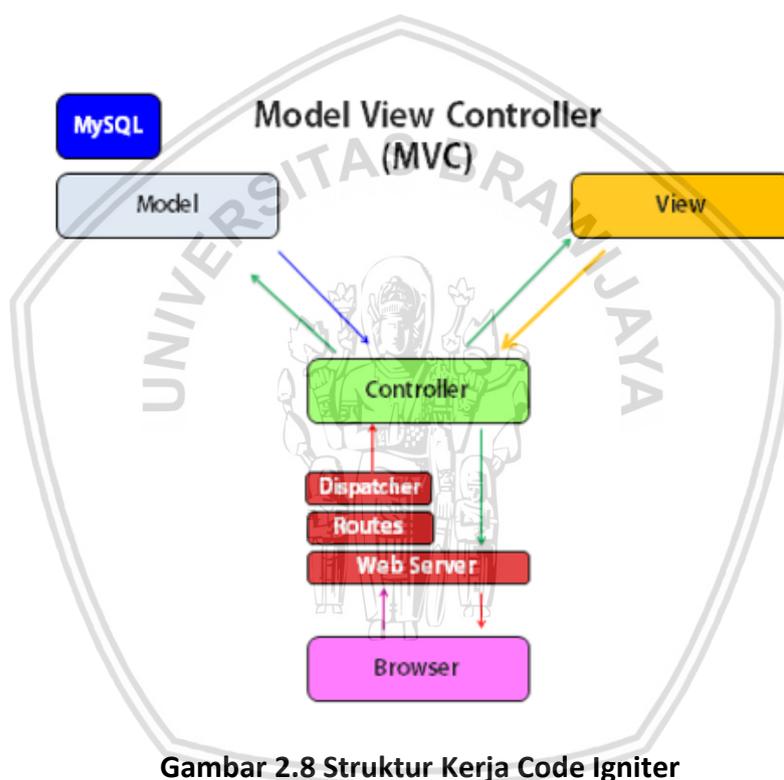
2.11.1 Framework CodeIgniter

Framework merupakan sekumpulan *library* yang terorganisir dalam suatu desain arsitektur yang mampu untuk memberikan kecepatan, ketepatan dan kenyamanan dalam pengembangan suatu aplikasi (Hustinawati, et al., 2014), serta terdiri dari elemen *Architecture*, *File Library* dan *Methodology*.

CodeIgniter adalah sebuah *framework* aplikasi web yang dikenalkan pada tanggal 28 Februari 2006 digunakan untuk membangun suatu aplikasi PHP dinamis

dan bersifat *open source*. Alasan yang mendasari CodeIgniter menjadi suatu *framework* yang bersifat *open source* adalah agar dapat membantu pengembangan dari pengembang CodeIgniter dalam mengerjakan suatu aplikasi menjadi lebih cepat dibandingkan dengan menulis seluruh kode program dari awal. Selain itu, CodeIgniter sendiri digunakan karena *framework* ini memiliki beberapa karakteristik sebagai berikut :

1. Bersifat fleksibel dan ringan sehingga memudahkan untuk dipelajari, dimodifikasi, dan diintegrasikan dengan *library* dan *helper*
2. Menggunakan pola *Model-View-Controller* (MVC) yang memungkinkan pemisahan kode program bagian logika dengan tampilan, sehingga mampu menghasilkan struktur kode program yang lebih terstruktur dan lebih baku



Gambar 2.8 Struktur Kerja Code Igniter

Sumber : Hustinawati, et al. (2014)

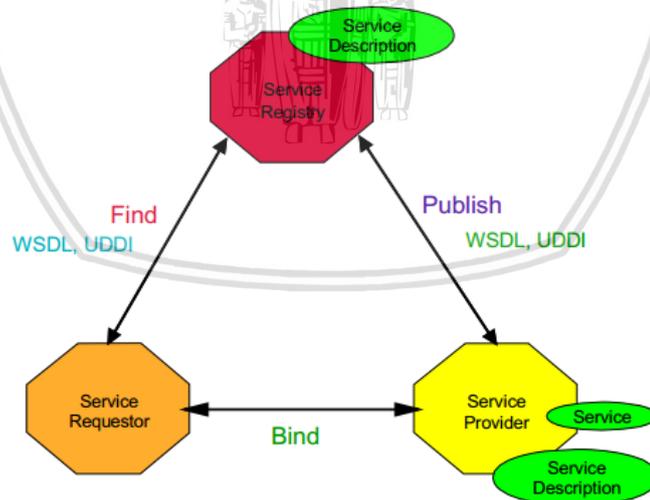
Struktur kerja dari *CodeIgniter* seperti yang digambarkan pada Gambar 2.8, diawali dari *browser* yang akan berinteraksi dengan *controller* melalui *web server*, *routes* lalu kemudian dengan *dispatcher*. *Controller* disini berperan untuk menerima dan membalas seluruh permintaan dari *browser*. Untuk keperluan mendapatkan dan melakukan perubahan data dari basis data dan sumber data lain, *controller* akan melakukan *request* ke *model*. Sedangkan untuk keperluan tampilan antarmuka pengguna *controller* akan melakukan request ke *view*. Ketika *browser* melakukan *request* sebuah halaman *web*, *router* akan mencari *controller* yang bertanggung jawab untuk menangani permintaan tersebut, kemudian *controller* tersebut akan menggunakan *model* untuk mengakses data yang diperlukan lalu menggunakan *view* untuk menampilkan data yang telah diakses.

Pada penelitian ini, *Framework CodeIgniter* akan digunakan sebagai kerangka kerja. Pengembangan sistem informasi manajemen perpustakaan dengan *Framework CodeIgniter* ini akan menggunakan PHP sebagai bahasa pemrograman *web* sisi *server*nya. Sedangkan bahasa pemrograman *web* yang akan digunakan untuk sisi *client*nya antara lain adalah *Hypertext Markup Language* (HTML) dan *Cascading Style Sheet* (CSS). HTML digunakan untuk menyusun struktur dari halaman *web* yang akan dibangun dan CSS digunakan untuk menyusun tata letak dari halaman *web* tersebut.

2.11.2 Web Service

Web service adalah layanan antarmuka yang diakses melalui jaringan untuk fungsionalitas aplikasi yang dibangun dengan standar teknologi internet. (Tidwell, et al., 2001). Berikut ini merupakan karakteristik *web service* (Yasin, 2012):

1. Merupakan *application logic* yang dapat diakses dan dipublikasikan menggunakan standar Internet (TCP/ IP, HTTP, web).
2. Dideskripsikan dalam format XML.
3. Diidentifikasi dengan *Universal Resources Identifier* (URI).
4. Bersifat *Loosely coupled, self-contained*, modular dan terbuka (*nonproprietary*)
5. Digunakan untuk mendukung interoperabilitas interaksi *machine-to-machine* melalui jaringan Internet/Intranet.



Gambar 2.9 Permodelan Web Service

Sumber : *Web Service Conceptual Architecture* (2011)

Pada gambar 2.9 merupakan *permodelan web service* dimana memiliki 3 peranan dalam berinteraksi yaitu:

1. *Service provider* (penyedia layanan)



- berfungsi untuk menyediakan layanan atau *service* dan mengolah sebuah *registry* agar layanan-layanan tersebut dapat tersedia.
2. *Service registry* (daftar layanan)
berfungsi sebagai lokasi utama yang mendeskripsikan semua layanan atau *service* yang telah didaftarkan.
 3. *Service requestor* (pemohon layanan)
berfungsi sebagai pemohon layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

Dalam mengimplementasikan *web service*, terdapat berbagai macam teknologi yang mampu mendukungnya. Teknologi yang paling sering digunakan untuk keperluan ini antara lain adalah menggunakan *Representational State Transfer*(REST) serta *Simple Object Access Protocol*(SOAP) sebagai alternatif yang juga sering digunakan. REST *Web Service* lebih sering digunakan karena penggunaannya yang lebih sederhana, lebih mudah, lebih efisien biaya pengiriman serta menggunakan teknologi berbasis web yang ekstensif seperti HTTP yang telah diketahui dan lebih populer di kalangan pengembang aplikasi *web*. Cara kerja dari REST *Web Service* adalah dengan menerapkan prinsip-prinsip dari *World Wide Web*, dalam artian bahwa sebuah sistem di tempat yang berbeda menawarkan akses kepada aplikasi *client* seperti *browser* atau *mobile apps* ke halaman penyedia *resource* dengan mengandalkan serangkaian tindakan standar seperti GET, POST, PUT, DELETE (Arroqui, et al., 2012). Kemudian dalam mengirimkan pesan atau *resource*, REST *web service* menggunakan teknologi yang sudah menjadi standar seperti HTTP, HTML, XML dan JSON.

2.11.3 JSON

JavaScript Object Notation(JSON) merupakan *format* representasi dan pertukaran data berbasis teks yang digunakan untuk mewakili suatu objek sebagai kumpulan nilai dalam bentuk *String* (Deitel & Deitel, 2012). JSON merupakan *format* representasi sederhana yang membuat suatu objek dapat mudah dibaca, dibuat atau diurai. JSON menggunakan *format* teks biasa untuk mempermudah pengiriman, pengolahan dan pertukaran Informasi antar layanan dan aplikasi yang berbeda-beda (Santos & Serrao, 2016). Penggunaan dari JSON melalui HTTP telah menjadi suatu solusi teknologi yang memungkinkan untuk diberlakukannya pertukaran Informasi antara layanan berbasis *web* dengan suatu aplikasi. JSON banyak digunakan untuk mendukung komunikasi antar beberapa *Application Programming Interface* (API) yang tersedia di internet saat ini, terutama keperluannya untuk mendukung fungsi dari REST *web service*.

JSON membuat *format* representasi suatu objek yang bisa mengandung kumpulan dari beberapa nilai-nilai yang dapat berupa data *string*, angka dan *boolean*. Kumpulan nama dan nilai yang dikandung dari setiap objek JSON diletakkan dan dituliskan di dalam tanda kurung kurawal seperti pada Gambar 2.10:

```
{
  "id" : "1",
  "f_name" : "Adam",
  "l_name" : "Smith",
  "skills" : [
    "web development",
    "ux design",
    "data science"
  ],
  "team": "engineering"
}
```

Gambar 2.10 Contoh Representasi Objek JSON

Sumber : www.json.com (2013)

Pada Gambar 2.10 dijelaskan bahwa selain nilai yang dikandung dari setiap objek JSON diletakkan di dalam tanda kurung kurawal, masing masing atribut dari nilai yang dikandung oleh suatu objek diwakilkan oleh *keys* yang dituliskan dalam bentuk data *string*. Contoh *keys* yang dimaksud antara lain adalah "id", "f_name", "l_name", "skills" dan "team". Sedangkan untuk penulisan nilai *array* pada JSON di letakkan didalam tanda kurung siku atau seperti yang dicontohkan nilai yang dikandung pada key "skills". Selain itu, atribut *key* "skills" pada Gambar 2.10 juga menjelaskan bahwa struktur nilai seperti *string*, angka, *boolean*, *array* atau objek lain yang dikandung dari setiap objek JSON juga dapat dibuat bersarang. Pada atribut *key* "skills" mengandung sebuah *array* JSON yang mengandung 3 buah *string* lain.

2.12 Tahap Pengujian Sistem dan Integrasi

Tahap Pengujian merupakan proses untuk memeriksa apakah solusi yang ditawarkan perangkat lunak tersebut telah memenuhi persyaratan dan spesifikasi yang telah disepakati di awal serta memenuhi tujuannya. Pengujian perangkat lunak berisi kegiatan yang bertujuan untuk mencari dan mengevaluasi kesalahan atau *error* yang dapat ditemukan pada suatu perangkat lunak. Sasaran dari *testing* adalah untuk mencari *error*, sehingga pengujian yang baik adalah pengujian yang memiliki peluang yang besar untuk menemukan sebuah *error* (Pressman, 2010). Dalam penelitian ini, peneliti melakukan pengujian sistem dengan menggunakan pengujian *black-box testing*.

2.12.1 Pengujian Black-Box

Pegujian *black-box* berfokus pada persyaratan fungsional dari sebuah perangkat lunak. Pada teknik pengujian ini akan ditentukan berbagai kondisi inputan yang akan dimasukan pada program, dimana *input* tersebut akan memenuhi semua tugas (menguji) dari tiap fungsional program. Pengujian ini bermaksud untuk mengetahui apabila terdapat kesalahan dalam fungsi program, antarmuka, kesalahan pada data atau akses basis data, kesalahan performa serta kesalahan pada inisialisasi dan terminasi program (Pressman, 2010). Dalam

melakukan *black-box testing* terdapat beberapa metode yang dapat dilakukan, antara lain adalah *Validation Testing* dan *User Acceptance Testing*

2.12.1.1 Validation Testing

Pengujian validasi dilakukan untuk mengetahui apakah sistem yang dibangun berjalan sesuai dengan kebutuhan yang telah didefinisikan. Pengujian validasi termasuk kedalam *black-box testing*, karena pengujian dilakukan tanpa menunjukkan proses berjalannya sebuah fungsi sebenarnya tetapi pengujian validasi melihat hasil yang dikeluarkan apakah sesuai dengan kebutuhan atau tidak dan lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan daftar kebutuhan (Indriati, 2010).

2.12.1.2 User Acceptance Testing

User Acceptance Testing (UAT) merupakan pengujian yang menangani kebutuhan pengguna, *requirement*, dan *business process*. UAT dilakukan untuk mengetahui apakah suatu sistem telah memenuhi kriteria untuk diterima oleh pengguna atau *customer* (Hambling, 2013). Pengujian UAT merupakan pengujian perangkat lunak yang bergantung kepada keterlibatan pengguna. Hal ini disebabkan karena untuk mengetahui kesesuaian penggunaan suatu perangkat lunak dengan konteks bisnis yang telah dirancang, membutuhkan sudut pandang yang unik dari pengguna. Untuk keperluan tersebut, maka dalam melakukan UAT diperlukan keterlibatan *stakeholder* dari perangkat lunak yang dibangun.

Kriteria *user acceptance testing* yang digunakan menyesuaikan dengan kebutuhan spesifikasi *stakeholder* yaitu, *functional correctness and completeness*, *confidentiality and availability*, dan *usability*. Pengujian akan dilaksanakan dengan menghampiri *tester*(penguji) untuk melakukan pengujian terhadap sistem yang telah dibuat (Naik & Kshirasagar, 2008).

Tabel 2.16 Kriteria UAT

No	Kriteria	Definisi
1	<i>Functional Correctness and Completeness</i>	Pengujian untuk mengetahui apakah sistem yang dibuat sudah sesuai dengan kebutuhan yang diinginkan pengguna dan semua fitur yang dijelaskan dalam spesifikasi persyaratan harus ada dalam sistem yang dikirimkan
2	<i>Usability</i>	Evaluasi produk atau layanan dengan mengujinya untuk mengetahui bahwa sistem dapat dengan mudah digunakan dan kepuasan pengguna.
3	<i>Confidentiality and Availability</i>	Confidentiality adalah sebuah aturan yang memiliki kemampuan untuk

Tabel 2.16 Kriteria UAT (Lanjutan)

No	Kriteria	Definisi
		membatasi hak akses terhadap informasi tertentu. Availability adalah jaminan akses terhadap data yang dapat dipercaya berkaitan dengan informasi yang dipegang oleh orang-orang yang berwenang

Sumber: Naik & Kshirasagar (2008)

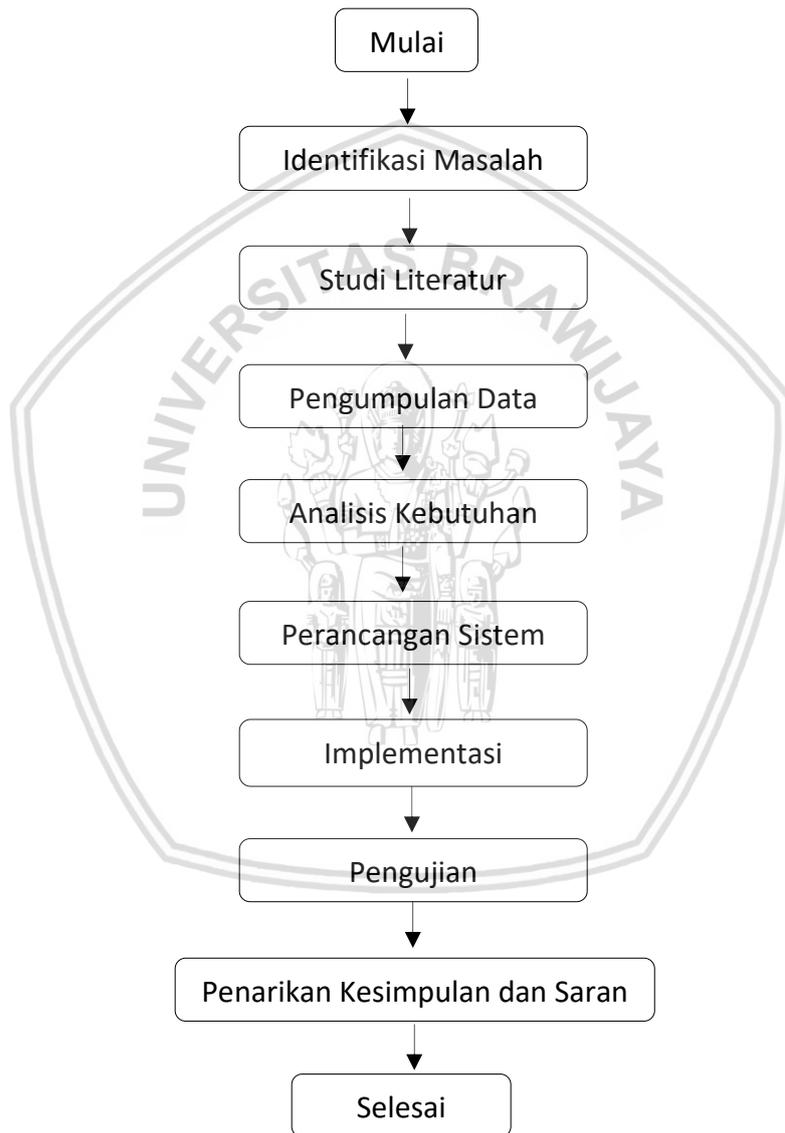
Dalam proses penilaian pada *user acceptance testing* menggunakan skala guttman. Skala Guttman, skala pengukuran dengan tipe ini akan didapat jawaban yang tegas yaitu : benar-salah, pernah-tidak pernah, ya-tidak. Skala ini dapat dibuat dengan bentuk centang maupun pilihan ganda (Sugiyono, 2012).

2.12.1.3 Compatibility Testing

Pengujian kompatibilitas merupakan salah satu dari beberapa pengujian yang perlu dilakukan pada antarmuka pengguna aplikasi berbasis web untuk memastikan apakah aplikasi tidak memiliki masalah ketika dijalankan pada komputer, perangkat tampilan, sistem operasi, aplikasi peramban, atau kecepatan jaringan yang berbeda (Pressman, 2010).

BAB 3 METODOLOGI PENELITIAN

Pada bab ini merupakan penjabaran dari langkah-langkah yang diambil dalam penyusunan penelitian pengembangan sistem informasi manajemen perpustakaan pada Ruang Baca Fakultas Ilmu Komputer Universitas Brawijaya. Untuk langkah-langkah tersebut awalnya adalah identifikasi masalah, studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan kesimpulan dan saran.



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1 Identifikasi Masalah

Analisis domain masalah yang dibahas meliputi masalah-masalah yang akan dihadapi oleh pengguna sistem informasi manajemen perpustakaan pada Ruang Baca Fakultas Ilmu Komputer Universitas Brawijaya. Setelah menganalisis dan mempelajari proses bisnis, penulis dapat merumuskan permasalahan yang dihadapi oleh petugas ruang baca dan mahasiswa. Berikut ini merupakan masalah-masalah penulis yang didapatkan:

1. Sistem informasi yang diterapkan saat ini berbasis desktop yang dapat memberikan informasi kepada mahasiswa mengenai koleksi buku yang hanya tersedia di ruang baca, sehingga mahasiswa yang ingin menggunakan sistem tersebut harus datang langsung ke ruang baca pada waktu operasional ruang baca yaitu setiap hari senin hingga jumat mulai pukul 08.00 hingga pukul 16.00.
2. Belum adanya informasi terkait jumlah ketersediaan pada setiap buku yang ada dan penataan susunan buku, sehingga mahasiswa harus mencari buku pada setiap rak yang ada, dan membutuhkan waktu yang lama.
3. Belum adanya pengelolaan denda terkait peminjaman buku, sehingga adanya mahasiswa yang mengembalikan buku melebihi batas waktu yang telah ditentukan.
4. Petugas ruang baca masih mengalami kesulitan pada saat mahasiswa dalam melakukan peminjaman buku dan pengembalian buku. Dalam proses peminjaman buku, mahasiswa mengisikan *form* peminjaman buku pada selembar kertas dan menyerahkan KTM (Kartu Tanda Mahasiswa) sebagai jaminan peminjaman buku kepada petugas ruang baca. Dalam proses pengembalian buku, mahasiswa menyerahkan buku yang dipinjam kepada petugas ruang baca dan petugas ruang baca melakukan pencarian *form* peminjaman buku serta KTM mahasiswa secara satu persatu pada tempat penyimpanan, sehingga membutuhkan waktu yang lama dan beresiko hilang atau rusak jika terjadi bencana yang tidak bisa diperkirakan.

3.2 Studi Literatur

Studi literatur dilakukan dengan mencari referensi untuk mendapatkan landasan kerangka berpikir yang berkaitan dengan penelitian yang dilakukan oleh penulis sehingga didapatkan konsep dasar, dasar teori serta metode yang akan membantu dalam pengembangan sistem informasi manajemen perpustakaan. Beberapa teori yang dipelajari meliputi:

1. Penelitian sebelumnya
2. Sistem informasi
3. SDLC model *waterfall*

4. Proses Bisnis
5. OOAD (*Object Oriented Analysis and Design*)
6. UML (*Unified Modeling Language*)
7. OOP (*Object Oriented Programming*)
8. Web Service
9. Black box testing
10. UAT (*User Acceptance Testing*)

3.3 Pengumpulan Data

Pengumpulan data bertujuan untuk memperoleh data yang akan mendukung penelitian pada pengembangan sistem informasi manajemen perpustakaan pada Ruang Baca Fakultas Ilmu Komputer Universitas Brawijaya. Dalam pengumpulan data, peneliti menggunakan beberapa metode dalam pengumpulan data yang mendukung penelitian:

1. Metode wawancara : Penulis melakukan wawancara kepada bapak Syaifudin yang berperan sebagai petugas ruang baca dan mahasiswa yang berperan sebagai pengguna ruang baca.
2. Metode observasi : pengamatan secara langsung yang dilakukan penulis bisa memberikan pemahaman lebih terkait dengan proses bisnis dari ruang baca Fakultas Ilmu Komputer Universitas Brawijaya.
3. Studi dokumen : merupakan teknik yang digunakan untuk mempelajari dokumen yang berkaitan dalam penelitian ini. Beberapa dokumen yang digunakan merupakan dokumen skripsi terdahulu yang mendukung pengembangan sistem informasi manajemen perpustakaan.

3.4 Analisis Kebutuhan

Analisis kebutuhan sistem merupakan tahapan yang dibutuhkan dalam mendapatkan batasan, tujuan dan kebutuhan sistem dengan melakukan konsultasi kepada pemangku kepentingan dan pengguna sistem. Tahapan yang dilakukan adalah memodelkan proses bisnis yang sedang berjalan dengan menggunakan BPMN (*Business Process Modeling and Notation*), identifikasi permasalahan yang terjadi pada proses bisnis yang sedang berjalan serta memodelkan proses bisnis yang diusulkan untuk menyelesaikan permasalahan menggunakan BPMN (*Business Process Modeling and Notation*). Kemudian mengidentifikasi kebutuhan pemangku kepentingan untuk mengetahui apa yang dibutuhkan oleh pemangku kepentingan dan pengguna sistem dalam menyelesaikan masalah.

Tahapan selanjutnya mendefinisikan fitur yang ada dalam sistem. Fitur merupakan solusi dari kebutuhan pemangku kepentingan dan pengguna sistem. Kemudian akan mendapatkan persyaratan fungsional dan nonfungsional dari

sistem yang akan dikembangkan dan dibangun. Setelah itu melakukan permodelan *use case diagram*, *use case spesification*, dan *activity diagram*.

3.5 Perancangan Sistem

Perancangan sistem dilakukan untuk memodelkan sistem ke dalam notasi-notasi pada *Unified Modeling Language* (UML) berdasarkan hasil tahap analisis kebutuhan. Hasil dari fase ini digunakan sebagai referensi untuk merancang arsitektur yang akan diimplementasikan dengan menggunakan pola perancangan MVC (*Model-View-Controller*). Pada perancangan sistem ini dilakukan perancangan arsitektur sistem, perancangan *sequence diagram*, perancangan *class diagram*, perancangan basis data, perancangan algoritma, dan perancangan antarmuka sistem.

3.6 Implementasi

Tahap implementasi merupakan tahap pembuatan sistem informasi manajemen perpustakaan berdasarkan hasil perancangan sistem. Dalam penelitian ini, proses implementasi menggunakan bahasa pemrograman *PHP*, *MySQL* sebagai *DBMS* dalam mengimplementasikan basis data, dan dengan kerangka kerja *Codeigniter*. Hasil tahap pembuatan sistem adalah sistem informasi manajemen perpustakaan berbasis web yang menyediakan fitur sesuai dengan yang telah teridentifikasi pada tahap analisis kebutuhan.

3.7 Pengujian

Tahap pengujian dilakukan untuk mengetahui bahwa sistem informasi yang dibangun telah berjalan sesuai kebutuhan pengguna yang telah didefinisikan. Pengujian dilakukan dengan menggunakan pengujian validasi, pengujian UAT (*User Acceptance Testing*), dan pengujian kompatibilitas. Pengujian validasi dilakukan untuk mengetahui sistem yang telah dikembangkan sesuai dengan kebutuhan pengguna, pengujian UAT (*User Acceptance Testing*) dilakukan untuk mengetahui apakah sistem yang dibangun telah dapat diterima oleh pengguna, dan pengujian kompatibilitas dilakukan untuk mengetahui apakah sistem sudah dapat digunakan dengan baik pada browser dan sistem operasi yang berbeda.

3.8 Penarikan Kesimpulan dan Saran

Proses pengambilan kesimpulan merupakan proses terakhir setelah sistem yang dibangun dapat digunakan dengan baik. Pengambilan kesimpulan diambil dari hasil pengujian dan analisa yang telah dilakukan sebelumnya. Kemudian penulisan saran untuk memperbaiki kesalahan-kesalahan yang ada sehingga dapat disempurnakan dan saran bagi pengembang sistem selanjutnya.

BAB 4 ANALISIS KEBUTUHAN

Pada bab ini akan membahas mengenai analisis kebutuhan sistem informasi manajemen pada ruang baca Fakultas Ilmu Komputer Universitas Brawijaya yang terdiri dari identifikasi kebutuhan dari sudut pandang bisnis dan sudut pandang pengguna. Kemudian dilakukan permodelan *use case diagram* beserta *use case scenario*. Berdasarkan *use case scenario* dapat dibuat activity diagram.

4.1 Analisis Proses Bisnis

Pada sub bab ini dilakukan analisis terhadap sistem yang digunakan saat dilakukan penelitian. Hasil dari analisis ini didapatkan alur proses bisnis yang sedang berjalan (*as-is*), analisis permasalahan dan proses bisnis usulan (*to-be*) sebagai solusi dalam permasalahan.

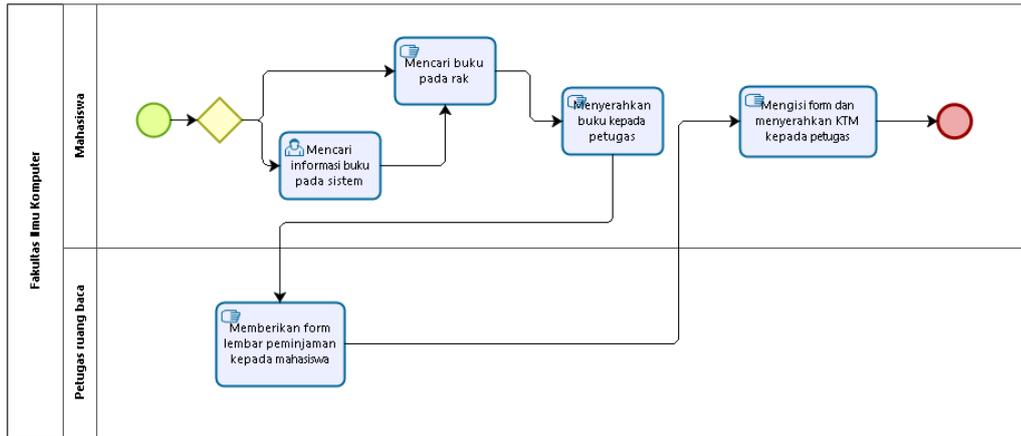
4.1.1 Analisis Proses Bisnis *As-Is*

Analisis proses bisnis *as-is* dilakukan untuk mengetahui proses bisnis yang sedang berjalan. Analisis proses bisnis *as-is* didapatkan dari hasil wawancara dengan bapak Syaifudin selaku petugas Ruang Baca dan observasi di Ruang Baca Fakultas Ilmu Komputer. Berikut ini merupakan proses bisnis yang sedang berjalan di Ruang Baca Fakultas Ilmu Komputer:

4.1.1.1 Permodelan Proses Bisnis *As-Is* Peminjaman Buku

Permodelan proses bisnis *as-is* peminjaman buku merupakan gambarkan proses bisnis peminjaman buku yang sedang berjalan. Pada gambar 4.1 merupakan permodelan proses bisnis *as-is* peminjaman buku di Ruang Baca. Berikut ini merupakan penjelasan dari alur proses bisnis *as-is* peminjaman buku:

1. Mahasiswa datang ke Ruang Baca.
2. Mahasiswa dapat langsung mencari buku pada rak atau mencari informasi data buku pada sistem, setelah itu mencari buku pada rak.
3. Mahasiswa menyerahkan buku kepada petugas ruang baca.
4. Petugas ruang baca memberikan *form* lembar peminjaman kepada mahasiswa.
5. Mahasiswa mengisi *form* lembar peminjaman dan menyerahkan KTM kepada petugas ruang baca sebagai jaminan peminjaman buku.

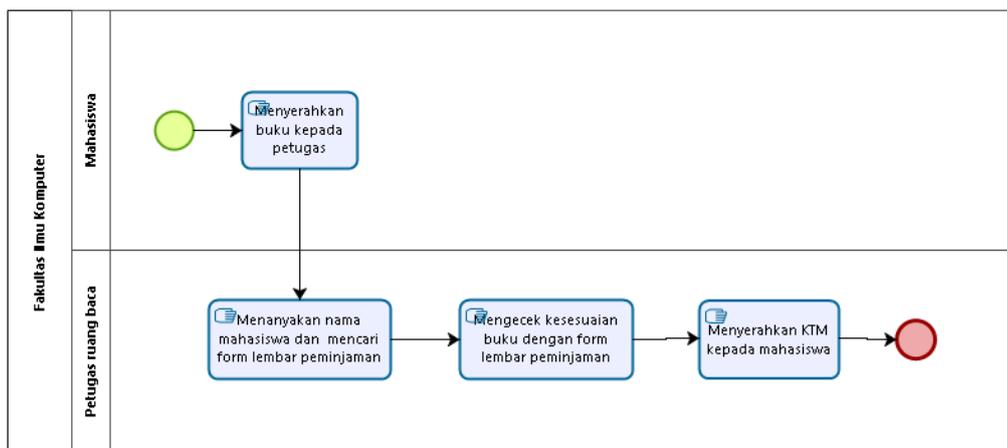


Gambar 4.1 Proses Bisnis As-Is Peminjaman Buku

4.1.1.2 Permodelan Proses Bisnis As-Is Pengembalian Buku

Permodelan proses bisnis *as-is* pengembalian buku merupakan gambarkan proses bisnis pengembalian buku yang sedang berjalan. Pada gambar 4.2 merupakan permodelan proses bisnis *as-is* pengembalian buku di Ruang Baca. Berikut ini merupakan penjelasan dari alur proses bisnis *as-is* pengembalian buku:

1. Mahasiswa datang ke Ruang Baca.
2. Mahasiswa menyerahkan buku yang akan dikembalikan kepada petugas ruang baca.
3. Petugas ruang baca menanyakan nama mahasiswa dan mencari *form* lembar peminjaman.
4. Petugas ruang baca melakukan pengecekan terkait kesesuaian buku yang dikembalikan dengan *form* lembar peminjaman.
5. Petugas ruang baca menyerahkan KTM kepada mahasiswa.



Gambar 4.2 Proses Bisnis As-Is Pengembalian Buku

4.1.2 Analisis Permasalahan

Analisis permasalahan menjelaskan masalah yang terjadi saat ini dalam proses peminjaman dan pengembalian buku sehingga ditemukan solusi yang dapat mengatasi masalah tersebut. Analisis permasalahan dilakukan berdasarkan hasil wawancara dengan bapak Syaifudin selaku petugas ruang baca serta proses bisnis *as-is*. Tabel 4.1 menjelaskan tentang analisis permasalahan mengenai peminjaman dan pengembalian buku di Ruang Baca Fakultas Ilmu Komputer.

Tabel 4.1 Analisis Permasalahan

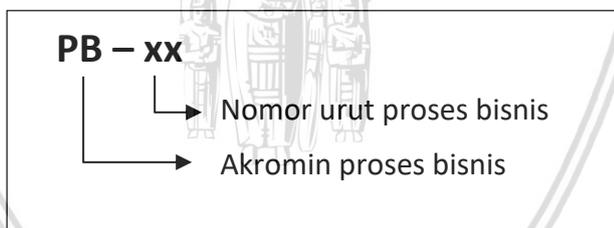
Masalah	<ol style="list-style-type: none"> 1. Sistem informasi di Ruang Baca berbasis desktop yang hanya dapat digunakan saat berada di Ruang Baca sesuai waktu operasional ruang baca. 2. Tidak adanya informasi mengenai jumlah ketersediaan jumlah buku dan penataan lokasi buku pada rak. 3. Belum adanya sistem pengelolaan denda terkait keterlambatan pengembalian buku oleh mahasiswa. 4. Dalam proses peminjaman buku, mahasiswa mengisi <i>form</i> peminjaman berupa kertas dan menyerahkan KTM kepada petugas ruang baca sebagai jaminan peminjaman buku.
Mempengaruhi	Petugas ruang baca, Mahasiswa
Dampak	<ol style="list-style-type: none"> 1. Mahasiswa harus datang langsung ke Ruang Baca untuk mengetahui koleksi buku yang ada di Ruang Baca. 2. Mahasiswa kesulitan mencari buku yang dicari karena informasi ketersediaan buku tidak tersedia dan harus mencari buku pada setiap rak yang ada.



Tabel 4.1 Analisis Permasalahan (lanjutan)

	<ol style="list-style-type: none"> 3. Adanya beberapa mahasiswa yang telat mengembalikan buku yang dipinjam, karena belum ada denda terkait pengembalian buku. 4. Petugas ruang baca harus mencari <i>form</i> peminjaman buku serta KTM secara satu-persatu ditempat penyimpanan ketika mahasiswa ingin mengembalikan buku yang telah dipinjam.
Solusi	<ol style="list-style-type: none"> 1. Menyediakan sistem informasi yang dapat diakses secara <i>online</i> sehingga dapat diakses 7 x 24 jam dan dimanapun mahasiswa berada. 2. Menyediakan sistem informasi yang dapat mengelola informasi koleksi buku, ketersediaan buku, transaksi peminjaman buku, dan pengembalian buku. 3. Menyediakan sistem informasi yang dapat mengelola denda terkait pengembalian buku. 4. Menyediakan sistem informasi yang dapat mengelola peminjaman dan pengembalian buku.

4.1.3 Permodelan Proses Bisnis *To-Be*



Gambar 4.3 Kodifikasi Proses Bisnis

Analisis proses bisnis usulan atau *to-be* didapatkan dari hasil analisis permasalahan diperoleh. Berdasarkan hasil analisis permasalahan yang telah diperoleh, didapatkan proses bisnis usulan yang telah di usulkan oleh peneliti. Proses bisnis usulan akan ditunjukkan pada tabel 4.2. Proses bisnis memiliki kode sesuai kodifikasi dalam gambar 4.3.

Tabel 4.2 Daftar Proses Bisnis Usulan (*To-Be*)

Kode Proses Bisnis	Nama	Deskripsi
PB-01	Peminjaman Buku	Proses bisnis terkait peminjaman buku dengan menggunakan Sistem Informasi Manajemen Perpustakaan.

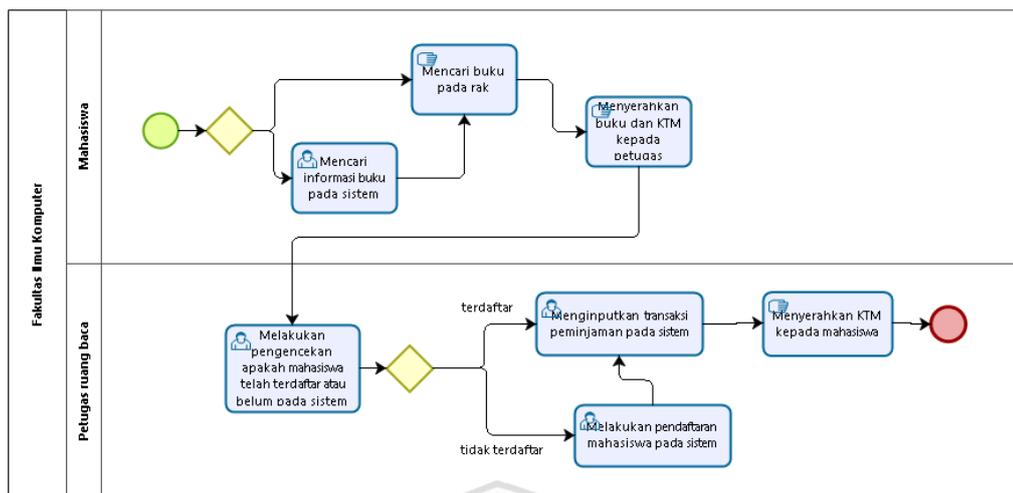
Tabel 4.2 Daftar Proses Bisnis Usulan (*To-Be*)

Kode Proses Bisnis	Nama	Deskripsi
PB-02	Pengembalian Buku	Proses bisnis terakit pengembalian buku dengan menggunakan Sistem Informasi Manajemen Perpustakaan.
PB-03	Pengelolaan Denda	Proses bisnis terakit pengelolaan denda dengan menggunakan <i>Web Sevice</i> Sistem Informasi Manajemen Perpustakaan.

4.1.3.1 Permodelan Proses Bisnis *To-Be* Peminjaman Buku

Permodelan proses bisnis *to-be* peminjaman buku merupakan gambarkan proses bisnis peminjaman buku yang diusukan oleh peneliti. Pada gambar 4.4 merupakan permodelan proses bisnis *to-be* peminjaman buku di Ruang Baca. Berikut ini merupakan penjelasan dari alur proses bisnis *to-be* peminjaman buku:

1. Mahasiswa datang ke Ruang Baca.
2. Mahasiswa dapat langsung mencari buku pada rak atau mencari informasi buku pada sistem, setelah itu mencari buku pada rak.
3. Mahasiswa menyerahkan buku dan KTM kepada petugas ruang baca.
4. Petugas ruang baca melakukan pengecekan apakah mahasiswa telah terdaftar sebelumnya atau belum pada sistem.
 - a. Jika mahasiswa telah terdaftar sebelumnya pada sistem.
 1. Petugas ruang baca menginputkan transaksi peminjaman pada sistem.
 2. Petugas ruang baca menyerahkan KTM kepada mahasiswa.
 - b. Jika mahasiswa belum terdaftar pada sistem.
 1. Petugas melakukan pendaftaran mahasiswa pada sistem.
 2. Petugas ruang baca menginputkan transaksi peminjaman pada sistem.
 3. Petugas ruang baca menyerahkan KTM kepada mahasiswa.



Gambar 4.4 Proses Bisnis To-Be Peminjaman Buku

Tabel 4.3 Perubahan Aktivitas Proses Bisnis Peminjaman Buku Berserta Perubahan Waktu

No.	Aktor	Task pada Proses Bisnis As-Is	Task pada Proses Bisnis To-Be	Waktu Sebelum	Waktu Sesudah
1	Mahasiswa	Mencari informasi buku pada sistem	Mencari informasi buku pada sistem	1 menit	1 menit
2	Mahasiswa	Mencari buku pada rak	Mencari buku pada rak	7 menit	1 menit
3	Mahasiswa	Menyerahkan buku pada petugas ruang baca	Menyerahkan buku dan KTM pada petugas ruang baca	3 detik	3 detik
4	Petugas ruang baca	Memberikan form lembar peminjaman pada mahasiswa	Melakukan pengecekan apakah mahasiswa telah terdaftar pada sistem	1 detik	3 detik
5	Mahasiswa	Mengisi form dan menyerahkan KTM kepada petugas ruang baca	-	2 menit	-
6	Petugas ruang baca	-	Melakukan pendaftaran mahasiswa pada sistem	-	2 mnt
7	Petugas ruang baca	-	Menginputkan transaksi peminjaman pada sistem	-	10 detik
8	Petugas ruang baca	-	Menyerahkan KTM kepada mahasiswa	-	1 detik

**Tabel 4.3 Perubahan Aktivitas Proses Bisnis Peminjaman Buku Berserta
Perubahan Waktu**

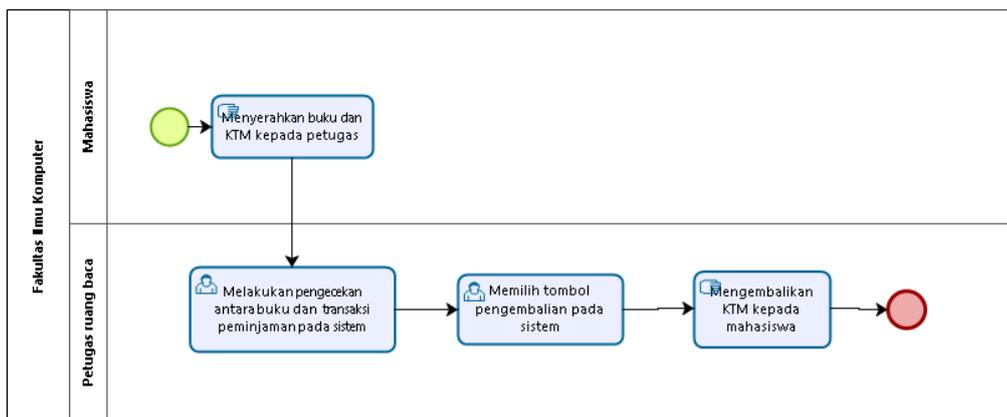
No.	Aktor	Task pada Proses Bisnis As-Is	Task pada Proses Bisnis To-Be	Waktu Sebelum	Waktu Sesudah
Total Waktu Sebelum				9 menit 5 detik	
Total Waktu Sesudah				4 menit 17 detik	
Selisih Waktu Sebelum dan Sesudah				4 menit 48 detik	

Pada Tabel 4.3 merupakan perubahan aktivitas proses bisnis peminjaman buku beserta perubahan waktu. Terdapat beberapa task yang mengalami perubahan pada aktor mahasiswa dan petugas ruang baca. Total waktu yang di habiskan oleh semua aktivitas proses bisnis *as-is* peminjaman buku sekitar 9 menit 5 detik. Total waktu yang di habiskan oleh semua aktivitas proses bisnis *to-be* peminjaman buku sekitar 4 menit 17 detik. Selisih perubahan waktu yang di habiskan semua aktivitas oleh proses bisnis *to-be* adalah 4 menit 48 detik.

4.1.3.2 Permodelan Proses Bisnis *To-Be* Pengembalian Buku

Permodelan proses bisnis *to-be* pengembalian buku merupakan gambarkan proses bisnis pengembalian buku yang diusukan oleh peneliti. Pada gambar 4.5 merupakan permodelan proses bisnis *to-be* pengembalian buku di Ruang Baca. Berikut ini merupakan penjelasan dari alur proses bisnis *to-be* pengembalian buku:

1. Mahasiswa datang ke Ruang Baca.
2. Mahasiswa menyerahkan buku yang akan dikembalikan kepada petugas ruang baca.
3. Petugas ruang baca melakukan pengecekan antara buku dengan transaksi peminjaman pada sistem.
4. Petugas ruang baca memilih tombol pengembalian pada sistem.
5. Petugas ruang baca mengembalikan KTM kepada mahasiswa.



Gambar 4.5 Proses Bisnis *To-Be* Pengembalian Buku

Tabel 4.4 Perubahan Aktivitas Proses Bisnis Pengembalian Buku Berserta Perubahan Waktu

No.	Aktor	Task pada Proses Bisnis As-Is	Task pada Proses Bisnis To-Be	Waktu Sebelum	Waktu Sesudah
1	Mahasiswa	Menyerahkan buku kepada petugas ruang baca	Menyerahkan buku dan KTM kepada petugas ruang baca	3 detik	3 detik
2	Petugas ruang baca	Menanyakan nama mahasiswa dan mencari form lembar peminjaman	Melakukan pengecekan antara buku dan transaksi peminjaman buku	3 menit	10 detik
3	Petugas ruang baca	Mengecek kesesuaian buku dengan form lembar peminjaman	Memilih tombol pengembalian pada sistem	5 detik	1 detik
4	Petugas ruang baca	Menyerahkan KTM kepada mahasiswa	Mengembalikan KTM kepada mahasiswa	1 detik	1 detik
Total Waktu Sebelum				3 menit 9 detik	
Total Waktu Sesudah				15 detik	
Selisih Waktu Sebelum dan Sesudah				2 menit 54 detik	

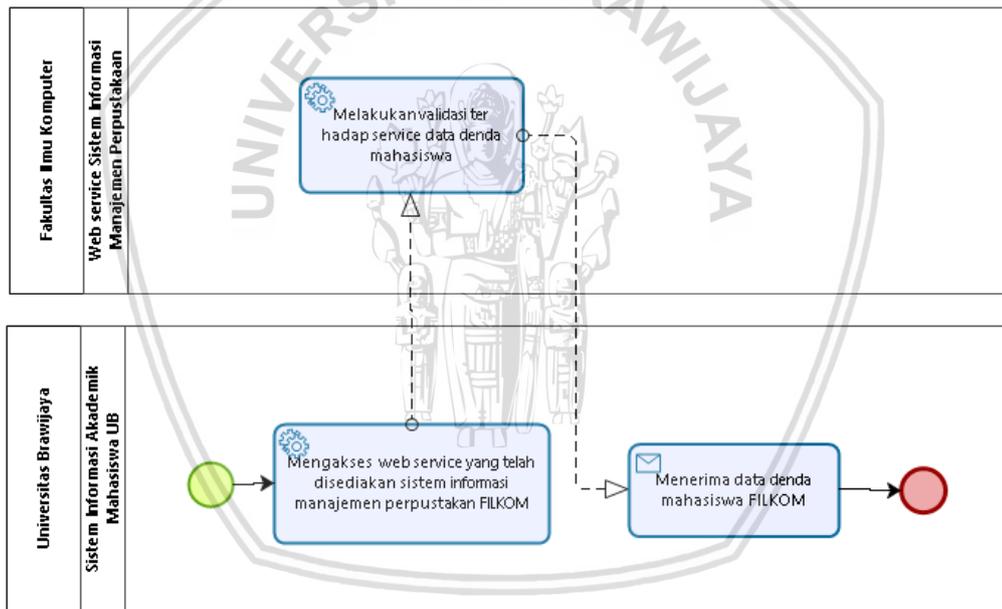
Pada Tabel 4.4 merupakan perubahan aktivitas proses bisnis pengembalian buku beserta perubahan waktu. Terdapat beberapa task yang mengalami perubahan pada aktor mahasiswa dan petugas ruang baca. Total waktu yang di habiskan oleh semua aktivitas proses bisnis *as-is* pengembalian buku sekitar 3 menit 9 detik. Total waktu yang di habiskan oleh semua aktivitas proses bisnis *to-*

be pengembalian buku sekitar 15 detik. Selisih perubahan waktu yang di habiskan semua aktivitas oleh proses bisnis *to-be* adalah 2 menit 54 detik.

4.1.3.3 Permodelan Proses Bisnis *To-Be* Pengelolaan Denda

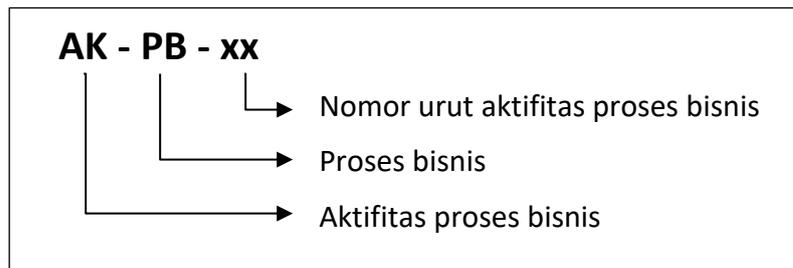
Permodelan proses bisnis *to-be* pengelolaan denda merupakan gambarkan proses bisnis pengelolaan denda yang diusukan oleh peneliti. Pada gambar 4.6 merupakan permodelan proses bisnis *to-be* pengelolaan denda di Ruang Baca. Berikut ini merupakan penjelasan dari alur proses bisnis *to-be* pengelolaan denda:

1. SIAM UB (Sistem Informasi Akademik Mahasiswa) melakukan akses terhadap *web service* Sistem Informasi Manajemen Perpustakaan.
2. *Web service* Sistem Informasi Manajemen Perpustakaan melakukan validasi terhadap *service* data denda mahasiswa.
3. SIAM UB (Sistem Informasi Akademik Mahasiswa) menerima data denda mahasiswa FILKOM yang nantinya akan ditampilkan pada biaya UKT (Uang Kuliah Tunggal) mahasiswa pada tiap semester.



Gambar 4.6 Proses Bisnis *To-Be* Pengelolaan Denda

4.2 Pemetaan Proses Bisnis



Gambar 4.7 Kodifikasi Aktivitas Proses Bisnis

Berdasarkan analisis proses bisnis usulan (*to-be*) yang telah dimodelkan oleh penulis, selanjutnya akan dijelaskan mengenai aktivitas-aktivitas yang akan didukung dengan penggunaan Sistem Informasi Manajemen Perpustakaan didalam tabel daftar aktifitas proses bisnis. Tabel 4.5 menunjukkan aktifitas proses bisnis berisi kode aktivitas proses bisnis, nama aktifitas proses bisnis, dan kode proses bisnis, dimana setiap aktifitas dapat dilakukan *traceability* terhadap proses bisnis yang telah diusulkan. Aktifitas proses bisnis memiliki kode sesuai dengan kodifikasi dalam gambar 4.7.

Tabel 4.5 Datar Aktivitas Proses Bisnis

Kode Proses Bisnis	Kode Aktifitas	Nama Aktifitas
PB-01	AK-PB-01	Mencari informasi data buku pada sistem.
PB-01	AK-PB-02	Menginputkan transaksi peminjaman buku pada sistem.
PB-02	AK-PB-03	Memilih tombol pengembalian buku pada sistem.
PB-02	AK-PB-04	Melakukan pendaftaran mahasiswa pada sistem.
PB-03	AK-PB-05	Melakukan validasi terhadap <i>service</i> data denda mahasiswa.

4.3 Identifikasi Tipe Pemangku Kepentingan

Identifikasi tipe pemangku pemangku kepentingan dilakukan untuk mengetahui dan mengelompokan beberapa pemangku kepentingan sesuai dengan karakteristik pemangku kepentingan dan hubungan kepentingan dengan sistem yang akan dikembangkan. Informasi mengenai pemangku kepentingan diperoleh dengan melakukan analisis terhadap wawancara dengan bapak Syaifudin selaku petugas ruang baca.

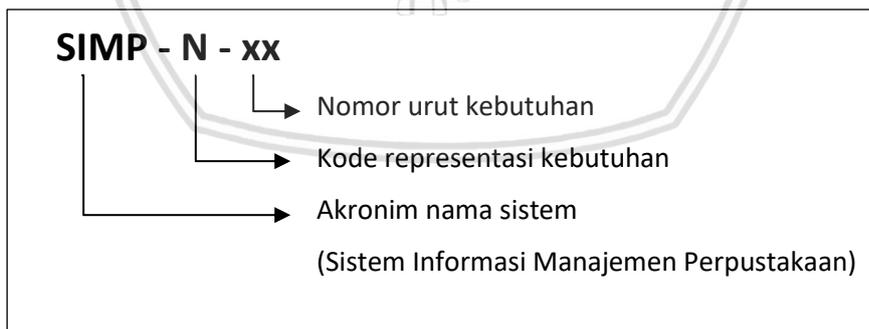
Hasil identifikasi tipe pemangku kepentingan daftar adalah pemangku kepentingan yang telah dikelompokan sesuai karakteristik dan hubungan pemangku kepentingan terhadap sistem yang akan dikembangkan dalam penelitian ini. Hasil identifikasi tipe pemangku kepentingan digunakan sebagai informasi untuk kegiatan identifikasi kebutuhan pemangku kepentingan dan pengguna sehingga peneliti dapat memperoleh masalah dari berbagai pemangku kepentingan yang ada pada lokasi studi kasus. Penjelasan tipe pemangku kepentingan terdapat pada tabel 4.6.

Tabel 4.6 Tipe Pemangku Kepentingan

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
Pengguna	Individu yang berinteraksi dengan sistem informasi secara langsung dengan berperan sabagai aktor dari <i>use case</i> .	Petugas ruang baca dan mahasiswa
Pengembang	Organisasi atau individu yang melakukan pengembangan sistem informasi manajemen perpustakaan.	Peneliti
Pihak yang berwenang	Organisasi atau individu yang memiliki kemampuan untuk mendukung pengembangan solusi dengan cara memberikan informasi mengenai proses bisnis yang berjalan.	Ruang Baca Fakultas Ilmu Komputer

4.4 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna

Informasi mengenai kebutuhan pengguna diperoleh dengan melakukan analisis terhadap permasalahan yang telah dijabarkan pada sub bab analisis permasalahan dan hasil wawancara dengan bapak Syaifudin selaku petugas ruang baca. Hasil dari identifikasi kebutuhan pengguna digunakan sebagai dasar informasi tahap analisis fitur, yang merupakan tahap untuk mengidentifikasi solusi yang ditawarkan untuk memenuhi kebutuhan pengguna. Masing-masing kebutuhan pengguna diberi kode sebagai identitas, keterangan kodifikasi kebutuhan pengguna terdapat pada gambar 4.8.



Gambar 4.8 Kodifikasi Kebutuhan Pengguna

Pada tabel 4.7 terdapat kode kebutuhan, pernyataan kebutuhan, pemangku kepentingan yang termasuk dalam kategori pengguna dan pengembang yang memiliki keterkaitan dengan kebutuhan yang telah teridentifikasi, 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 4.7 Kebutuhan Pemangku Kepentingan dan Pengguna

Kode Kebutuhan	Kebutuhan	Pemangku kepentingan	Situasi Saat Ini	Solusi yang Ditawakan	Kode Aktifitas
SIMP-N-01	Memberikan kemudahan dalam pencarian informasi buku kapanpun dan dimanapun.	Mahasiswa	Tidak ada	Sistem dapat diakses secara <i>online</i> dan dapat diakses selama terhubung dengan koneksi internet dan memiliki <i>browser</i> pada <i>device</i> yang digunakan.	AK-PB-01
SIMP-N-02	Memberikan informasi terkait data buku dan ketersediaan jumlah buku.	Mahasiswa	Sistem informasi berbasis desktop telah memberikan informasi buku dengan jelas tetapi tidak dilengkapi dengan informasi total ketersediaan buku.	Terdapat fitur untuk menginformasikan kepada mahasiswa terkait data informasi buku serta total ketersediaan buku yang masih ada.	AK-PB-01
SIMP-N-03	Memberikan kemudahan dalam melihat informasi transaksi peminjaman dan informasi denda yang diperoleh.	Mahasiswa	Tidak ada	Terdapat fitur untuk menampilkan informasi transaksi peminjaman yang pernah dilakukan serta denda yang diperoleh oleh mahasiswa.	-
SIMP-N-04	Memberikan kemudahan dalam mengelola peminjaman buku.	Petugas ruang baca	Dalam mengelola peminjaman buku masih bersifat manual dengan menggunakan lembar <i>form</i> peminjaman dan KTM sebagai jaminan peminjaman .	Terdapat fitur untuk mengelola peminjaman buku pada sistem, sehingga tidak perlu dilakukan secara manual.	AK-PB-02
SIMP-N-05	Memberikan kemudahan dalam mengelola pengembalian buku.	Petugas ruang baca	Dalam mengelola pengembalian buku masih bersifat manual dengan mencari lembar form	Terdapat fitur untuk mengelola pengembalian buku.	AK-PB-03

Tabel 4.7 Kebutuhan Pemangku Kepentingan dan Pengguna (Lanjutan)

Kode Kebutuhan	Kebutuhan	Pemangku kepentingan	Situasi Saat Ini	Solusi yang Ditawarkan	Kode Aktifitas
			peminjaman satu-persatu dalam melakukan <i>verifikasi</i>		
SIMP-N-06	Memberikan kemudahan dalam mengelola data buku.	Petugas ruang baca	Menggunakan sistem informasi berbasis desktop yang telah ada saat ini.	Terdapat fitur untuk mengelola data buku yang ada di Ruang Baca.	-
SIMP-N-07	Memberikan kemudahan dalam mencetak kode registrasi.	Petugas ruang baca	Petugas ruang baca dalam mencetak kode registrasi tidak sesuai format dan dibikin secara manual.	Terdapat fitur untuk mencetak kode registrasi secara otomatis dan sesuai format yang diberikan.	-
SIMP-N-08	Memberikan kemudahan dalam mengelola pengguna ruang baca dalam melakukan aktifitas peminjaman dan pengembalian buku.	Petugas ruang baca	Tidak ada	Terdapat fitur untuk pendaftaran kepada pengguna ruang baca.	AK-PB-04

4.5 Identifikasi Pengguna

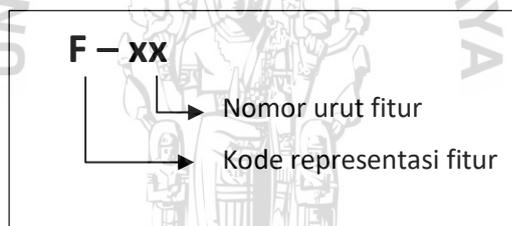
Identifikasi pengguna merupakan tahap yang dilakukan untuk mengetahui individu yang akan menggunakan sistem informasi secara langsung. Pengguna sistem merupakan bagian dari kelompok kategori pengguna dan pengembang dalam tipe pemangku kepentingan.

Hasil identifikasi pengguna diperoleh berdasarkan pemangku kepentingan yang tergolong didalam kategori pengguna dan pengembang dalam tipe pemangku kepentingan. Informasi mengenai identifikasi pengguna digunakan untuk melakukan identifikasi aktor yang akan dimodelkan kedalam *use case* diagram. Penjelasan hasil identifikasi pengguna akan ditunjukkan pada tabel 4.8.

Tabel 4.8 Identifikasi Pengguna

Tipe Pemangku Kepentingan	Perwakilan Pemangku Kepentingan	Deskripsi
Pengguna	Mahasiswa	Mahasiswa yang menggunakan sistem untuk melakukan pencarian informasi buku, informasi buku yang sedang dipinjam, dan informasi riwayat peminjaman.
	Petugas Ruang Baca	Petugas ruang baca yang bertugas untuk mengelola data buku, peminjaman buku, pengembalian buku, dan pendaftaran mahasiswa.
Pengembang	Petugas ruang baca	Orang yang melayani mahasiswa di Ruang Baca seperti pengelolaan buku, peminjaman buku, pengembalian buku, serta informasi buku.

4.6 Identifikasi Fitur



Gambar 4.9 Kodifikasi Fitur

Fitur yang teridentifikasi dalam penelitian ini merepresentasikan beberapa solusi yang ditawarkan untuk memenuhi kebutuhan pengguna, sehingga masalah yang dihadapi oleh pemangku kepentingan dapat diselesaikan. Hasil identifikasi fitur akan digunakan sebagai informasi untuk melakukan identifikasi persyaratan fungsional dan non-fungsional pada sistem informasi yang akan dikembangkan pada penelitian ini ditunjukkan pada tabel 4.9. Keterangan kodifikasi fitur terdapat pada gambar 4.9.

Tabel 4.9 Fitur Sistem

Kode Fitur	Nama Fitur	Deskripsi	Kode Kebutuhan
F-01	Mengelola peminjaman buku	Sistem dapat digunakan untuk mengelola peminjaman buku yang terdiri dari penyajian data peminjaman, penambahan data	SIMP-N-05

Tabel 4.9 Fitur Sistem (Lanjutan)

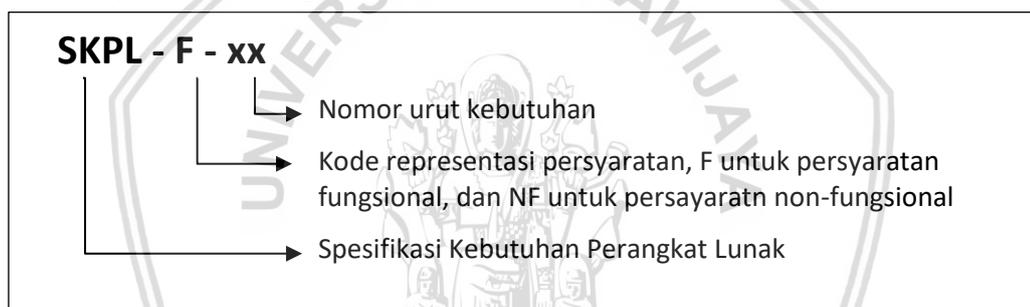
Kode Fitur	Nama Fitur	Deskripsi	Kode Kebutuhan
		peminjaman, dan penghapusan data peminjaman buku.	
F-02	Mengelola pengembalian buku	Sistem dapat digunakan untuk mengelola pengembalian buku dengan cara melakukan verifikasi pengembalian buku.	SIMP-N-06
F-03	Melakukan pencarian informasi buku	Sistem dapat digunakan untuk melakukan pencarian informasi buku.	SIMP-N-01, SIMP-N-02
F-04	Mengelola data buku	Sistem dapat digunakan untuk mengelola data buku yang terdiri dari penambahan data buku, penyajian data buku, perbaruan data buku, penghapusan data buku, dan mencetak kode registrasi buku	SIMP-N-07
F-05	Mengelola data mahasiswa	Sistem dapat digunakan untuk mengelola data mahasiswa yang terdiri dari penambahan, penyajian, perbaruan, dan penghapusan data mahasiswa.	SIMP-N-08
F-06	Mengelola data denda	Sistem dapat digunakan untuk mengelola data denda yang diperoleh oleh mahasiswa	SIMP-N-03
F-07	Melihat informasi buku yang dipinjam	Sistem dapat memberikan informasi terkait buku yang dipinjam oleh mahasiswa	SIMP-N-04
F-08	Melihat riwayat transaksi	Sistem dapat memberikan informasi terkait riwayat transaksi peminjaman dan pengembalian.	SIMP-N-03
F-09	Melakukan perubahan password	Sistem dapat digunakan untuk melakukan perubahan <i>password</i> mahasiswa.	SIMP-N-08
F-10	Melihat riwayat peminjaman dan denda	Sistem dapat memberikan informasi terkait riwayat peminjaman yang pernah dilakukan mahasiswa beserta denda yang diperoleh.	SIMP-N-04
F-11	Mengelola data rak buku	Sistem dapat digunakan untuk mengelola data rak buku yang terdiri dari penambahan, penyajian, perbaruan, dan penghapusan data rak buku.	SIMP-N-06



Tabel 4.9 Fitur Sistem (Lanjutan)

Kode Fitur	Nama Fitur	Deskripsi	Kode Kebutuhan
F-12	Melakukan proses otentikasi	Sistem dapat mengenali identitas pengguna sistem ketika masuk ke dalam sistem dan membatasi akses pengguna terhadap informasi serta layanan-layanan yang disediakan oleh sistem berdasarkan hak akses pengguna.	-
F-13	Melakukan proses keluar dari sistem	Sistem dapat digunakan oleh pengguna sistem untuk keluar dari sistem saat pengguna sistem meninggalkan sistem.	-

4.7 Persyaratan Fungsional



Gambar 4.10 Kodifikasi Persyaratan Fungsional

Bagian ini menjelaskan persyaratan fungsional sistem yang akan dikembangkan dalam penelitian ini. Persyaratan fungsional merupakan kondisi atau kemampuan yang harus dipenuhi oleh sistem agar fitur sistem dapat berjalan dan dapat memenuhi kebutuhan pengguna. Tabel 4.10 menunjukkan hubungan antara fitur dengan persyaratan fungsional pada sistem informasi manajemen perpustakaan yang akan dikembangkan. Persyaratan fungsional dan non-fungsional memiliki kode sesuai dengan kodifikasi dalam gambar 4.10.

Tabel 4.10 Persyaratan Fungsional

Kode Fitur	Kode Persyaratan Kebutuhan	Deskripsi
F-01	SKPL-F-01	Sistem harus menyediakan fungsi untuk menampilkan informasi data peminjaman buku.
	SKPL-F-02	Sistem harus menyediakan fungsi untuk menambahkan informasi data peminjaman buku.

Tabel 4.10 Persyaratan Fungsional(Lanjutan)

Kode Fitur	Kode Persyaratan Kebutuhan	Deskripsi
	SKPL-F-03	Sistem harus menyediakan fungsi untuk menghapus informasi data peminjaman buku.
F-02	SKPL-F-04	Sistem harus menyediakan fungsi untuk melakukan pengembalian buku.
F-03	SKPL-F-05	Sistem harus menyediakan fungsi untuk melakukan pencarian buku yang ada di Ruang Baca .
	SKPL-F-06	Sistem harus menyediakan fungsi untuk menampilkan informasi koleksi buku yang ada di Ruang Baca .
	SKPL-F-07	Sistem harus menyediakan fungsi untuk menampilkan informasi detail buku dan juga ketersediaan buku.
F-04	SKPL-F-08	Sistem harus menyediakan fungsi untuk menampilkan data buku.
	SKPL-F-09	Sistem harus menyediakan fungsi untuk menambahkan data buku.
	SKPL-F-10	Sistem harus menyediakan fungsi untuk memperbarui informasi data buku.
	SKPL-F-11	Sistem harus menyediakan fungsi untuk menghapus informasi data buku.
	SKPL-F-12	Sistem harus menyediakan fungsi untuk mencetak barcode buku.
F-05	SKPL-F-13	Sistem harus menyediakan fungsi untuk menampilkan data mahasiswa.
	SKPL-F-14	Sistem harus menyediakan fungsi untuk menambahkan data mahasiswa.
	SKPL-F-15	Sistem harus menyediakan fungsi untuk memperbarui data mahasiswa.
	SKPL-F-16	Sistem harus menyediakan fungsi untuk menghapus data mahasiswa.
F-06	SKPL-F-17	Sistem harus menyediakan fungsi untuk menampilkan total denda keseluruhan setiap mahasiswa.
	SKPL-F-18	Sistem harus menyediakan fungsi untuk menampilkan detail denda yang diperoleh oleh mahasiswa .

Tabel 4.10 Persyaratan Fungsional(Lanjutan)

Kode Fitur	Kode Persyaratan Kebutuhan	Deskripsi
F-07	SKPL-F-19	Sistem harus menyediakan fungsi untuk menampilkan informasi buku yang dipinjam mahasiswa saat ini.
F-08	SKPL-F-20	Sistem harus menyediakan fungsi untuk menampilkan informasi riwayat transaksi peminjaman dan pengembalian.
F-09	SKPL-F-21	Sistem harus menyediakan fungsi untuk melakukan perubahan <i>password</i> mahasiswa
F-10	SKPL-F-22	Sistem dapat memberikan informasi terkait riwayat peminjaman mahasiswa yang pernah dilakukan beserta denda yang diperoleh.
F-11	SKPL-F-23	Sistem harus menyediakan fungsi untuk menampilkan data rak buku
	SKPL-F-24	Sistem harus menyediakan fungsi untuk menambahkan data rak buku.
	SKPL-F-25	Sistem harus menyediakan fungsi untuk memperbarui informasi data rak buku.
	SKPL-F-26	Sistem harus menyediakan fungsi untuk menghapus informasi data rak buku.
F-12	SKPL-F-27	Sistem harus mengenali dan memberikan hak akses terhadap pengguna sesuai otorisasinya
F-13	SKPL-F-28	Sistem harus menghapus otorisasi pengguna saat meninggalkan sistem.

4.8 Persyaratan Non-Fungsional

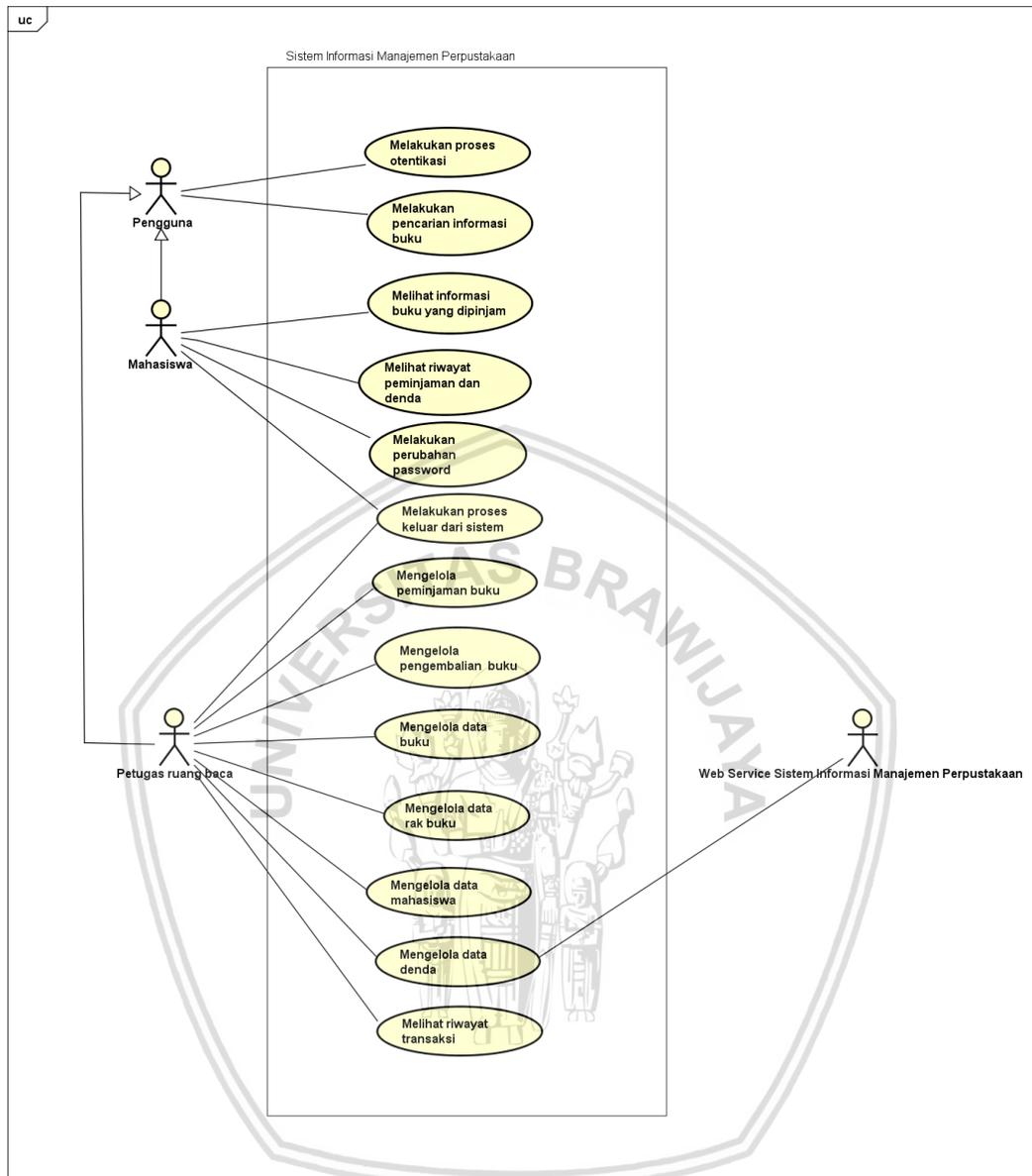
Bagian ini menjelaskan persyaratan yang harus dipenuhi oleh sistem untuk mendukung berjalannya sistem yang akan digunakan. Identifikasi persyaratan non-fungsional didapatkan berdasarkan observasi penggunaan aplikasi berbasis desktop yang digunakan pada ruang baca. Persyaratan non-fungsional tersebut dijelaskan pada Tabel 4.11.

Tabel 4.11 Persyaratan Non-Fungsional

Kode Fungsi	Nama Fungsi	Deskripsi
NF-01	Kompatibilitas	Sistem harus dapat berjalan dengan benar di <i>google chrome</i> , <i>mozilla firefox</i> , dan <i>microsoft edge</i> .



4.9 Pemodelan Use case Diagram



Gambar 4.11 Use Case Diagram

Use case diagram ditunjukkan pada gambar 4.11. Permodelan use case dilakukan setelah tahap analisis persyaratan. Pada pemodelan *use case*, dilakukan identifikasi aktor, identifikasi *use case*, dan spesifikasi untuk masing-masing *use case*. Identifikasi aktor dilakukan dengan mengelompokkan pengguna sistem berdasarkan karakteristik yang dimiliki, sedangkan identifikasi *use case* dilakukan untuk menentukan tujuan aktor dalam menggunakan sistem. Penentuan tujuan aktor dalam menggunakan sistem mengacu kepada pada fitur-fitur yang diharapkan oleh menggunakan sistem.

4.9.1 Spesifikasi *Use case*

Berikut ini merupakan spesifikasi dari masing-masing *use case* yang telah dijabarkan sebelumnya pada gambar 4.11 mengenai *use case* diagram sistem informasi manajemen perpustakaan:

4.9.1.1 Spesifikasi *use case* mengelola peminjaman buku

Tabel 4.12 Spesifikasi *Use case* mengelola peminjaman buku

<i>Flow of Events</i> untuk <i>use case</i> mengelola peminjaman buku	
Kode Fitur	F-01
Brief Description	<i>Use case</i> ini menjelaskan bagaimana aktor menggunakan sistem dalam mengelola peminjaman buku.
Actor	Petugas ruang baca
Pre-Condition	<ul style="list-style-type: none"> • Komputer yang digunakan terhubung dengan internet. • Sistem terhubung dengan server • Aktor pengguna berhasil melakukan proses otentikasi menjadi petugas ruang baca.
Basic Flow of Events	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor memilih menu pengelolaan transaksi dan submenu data peminjaman buku. 2. Sistem menampilkan data peminjaman buku. {menampilkan data peminjaman} 3. Jika aktor ingin melakukan penambahan data peminjaman, maka <i>sub flow menambahkan peminjaman</i> akan dijalankan. 4. Jika aktor ingin menghapus peminjaman, maka <i>sub flow menghapus peminjaman</i> akan dijalankan. <p>{use case selesai}</p>
Alternative Flows	<ul style="list-style-type: none"> • Form yang diisikan tidak lengkap Pada {nim,kode buku} pada <i>subflow</i> menambah peminjaman, jika aktor tidak mengisikan salah satu antara nim atau kode buku, maka sistem akan menampilkan pesan nim atau kode buku harus terisi. • Mahasiswa meminjam lebih dari 1 buku yang sama Pada {nim,kode buku} pada <i>subflow</i> menambah peminjaman, jika mahasiswa meminjam lebih dari 1 buku yang sama, maka sistem akan

	<p>menampilkan pesan mahasiswa tidak boleh meminjam lebih dari 1 buku yang sama.</p> <ul style="list-style-type: none"> • Mahasiswa meminjam lebih dari 3 buku yang berbeda Pada {nim,kode buku} pada <i>subflow</i> menambah peminjaman, jika mahasiswa meminjam lebih dari 3buku yang berbeda, maka sistem akan menampilkan pesan mahasiswa tidak boleh meminjam lebih dari 3 buku yang berbeda
Subflow	<ul style="list-style-type: none"> • Menambah peminjaman <ol style="list-style-type: none"> 1. Aktor memilih menu tambah peminjaman 2. Sistem menampilkan formulir peminjaman buku. 3. Aktor mengisikan nim dan kode buku{nim,kode buku} 4. Aktor melakukan verifikasi terhadap nama mahasiswa dan judul buku 5. Aktor menyimpan peminjaman buku 6. Sistem menyimpan data peminjaman buku dan menampilkan pesan peminjaman buku berhasil • Menghapus peminjaman <ol style="list-style-type: none"> 1. Aktor memilih salah satu data peminjaman buku yang akan dihapus pada {menampilkan data peminjaman} 2. Sistem menampilkan pesan konfirmasi penghapusan data peminjaman 3. Aktor melakukan verifikasi penghapusan data peminjaman 4. Data peminjaman terhapus dan sistem menampilkan pesan hapus peminjaman buku berhasil.
Post-Conditions	Aktor berhasil mengelola peminjaman buku.

4.9.1.2 Spesifikasi use case mengelola pengembalian buku

Tabel 4.13 Spesifikasi Use case mengelola pengembalian buku

Flow of Events untuk use case mengelola pengembalian buku	
Kode Fitur	F-02
Brief Description	Use case ini menjelaskan bagaimana aktor menggunakan sistem dalam mengelola pegembalian buku.



Actor	Petugas ruang baca
Pre-Condition	<ul style="list-style-type: none"> • Komputer yang digunakan terhubung dengan internet. • Sistem terhubung dengan server. • Aktor pengguna berhasil melakukan proses otentikasi menjadi petugas ruang baca.
Basic Flow of Events	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor memilih menu pengelolaan transaksi dan submenu data peminjaman buku. 2. Sistem menampilkan data peminjaman buku. 3. Aktor memilih salah satu data peminjaman buku yang akan dikembalikan. 4. Sistem menampilkan pesan konfirmasi pengembalian buku. 5. Aktor melakukan konfirmasi pengembalian buku. 6. Buku berhasil dikembalikan dan sistem menampilkan pesan pengembalian buku berhasil. <p>{use case selesai}</p>
Alternative Flows	Tidak ada
Subflow	Tidak ada
Post-Conditions	Aktor berhasil mengelola pengembalian buku.

4.9.1.3 Spesifikasi use case melakukan pencarian informasi buku

Tabel 4.14 Spesifikasi Use case melakukan pencarian informasi buku

Flow of Events untuk use case melakukan pencarian informasi buku	
Kode Fitur	F-03
Brief Description	Use case ini menjelaskan bagaimana aktor menggunakan sistem dalam melakukan pencarian informasi buku.
Actor	Pengguna
Pre-Condition	<ul style="list-style-type: none"> • Komputer yang digunakan terhubung dengan internet. • Sistem terhubung dengan server.
Basic Flow of Events	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor memilih koleksi buku yang akan dicari pada sistem. 2. Sistem menampilkan pilihan data buku berdasarkan koleksi buku dipilih. {menampilkan data buku} 3. Aktor memilih salah satu buku pada pilihan buku yang ada pada sistem

	<p>4. Sistem menampilkan informasi detail data buku. 5. Aktor mendapatkan informasi buku yang dicari. {use case selesai}</p>
Alternative Flows	<ul style="list-style-type: none"> • Melakukan pencarian informasi buku melalui pencarian buku Pada {use case dimulai} pada basic flow, jika aktor ingin melakukan pencarian informasi buku melalui pencarian buku, maka : <ol style="list-style-type: none"> 1. Aktor mengisikan kata kunci, jenis koleksi, dan pencarian berdasarkan. {kata kunci, jenis koleksi, pencarian berdasarkan} 2. Aktor melakukan pencarian buku. 3. Sistem menampilkan pilihan data buku. . {menampilkan data buku} 4. Aktor memilih salah satu buku pada pilihan buku pada sistem. 5. Sistem menampilkan informasi detail data buku. 6. Aktor mendapatkan informasi buku yang dicari. 7. <i>Use case selesai.</i> • {kata kunci, jenis koleksi, pencarian berdasarkan} tidak terisi Pada {kata kunci, jenis koleksi, pencarian berdasarkan} pada <i>alternatif flow</i>, jika aktor tidak mengisikan salah satu, maka sistem akan menampilkan pesan bahwa kata kunci atau jenis koleksi, pencarian berdasarkan harus terisi. • Data buku tidak tersedia Pada {menampilkan data buku} di <i>basic flow</i> dan <i>alternatif flow</i>, jika data buku tidak tersedia, maka sistem akan menampilkan informasi data buku tidak tersedia.
Subflow	Tidak ada
Post-Conditions	Aktor berhasil melakukan pencarian informasi buku

4.9.1.4 Spesifikasi *use case* mengelola data buku

Tabel 4.15 Spesifikasi *Use case* mengelola data buku

Flow of Events untuk use case mengelola data buku	
Kode Fitur	F-04



Brief Description	Use case ini menjelaskan bagaimana aktor menggunakan sistem dalam mengelola data buku.
Actor	Petugas ruang baca
Pre-Condition	<ul style="list-style-type: none"> • Komputer yang digunakan terhubung dengan internet. • Sistem terhubung dengan server • Aktor pengguna berhasil melakukan proses otentikasi menjadi petugas ruang baca.
Basic Flow of Events	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor memilih menu pengelolaan buku dan submenu data buku. 2. Sistem menampilkan koleksi data buku. <p>{menampilkan data buku}</p> <ol style="list-style-type: none"> 3. Jika aktor ingin melakukan penambahan data buku, maka <i>sub flow menambahkan data buku</i> akan dijalankan. 4. Jika aktor ingin melakukan perubahan data buku, maka <i>sub flow memperbarui data buku</i> akan dijalankan. 5. Jika aktor ingin menghapus data buku, maka <i>sub flow menghapus data buku</i> akan dijalankan. 6. Jika aktor ingin melihat detail data buku, maka <i>sub flow melihat detail data buku</i> akan dijalankan. 7. Jika aktor ingin mencetak barcode buku, maka <i>sub flow mencetak barcode data buku</i> akan dijalankan. <p>{use case selesai}</p>
Alternative Flows	<ul style="list-style-type: none"> • Form yang diisikan tidak lengkap Pada subflow Menambah data buku dan Memperbarui data buku, jika form yang diisikan aktor tidak lengkap, maka sistem akan menampilkan pesan form harus terisi.
Subflow	<ul style="list-style-type: none"> • Menambah data buku <ol style="list-style-type: none"> 1. Aktor memilih menu tambah buku. 2. Sistem menampilkan formulir tambah buku 3. Aktor mengisikan form tambah buku. 4. Aktor memilih tombol simpan 5. Sistem menyimpan data buku dan menampilkan pesan tambah buku berhasil • Memperbarui data buku



	<ol style="list-style-type: none"> 1. Aktor memilih tombol perbarui pada salah satu judul buku yang akan diperbarui {menampilkan data buku} 2. Sistem menampilkan form perbarui data buku 3. Aktor melakukan perubahan terhadap data buku. 4. Aktor memilih tombol simpan 5. Sistem menyimpan perubahan data buku dan menampilkan pesan perbarui data buku berhasil <ul style="list-style-type: none"> • Menghapus data buku <ol style="list-style-type: none"> 1. Aktor memilih tombol hapus pada salah satu data buku yang akan dihapus pada {menampilkan data buku} 2. Sistem menampilkan pesan konfirmasi penghapusan data buku 3. Aktor melakukan verifikasi penghapusan data buku 4. Data buku terhapus dan sistem menampilkan pesan data buku berhasil dihapus. • Melihat detail data buku <ol style="list-style-type: none"> 1. Aktor memilih salah satu judul buku yang akan dilihat detail data buku {menampilkan data buku} 2. Sistem menampilkan halaman detail data buku 3. Aktor melihat detail data buku. • Mencetak barcode buku <ol style="list-style-type: none"> 1. Aktor harus melakukan melihat detail data buku terlebih dahulu {melihat detail data buku} 2. Aktor memilih tombol cetak barcode 3. Sistem menampilkan barcode buku.
Post-Conditions	Aktor berhasil mengelola data buku.

4.9.1.5 Spesifikasi use case mengelola data mahasiswa

Tabel 4.16 Spesifikasi Use case mengelola data mahasiswa

Flow of Events untuk use case mengelola data mahasiswa	
Kode Fitur	F-05



Brief Description	Use case ini menjelaskan bagaimana aktor menggunakan sistem dalam mengelola data mahasiswa.
Actor	Petugas ruang baca
Pre-Condition	<ul style="list-style-type: none"> • Komputer yang digunakan terhubung dengan internet. • Sistem terhubung dengan server • Aktor pengguna berhasil melakukan proses otentikasi menjadi petugas ruang baca.
Basic Flow of Events	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor memilih menu pengelolaan mahasiswa. 2. Sistem menampilkan data mahasiswa. <p>{menampilkan data mahasiswa}</p> <ol style="list-style-type: none"> 3. Jika aktor ingin melakukan penambahan data mahasiswa, maka <i>sub flow menambahkan data mahasiswa</i> akan dijalankan. 4. Jika aktor ingin melakukan perubahan data mahasiswa, maka <i>sub flow memperbaiki data mahasiswa</i> akan dijalankan. 5. Jika aktor ingin menghapus data buku, maka <i>sub flow menghapus data mahasiswa</i> akan dijalankan. <p>{use case selesai}</p>
Alternative Flows	<ul style="list-style-type: none"> • Form yang diisikan tidak lengkap Pada subflow Menambah data mahasiswa dan Memperbarui data mahasiswa, jika form yang diisikan aktor tidak lengkap, maka sistem akan menampilkan pesan form harus terisi
Subflow	<ul style="list-style-type: none"> • Menambah data mahasiswa <ol style="list-style-type: none"> 1. Aktor memilih menu tambah buku. 2. Sistem menampilkan formulir tambah mahasiswa 3. Aktor mengisikan form tambah mahasiswa. 4. Aktor memilih tombol simpan 5. Sistem menyimpan data mahasiswa dan menampilkan pesan tambah data mahasiswa berhasil • Memperbarui data mahasiswa <ol style="list-style-type: none"> 1. Aktor memilih tombol perbarui pada salah satu data mahasiswa yang akan diperbarui <p>{menampilkan data mahasiswa}</p>

	<ol style="list-style-type: none"> 2. Sistem menampilkan form perbarui data mahasiswa 3. Aktor melakukan perubahan terhadap data mahasiswa. 4. Aktor memilih tombol simpan 5. Sistem menyimpan perubahan data mahasiswa dan menampilkan pesan perbarui data mahasiswa berhasil. <ul style="list-style-type: none"> • Menghapus data mahasiswa <ol style="list-style-type: none"> 1. Aktor memilih tombol hapus pada salah satu data mahasiswa yang akan dihapus pada {menampilkan data mahasiswa} 2. Sistem menampilkan pesan konfirmasi penghapusan data mahasiswa 3. Aktor melakukan verifikasi penghapusan data mahasiswa 4. Data mahasiswa terhapus dan sistem menampilkan pesan data mahasiswa berhasil dihapus.
Post-Conditions	Aktor berhasil mengelola data mahasiswa.

4.9.1.6 Spesifikasi use case mengelola data denda

Tabel 4.17 Spesifikasi Use case mengelola data denda

Flow of Events untuk use case mengelola data denda	
Kode Fitur	F-06
Brief Description	Use case ini menjelaskan bagaimana aktor menggunakan sistem dalam mengelola data denda yang diperoleh oleh mahasiswa.
Actor	Petugas Ruang Baca
Pre-Condition	<ul style="list-style-type: none"> • Komputer yang digunakan terhubung dengan internet. • Sistem terhubung dengan server. • Aktor pengguna berhasil melakukan proses otentikasi menjadi petugas ruang baca.
Basic Flow of Events	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor memilih menu pengelolaan denda. 2. Sistem menampilkan data denda setiap mahasiswa. 3. Jika aktor ingin melihat detail denda mahasiswa, maka <i>sub flow melihat detail denda</i> akan dijalankan



	{use case selesai}
Alternative Flows	Tidak ada
Subflow	<ul style="list-style-type: none"> • Melihat detail denda <ol style="list-style-type: none"> 1. Aktor memilih tombol detail pada salah satu mahasiswa 2. Sistem menampilkan data detail transaksi mahasiswa beserta denda yang diperoleh.
Post-Conditions	Aktor berhasil mengelola data denda.

4.9.1.7 Spesifikasi use case melihat informasi buku yang dipinjam

Tabel 4.18 Spesifikasi Use case melihat informasi buku yang dipinjam

Flow of Events untuk use case melihat informasi buku yang dipinjam	
Kode Fitur	F-07
Brief Description	Use case ini menjelaskan bagaimana aktor menggunakan sistem untuk melihat informasi terkait buku yang dipinjam .
Actor	Mahasiswa
Pre-Condition	<ul style="list-style-type: none"> • Komputer yang digunakan terhubung dengan internet. • Sistem terhubung dengan server. • Aktor pengguna berhasil melakukan proses otentikasi menjadi mahasiswa.
Basic Flow of Events	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor memilih data transaksi pada menu sistem. 2. Sistem menampilkan informasi buku yang dipinjam oleh mahasiswa. 3. Aktor melihat informasi buku yang dipinjam. <p>{use case selesai}</p>
Alternative Flows	Tidak ada
Subflow	Tidak ada
Post-Conditions	Aktor berhasil melihat informasi buku yang dipinjam.

4.9.1.8 Spesifikasi use case melihat riwayat transaksi

Tabel 4.19 Spesifikasi Use case melihat riwayat transaksi

Flow of Events untuk use case melihat riwayat transaksi	
Kode Fitur	F-08



Brief Description	<i>Use case</i> ini menjelaskan bagaimana aktor menggunakan sistem dalam melihat transaksi peminjaman dan pengembalian buku
Actor	Petugas ruang baca
Pre-Condition	<ul style="list-style-type: none"> • Komputer yang digunakan terhubung dengan internet. • Sistem terhubung dengan server • Aktor pengguna berhasil melakukan proses otentikasi menjadi petugas ruang baca.
Basic Flow of Events	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor memilih menu pengelolaan transaksi dan submenu data riwayat transaksi. 2. Sistem menampilkan data riwayat transaksi peminjaman dan pengembalian buku. <p>{use case selesai}</p>
Alternative Flows	Tidak ada
Subflow	Tidak ada
Post-Conditions	Aktor berhasil melihat riwayat transaksi.

4.9.1.9 Spesifikasi *use case* melakukan perubahan *password*

Tabel 4.20 Spesifikasi *Use case* melakukan perubahan *password*

Flow of Events untuk <i>use case</i> melakukan perubahan <i>password</i>	
Kode Fitur	F-09
Brief Description	<i>Use case</i> ini menjelaskan bagaimana aktor menggunakan sistem untuk melakukan perubahan <i>password</i> .
Actor	Mahasiswa
Pre-Condition	<ul style="list-style-type: none"> • Komputer yang digunakan terhubung dengan internet. • Sistem terhubung dengan server. • Aktor pengguna berhasil melakukan proses otentikasi menjadi mahasiswa.
Basic Flow of Events	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor memilih menu ubah <i>password</i> pada menu sistem. 2. Sistem menampilkan informasi terkait data diri aktor. 3. Aktor mengisikan <i>password</i> baru dan ulangi <i>password</i> baru. {mengisikan <i>password</i> baru}

	<p>4. Aktor melakukan penyimpanan perubahan password.</p> <p>5. Sistem menampilkan pesan perubahan password berhasil.</p> <p>{use case selesai}</p>
Alternative Flows	<ul style="list-style-type: none"> • { mengisikan password baru } tidak terisi Pada {mengisikan password baru}, jika aktor tidak mengisikan salah satu diantara <i>password</i> baru atau ulangi <i>password</i> baru, maka sistem akan menampilkan pesan <i>password</i> baru atau ulangi <i>password</i> baru harus terisi. • { mengisikan password baru } tidak terisi Pada {mengisikan password baru}, jika aktor tidak mengisikan <i>password</i> baru dan ulangi <i>password</i> baru tidak sesuai , maka sistem akan menampilkan pesan perubahan <i>password</i> gagal.
Subflow	Tidak ada
Post-Conditions	Aktor berhasil melihat melakukan perubahan <i>password</i> .

4.9.1.10 Spesifikasi use case melihat riwayat peminjaman dan denda

Tabel 4.21 Spesifikasi Use case melihat riwayat peminjaman dan denda

Flow of Events untuk use case melihat riwayat peminjaman dan denda	
Kode Fitur	F-10
Brief Description	<i>Use case</i> ini menjelaskan bagaimana aktor menggunakan sistem untuk melihat informasi terkait riwayat peminjaman dan denda yang diperoleh .
Actor	Mahasiswa
Pre-Condition	<ul style="list-style-type: none"> • Komputer yang digunakan terhubung dengan internet. • Sistem terhubung dengan server. • Aktor pengguna berhasil melakukan proses otentikasi menjadi mahasiswa.
Basic Flow of Events	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor memilih data transaksi pada menu sistem. 2. Sistem menampilkan informasi buku yang dipinjam oleh mahasiswa. 3. Aktor memilih tab menu riwayat peminjaman dan denda



	<p>4. Sistem menampilkan informasi terkait riwayat peminjaman dan denda aktor.</p> <p>5. Aktor melihat informasi riwayat peminjaman dan denda yang diperoleh.</p> <p>{use case selesai}</p>
Alternative Flows	Tidak ada
Subflow	Tidak ada
Post-Conditions	Aktor berhasil melihat informasi buku yang dipinjam.

4.9.1.11 Spesifikasi *use case* mengelola data rak buku

Tabel 4.22 Spesifikasi *Use case* mengelola data rak buku

Flow of Events untuk <i>use case</i> mengelola data rak buku	
Kode Fitur	F-11
Brief Description	<i>Use case</i> ini menjelaskan bagaimana aktor menggunakan sistem dalam mengelola data rak buku.
Actor	Petugas ruang baca
Pre-Condition	<ul style="list-style-type: none"> • Komputer yang digunakan terhubung dengan internet. • Sistem terhubung dengan server • Aktor pengguna berhasil melakukan proses otentikasi menjadi petugas ruang baca.
Basic Flow of Events	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor memilih menu pengelolaan buku dan submenu data rak.. 2. Sistem menampilkan koleksi data rak buku. {menampilkan data rak buku} 3. Jika aktor ingin melakukan penambahan data rak buku, maka <i>sub flow</i> menambahkan datarak buku akan dijalankan. 4. Jika aktor ingin melakukan perubahan data rak buku, maka <i>sub flow</i> memperbarui data rak buku akan dijalankan. 5. Jika aktor ingin menghapus data rak buku, maka <i>sub flow</i> menghapus data rak buku akan dijalankan. <p>{use case selesai}</p>
Alternative Flows	<ul style="list-style-type: none"> • Form yang diisikan tidak lengkap Pada subflow Menambah data rak buku dan Memperbarui data rak buku, jika form yang diisikan aktor tidak lengkap, maka sistem akan menampilkan pesan form harus terisi.



<p>Subflow</p>	<ul style="list-style-type: none"> • Menambah data rak buku <ol style="list-style-type: none"> 1. Aktor memilih menu tambah rak buku. 2. Sistem menampilkan formulir tambah rak buku 3. Aktor mengisikan form tambah rak buku. 4. Aktor memilih tombol simpan 5. Sistem menyimpan data rak buku dan menampilkan pesan tambah rak buku berhasil • Memperbarui data rak buku <ol style="list-style-type: none"> 1. Aktor memilih tombol perbarui pada salah satu rak buku yang akan diperbarui {menampilkan data rak buku} 2. Sistem menampilkan form perbarui data rak buku 3. Aktor melakukan perubahan terhadap data rak buku. 4. Aktor memilih tombol simpan 5. Sistem menyimpan perubahan data rak buku dan menampilkan pesan perbarui data rak buku berhasil • Menghapus data rak buku <ol style="list-style-type: none"> 1. Aktor memilih tombol hapus pada salah satu data rak buku yang akan dihapus pada {menampilkan data rak buku} 2. Sistem menampilkan pesan konfirmasi penghapusan data rak buku 3. Aktor melakukan verifikasi penghapusan data rak buku 4. Data buku terhapus dan sistem menampilkan pesan data rak buku berhasil dihapus.
<p>Post-Conditions</p>	<p>Aktor berhasil mengelola data rak buku.</p>

4.9.1.12 Spesifikasi use case melakukan proses autentikasi

Tabel 4.23 Spesifikasi Use case melakukan proses autentikasi

<p>Flow of Events untuk use case melakukan proses autentikasi</p>	
<p>Kode Fitur</p>	<p>F-12</p>



Brief Description	Use case ini menjelaskan bagaimana aktor menggunakan sistem untuk melakukan verifikasi terhadap identitasnya, sehingga aktor dapat mengakses sistem sesuai dengan hak aksesnya.
Actor	Pengguna
Pre-Condition	<ul style="list-style-type: none"> • Komputer yang digunakan terhubung dengan internet. • Sistem terhubung dengan server.
Basic Flow of Events	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor mengisikan nim dan <i>password</i>. {nim dan password} 2. Aktor mengirimkan nim dan <i>password</i> pada sistem. 3. Sistem mengidentifikasi identitas pengguna. {mengidentifikasi pengguna} 4. Sistem menyimpan informasi pengguna. 5. Sistem menampilkan informasi berdasarkan hak akses pengguna. <p>{use case selesai}</p>
Alternative Flows	<ul style="list-style-type: none"> • {nim dan password } tidak terisi Pada {nim dan password} di <i>basic flow</i>, jika aktor tidak mengisikan nim atau password, maka sistem akan menampilkan pesan bahwa nim atau password harus terisi. • {nim dan password} tidak sesuai Pada {nim dan password} di <i>basic flow</i>, jika aktor mengisikan nim dan <i>password</i> tidak sesuai, maka sistem akan menampilkan pesan bahwa nim atau <i>password</i> salah.
Subflow	Tidak ada
Post-Conditions	<ul style="list-style-type: none"> • Aktor berhasil melakukan otentikasi. • Sistem menampilkan informasi sesuai dengan hak aksesnya.

4.9.1.13 Spesifikasi use case melakukan proses keluar dari sistem

Tabel 4.24 Spesifikasi Use case melakukan proses keluar dari sistem

Flow of Events untuk use case melakukan proses keluar dari sistem	
Kode Fitur	F-13
Brief Description	Use case ini menjelaskan bagaimana aktor dapat keluar dari sistem.



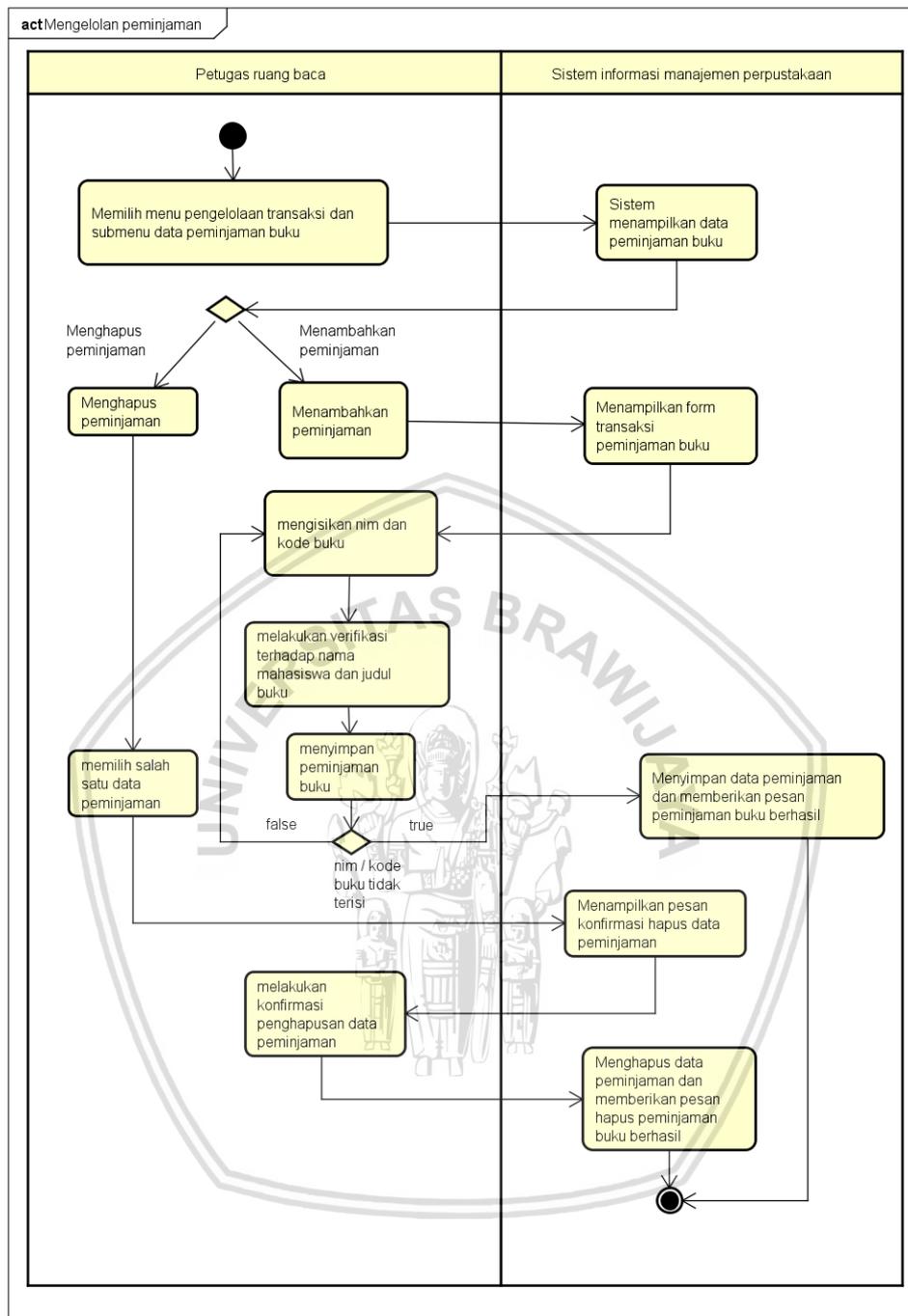
Actor	Petugas ruang baca dan mahasiswa
Pre-Condition	<ul style="list-style-type: none"> • Komputer yang digunakan terhubung dengan internet. • Sistem terhubung dengan server. • Aktor pengguna berhasil melakukan proses otentikasi menjadi petugas ruang baca atau mahasiswa
Basic Flow of Events	{use case dimulai} 1. Aktor memilih tombol logout 2. Sistem menampilkan halaman utama sistem {use case selesai}
Alternative Flows	Tidak ada
Subflow	Tidak ada
Post-Conditions	Aktor berhasil keluar dari sistem

4.10 Permodelan Activity Diagram

Activity diagram menggambarkan alur aktivitas dari sebuah proses atau aktivitas. *Activity diagram* mengacu kepada *use case scenario*. Berikut ini merupakan activity diagram sistem informasi manajemen perpustakaan:

4.10.1 Activity diagram mengelola peminjaman buku

Pada Gambar 4.12 merupakan diagram aktivitas dari *use case* mengelola peminjaman buku. Diagram tersebut menggambarkan alur interaksi yang dilakukan petugas ruang baca dengan sistem informasi manajemen perpustakaan pada saat mahasiswa melakukan peminjaman buku.

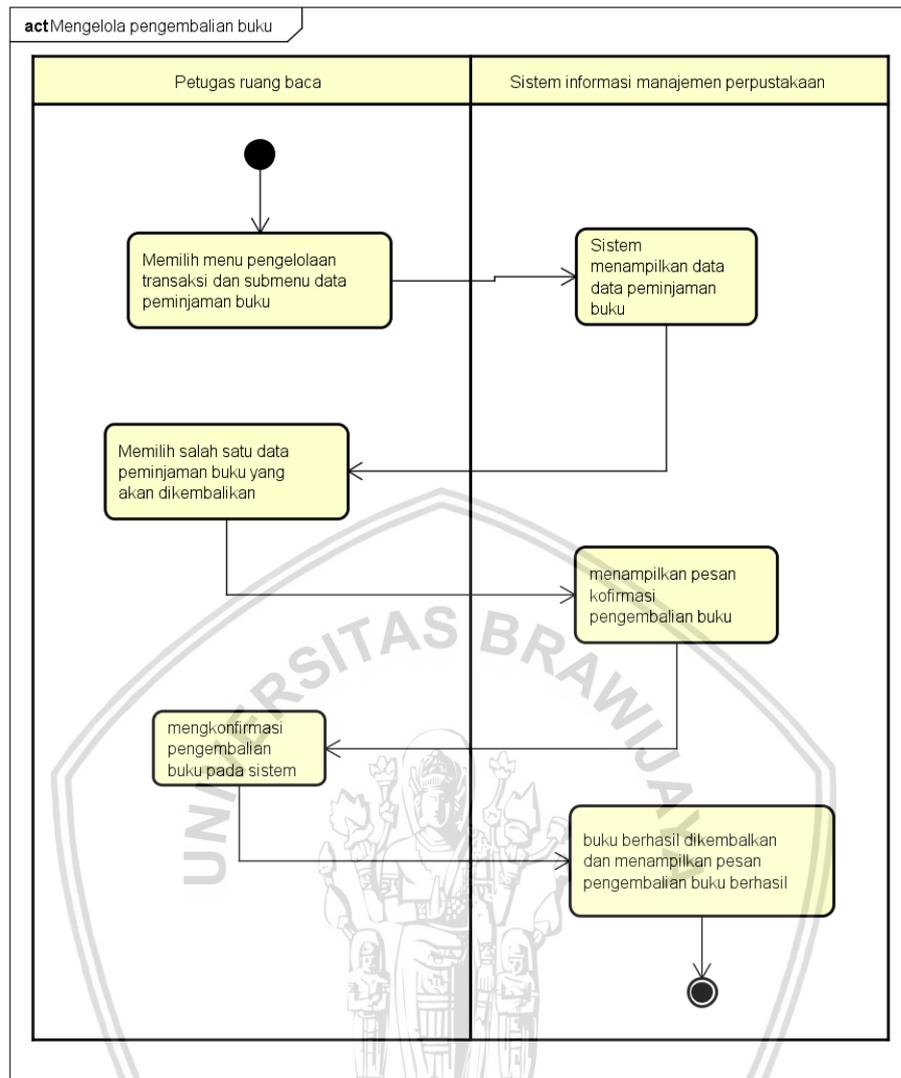


Gambar 4.12 Activity Diagram Mengelola Peminjaman Buku

4.10.2 Activity diagram mengelola pengembalian buku

Pada Gambar 4.13 merupakan diagram aktivitas dari *use case* mengelola pengembalian buku. Diagram tersebut menggambarkan alur interaksi yang dilakukan petugas ruang baca dengan sistem informasi manajemen perpustakaan pada saat mahasiswa melakukan pengembalian buku.

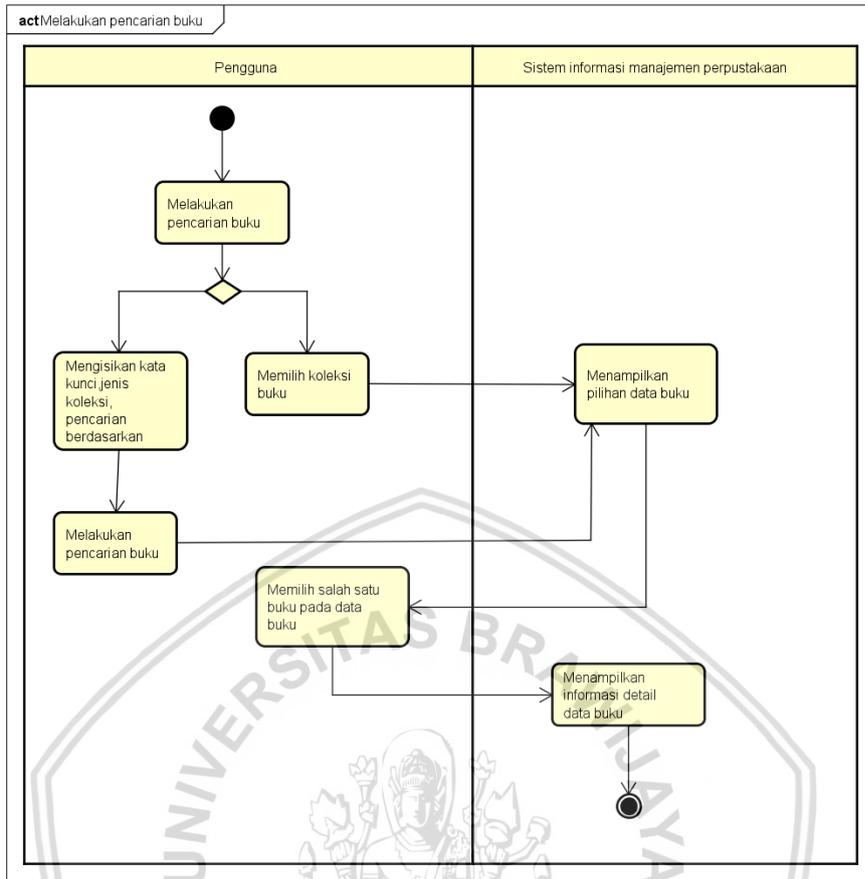




Gambar 4.13 Activity Diagram Mengelola Pengembalian Buku

4.10.3 Activity diagram melakukan pencarian buku

Pada Gambar 4.14 merupakan diagram aktivitas dari *use case* melakukan pencarian buku. Diagram tersebut menggambarkan alur interaksi yang dilakukan pengguna dengan sistem informasi manajemen perpustakaan pada saat ingin melakukan pencarian buku. Pengguna merupakan generalisasi dari mahasiswa dan petugas ruang baca yang berperan sebagai pengunjung sistem informasi manajemen perpustakaan.

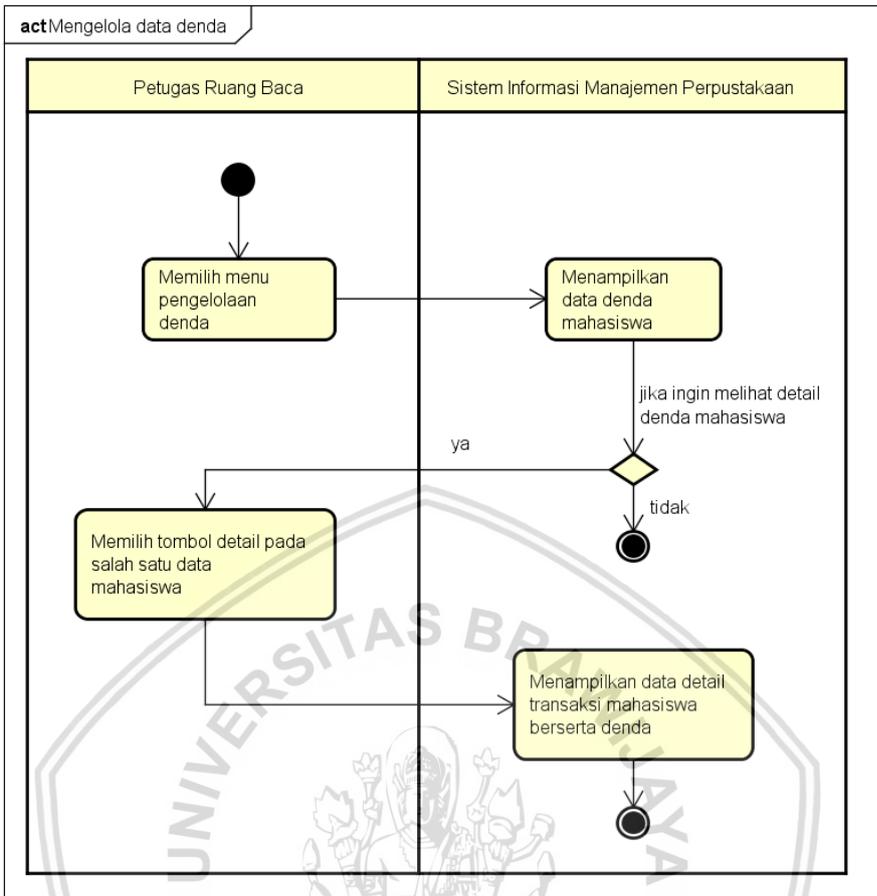


Gambar 4.14 Activity Diagram Melakukan Pencarian Buku

4.10.4 Activity diagram mengelola data denda

Pada Gambar 4.15 merupakan diagram aktivitas dari *use case* untuk mengelola data denda. Diagram tersebut menggambarkan alur interaksi yang dilakukan petugas ruang baca dengan sistem informasi manajemen perpustakaan pada saat ingin melakukan pencarian buku. Petugas ruang baca dapat melihat data denda setiap mahasiswa dan dapat melihat data detail denda dari setiap mahasiswa.





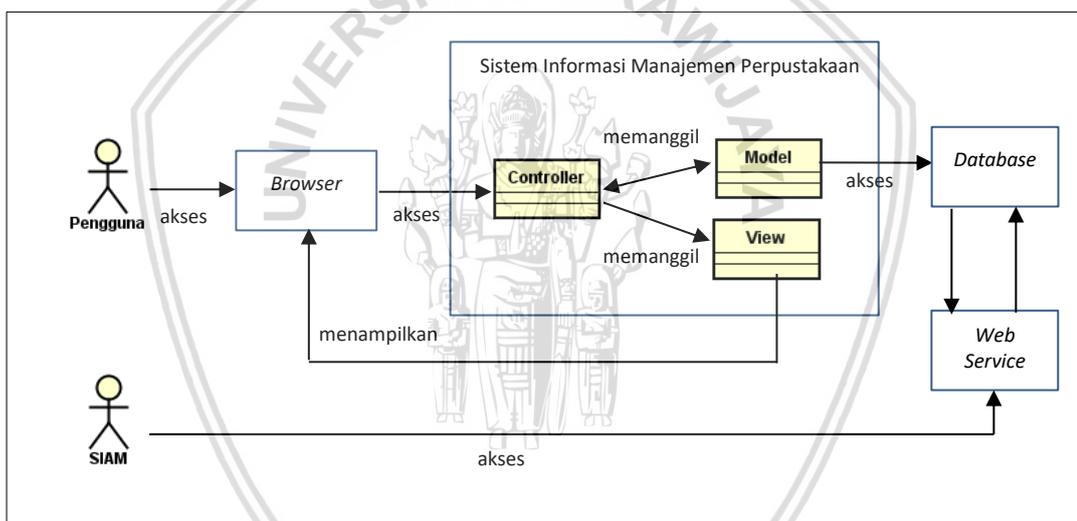
Gambar 4.15 Activity Diagram Mengelola Data Denda

BAB 5 PERANCANGAN

Pada bab perancangan berisi pembahasan mengenai hasil pemodelan dengan sistem dengan menggunakan *Unified Modelling Language* (UML) sebagai notasi pemodelan. Hasil pemodelan yang dijelaskan dalam bab ini antara lain sequence diagram, class diagram, perancangan basis data yang didokumentasikan ke dalam PDM, perancangan algoritma yang didokumentasikan ke dalam *pseudocode*, dan perancangan antarmuka pengguna.

5.1 Perancangan Arsitektur Sistem

Perancangan arsitektur sistem bertujuan untuk menggambarkan bagaimana interaksi antar komponen-komponen Sistem Informasi Manajemen Perpustakaan dalam membantu pengguna sistem dalam menjalankan aktifitasnya. Perancangan arsitektur Sistem Informasi Manajemen Perpustakaan dapat dilihat pada Gambar 5.1 berikut.



Gambar 5.1 Arsitektur Sistem

Pada gambar 5.1 sistem diakses oleh pengguna melalui *browser*. Sistem dibangun dan dikembangkan dengan menerapkan *framework* Codeigniter, yang mana *framework* ini menerapkan pola perancangan *Model-View-Controller* (MVC). Sistem menyediakan *web service* untuk pertukaran data dengan sistem lain dengan menerapkan REST API. Web Service yang ditujukan pada sistem lain yaitu Sistem Informasi Akademik Mahasiswa (SIAM) Universitas Brawijaya agar dapat mengakses dan mendapatkan data denda yang didapatkan oleh mahasiswa ketika telat dalam melakukan pengembalian buku. Data denda tersebut nantinya akan dimasukkan kedalam tagihan biaya uang kuliah tunggal (UKT) mahasiswa pada tiap semester.

5.2 Perancangan *Sequence Diagram*

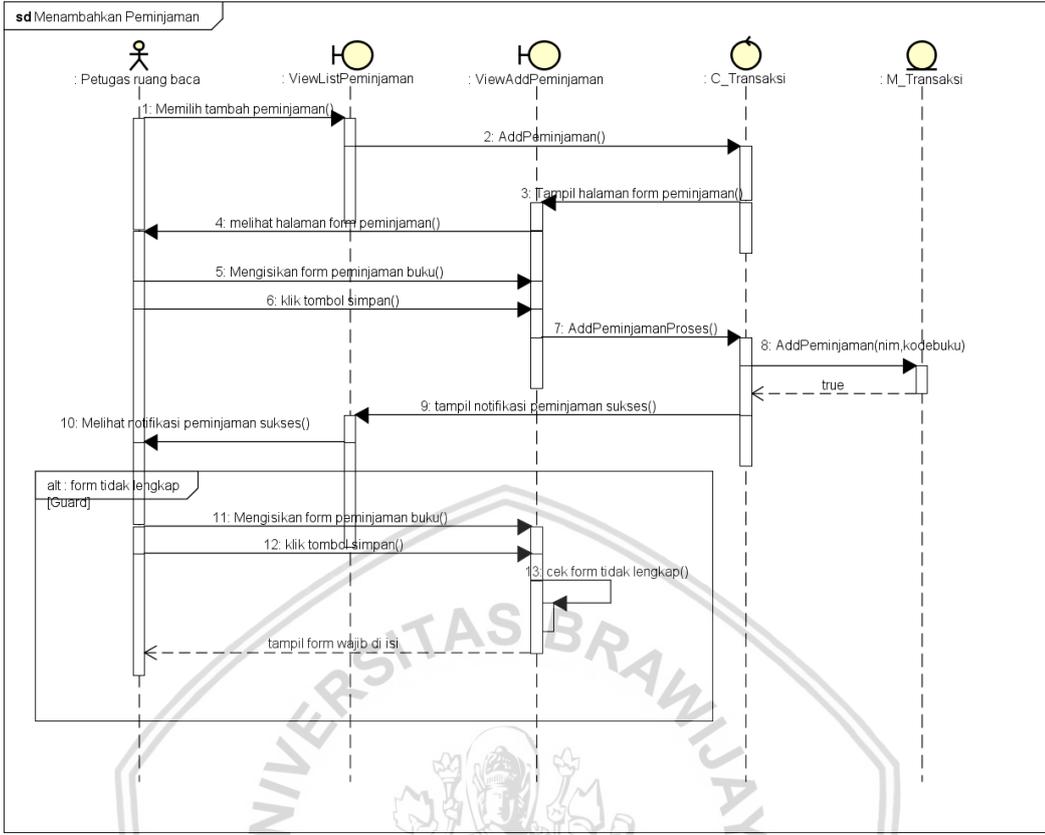
Pada bagian ini berisi penggambaran alur proses yang terjadi di dalam Sistem Informasi Manajemen Perpustakaan. Pada perancangan *sequence diagram* berdasarkan spesifikasi *use case* spesifikasi yang telah didefinisikan pada saat proses analisis persyaratan fungsional Sistem Informasi Manajemen Perpustakaan. Berikut merupakan beberapa *sequence diagram* yang diambil berdasarkan fitur utama yang mewaliki dari Sistem Informasi Manajemen Perpustakaan.

5.2.1 *Sequence Diagram* Mengelola Peminjaman Buku

Pada subbab ini berisi beberapa *sequence diagram* yang memvisualisasi interaksi objek yang terlibat dalam proses peminjaman buku. Proses mengelola peminjaman buku dibagi menjadi beberapa *sequence diagram* untuk memudahkan memvisualisasi proses. *Sequence diagram* mengelola peminjaman buku terdiri dari *sequence diagram* menambahkan peminjaman, *sequence diagram* menghapus peminjaman, *sequence diagram* menampilkan peminjaman.

5.2.1.1 *Sequence Diagram* Menambahkan Peminjaman Buku

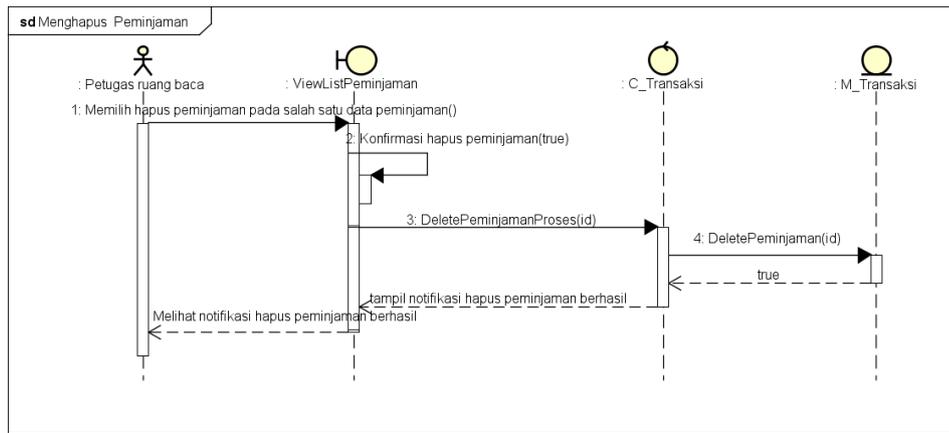
Sequence diagram yang terdapat pada gambar 5.2 merupakan visualisasi interaksi antar objek pada saat petugas ruang baca menggunakan sistem untuk menambahkan peminjaman buku. Beberapa objek yang terlibat di dalam interaksi untuk menambahkan peminjaman diantaranya adalah aktor petugas ruang baca, *ViewListPeminjaman* sebagai objek *boundary*, *ViewAddPeminjaman* sebagai objek *boundary*, *C_Transaksi* sebagai objek *control*, dan *M_Transaksi* sebagai objek *model*.



Gambar 5.2 Sequence Diagram Menambahkan Peminjaman Buku

5.2.1.2 Sequence Diagram Menghapus Peminjaman

Sequence diagram yang terdapat pada gambar 5.3 merupakan visualisasi interaksi antar objek pada saat petugas ruang baca menggunakan sistem untuk menghapus peminjaman buku. Pertukaran pesan dimulai ketika aktor petugas ruang baca memilih salah satu data peminjaman melalui object ViewListPeminjaman sebagai objek *boundary* sistem. Sistem memberikan notifikasi konfirmasi terkait penghapusan data peminjaman kepada petugas ruang baca. Ketika petugas ruang baca sudah mengkonfirmasi, maka pesan diteruskan kepada kelas C_Transaksi sebagai objek *control*. Kemudian objek *control* berinteraksi dengan objek model yaitu M_Transaksi.

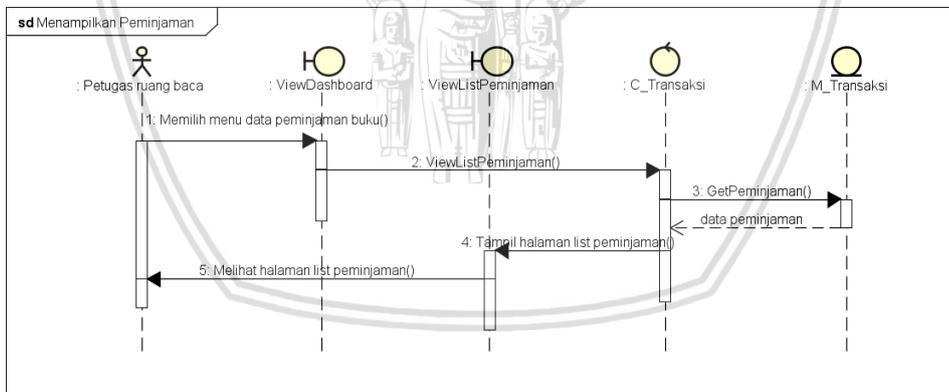


powered by Astah

Gambar 5.3 Sequence Diagram Menghapus Peminjaman Buku

5.2.1.3 Sequence Diagram Melihat Data Peminjaman

Sequence diagram yang terdapat pada gambar 5.4 merupakan visualisasi interaksi antar objek pada saat petugas ruang baca menggunakan sistem untuk melihat peminjaman buku. Pertukaran pesan dimulai ketika aktor petugas ruang baca memilih menu data peminjaman buku melalui object ViewDashboard sebagai objek *boundary* sistem. Kemudian pesan diteruskan kepada kelas C_Transaksi sebagai objek *control*. Objek *control* berinteraksi dengan objek model yaitu M_Transaksi. Setelah itu objek *control* berinteraksi dengan Objek *boundary* yaitu ViewListPeminjaman.



powered by Astah

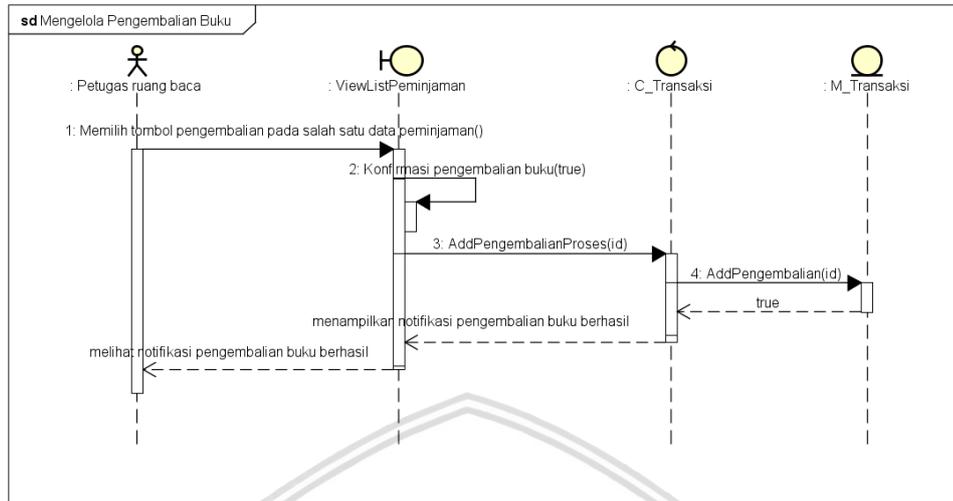
Gambar 5.4 Sequence Diagram Melihat Peminjaman Buku

5.2.2 Sequence Diagram Mengelola Pengembalian Buku

Sequence diagram yang terdapat pada gambar 5.5 merupakan visualisasi interaksi antar objek pada saat petugas ruang baca menggunakan sistem untuk pengembalian buku. Pertukaran pesan dimulai ketika aktor petugas ruang baca memilih salah satu data peminjaman melalui object ViewListPeminjaman sebagai objek *boundary* sistem. Sistem memberikan notifikasi konfirmasi terkait pengembalian data buku kepada petugas ruang baca. Ketika petugas ruang baca sudah mengkonfirmasi, maka pesan diteruskan kepada kelas C_Transaksi sebagai



objek *control*. Kemudian objek *control* berinteraksi dengan objek model yaitu M_Transaksi.



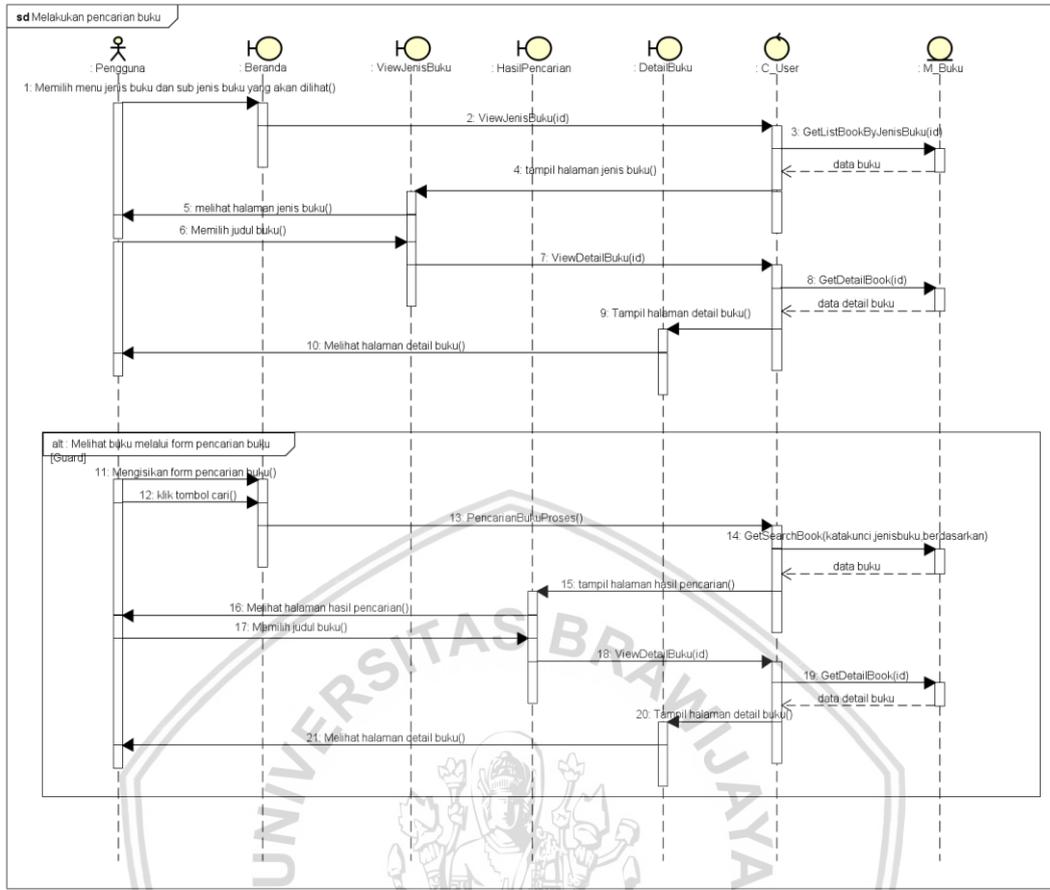
Gambar 5.5 Sequence Diagram Pengembalian Buku

5.2.3 Sequence Diagram Melakukan Pencarian Buku

Sequence diagram yang terdapat pada gambar 5.6 merupakan visualisasi interaksi antar objek pada saat pengguna menggunakan sistem untuk melakukan pencarian buku. Dalam melakukan pencarian buku dapat dilakukan melalui menu jenis buku dan melalui *form* pencarian buku. Pertukaran pesan pencarian buku melalui menu jenis buku dimulai ketika aktor pengguna memilih jenis buku dan sub jenis buku sesuai buku yang akan dilihat pada Beranda sebagai objek *boundary*. Lalu pesan diteruskan kepada kelas C_User sebagai objek *control*. Kemudian objek *control* berinteraksi dengan objek model yaitu M_Buku. Setelah itu akan ditampilkan data buku berdasarkan jenis buku dan sub jenis buku yang telah dipilih pada ViewJenisBuku sebagai objek *boundary*. Jika pengguna ingin melihat detail data buku, pengguna memilih judul buku yang diinginkan. Pesan akan diteruskan kepada objek control yaitu C_User dan M_Buku dalam pengambilan data detail buku. Data detail buku tersebut akan ditampilkan pada DetailBuku sebagai objek *boundary*.

Sedangkan pertukaran pesan pencarian buku melalui *form* pencarian dimulai ketika aktor pengguna mengisikan *form* pencarian dan memilih tombol cari pada Beranda sebagai objek *boundary*. Setelah itu pesan diteruskan kepada C_User sebagai objek *control*. Objek *control* berkomunikasi kepada M_Buku sebagai objek *model* dalam pengambilan data buku berdasarkan data *form* pencarian yang telah diisikan. Hasil data buku tersebut akan ditampilkan melalui HasilPencarian sebagai objek *boundary*. Jika pengguna ingin melihat detail data buku, pengguna memilih judul buku yang diinginkan. Pesan akan diteruskan kepada objek control yaitu C_User dan M_Buku dalam pengambilan data detail buku. Data detail buku tersebut akan ditampilkan pada DetailBuku sebagai objek *boundary*.





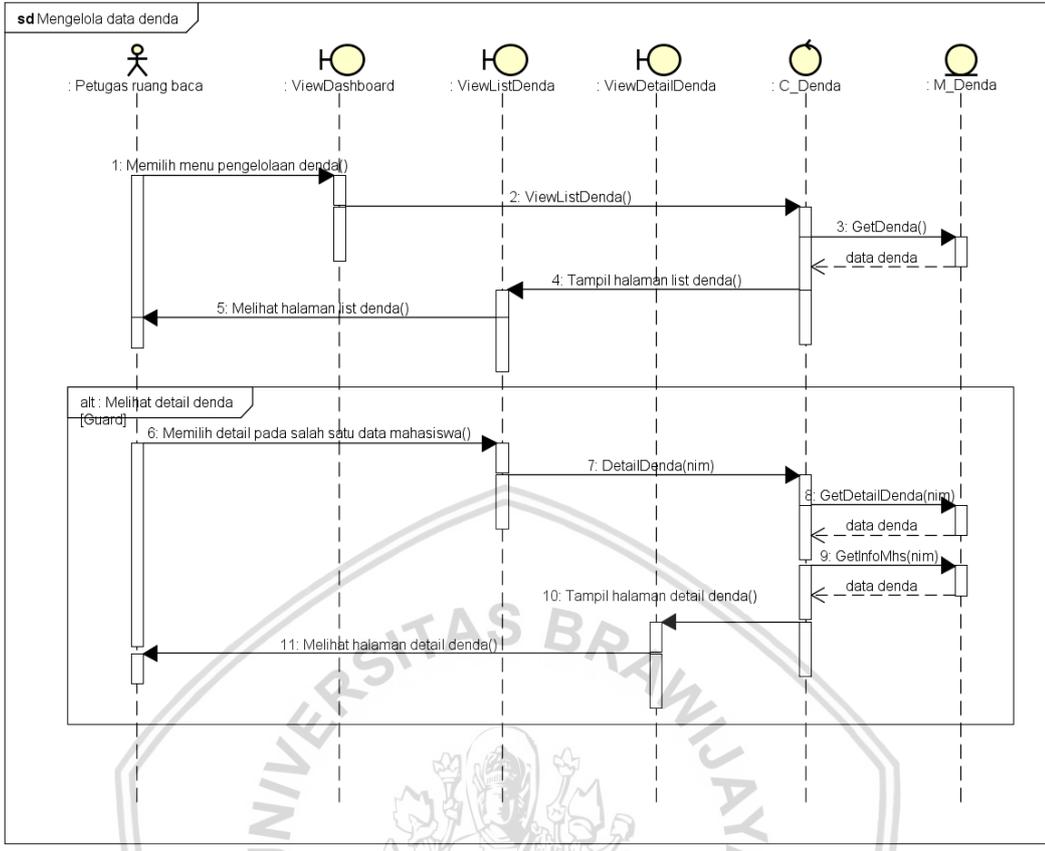
Gambar 5.6 Sequence Diagram Melakukan Pencarian Buku

5.2.4 Sequence Diagram Mengelola Data Denda

Sequence diagram yang terdapat pada gambar 5.7 merupakan visualisasi interaksi antar objek pada saat petugas ruang baca menggunakan sistem untuk melakukan mengelola data denda. Pertukaran pesan dimulai ketika petugas ruang baca memilih pengelolaan denda yang akan dilihat pada ViewDashboard sebagai objek *boundary*. Lalu pesan diteruskan kepada kelas C_Denda sebagai objek *control*. Kemudian objek *control* berinteraksi dengan objek model yaitu M_Denda untuk mengambil data denda pada basis data. Setelah itu akan ditampilkan data denda setiap mahasiswa pada ViewListDenda sebagai objek *boundary*.

Dalam mengelola data denda, petugas ruang baca juga dapat melihat detail denda setiap mahasiswa. Pertukaran pesan dimulai ketika petugas ruang baca memilih detail pada salah satu data mahasiswa yang akan dilihat pada ViewListDenda sebagai objek *boundary*. Lalu pesan diteruskan kepada kelas C_Denda sebagai objek *control*. Kemudian objek *control* berinteraksi dengan objek model yaitu M_Denda untuk mengambil detail data denda setiap mahasiswa berdasarkan nim pada basis data. Setelah itu akan ditampilkan data detail denda setiap mahasiswa pada ViewDetailDenda sebagai objek *boundary*.





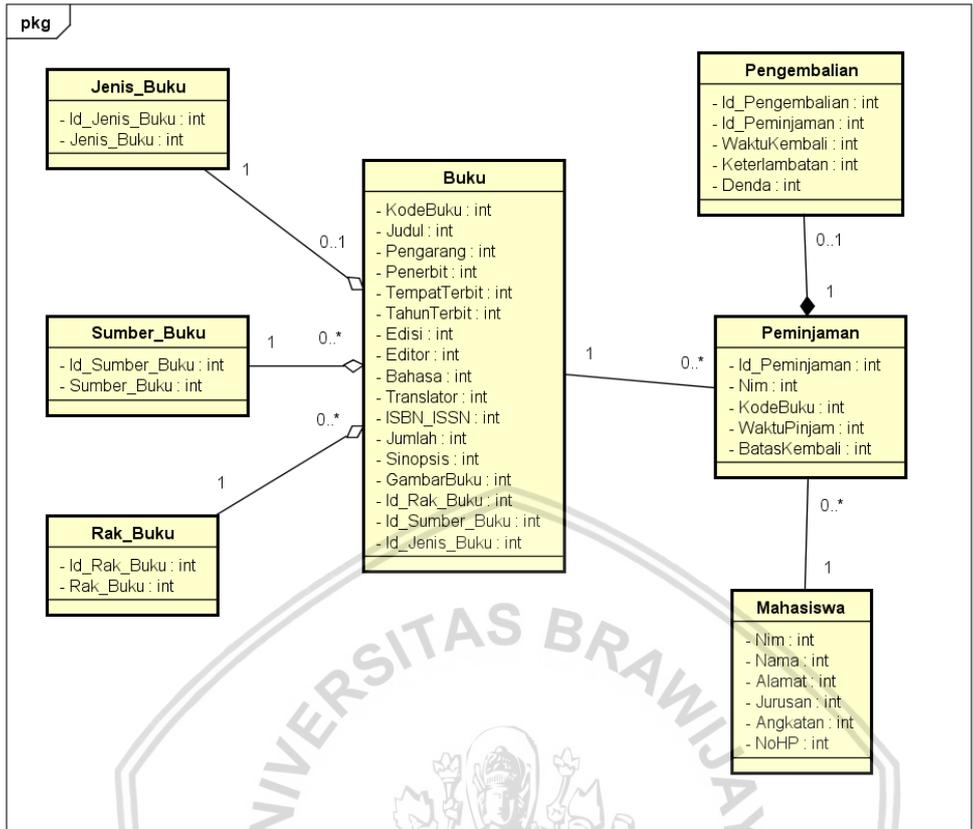
Gambar 5.7 Sequence Diagram Mengelola Data Denda

5.3 Perancangan Kelas Diagram

Pada perancangan kelas diagram sistem informasi manajemen perpustakaan, akan dijelaskan diagram kelas analisis dan diagram kelas perancangan. Diagram kelas analisis akan menjadi panduan untuk melakukan pembuatan diagram kelas objek dan perancangan basis data. Diagram kelas perancangan merupakan visualisasi kelas dan hubungan di antara setiap kelas sebagai panduan untuk melakukan pengembangan sistem

5.3.1 Diagram Kelas Analisis

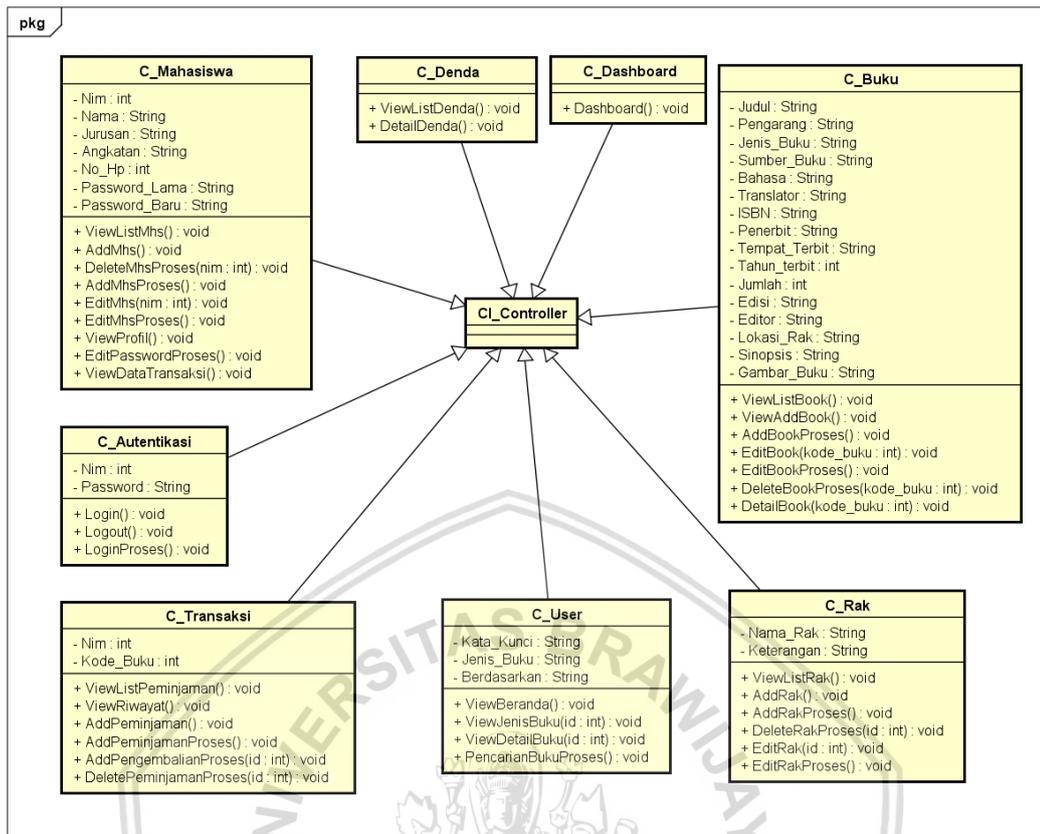
Diagram kelas analisis berisi visualisasi hubungan antar kelas sebagai representasi objek yang menjadi komponen penyusun sistem yang akan dikembangkan. Pada gambar 5.8 merupakan diagram kelas analisis yang menunjukkan bahwa terdapat 7 kelas yaitu Jenis_Buku, Sumber_Buku, Rak_Buku, Buku, Mahasiswa, Peminjaman, dan Pengembalian, dimana masing-masing kelas tersebut memiliki atribut-atribut tertentu. Masing-masing kelas memiliki hubungan dengan kelas yang lain. Diagram kelas analisis akan digunakan sebagai panduan pemodelan kelas perancangan dan perancangan basis data.



Gambar 5.8 Diagram Kelas Analisis

5.3.2 Diagram Kelas Perancangan

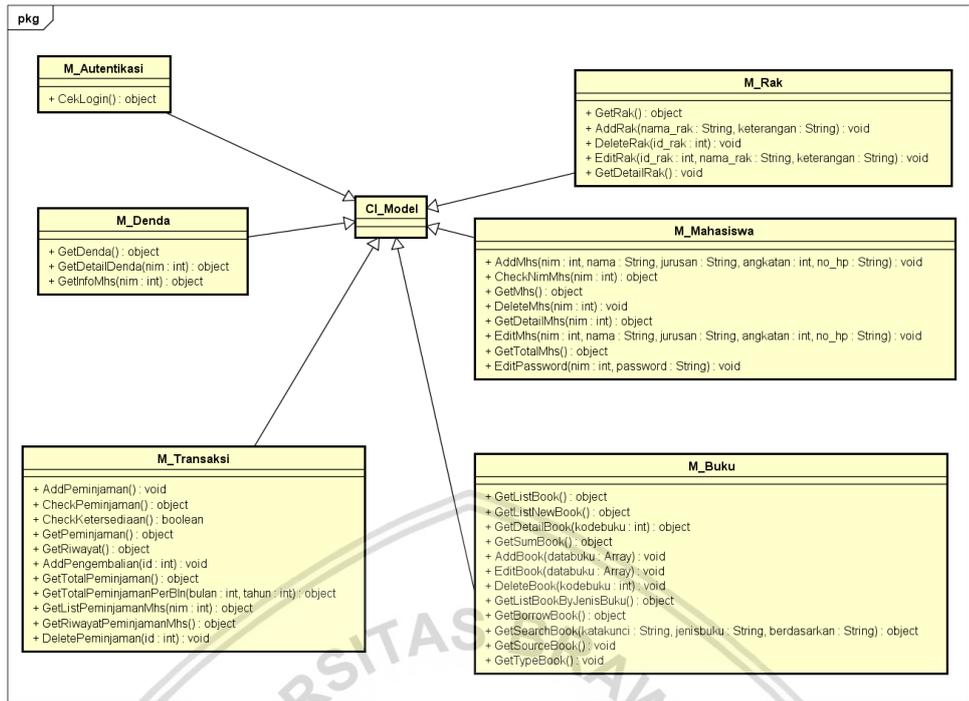
Diagram kelas perancangan merupakan hasil dari pengembangan diagram kelas analisis yang telah dibuat sebelumnya. Diagram kelas perancangan dibuat dengan menyesuaikan diagram kelas analisis, objek-objek yang teridentifikasi pada *sequence diagram* dengan pola MVC (Model-View-Controller) pada kerangka kerja pemrograman PHP Codeigniter. Pada diagram kelas perancangan menggambarkan 8 subkelas *controller* dan 6 subkelas *model*. Hasil pemodelan diagram kelas akan menjadi panduan untuk membangun perangkat lunak



Gambar 5.9 Diagram Kelas Perancangan *Controller*

Subkelas controller menyediakan fungsi logika yang merepresentasikan kemampuan sistem untuk memenuhi kebutuhan pengguna. Pada gambar 5.9 menjelaskan bahwa diagram kelas perancangan *controller* untuk implementasi sistem informasi manajemen perpustakaan memiliki 8 *subclass controller* yang terdiri dari C_Mahasiswa digunakan untuk mengelola mahasiswa, C_Denda digunakan untuk mengelola denda, C_Dashboard digunakan untuk memberikan informasi secara umum terkait total koleksi buku, total buku terpinjam dan total mahasiswa yang terdaftar dalam sistem, C_Buku digunakan untuk mengelola buku, C_Autentikasi digunakan untuk penentuan hak akses yang dapatkan oleh pengguna sistem, C_Transaksi digunakan untuk mengelola peminjaman dan pengembalian, C_User digunakan untuk melakukan pencarian buku, dan informasi detail buku, dan C_Rak digunakan untuk mengelola rak buku.

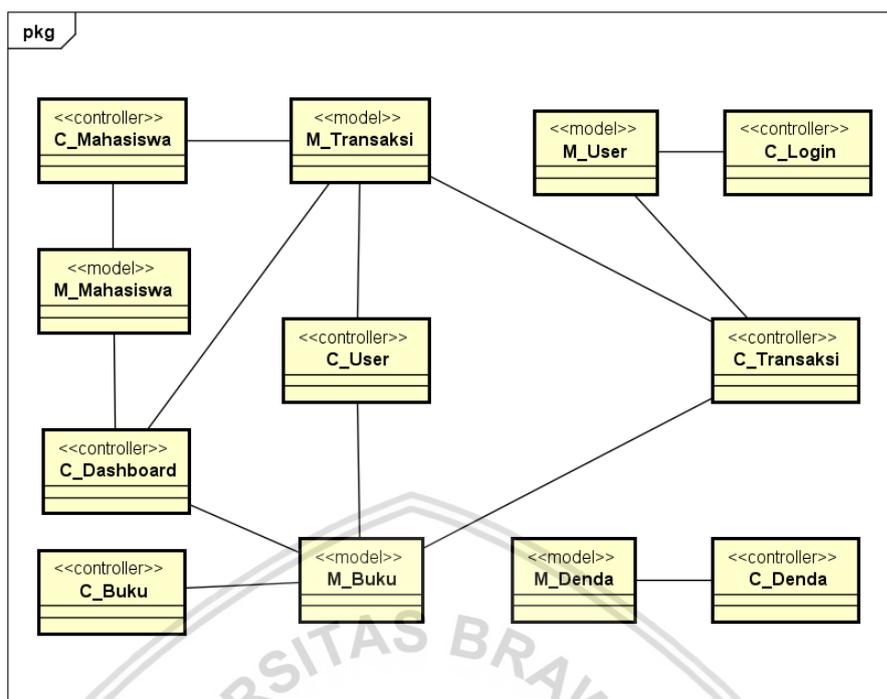




Gambar 5.10 Diagram Kelas Perancangan Model

Subkelas *model* menyediakan fungsi untuk melakukan akses data yang tersimpan pada basis data. Pada gambar 5.10 menjelaskan bahwa diagram kelas perancangan *model* untuk implementasi sistem informasi manajemen perpustakaan memiliki 6 *subclass model* yang terdiri dari M_Denda, M_Transaksi, M_Buku, M_Mahasiswa, M_Rak, dan M_Autentikasi.



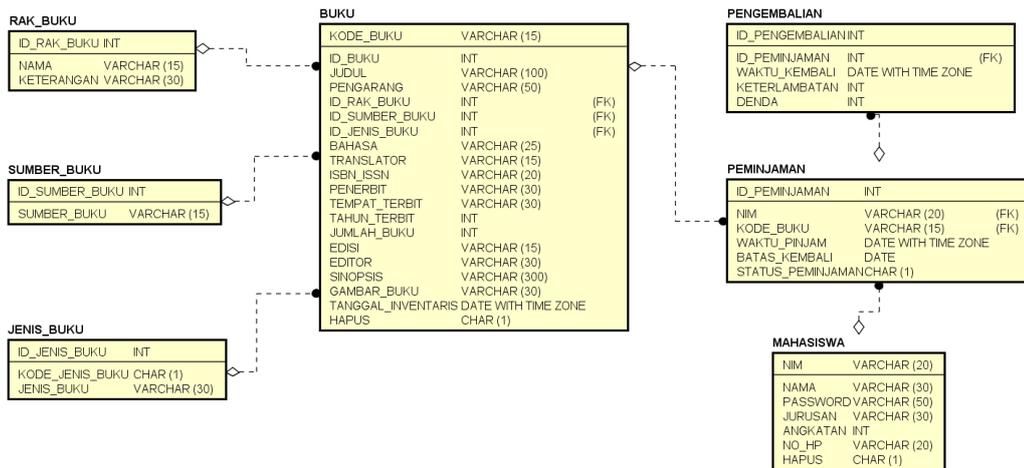


Gambar 5.11 Diagram Kelas Perancangan Relasi *Controller* dan *Model*

Pada gambar 5.11 merupakan diagram kelas perancangan relasi *controller* dan *model* pada sistem informasi manajemen perpustakaan. Sistem dibangun dengan menggunakan *framework Codeigniter*. Framework ini didasarkan pada pola perancangan *Model-View-Controller* sehingga dalam penerapannya kelas-kelas akan dibagi menjadi kelas *controller* dan kelas *model*.

5.4 Perancangan Basis Data

Bagian ini merupakan penggambaran skema basis data yang digunakan sebagai acuan untuk membangun basis data dari Sistem Informasi Manajemen Perpustakaan. Rancangan basis data dengan menggunakan *physical data model* (PDM) dibuat berdasarkan pemodelan kelas diagram analisis yang telah dibuat sebelumnya. Masing-masing kelas pada diagram kelas analisis direpresentasikan oleh tabel yang terdapat pada PDM, sedangkan atribut yang terdapat di dalam masing-masing 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. Hasil dari pemetaan diagram kelas analisis ke PDM didapatkan 7 tabel yang terdiri dari tabel RAK_BUKU, SUMBER_BUKU, JENIS_BUKU, BUKU, PEMINJAMAN, PENGEMBALIAN, dan MAHASISWA akan ditunjukkan pada gambar 5.12.



Gambar 5.12 PDM Sistem Informasi Manajemen Perpustakaan

5.5 Perancangan Antarmuka Pengguna

Pada bagian ini akan ditunjukkan perancangan antarmuka pengguna yang dilakukan untuk menggambarkan tampilan dari sistem yang akan dikembangkan dalam penelitian ini. Beberapa sketsa antarmuka pengguna sistem di antaranya adalah antarmuka login, antarmuka melihat detail buku, antarmuka melihat peminjaman buku, antarmuka melihat informasi buku yang dipinjam, dan antarmuka melihat riwayat peminjaman dan denda.

5.5.1 Antarmuka Login

Antarmuka login merupakan tampilan yang akan dilihat pengguna untuk melakukan otentikasi untuk melakukan aktifitas pada sistem sesuai dengan berdasarkan hak aksesnya. Komponen antarmuka login di antaranya adalah logo, kolom untuk memasukkan nim, kolom untuk memasukkan *password*, dan tombol *login* untuk menjalankan fungsi otentikasi pengguna. Antarmuka login akan ditunjukkan dalam gambar 5.13.

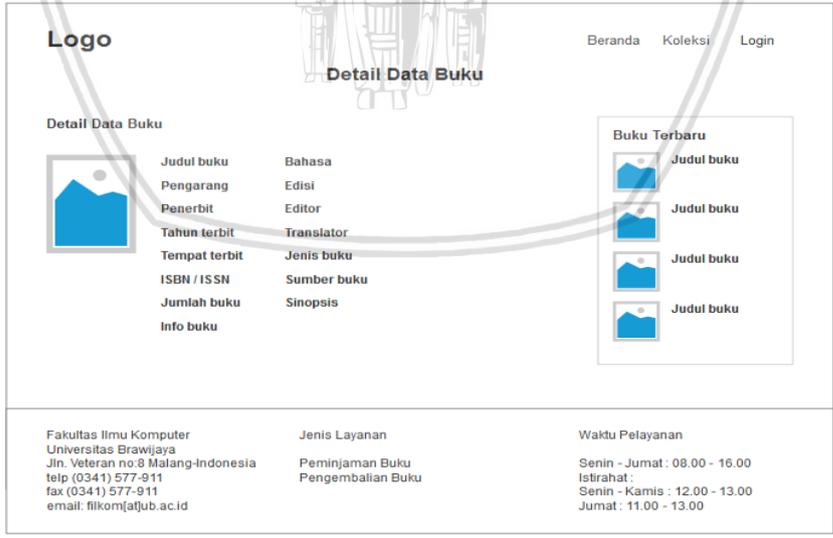




Gambar 5.13 Antarmuka Login

5.5.2 Antarmuka Melihat Detail Buku

Antarmuka melihat detail buku merupakan tampilan yang akan dilihat pengguna, pada saat pengguna menggunakan sistem untuk melihat informasi detail data buku. Komponen antarmuka melihat detail buku di antaranya adalah *header* yang terdiri dari logo dan menu bar yaitu beranda, koleksi dan login. Komponen selanjutnya berisi gambar buku dan data informasi detail buku. Selain itu terdapat komponen *side bar* yang berisi informasi terkait data buku terbaru dan *footer* yang berisi informasi terkait kontak dari FILKOM (Fakultas Ilmu Komputer), jenis pelayanan, dan waktu pelayanan ruang baca. Antarmuka melihat detail buku akan ditunjukkan dalam gambar 5.14.

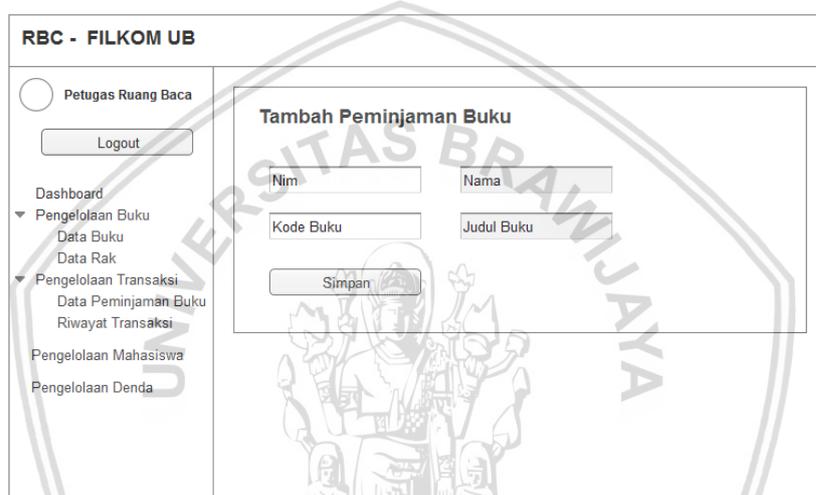


Gambar 5.14 Antarmuka Melihat Detail Buku



5.5.3 Antarmuka Menambahkan Peminjaman Buku

Antarmuka menambahkan peminjaman buku merupakan tampilan yang akan dilihat pengguna, pada saat pengguna menggunakan sistem untuk menambahkan peminjaman buku. Pada antarmuka melihat peminjaman buku terdapat dua kelompok komponen yang berada di dalam persegi di sisi kiri dan kanan. Pada sisi kiri merupakan *sidebar* yang berfungsi untuk menunjukkan informasi mengenai pengguna dan untuk menampilkan beberapa menu sistem. Pada sisi kanan terdapat *form* tambah peminjaman buku yang terdapat kolom untuk memasukan nim dan kolom untuk memasukan kode buku, sedangkan kolom nama dan judul hanya menampilkan informasi berdasarkan inputkan pada kolom nim dan kode buku. Antarmuka menambahkan peminjaman buku akan ditunjukan dalam gambar 5.15.

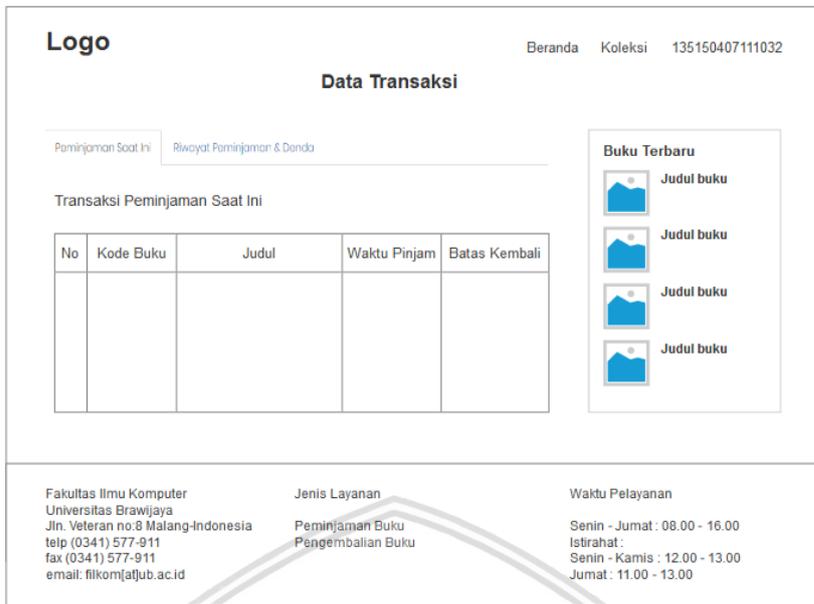


The screenshot shows a web interface for 'RBC - FILKOM UB'. On the left is a sidebar with a user profile 'Petugas Ruang Baca' and a 'Logout' button. Below that is a navigation menu with categories: 'Dashboard', 'Pengelolaan Buku' (sub-menu: 'Data Buku', 'Data Rak'), 'Pengelolaan Transaksi' (sub-menu: 'Data Peminjaman Buku', 'Riwayat Transaksi'), 'Pengelolaan Mahasiswa', and 'Pengelolaan Denda'. The main content area is titled 'Tambah Peminjaman Buku' and contains a form with four input fields: 'Nim', 'Nama', 'Kode Buku', and 'Judul Buku'. A 'Simpan' button is located below the 'Kode Buku' field. A large, faint watermark of the Universitas Brawijaya logo is visible in the background.

Gambar 5.15 Antarmuka Menambahkan Peminjaman Buku

5.5.4 Antarmuka Melihat Informasi Buku yang dipinjam

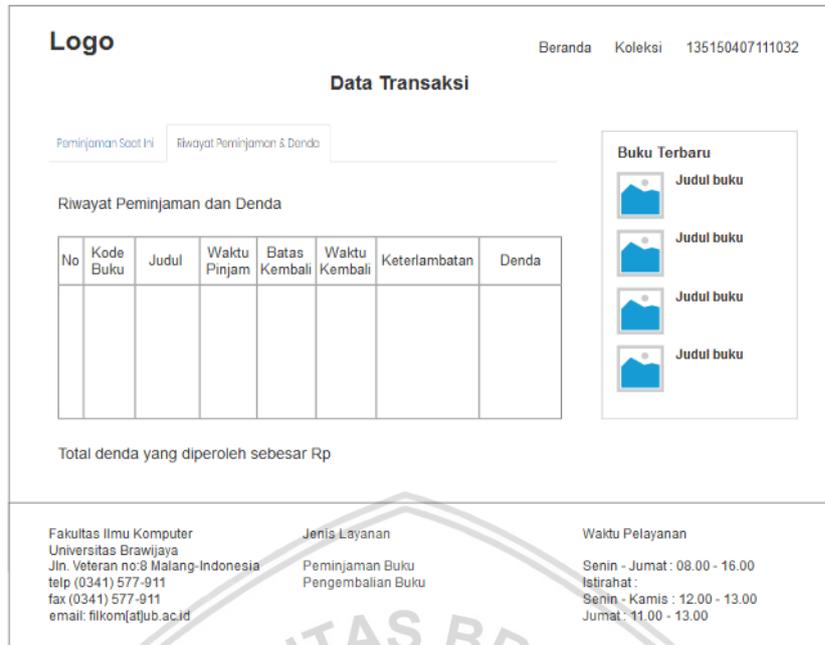
Antarmuka melihat informasi buku yang dipinjam merupakan tampilan yang akan dilihat pengguna, pada saat pengguna menggunakan sistem untuk melihat informasi buku yang dipinjam. Komponen antarmuka melihat informasi buku yang dipinjam di antaranya adalah *header* yang terdiri dari logo dan menu bar yaitu beranda, koleksi dan login. Komponen selanjutnya berisi *tab menu* yaitu peminjaman saat ini dan riwayat peminjaman dan denda. Pada antarmuka melihat informasi buku ditampilkan *tab menu* peminjaman saat ini yang berisi informasi transaksi peminjaman saat ini yang ditampilkan dalam bentuk tabel. Selain itu terdapat komponen *side bar* yang berisi informasi terkait data buku terbaru dan *footer* yang berisi informasi terkait kontak dari FILKOM (Fakultas Ilmu Komputer), jenis pelayanan, dan waktu pelayanan ruang baca. Antarmuka melihat informasi buku yang dipinjam akan ditunjukan dalam gambar 5.16.



Gambar 5.16 Antarmuka Melihat Informasi Buku yang dipinjam

5.5.5 Antarmuka Melihat Riwayat Peminjaman dan Denda

Antarmuka melihat riwayat peminjaman dan denda merupakan tampilan yang akan dilihat pengguna, pada saat pengguna menggunakan sistem untuk melihat riwayat peminjaman dan denda. Komponen antarmuka melihat informasi buku yang dipinjam di antaranya adalah *header* yang terdiri dari logo dan menu bar yaitu beranda, koleksi dan login. Komponen selanjutnya berisi *tab menu* yaitu peminjaman saat ini dan riwayat peminjaman dan denda. Pada antarmuka ini ditampilkan *tab menu* riwayat peminjaman dan denda yang berisi informasi riwayat transaksi peminjaman dan denda yang diperoleh ditampilkan dalam bentuk tabel. Selain itu terdapat komponen *side bar* yang berisi informasi terkait data buku terbaru dan *footer* yang berisi informasi terkait kontak dari FILKOM (Fakultas Ilmu Komputer), jenis pelayanan, dan waktu pelayanan ruang baca. Antarmuka melihat riwayat peminjaman dan denda akan ditunjukkan dalam gambar 5.17.



Gambar 5.17 Antarmuka Melihat Riwayat Peminjaman dan Denda

5.6 Perancangan Algoritme

Pada perancangan algoritme dijelaskan mengenai algoritme yang akan diterapkan untuk implementasi Sistem Informasi Manajemen Perpustakaan. Hasil perancangan algoritme dalam penelitian ini di dokumentasikan ke dalam beberapa *pseudocode*. Algoritme yang telah didokumentasikan akan menjadi panduan untuk pembangunan sistem informasi dalam tahap implementasi. Terdapat beberapa algoritma yang akan dijelaskan dalam sub bab ini yaitu algoritme mengelola peminjaman buku, mengelola pengembalian buku, dan melakukan pencarian buku.

5.6.1 Mengelola Peminjaman Buku

Subbab ini berisi rancangan algoritme dari sebuah fungsi yang disediakan oleh sistem kepada petugas ruang baca saat menggunakan sistem untuk mengelola peminjaman buku. *Pseudocode* yang terdapat pada tabel 5.1 merupakan rancangan algoritme fungsi untuk menambahkan peminjaman buku.

Tabel 5.1 *Pseudocode* Menambahkan Peminjaman Buku

No	<i>Pseudocode</i>
1	Mulai
2	Mengambil data nim pada form peminjaman buku
3	Mengambil data kode buku pada form peminjaman buku
4	Memanggil model M_Transaksi
5	Menjalankan fungsi CheckPeminjaman dengan parameter nim untuk mengecek peminjaman mahasiswa
6	Menjalankan fungsi Check Ketersediaan dengan parameter kode buku untuk mengecek ketersediaan buku
8	
9	Jika peminjaman kurang dari 3



Tabel 5.1 Psdeudocode Menambahkan Peminjaman Buku

No	Pseudocode
10	Jika buku yang dipinjam tersedia
11	Jika buku yang dipinjam tidak sedang dipinjam
12	Jalankan fungsi AddPeminjaman dengan parameter nim dan kode buku
14	Tampil halaman peminjaman buku dan pesan peminjaman buku berhasil
15	Lainnya
16	Tampil halaman tambah peminjaman dan pesan mahasiswa tidak diperbolehkan meminjam lebih dari 1 buku yang sama
17	Lainnya
18	Tampil halaman tambah peminjaman dan pesan maaf buku tidak tersedia, karena telah terpinjam semua
	Lainnya
19	Tampil halaman tambah peminjaman dan pesan mahasiswa dengan nim
20	tersebut telah meminjam 3 buku
21	Selesai

5.6.2 Mengelola Pengembalian Buku

Subbab ini berisi rancangan algoritme dari sebuah fungsi yang disediakan oleh sistem kepada petugas ruang baca saat menggunakan sistem untuk mengelola pengembalian buku. *Pseudocode* yang terdapat pada tabel 5.1 merupakan rancangan algoritma fungsi untuk mengelola pengembalian buku.

Tabel 5.2 Psdeudocode Mengelola Pengembalian Buku

No	Pseudocode
1	Mulai
2	Mengambil id_peminjaman pada data peminjaman buku
3	Memanggil model M_Transaksi
4	Menjalakan fungsi AddPengembalian dengan parameter id_peminjaman pada model M_Transaksi
5	Tampil halaman data peminjaman buku dan menampilkan pesan buku berhasil dikembalikan
6	Selesai

5.6.3 Melakukan Pencarian Buku

Subbab ini berisi rancangan algoritme dari sebuah fungsi yang disediakan oleh sistem kepada pengguna saat menggunakan sistem untuk melakukan pencarian buku. *Pseudocode* yang terdapat pada tabel 5.3 merupakan rancangan algoritma fungsi untuk melakukan pencarian buku.

Tabel 5.3 Psdeudocode Melakukan Pencarian Buku

No	Pseudocode
1	Mulai
2	Mengambil data kata kunci pada form pencarian
3	Mengambil data jenis buku pada form pencarian
4	Mengambil data berdasarkan pada form pencarian
5	Memanggil model M_Buku
6	
7	

Tabel 5.3 *Pseudocode* Melakukan Pencarian Buku

No	<i>Pseudocode</i>
8	Memuat data hasil pencarian dengan menjalankan fungsi
9	GetSearchBook dengan parameter kata kunci, jenis buku dan
10	berdasarkan pada model M_Buku
	Tampil halaman hasil pencarian buku
	Selesai

5.6.4 Mengelola Data Denda

Subbab ini berisi rancangan algoritme dari sebuah fungsi yang disediakan oleh sistem kepada petugas ruang baca saat menggunakan sistem untuk mengelola data denda. *Pseudocode* yang terdapat pada tabel 5.4 merupakan rancangan algoritma fungsi untuk melihat data denda mahasiswa.

Tabel 5.4 *Pseudocode* Melihat Data Denda

No	<i>Pseudocode</i>
1	Mulai
2	Set variabel array dengan kata 'denda'
3	Memanggil model M_Denda
4	Memuat data denda dengan menjalankan fungsi GetDenda pada model
5	M_Denda
6	Tampil halaman data denda mahasiswa
	Selesai

5.7 Perancangan *Web Service*

Pada sub bab ini menjelaskan mengenai perancangan *web service* yang disediakan oleh Sistem Informasi Manajemen Perpustakaan. Pengembangan *web service* dilakukan dengan menggunakan REST API yang disediakan oleh CodeIgniter. Peran dari *web service* disini sebagai perantara untuk melakukan komunikasi data. Data yang diberikan oleh *web service* dalam bentuk format JSON. *Web Service* yang ditujukan kepada Sistem Informasi Akademik Mahasiswa Universitas Brawijaya (SIAM UB).

Peran dari Sistem Informasi Manajemen Perpustakaan sebagai penyedia layanan sedangkan Sistem Informasi Akademik Mahasiswa Universitas Brawijaya (SIAM UB) berperan sebagai pemohon layanan. Daftar layanan yang sediakan yaitu data denda yang didapatkan oleh mahasiswa ketika telat dalam melakukan pengembalian buku. Data denda tersebut nantinya akan dimasukkan kedalam tagihan biaya uang kuliah tunggal (UKT) mahasiswa pada tiap semester.

5.7.1 *Service* Menampilkan Data Denda Mahasiswa

Tabel 5.5 *Service* Untuk Menampilkan Data Denda

Nama	Denda
Tujuan	Meminta <i>web service</i> untuk menampilkan data denda pengembalian buku mahasiswa.
<i>Input</i>	Nim mahasiswa

Tabel 5.5 *Service* Untuk Menampilkan Data Denda (Lanjutan)

<i>Method</i>	GET
<i>Output</i>	<pre>[{ "Nim": "135150407111032", "Nama ": "I Gde Yuda Pratama", "Jurusan": "Sistem Informasi", "Angkatan": "2013", "Total_denda": "500000" }]</pre>

Pada tabel 5.5 menjelaskan mengenai *service* `getDataDenda` menyediakan layanan yang bertujuan untuk mendapatkan total denda yang diperoleh oleh mahasiswa. *Service* tersebut bisa diakses dengan memenuhi `key_API`. JSON yang diterima berupa JSON Array. JSON Array tersebut memiliki beberapa JSON Object yang berisi:

1. Nim : berisi informasi nim setiap mahasiswa.
2. Nama : berisi nama mahasiswa.
3. Jurusan : berisi informasi jurusan asal mahasiswa.
4. Angkatan : berisi informasi tahun masuk kuliah mahasiswa.
5. Total_denda mahasiswa : berisi informasi biaya total denda yang diperoleh mahasiswa.

BAB 6 IMPLEMENTASI

Bab implementasi berisi penjelasan mengenai kebutuhan implementasi dan hasil implementasi sistem informasi yang dikembangkan dalam penelitian ini. Beberapa hal yang dijelaskan di dalam bagian ini di antaranya adalah spesifikasi lingkungan implementasi, beberapa kode program, dan beberapa gambar antarmuka pengguna sistem informasi yang dikembangkan. Implementasi dilakukan berdasarkan hasil perancangan yang telah dijelaskan pada bab sebelumnya.

6.1 Spesifikasi Lingkungan Implementasi

Subbab ini menjelaskan spesifikasi perangkat keras dan perangkat lunak yang digunakan selama proses pembangunan sistem informasi. Perangkat keras yang digunakan pada proses pembangunan sistem informasi adalah sebuah unit laptop dengan spesifikasi yang dijelaskan pada Tabel 6.1. Selain itu terdapat informasi spesifikasi perangkat lunak yang digunakan untuk pengembangan ditunjukkan pada Tabel 6.2.

Tabel 6.1 Spesifikasi Perangkat Keras

Unit Komputasi	Laptop ASUS K46CB
CPU	Intel(R) Core™ i3 CPU @ 1.80 GHz
Kapasitas RAM	8 GB
Kapasitas Penyimpanan	500 GB
Kartu Grafis	NVidia GeForce GT 740 M
Resolusi Layar	1366 x 768 pixels

Tabel 6.2 Spesifikasi Perangkat Lunak

Sistem Operasi	Windows 10 Pro 64-bit
Web Server	Apache 2.4.7
DBMS	MySQL
Bahasa Pemrograman	PHP
Editor Kode Program	Sublime Text 2
Peramban	Google Chrome Version 61.0.3163.100 (Official Build) (64-bit)
Perangkat Lunak atau Bahasa Pemrograman Pendukung	Javascript
	HTML



6.2 Implementasi Algoritme

Subbab implementasi algoritme berisi beberapa kode program yang dibuat berdasarkan algoritme dalam *pseudocode* yang telah dibuat pada tahap perancangan.

6.2.1 Mengelola Peminjaman Buku

Kode program yang terdapat pada tabel 6.3 merupakan implementasi algoritme fungsi untuk menambahkan peminjaman buku yang merupakan bagian dari layanan pengelolaan peminjaman buku yang disediakan oleh sistem. implementasi algoritme menambahkan peminjaman buku dibuat berdasarkan hasil rancangan algoritme menambahkan peminjaman buku.

Tabel 6.3 Implementasi Algoritme Menambahkan Peminjaman Buku

No	Kode Program
1	public function AddPeminjamanProses()
2	{
3	\$nim=\$this->input->post('nimm');
4	\$kode_buku=\$this->input->post('kode_buku');
5	\$this->load->model('M_Transaksi');
6	\$cekPeminjaman=\$this->M_Transaksi->CheckPeminjaman(\$nim);
7	\$cekKetersediaan=\$this->M_Transaksi->CheckKetersediaan(\$kode_buku);
8	
9	if (count(\$cekPeminjaman) < 3) {
10	if (\$cekKetersediaan == true) {
11	//cek apakah mhs telah meminjam buku sebelumnya
12	\$cekBukuTerpijam= false;
13	foreach (\$cekPeminjaman as \$key) {
14	if (\$key->KODE_BUKU == \$kode_buku) {
15	\$cekBukuTerpijam= true;
16	}
17	} //end
18	
19	if (\$cekBukuTerpijam == false) {
20	\$this->M_Transaksi->AddPeminjaman(\$nim,\$kode_buku);
21	\$this->session->set_flashdata('info', 'Data peminjaman buku berhasil ditambahkan');
22	
23	redirect('C_Transaksi/ViewListPeminjaman','refresh');
24	}else {
25	\$this->session->set_flashdata('info', 'Maaf, mahasiswa tidak diperbolehkan meminjam lebih dari 1 buku yang sama');
26	
27	redirect('C_Transaksi/AddPeminjaman');
28	}
29	}else {
30	\$this->session->set_flashdata('info', 'Maaf buku tidak tersedia, karena telah terpinjam semua');
31	redirect('C_Transaksi/AddPeminjaman');
32	}
33	}else {
34	



Tabel 6.3 Implementasi Algoritme Menambahkan Peminjaman Buku

No	Kode Program
35	\$pesan='Mahasiswa dengan nim '.\$nim.' telah
36	meminjam 3 buku';
37	\$this->session->set_flashdata('info', \$pesan);
38	redirect('C_Transaksi/AddPeminjaman');
39	}

6.2.2 Mengelola Pengembalian Buku

Implementasi algoritme dalam bentuk kode program yang terdapat pada tabel 6.4 merupakan implementasi fungsi mengelola pengembalian buku yang disediakan oleh sistem. Implementasi algoritme dilakukan berdasarkan hasil rancangan algoritme mengelola pengembalian buku.

Tabel 6.4 Implementasi Algoritme Mengelola Pengembalian Buku

No	Kode Program
1	public function AddPengembalianProses(\$id)
2	{
3	\$id_peminjaman=\$id;
4	\$this->load->model('M_Transaksi');
5	\$this->M_Transaksi->AddPengembalian(\$id_peminjaman);
6	\$this->session->set_flashdata('info', 'Buku berhasil
7	dikembalikan');
8	redirect('C_Transaksi/ViewListPeminjaman','refresh');
	}

6.2.3 Melakukan Pencarian Buku

Kode program pada Tabel 6.5 merupakan hasil implementasi algoritme dari fungsi melakukan pencarian buku yang disediakan oleh sistem. Implementasi algoritme melakukan pencarian buku dilakukan berdasarkan hasil rancangan algoritme melakukan pencarian buku.

Tabel 6.5 Implementasi Algoritme Melihat Data Denda

No	Kode Program
1	public function ViewListDenda()
2	{
3	\$data['page']='Denda';
4	\$this->load->model('M_Denda');
5	\$data['denda']=\$this->M_Denda->GetDenda();
6	\$this->load->view('admin/ViewListDenda',\$data);
7	}

6.2.4 Mengelola Data Denda

Kode program yang terdapat pada tabel 6.6 merupakan implementasi algoritme fungsi untuk melihat data denda yang merupakan bagian dari layanan pengelolaan data denda yang disediakan oleh sistem. implementasi algoritme melihat data denda dibuat berdasarkan hasil rancangan algoritme melihat data denda.



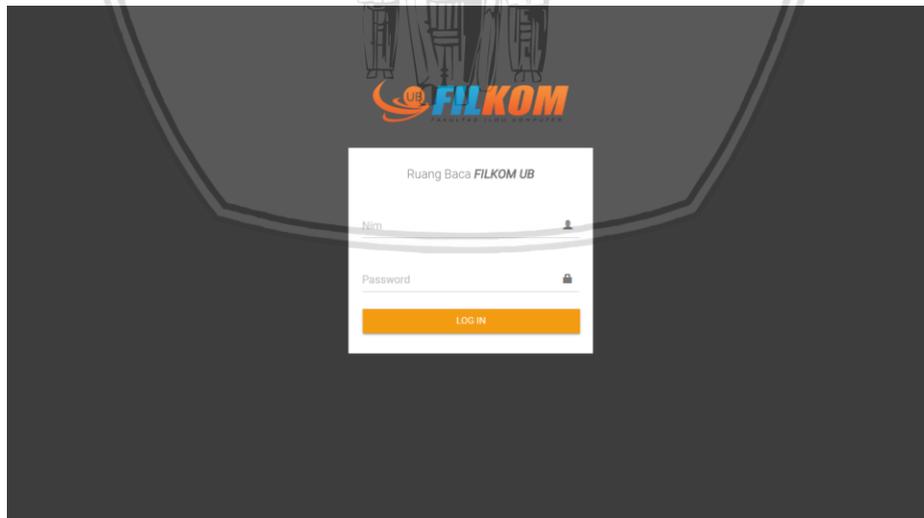
Tabel 6.6 Implementasi Algoritme Melakukan Pencarian Buku

No	Kode Program
1	public function PencarianBukuProses()
2	{
3	\$kata_kunci=\$this->input->post('kata_kunci');
4	\$jenis_buku=\$this->input->post('jenis_buku');
5	\$berdasarkan=\$this->input->post('berdasarkan');
6	
7	\$this->load->model('M_Buku');
8	\$data['kata_kunci']=\$kata_kunci;
9	\$data['jenis_buku']=\$this->M_Buku->GetJenisBuku();
10	\$data['buku_terbaru']=\$this->M_Buku->GetListNewBook();
11	\$data['pencarian']=\$this->M_Buku-
12	>GetSearchBook(\$kata_kunci,\$jenis_buku,\$berdasarkan);
13	\$this->load->view('front/HasilPencarian', \$data);
	}

6.3 Implementasi Antarmuka Pengguna

Subbab implementasi antarmuka pengguna menjelaskan beberapa hasil implementasi yang telah dirancang dalam subbab 5.5. Subbab ini hanya menyertakan beberapa hasil implementasi antarmuka pengguna di antaranya adalah antarmuka *login*, antarmuka melihat detail buku, antarmuka menambahkan peminjaman buku, antarmuka melihat informasi buku yang dipinjam, dan antarmuka melihat riwayat peminjaman dan denda. Implementasi antarmuka pengguna dilakukan berdasarkan hasil perancangan antarmuka pengguna yang telah dilakukan dalam penelitian ini.

6.3.1 Antarmuka Login



Gambar 6.1 Antarmuka Login

Antarmuka *login* ditampilkan kepada pengguna pada laptop atau komputer pada saat pengguna sistem menggunakan sistem dan harus melakukan otentikasi identitas pengguna. Hasil implementasi antarmuka *login* yang terdapat dalam gambar 6.1 bersadarkan hasil perancangan antarmuka pada gambar 5.11.

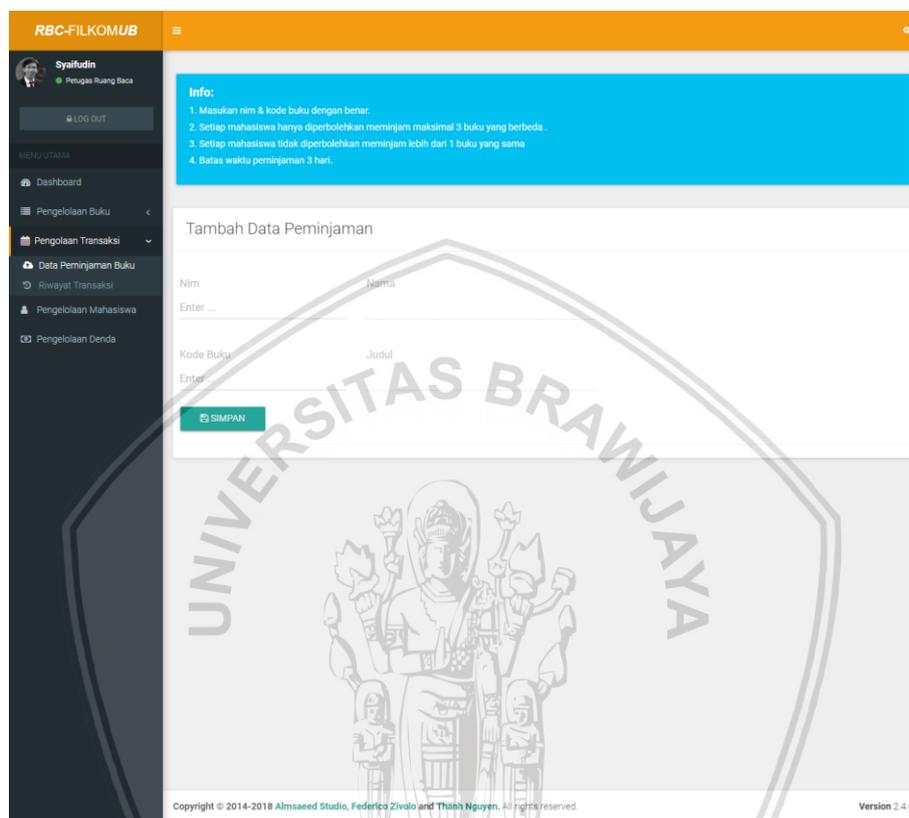
Terdapat beberapa komponen antarmuka di antaranya judul atau logo sistem, kolom untuk memasukan nim, kolom untuk memasukkan *password*, dan tombol untuk menjalankan fungsi otentikasi identitas pengguna sistem.

6.3.2 Antarmuka Melihat Detail Buku

Antarmuka melihat detail buku merupakan tampilan yang dilihat oleh pengguna sistem pada saat menggunakan sistem untuk melihat detail buku. Hasil implementasi antarmuka melihat detail buku terdapat dalam gambar 6.2. Pada halaman ini ditampilkan menu navigasi untuk mengakses fitur-fitur lain dari sistem informasi manajemen perpustakaan dan informasi terkait detail data buku. Terdapat juga komponen *sidebar* yang menampilkan informasi koleksi buku terbaru.



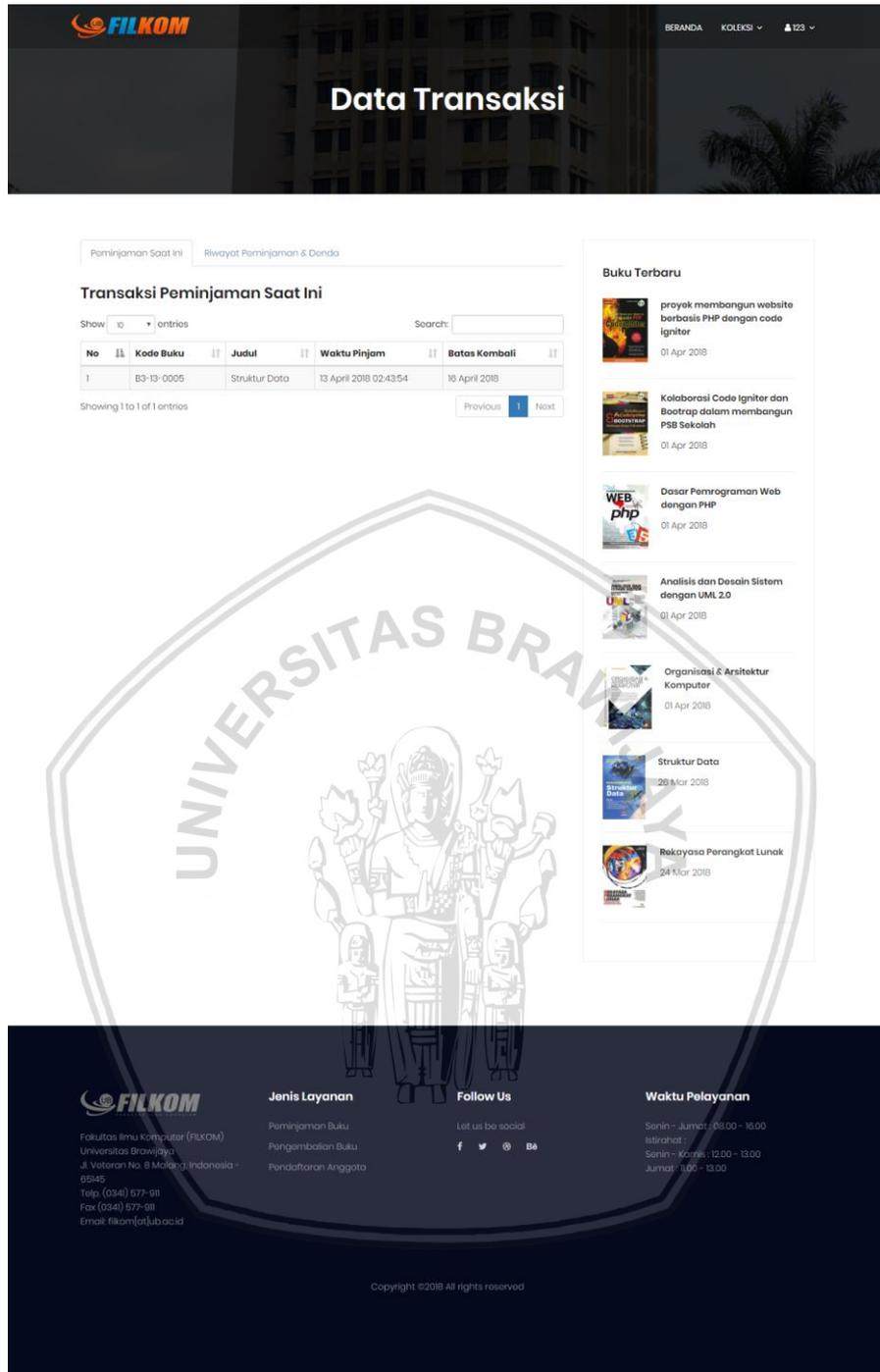
lain dari sistem. Pada bagian utama halaman terdapat formulir tambah data peminjaman yang berisi kolom untuk memasukan nim, kode buku dan tombol simpan untuk menjalankan fungsi menambahkan peminjaman buku. Sedangkan kolom nama dan judul buku akan terisi secara otomatis untuk mengkonfirmasi data peminjaman buku.



Gambar 6.3 Antarmuka Menambahkan Peminjaman Buku

6.3.4 Antarmuka Melihat Informasi Buku yang dipinjam

Antarmuka melihat informasi buku yang dipinjam merupakan tampilan yang dilihat oleh mahasiswa pada saat menggunakan sistem untuk melihat informasi buku yang sedang dipinjam. Hasil implementasi antarmuka melihat informasi buku yang dipinjam terdapat dalam gambar 6.4.



Gambar 6.4 Antarmuka Melihat Informasi Buku yang dipinjam

Pada halaman ini ditampilkan menu navigasi untuk mengakses fitur-fitur lain dari sistem informasi manajemen perpustakaan. Pada bagian utama halaman terdapat tab menu yaitu peminjaman saat ini, dan riwayat peminjaman dan denda yang diperoleh mahasiswa. Pada antarmuka ini yang ditampilkan yaitu tab menu peminjaman saat ini, yang berisi informasi terkait buku yang sedang dipinjem oleh mahasiswa. Terdapat juga komponen *sidebar* yang menampilkan informasi koleksi buku terbaru.



6.3.5 Antarmuka Melihat Riwayat Peminjaman dan Denda

Antarmuka melihat riwayat peminjaman dan denda merupakan tampilan yang dilihat oleh mahasiswa pada saat menggunakan sistem untuk melihat informasi riwayat peminjaman dan denda yang telah diperoleh. Hasil implementasi antarmuka melihat informasi buku yang dipinjam terdapat dalam gambar 6.5.



Gambar 6.5 Antarmuka Melihat Riwayat Peminjaman dan Denda



Pada halaman ini ditampilkan menu navigasi untuk mengakses fitur-fitur lain dari sistem informasi manajemen perpustakaan. Pada bagian utama halaman terdapat tab menu yaitu peminjaman saat ini, dan riwayat peminjaman dan denda yang diperoleh mahasiswa. Pada antarmuka ini yang ditampilkan yaitu tab menu riwayat peminjaman dan denda, yang berisi informasi terkait riwayat buku-buku yang telah dipinjam oleh mahasiswa beserta denda yang diperoleh. Terdapat juga komponen *sidebar* yang menampilkan informasi koleksi buku terbaru.

6.4 Implementasi Web Service

Pada sub bab ini menjelaskan mengenai implelementasi *web service* yang disediakan oleh Sistem Informasi Manajemen Perpustakaan. Implementasi *web service* yang akan dijelaskan pada bagian ini terdiri dari penjelasan mengenai implementasi fungsi web service untuk menampilkan data denda mahasiswa.

6.4.1 Implementasi Web Service Menampilkan Data Denda Mahasiswa

Bagian ini merupakan bagian implementasi dari *web service* untuk fungsi menampilkan data denda mahasiswa. Implementasi *web service* yang akan dijelaskan pada bagian ini diwakilkan dengan fungsi untuk melihat data denda yang merupakan bagian dari implementasi dari *use case* mengelola data denda. Implementasi kode program *web service* untuk fungsi melihat data denda mahasiswa dijelaskan pada Tabel 6.7 berikut.

Tabel 6.7 Implementasi Controller Denda

No	Kode Program
1	<?php
2	defined('BASEPATH') OR exit('No direct script access allowed');
3	
4	require APPPATH . '/libraries/REST_Controller.php';
5	class Denda extends REST_Controller {
6	
7	function __construct(\$config = 'rest') {
8	parent::__construct(\$config);
9	\$this->load->database();
10	}
11	
12	//Menampilkan data denda mahasiswa berdasarkan nim
13	function index_get() {
14	\$nim = \$this->get('NIM');
15	\$this->load->model('M_ServiceDenda');
16	\$kontak=\$this->M_ServiceDenda->GetDendaNim(\$nim);
17	\$this->response(\$kontak, 200);
18	}
19	}

Kode program pada Tabel 6.7 merupakan fungsi yang terdapat pada kelas *controller* Denda dan dapat diakses dengan mengirimkan parameter masukan melalui metode GET untuk mendapatkan data denda mahasiswa berdasarkan NIM. Kode program baris 13 menjelaskan fungsi untuk mendapatkan data denda mahasiswa melalui metode GET. Selanjutnya pada baris 14 memuat parameter masukan yang perlu dikirimkan dari pengguna *web service* untuk mendapatkan

data denda mahasiswa berdasarkan nim mahasiswa. Pada baris 15-16 terjadi pemanggilan model `M_ServiceDenda` dan memanggil fungsi `GetDendaNim` dengan parameter `nim` untuk mendapatkan data denda mahasiswa. Data denda mahasiswa yang diperoleh dalam format JSON. Implementasi fungsi dari kelas `model M_ServiceDenda` dijelaskan pada Tabel 6.8 berikut:

Tabel 6.8 Implementasi Model `M_ServiceDenda`

No	Kode Program
1	<?php
2	defined('BASEPATH') OR exit('No direct script access allowed');
3	
4	class M_ServiceDenda extends CI_Model {
5	
6	public function GetDendaNim(\$nim)
7	{
8	\$query="SELECT
9	U.NIM,U.NAMA,U.JURUSAN,U.ANGKATAN,COALESCE(SUM(PM.DENDA),0) AS
10	TOTAL_DENDA
11	FROM USER U
12	LEFT JOIN PEMINJAMAN P
13	ON U.NIM=P.NIM
14	LEFT JOIN PENGEMBALIAN PM
15	ON P.ID_PEMINJAMAN=PM.ID_PEMINJAMAN
16	WHERE U.STATUS IN(1,2) AND U.NIM='\$nim'
17	GROUP BY
18	U.NIM,U.NAMA,U.JURUSAN,U.ANGKATAN";
19	\$result=\$this->db->query(\$query);
20	return \$result->result();
21	}
22	}

Kode program pada Tabel 6.8 merupakan fungsi yang terdapat pada kelas `model M_ServiceDenda` untuk mendapatkan data denda mahasiswa berdasarkan `nim` dari basis data. Kode program pada baris 8 dan 20 merupakan `query` yang mengembalikan nilai berupa data denda mahasiswa berdasarkan `nim` mahasiswa.



Gambar 6.6 Hasil Implementasi Web Service Melihat Data Denda Mahasiswa

Pada Gambar 6.6 merupakan implementasi dari format data kembalian dalam JSON dari fungsi `web service` melihat data denda mahasiswa. Bagian paling atas dari Gambar 6.6 merupakan alamat URL dari `web service` yang bisa diakses oleh aplikasi untuk mengakses `web service` melihat data denda mahasiswa.

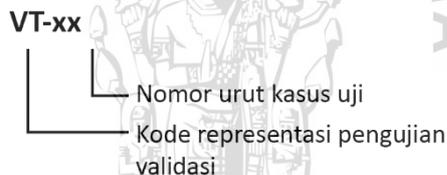
BAB 7 PENGUJIAN DAN ANALISIS PENGUJIAN

Bab ini membahas tahapan pengujian terhadap sistem yang telah dibangun pada penelitian ini. Pengujian sistem ini bertujuan untuk menguji bahwa setiap bagian dari sistem telah sesuai dengan rancangan dan persyaratan yang telah ditetapkan sebelumnya. Pembahasan pada bab pengujian dan analisis hasil ini diuraikan menjadi pengujian *black-box*, dan pengujian *user acceptance testing* (UAT).

7.1 Pengujian *Black-Box*

Pengujian *black-box* ini bertujuan untuk memastikan bahwa setiap persyaratan fungsional yang telah didefinisikan sebelumnya mampu dipenuhi oleh sistem yang dibangun. Setiap persyaratan fungsional yang telah didefinisikan pada bagian analisis persyaratan akan diuji pada pengujian ini. Pengujian pada bagian ini dilakukan dengan mendefinisikan kasus uji yang merujuk pada *use case scenario* yang telah didefinisikan pada tahap analisis kebutuhan. Kemudian setiap hasil dari pengujian yang dilakukan akan dibandingkan dengan hasil yang telah didefinisikan pada *use case scenario*.

7.1.1 Pengujian Validasi



Gambar 7.1 Kodifikasi Pengujian Validasi

Pengujian validasi merupakan bagian dari pengujian *black-box testing*. Pengujian validasi dilakukan berdasarkan berdasarkan skenario uji yang diperoleh dari beberapa alur *use case*. Pada Gambar 7.1 merupakan aturan penomoran yang diterapkan untuk pengujian validasi ini.

7.1.2 Pengujian Validasi Mengelola Peminjaman Buku

Bagian ini menjelaskan pengujian validasi pada beberapa fungsi yang dapat digunakan oleh petugas rung baca untuk mengelola peminjaman buku melalui sistem yang telah dikembangkan. Pada Tabel 7.1 menunjukkan beberapa skenario yang bersumber dari alur *use-case* Mengelola Peminjaman Buku yang akan digunakan untuk mengidentifikasi kasus uji.

Tabel 7.1 Matriks Skenario Pengujian Validasi Mengelola Peminjaman Buku

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic Flow</i>	-

Tabel 7.1 Matriks Skenario Pengujian Validasi Mengelola Peminjaman Buku

No. Skenario	Alur Awal	Alur Alternatif
Skenario 2	<i>Basic Flow</i>	Menambahkan Peminjaman Buku
Skenario 3	<i>Basic Flow</i>	Menambahkan Peminjaman Buku Alternatif (<i>form</i> yang diisikan tidak lengkap)

7.1.2.1 Pengujian Validasi Fungsi Menampilkan Peminjaman Buku

Pada Tabel 7.2 berisi informasi rencana pengujian dan kasus uji validasi dengan kode VT-01. Kasus uji VT-01 dilakukan untuk memastikan bahwa sistem dapat menampilkan data peminjaman buku dengan kode persyaratan SKPL-F-01. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.1. Pengujian validasi fungsi menampilkan peminjaman buku menunjukkan hasil valid.

Tabel 7.2 Rencana Pengujian dan Kasus Uji Validasi Fungsi Menampilkan Peminjaman Buku

Kode Pengujian	VT-01
Nomor Skenario	Skenario 1
Kode Persyaratan	SKPL-F-01
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menampilkan data peminjaman buku.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai petugas ruang baca. 2. Penguji memilih menu pengelolaan buku. 3. Penguji memilih submenu data peminjaman buku.
Hasil yang Diharapkan	Sistem menampilkan data peminjaman buku.
Hasil Pengujian	Sistem berhasil menampilkan data peminjaman buku.
Status Uji	Valid

7.1.2.2 Pengujian Validasi Fungsi Menambahkan Peminjaman Buku

Pada Tabel 7.23 berisi informasi rencana pengujian dan kasus uji validasi dengan kode VT-02. Kasus uji VT-02 dilakukan untuk memastikan bahwa sistem dapat menambahkan data peminjaman buku dengan kode persyaratan SKPL-F-02. Kasus uji diidentifikasi berdasarkan skenario 2 pada Tabel 7.1. Pengujian validasi fungsi menambahkan peminjaman buku menunjukkan hasil valid.



Tabel 7.3 Rencana Pengujian dan Kasus Uji Validasi Fungsi Menambahkan Peminjaman Buku

Kode Pengujian	VT-02
Nomor Skenario	Skenario 2
Kode Persyaratan	SKPL-F-02
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menambahkan data peminjaman buku.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai petugas ruang baca. 2. Penguji memilih menu tambah peminjaman. 3. Penguji mengisikan form tambah peminjaman. 4. Penguji memilih tombol simpan.
Hasil yang Diharapkan	Sistem dapat menyimpan data peminjaman buku ke <i>database</i> dan menampilkan notifikasi data peminjaman buku berhasil ditambahkan.
Hasil Pengujian	Sistem menyimpan data peminjaman buku ke <i>database</i> dan menampilkan notifikasi data peminjaman buku berhasil ditambahkan
Status Uji	Valid

7.1.2.3 Pengujian Validasi Fungsi Menambahkan Peminjaman Buku Alternatif (Form yang diisikan tidak lengkap)

Pada Tabel 7.24 berisi informasi rencana pengujian dan kasus uji validasi dengan kode VT-03. Kasus uji VT-03 dilakukan untuk memastikan bahwa sistem dapat melakukan alur alternatif dengan menunjukkan kemampuan sistem untuk menangani kegagalan menambahkan peminjaman buku jika form yang diisikan tidak lengkap. Kasus uji diidentifikasi berdasarkan skenario 3 pada Tabel 7.1. Pengujian validasi fungsi menambahkan peminjaman buku alternatif (*form* yang diisikan tidak lengkap) menunjukkan hasil valid.

Tabel 7.4 Rencana Pengujian dan Kasus Uji Validasi Fungsi Menambahkan Peminjaman Buku

Kode Pengujian	VT-03
Nomor Skenario	Skenario 3
Kode Persyaratan	SKPL-F-02

Tabel 7.4 Rencana Pengujian dan Kasus Uji Validasi Fungsi Menambahkan Peminjaman Buku (Lanjutan)

Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menampilkan notifikasi jika kolom pada <i>form</i> tambah peminjaman buku belum lengkap.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai petugas ruang baca. 2. Penguji memilih menu tambah peminjaman. 3. Penguji mengisikan form tambah peminjaman secara tidak lengkap. 4. Penguji memilih tombol simpan.
Hasil yang Diharapkan	Sistem dapat menampilkan notifikasi bahwa ada kolom pada <i>form</i> yang kosong harus diisi.
Hasil Pengujian	Sistem menampilkan notifikasi bahwa ada kolom pada <i>form</i> yang kosong harus diisi.
Status Uji	Valid

7.1.3 Pengujian *Validasi* Mengelola Pengembalian Buku

Pada Tabel 7.25 berisi informasi rencana pengujian dan kasus uji validasi dengan kode VT-04. Kasus uji VT-04 dilakukan untuk memastikan bahwa sistem dapat menyimpan data pengembalian buku dengan kode persyaratan SKPL-F-04. Pengujian validasi fungsi mengelola pengembalian buku menunjukkan hasil valid.

Tabel 7.5 Rencana Pengujian dan Kasus Uji Validasi Fungsi Mengelola Pengembalian Buku

Kode Pengujian	VT-04
Kode Persyaratan	SKPL-F-04
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menyimpan data pengembalian buku.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai petugas ruang baca. 2. Penguji memilih salah satu data peminjaman buku yang nantinya akan diproses untuk pengembalian buku. 3. Penguji memilih tombol pengembalian. 4. Pengujian mengkonfirmasi data pengembalian buku.
Hasil yang Diharapkan	Sistem dapat menyimpan data pengembalian buku ke <i>database</i> dan menampilkan notifikasi buku berhasil dikembalikan.

Tabel 7.5 Rencana Pengujian dan Kasus Uji Validasi Fungsi Mengelola Pengembalian Buku (Lanjutan)

Hasil Pengujian	Sistem menyimpan data pengembalian buku ke <i>database</i> dan menampilkan notifikasi buku berhasil dikembalikan.
Status Uji	Valid

7.1.4 Pengujian *Validasi* Mengelola Data Denda

Bagian ini menjelaskan pengujian *Validasi* pada beberapa fungsi yang dapat digunakan oleh petugas rung baca untuk mengelola data denda melalui sistem yang telah dikembangkan. Pada Tabel 7.16 menunjukkan beberapa skenario yang bersumber dari alur *use-case* Mengelola Data Denda yang akan digunakan untuk mengidentifikasi kasus uji.

Tabel 7.6 Matriks Skenario Pengujian *Validasi* Mengelola Data Denda

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic Flow</i>	-
Skenario 2	<i>Basic Flow</i>	Menampilkan detail denda

7.1.4.1 Pengujian *Validasi Fungsi Menampilkan Data Denda*

Pada Tabel 7.27 berisi informasi rencana pengujian dan kasus uji *validasi* dengan kode VT-05. Kasus uji VT-05 dilakukan untuk memastikan bahwa sistem dapat menampilkan data denda mahasiswa dengan kode persyaratan SKPL-F-17. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.16. Pengujian *validasi* fungsi menampilkan data denda menunjukkan hasil valid.

Tabel 7.7 Rencana Pengujian dan Kasus Uji *Validasi Fungsi Menampilkan Denda*

Kode Pengujian	VT-05
Nomor Skenario	Skenario 1
Kode Persyaratan	SKPL-F-17
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menampilkan data denda mahasiswa.
Prosedur Uji	1. Penguji teridentifikasi sebagai petugas ruang baca. 2. Penguji memilih menu pengelolaan denda.
Hasil yang Diharapkan	Sistem menampilkan data denda mahasiswa.

Tabel 7.7 Rencana Pengujian dan Kasus Uji Validasi Fungsi Menampilkan Denda(Lanjutan)

Hasil Pengujian	Sistem berhasil menampilkan data denda mahasiswa.
Status Uji	Valid

7.1.4.2 Pengujian Validasi Fungsi Menampilkan Detail Data Denda

Pada Tabel 7.28 berisi informasi rencana pengujian dan kasus uji validasi dengan kode VT-06. Kasus uji VT-06 dilakukan untuk memastikan bahwa sistem dapat menampilkan detail data denda mahasiswa dengan kode persyaratan SKPL-F-17. Kasus uji diidentifikasi berdasarkan skenario 2 pada Tabel 7.16. Pengujian validasi fungsi menampilkan detail data denda menunjukkan hasil valid.

Tabel 7.8 Rencana Pengujian dan Kasus Uji Validasi Fungsi Menampilkan Denda

Kode Pengujian	VT-06
Nomor Skenario	Skenario 2
Kode Persyaratan	SKPL-F-18
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menampilkan detail data denda mahasiswa.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai petugas ruang baca. 2. Penguji memilih menu pengelolaan denda. 3. Penguji memilih tombol detail pada salah satu data mahasiswa
Hasil yang Diharapkan	Sistem menampilkan detail data denda mahasiswa.
Hasil Pengujian	Sistem berhasil menampilkan detail data denda mahasiswa.
Status Uji	Valid

7.2 User Acceptance Testing

User acceptance testing dilakukan dengan cara membuat daftar pertanyaan yang bisa mewakili seluruh kebutuhan yang sudah didefinisikan di bab analisis kebutuhan. Selanjutnya masing-masing tester akan diberikan pertanyaan sesuai dengan fungsi yang diberikan kepada masing-masing tester. Sebelum tester diberikan pertanyaan, pengembang akan mendemonstrasikan program menyesuaikan dengan *task* yang akan diberikan ditunjukkan pada tabel 7.9.

Tabel 7.9 Task yang perlu dikerjakan tester

No	Tugas yang dikerjakan	Tester
1	Mengelola data peminjaman buku	Petugas ruang baca
2	Mengelola data pengembalian buku	Petugas ruang baca
3	Melakukan pencarian informasi buku	Petugas ruang baca dan Mahasiswa
4	Mengelola data buku	Petugas ruang baca
5	Mengelola data mahasiswa	Petugas ruang baca
6	Mengelola data denda	Petugas ruang baca
7	Melihat informasi buku yang dipinjam	Mahasiswa
8	Melihat riwayat transaksi	Petugas ruang baca
9	Melakukan perubahan password	Mahasiswa
10	Melihat riwayat peminjaman dan denda	Mahasiswa
11	Mengelola data rak buku	Petugas ruang baca
12	Melakukan proses otentikasi	Petugas ruang baca dan mahasiswa
13	Melakukan proses keluar dari sistem	Petugas ruang baca dan Mahasiswa

Pertanyaan akan diberikan kepada masing-masing tester, yaitu pertanyaan untuk petugas ruang baca dan mahasiswa. Hasil dari pertanyaan yang telah diberikan ditunjukkan pada lampiran. Pada tabel 7.10 menunjukkan pertanyaan yang akan diajukan kepada tester.

Tabel 7.10 Daftar Pertanyaan UAT

No	Pertanyaan	Kriteria	Tester
1	Apakah fitur pengelolaan data peminjaman buku dapat membantu petugas ruang baca untuk mengelola data peminjaman?	<i>Functional Correctness and Completeness</i>	Petugas ruang baca
2	Apakah fitur pengelolaan data pengembalian buku dapat membantu petugas ruang baca untuk mengelola data pengembalian?		Petugas ruang baca
3	Apakah fitur pengelolaan data buku dapat membantu petugas ruang baca untuk mengelola data buku?		Petugas ruang baca

Tabel 7.10 Daftar Pertanyaan UAT (Lanjutan)

No	Pertanyaan	Kriteria	Tester
4	Apakah fitur pengelolaan data mahasiswa dapat membantu petugas ruang baca untuk mengelola data mahasiswa?	Usability	Petugas ruang baca
5	Apakah fitur pengelolaan data denda dapat membantu petugas ruang baca untuk mengelola data denda?		Petugas ruang baca
6	Apakah fitur pengelolaan data rak buku dapat membantu petugas ruang baca untuk mengelola rak buku?		Petugas ruang baca
7	Apakah sistem informasi manajemen perpustakaan membantu dalam pencarian informasi buku?		Petugas ruang baca, Mahasiswa
8	Apakah sistem informasi manajemen perpustakaan membantu dalam melihat riwayat transaksi?		Petugas ruang baca
9	Apakah sistem informasi manajemen perpustakaan membantu dalam melihat informasi buku yang dipinjam?		Mahasiswa
10	Apakah sistem informasi manajemen perpustakaan membantu dalam melihat riwayat peminjaman dan denda ?		Mahasiswa
11	Apakah sistem informasi manajemen perpustakaan dapat melakukan perubahan password ?		Mahasiswa
12	Apakah dengan adanya sistem informasi manajemen perpustakaan dapat meningkatkan kinerja dalam melakukan pelayanan terhadap pengguna?		Petugas ruang baca, Mahasiswa
13	Apakah informasi yang diberikan kepada pengguna sudah sesuai dengan kebutuhan pengguna?		Petugas ruang baca, Mahasiswa

Tabel 7.10 Daftar Pertanyaan UAT (Lanjutan)

No	Pertanyaan	Kriteria	Tester
14	Apakah anda sebagai pengguna senang dengan tampilan web yang ada saat ini?		Petugas ruang baca, Mahasiswa
15	Apakah sistem informasi manajemen perpustakaan mempermudah dalam penyampaian informasi terkait dengan pelayanan yang ada?		Petugas ruang baca, Mahasiswa
16	Apakah anda sebagai mahasiswa merasa terjamin dengan keamanan data dari sistem informasi manajemen perpustakaan?	<i>Confidentiality and Availability</i>	Mahasiswa
17	Apakah anda sebagai petugas ruang baca merasa terjamin dengan keamanan data dari sistem informasi manajemen perpustakaan?		Petugas ruang baca

Berdasarkan data hasil pengujian UAT kepada petugas ruang baca dan mahasiswa, maka hasil analisis pengujian UAT akan disesuaikan dengan kriteria dan menghitung persentase jawaban pengguna.

Tabel 7.10 Hasil Pengujian UAT

No	Kriteria	Jumlah Pertanyaan Diajukan	Jawaban "YA"	Jawaban "TIDAK"	Presentase Jawaban "YA"
1	<i>Functional Correctness and Completeness</i>	28	28	0	100%
2	<i>Usability</i>	18	18	0	100%
3	<i>Confidentiality and Availability</i>	12	12	0	100%

Berdasarkan hasil pengujian UAT pada tabel 7.10 menunjukkan 100% dari keseluruhan pertanyaan yang diberikan memberikan jawaban "YA". Maka dapat diambil kesimpulan bahwa sistem informasi manajemen perpustakaan dapat diterima dengan baik dan sudah sesuai dengan kebutuhan yang disepakati diawal.

7.3 Compatibility Testing

Compatibility testing merupakan sebuah pengujian yang bertujuan untuk mengetahui kemampuan sistem dalam menunjukkan kesesuaian terhadap *web browser* yang digunakan. *Compatibility testing* dilakukan dengan menggunakan software SortSite versi 5.29.818.0. *Compatibility testing* sistem informasi manajemen perpustakaan dilakukan pada laman yang diakses oleh pengguna. Hasil *compatibility testing* yang terdapat dalam gambar 7.2

Browser Version	IE			Edge	Firefox		Safari		Opera	Chrome	iOS			Android		BlackBerry	
	9	10	11	16	60	≤ 10	11	51	66	≤ 9	10	11	≤ 3	4*	≤ 7.1	10.0	
Critical Issues	✓	✓	✓	✓	●	✓	✓	✓	●	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	●
Minor Issues	●	✓	✓	✓	●	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	●

Key
 ● Missing content or functionality
 ● Major layout or performance problems
 ● Minor layout or performance problems

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Gambar 7.2 Hasil pengujian kompatibilitas pada SortSite

Pada gambar 7.2 merupakan hasil pengujian kompatibilitas, diketahui bahwa terdapat *critical issue* pada jenis browser *firefox 60* dan *chrome 66*. *Critical issue* muncul akibat tidak diaktifkannya fitur ssl atau https pada saat melakukan pengujian pada halaman login. Kemudian terdapat *major issue* pada jenis browser blackberry <= 7 dikarenakan CSS tidak didukung oleh browser. Sedangkan minor issue pada jenis browser IE 9, firefox 60, dan blackberry <= 7 dikarenakan CSS tidak didukung oleh browser.

7.4 Analisis Hasil

Pada analisis hasil pengujian *black box* berupa validasi sistem berdasarkan dari hasil pengujian validasi pada subbab 7.1 yang menjelaskan bahwa nomor kode uji VT-01 hingga VT- 06 adalah valid, artinya dari semua kasus uji yang sudah dilakukan dalam penelitian pada hasil pengujian yang didapatkan sudah tidak terdapat kesalahan atau *error* dan sesuai dengan apa yang diharapkan oleh peneliti untuk membangun sistem informasi. Pengujian validasi dilakukan secara keseluruhan yang merujuk pada *use case* yang sudah dibuat. Hasil yang didapatkan dari *validation testing* dapat disimpulkan bahwa sistem informasi yang dibangun dapat berjalan sesuai dengan hasil yang diharapkan.

Pada analisis hasil perhitungan kriteria *user acceptance testing* ke petugas ruang baca dan mahasiswa yang telah dihitung dengan menggunakan *skala gutman* menghasilkan nilai 100% keseluruhan pertanyaan yang diberikan memberikan jawaban “YA”, yang berarti dalam interpretasi nilai berarti pihak petugas ruang baca dan mahasiswa dapat diterima dengan baik dan sudah sesuai dengan kebutuhan yang disepakati diawal.

Pada analisis hasil pengujian kompatibilitas pada gambar 7.2 merupakan hasil pengujian kompatibilitas, diketahui bahwa terdapat *critical issue* pada jenis browser *firefox 60* dan *chrome 66*. *Critical issue* muncul akibat tidak diaktifkannya fitur ssl atau https pada saat melakukan pengujian pada halaman login. Kemudian terdapat *major issue* pada jenis browser blackberry <= 7 dikarenakan CSS tidak



didukung oleh browser. Sedangkan minor issue pada jenis browser IE 9, firefox 60, dan blackberry <= 7 dikarenakan CSS tidak didukung oleh browser.



BAB 8 PENUTUP

8.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Hasil analisis proses bisnis pada penelitian ini terdapat 3 proses bisnis usulan (*to-be*) yang merupakan perbaikan pada aktivitas peminjaman buku, pengembalian buku, dan pengelolaan denda. Proses bisnis usulan (*to-be*) didapatkan berdasarkan solusi permasalahan proses bisnis yang sedang berjalan pada Ruang Baca Fakultas Ilmu Komputer (*as-is*). Hasil analisis permasalahan menghasilkan 28 kebutuhan fungsional, dan 1 kebutuhan non-fungsional memberikan gambaran umum kemampuan yang dapat dilakukan oleh sistem informasi manajemen perpustakaan sebagai berikut:
 - Kebutuhan fungsional (SKPL-F-01, SKPL-F-02, SKPL-F-03) untuk menyelesaikan permasalahan terkait peminjaman buku.
 - Kebutuhan fungsional SKPL-F-04 untuk menyelesaikan permasalahan dalam melakukan pengembalian buku,
 - Kebutuhan fungsional (SKPL-F-05, SKPL-F-06, SKPL-F-07) untuk menyelesaikan permasalahan dalam melakukan pencarian informasi buku
 - Kebutuhan fungsional (SKPL-F-08, SKPL-F-09, SKPL-F-10, SKPL-F-11, SKPL-F-12) untuk menyelesaikan permasalahan terkait pengelolaan data buku.
 - Kebutuhan fungsional (SKPL-F-13, SKPL-F-14, SKPL-F-15, SKPL-F-16, SKPL-F-12) untuk menyelesaikan permasalahan terkait pengelolaan data mahasiswa.
 - Kebutuhan fungsional (SKPL-F-17,SKPL-F-18) untuk menyelesaikan permasalahan terkait pengelolaan denda.
 - Kebutuhan fungsional SKPL-F-19 untuk menyelesaikan permasalahan dalam melihat informasi buku yang dipinjam mahasiswa.
 - Kebutuhan fungsional SKPL-F-20 untuk menyelesaikan permasalahan dalam melihat riwayat informasi transaksi.
 - Kebutuhan fungsional SKPL-F-21 untuk menyelesaikan permasalahan dalam melakukan perubahan *password*.
 - Kebutuhan fungsional SKPL-F-22 untuk menyelesaikan permasalahan dalam melihat riwayat peminjaman dan denda,
 - Kebutuhan fungsional (SKPL-F-23, SKPL-F-24, SKPL-F-25, SKPL-F-26) untuk menyelesaikan permasalahan terkait pengelolaan data rak buku.

- Kebutuhan fungsional SKPL-F-27 untuk menyelesaikan permasalahan dalam melakukan proses autentikasi.
 - Kebutuhan fungsional SKPL-F-28 untuk menyelesaikan permasalahan dalam melakukan proses keluar dari sistem.
 - Kebutuhan non-fungsional yaitu kompatibilitas dimana sistem harus dapat berjalan dengan benar di berbeda *browser* dan sistem operasi.
2. Hasil dari perancangan sistem berdasarkan hasil analisis kebutuhan menghasilkan 13 *use case*, 6 *sequence* diagram, 7 kelas pada diagram kelas analisis, 8 kelas *controller* dan 6 kelas *model* pada diagram kelas perancangan, 7 tabel pada perancangan basis data, perancangan antarmuka, dan perancangan *web service*.
 3. Proses implementasi sistem informasi manajemen perpustakaan dilakukan berdasarkan hasil perancangan sistem. Sistem dibangun dengan menggunakan HTML, CSS, PHP, Javascript, Codeigniter, dan MySQL sebagai database sistem. Sistem memberikan fitur identifikasi identitas pengguna, mengelola peminjaman buku, mengelola pengembalian buku, melakukan pencarian informasi buku, mengelola data buku, mengelola data mahasiswa, mengelola data denda, melihat informasi buku yang dipinjam, melihat informasi transaksi, melakukan perubahan password, melihat riwayat peminjaman dan denda, mengelola rak buku, dan proses keluar dari sistem.
 4. Hasil pengujian *black-box testing* berupa pengujian validasi pada beberapa fungsi yang berkaitan dengan pengelolaan peminjaman buku, pengelolaan pengembalian buku, dan pengelolaan data denda dinyatakan valid. Hasil pengujian *user acceptance test* (UAT) menunjukkan bahwa sistem dapat diterima 100% dari sisi *stackholder* sebagai petugas ruang baca, dan mahasiswa dengan seluruh jawaban “YA” pada kuisioner yang dibagikan. Hasil pengujian kompatibilitas menunjukkan bahwa terdapat *critical issue* pada jenis browser *firefox 60* dan *chrome 66*. *Critical issue* muncul akibat tidak diaktifkannya fitur ssl atau https pada saat melakukan pengujian pada halaman login. Kemudian terdapat *major issue* pada jenis browser blackberry <= 7 dikarenakan CSS tidak didukung oleh browser. Sedangkan minor issue pada jenis browser IE 9, firefox 60, dan blackberry <= 7 dikarenakan CSS tidak didukung oleh browser.

8.2 Saran

1. Perlu dilakukan penerapan lapangan Sistem Informasi Manajemen Perpustakaan pada Ruang Baca, kemudian dilakukan kegiatan monitoring dan evaluasi penerapan lapangan sehingga memungkinkan hasil pengembangan lanjut Sistem Informasi Manajemen Perpustakaan yang lebih baik.
2. Perlu dilakukan evaluasi pada antarmuka pengguna Sistem Informasi Manajemen Perpustakaan untuk meningkatkan pengalaman dan kenyamanan pengguna dalam menggunakan sistem.

DAFTAR PUSTAKA

- 1keydata, 2017. Physical Data Model. [Online] Tersedia di: <http://www.1keydata.com/datawarehousing/physical-data-model.html> [Diakses 6 Agustus 2017].
- A.S, R. dan M. Shalahudin. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika, 2018.
- Bittner, K. & Spence, I., 2002. *Use Case Modeling*. Boston: Addison Wesley.
- Booch, G., Rumbaugh, J. & Jacobson, I., 2005. *The Unified Modeling Language User Guide*. 2nd penyunt. Reading: Addison-Wesley Longman.
- Brandenburg, L., 2017. *Bridging The Gap*. [Online] Available at: <http://www.bridging-the-gap.com/as-is-business-process/> [Diakses 30 January 2018].
- Hall, J. A., 2008. *Accounting Information System*. 6th penyunt. Mason: Cengage Learning.
- Hambling, B. & Gothem, P. v., 2013. *User Acceptance Testing: A Step-by-step Guide*. Swindon: BCS Learning & Development Limited.
- Hustinawati, Himawan, A. K. & Latifah, 2014. Performance Analysis Framework Codeigniter and CakePHP in Website Creation. *International Journal of Computer Applications*, Volume 94, p. 6.
- Indriati. 2010. *Pengujian Validasi*. [Online] Tersedia di: <http://indryz.lecture.ub.ac.id/files/2010/11/Pengujian-Validasi.docx> [Diakses 25 Oktober 2017].
- Laudon, K. C. a. J. P., 2014. *Management Information Systems: Managing the Digital Firm*. 13th ed. United States of America: Pearson Education.
- Leffingwell, D. & Widrig, D., 2003. *Managing Software Requirements: A Use Case Approach*. 2nd penyunt. Boston: Addison Wesley.
- Monk, E. & Wagner, B., 2013. *Concepts in Enterprise Resource Planning*. 4th penyunt. Boston: Course Technology.
- Naik, K. & Tripathy, P., 2008. *Software Testing And Quality Assurance*. In: *Theory and Practice*. s.l.:s.n., p. 452.

- Object Management Group, 2011. *The Object Management Group® (OMG®)*. [Online]. Available at: <http://www.omg.org/spec/BPMN/2.0/PDF> [Diakses 10 Agustus 2017].
- O'Brien, J. A. a. G. M., 2007. *Introduction to Information System*. 5th ed. New York: McGraw-Hill/Irwin.
- Pressman, R. S., 2010. *Software Engineering: A Practitioner's Approach*. 7th Edition. New York: McGraw-Hill.
- Przybyłek, A., 2011. *Bridging the gap between business process models and use-case models*. Gdańsk, Gdańsk University of Technology.
- Sugiyono. 2012. *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung: Alfabeta
- Shodiqin, Faishal Faris. 2016. *Rancang Bangun Sistem Informasi Manajemen Perpustakaan Pada Ruang Baca Fakultas Ilmu Komputer Universitas Brawijaya*. S1. Universitas Brawijaya Malang
- Sommerville, Ian. 2011. *Software Engeneering, Ninth Edition*. New York: AddisonWesley.
- Suwarno, Wiji. 2016. *Ilmu Perpustakaan & Kode Etik Pustakawan*. Jogjakarta : Ar-Ruzz Media
- Undang-undang Republik Indonesia nomor 43 tahun 2007 tentang Perpustakaan, Jakarta: Kementrian Sekretariat Negara Republik Indonesia
- Weske, M., 2007. *Business Process Management : Concepts, Languages, Architectures*. Berlin: Springer.
- Zota, Rayvan Daniel dan Liviu Ciovita. 2015. *Designing Software Solutions Using Business Processes*. Gdańsk, Gdańsk University of Technology.