

**PENGEMBANGAN SISTEM INFORMASI PERPUSTAKAAN  
BERBASIS WEB PADA PERPUSTAKAAN KECAMATAN  
BUNGAH**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Nurur Rohmah  
NIM: 145150407111037



PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

# PENGESAHAN

**PENGEMBANGAN SISTEM INFORMASI PERPUSTAKAAN BERBASIS WEB PADA  
PERPUSTAKAAN KECAMATAN BUNGAH**

**SKRIPSI**

**Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer**

**Disusun oleh:  
Nurur Rohmah  
NIM: 145150407111037**

**Skripsi ini telah diuji dan dinyatakan lulus pada  
27 Desember 2018**

**Telah diperiksa dan disetujui oleh:**

**Pembimbing I**  
A-h

Himawat Aryadita, S.T., M.Sc.  
NIP: 198010182008011003

**Pembimbing II**

Adam Hendra Brata, S.Kom., M.T., M.Sc.  
NIK: 2016079001051001



**Mengetahui  
Ketua Jurusan Sistem Informasi**

Dr. Eng. Herman Tolle, S.T., M.T.  
NIP: 197408232000121001



## PERNYATAAN ORISINALITAS

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

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

Malang, 27 Desember 2018



Nurur Rohmah

NIM: 145150407111037



## KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Pengembangan Sistem Informasi Perpustakaan Berbasis Web Pada Perpustakaan Kecamatan Bungah” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Himawat Aryadita, S.T., M.Sc dan Bapak Adam Hendra Brata, M.T, M.Sc selaku pembimbing skripsi yang telah meluangkan waktu dan dengan sabar membimbing serta mengarahkan penulis sehingga dapat menyelesaikan skripsi ini,
2. Bapak Yusi Tyroni Mursityo, S.Kom., M.AB selaku Ketua Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya Malang,
3. Bapak Herman Tolle, Dr. Eng., S.T., M.T selaku Ketua Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya Malang,
4. Ibu Rochimah, S.Pd.I selaku Pengelola Perpustakaan Kecamatan Bungah yang telah memberi izin dan memperlancar pengambilan data dalam pelaksanaan penelitian skripsi,
5. Para pegawai kantor Kecamatan Bungah yang telah memberi bantuan memperlancar perizinan dalam pelaksanaan penelitian skripsi,
6. Bapak, Ibu, Kakak dan seluruh keluarga besar atas segala nasihat, dukungan dan senantiasa tiada henti-hentinya memberikan do’a kepada penulis sehingga penulis dapat mengerjakan skripsi dengan lancer,
7. Semua pihak secara langsung maupun tidak langsung yang telah membantu terselesainya laporan skripsi ini yang tidak dapat penulis sebut satu persatu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya

Malang, 17 Desember 2018

Penulis

Email: 145150407111037@ub.ac.id

## ABSTRAK

**Nurur Rohmah, Pengembangan Sistem Informasi Perpustakaan Berbasis Web Pada Perpustakaan Kecamatan Bungah**

**Pembimbing: Himawat Aryadita, S.T., M.Sc. dan Adam Hendra Brata, S.Kom, M.T., M.Sc.**

Perpustakaan Kecamatan Bungah memiliki kendala dalam pengelolaan data dan efisiensi pelayanan peminjaman dan pengembalian buku yang dilakukan oleh petugas perpustakaan. Kompleksnya koleksi perpustakaan, data pengunjung, serta transaksi peminjaman menuntut kebutuhan akan penggunaan teknologi informasi untuk otomatisasi proses bisnis di Perpustakaan Kecamatan Bungah. Penelitian ini bertujuan untuk mengembangkan sistem informasi perpustakaan berbasis web yang dapat membantu pengelolaan data dan meningkatkan kualitas pelayanan transaksi peminjaman pada perpustakaan. Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode *waterfall* yang mana pada tahap perancangan dan implementasi menerapkan pola perancangan *model view controller* atau MVC. Sedangkan pada proses pengujian dilakukan beberapa tahap pengujian secara berurutan yaitu *unit testing*, *integration testing*, *validation testing*, *reliability testing*, *compatibility testing*, dan *acceptance testing*. *Unit testing* menggunakan teknik *basis path* menunjukkan hasil bahwa sistem memiliki struktur yang sederhana, mudah dipahami dan diimplementasi. *Integration testing* menggunakan metode *white box* menunjukkan hasil yang *valid*. *Validation testing* menggunakan metode *black box* menunjukkan bahwa 26 fungsi berjalan dengan baik. *Reliability testing* dengan menggunakan perhitungan nelson menunjukkan bahwa sistem dapat berjalan dengan konsisten. *Compatibility testing* menunjukkan sistem dapat digunakan dengan baik pada 6 jenis *browser*. Dan *acceptance testing* menunjukkan bahwa semua fungsi berhasil dijalankan dengan baik dan telah diterima oleh *stakeholder*.

Kata kunci: *pengembangan, sistem informasi perpustakaan, waterfall, model view controller*

## ABSTRACT

**Nurur Rohmah, Development of Web-Based Library Information System In Bungah District Library**

**Mentor: Himawat Aryadita, S.T., M.Sc. and Adam Hendra Brata, S.Kom, M.T., M.Sc.**

Bungah District Library has problems in data management and efficiency of loaning and returning services by library staff. The complexity of library collections, library visitor data, and loan transactions demand the need of information technology to automate business processes in Bungah District Library. This study aims to build a web-based library information system that can help manage data and improve the quality of service loan transactions at the library. The software development method that used in this study is the waterfall method, which at the design and implementation stage applies the design pattern of model view controller or MVC pattern. While at the testing stage conducted sequentially from unit testing, integration testing, validation testing, reliability testing, compatibility testing, and acceptance testing. Unit tests using basis path technique shows that system has simple structure, easy to understand and implemented. Integration tests using white box method shows valid result. Validation tests using black box method shows that 26 functions of system can work properly. Reliability tests using nelson calculation shows that the system can run consistently. Compatibility testing result shows that the system can be used well on 6 types of browser. And the result of acceptance testing shows that all of functions can run well and accepted by stakeholder.

Keywords: *development, library information system, waterfall, model view controller*

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
<i>ABSTRACT</i> .....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiv
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	3
1.5 Batasan Masalah .....	3
1.6 Sistematika Pembahasan .....	3
<b>BAB 2 LANDASAN KEPUSTAKAAN .....</b>	<b>5</b>
2.1 Kajian Pustaka .....	5
2.2 Landasan Teori.....	6
2.2.1 Perpustakaan .....	6
2.2.2 Teknologi Informasi Perpustakaan .....	7
2.2.3 Sistem Informasi.....	8
2.2.4 Sistem Informasi Perpustakaan .....	8
2.2.5 <i>Bussines Process Model Notation</i> .....	9
2.2.6 <i>Physical Data Model (PDM)</i> .....	11
2.2.7 <i>Software Development Life Cycle (SDLC)</i> .....	12
2.2.8 Model Waterfall .....	13
2.2.9 <i>Design Pattern</i> .....	22
2.2.10 HTML .....	23
2.2.11 PHP .....	26
2.2.12 MySQL .....	27

BAB 3 METODOLOGI .....	28
3.1 Studi Pustaka dan Penyusunan Dasar Teori .....	28
3.2 Pengumpulan Data .....	29
3.3 Analisis Kebutuhan Sistem.....	30
3.4 Perancangan Sistem.....	31
3.5 Implementasi .....	31
3.6 Pengujian .....	31
3.7 Penarikan Kesimpulan dan Saran .....	31
BAB 4 ANALISIS KEBUTUHAN .....	32
4.1 <i>Bussiness Prespective</i> .....	32
4.1.1 Identifikasi Proses Bisnis As-Is.....	32
4.1.2 Analisis Permasalahan.....	34
4.1.3 Identifikasi Proses Bisnis To-Be.....	36
4.2 Deskripsi Perangkat Lunak.....	40
4.2.1 Fungsi Perangkat Lunak .....	40
4.2.2 Batasan Perangkat Lunak.....	40
4.2.3 Lingkungan Operasi.....	41
4.3 Identifikasi Kebutuhan .....	41
4.4 Identifikasi Pengguna.....	42
4.5 Identifikasi Fitur .....	43
4.6 Analisis Kebutuhan Fungsional .....	44
4.7 Analisis Kebutuhan Non-Fungsional .....	46
4.8 <i>Use Case Diagram</i> .....	47
4.9 <i>Use Case Scenario</i> .....	48
4.9.1 <i>Use Case Scenario</i> Petugas.....	48
4.9.2 <i>Use Case Scenario</i> Anggota .....	61
4.9.3 <i>Use Case Scenario</i> Pengunjung .....	64
BAB 5 PERANCANGAN DAN IMPLEMENTASI .....	67
5.1 <i>Class Diagram</i> .....	67
5.2 <i>Physical Data Model Diagram</i> .....	69
5.3 <i>Sequence Diagram</i> .....	70
5.3.1 <i>Sequence Diagram</i> Petugas.....	70

5.3.2	<i>Sequence Diagram</i> Anggota .....	82
5.3.3	<i>Sequence Diagram</i> Pengunjung .....	86
5.4	Perancangan Antarmuka .....	89
5.4.1	Halaman <i>Login</i> Petugas.....	89
5.4.2	Halaman <i>Dashboard</i> .....	89
5.4.3	Halaman Data Buku.....	90
5.4.4	Halaman Data Anggota .....	90
5.4.5	Halaman Data Tamu.....	91
5.4.6	Halaman Transaksi .....	91
5.4.7	Halaman Data Inventaris.....	92
5.4.8	Halaman Pengaturan Akun Petugas.....	92
5.4.9	Halaman Utama Pengguna .....	93
5.4.10	Halaman Presensi Tamu.....	93
5.4.11	Halaman Cari Buku.....	94
5.4.12	Halaman Registrasi Anggota .....	94
5.4.13	Halaman Login Anggota .....	95
5.4.14	Halaman Daftar Riwayat Peminjaman .....	95
5.4.15	Halaman Pengaturan Akun Anggota.....	96
5.5	Perangkat Keras .....	96
5.6	Perangkat Lunak .....	96
5.7	Implementasi Algoritma .....	96
5.7.1	Kelas Controller PetugasController.....	97
5.7.2	Kelas Model PetugasModel dan Kelas Anggota.....	98
5.8	<i>Screenshot</i> Program.....	100
BAB 6	PENGUJIAN .....	108
6.1	<i>Unit Testing</i> .....	108
6.1.1	<i>Basis Path Testing</i> .....	108
6.2	<i>Integration Testing</i> .....	112
6.3	<i>Validation Testing</i> .....	113
6.4	<i>Reliability Testing</i> .....	128
1.6	<i>Acceptance Testing</i> .....	130
1.7	Analisis Pengujian.....	137

BAB 7 PENUTUP .....	139
7.1 Kesimpulan.....	139
7.2 Saran .....	140
DAFTAR PUSTAKA.....	141
LAMPIRAN A HASIL WAWANCARA.....	143
LAMPIRAN B ACCEPTANCE TESTING .....	145
LAMPIRAN C SURAT KETERANGAN PENELITIAN .....	148
LAMPIRAN D DOKUMENTASI .....	149



## DAFTAR TABEL

Tabel 2.1 Notasi Pada <i>Use Case Diagram</i> .....	16
Tabel 2.2 Notasi <i>Sequence Diagram</i> .....	17
Tabel 2.3 Notasi <i>Class Diagram</i> .....	19
Tabel 2.4 Tag HTML.....	24
Tabel 3.1 Tahapan metodologi pengembangan sistem waterfall .....	30
Tabel 4.1 Analisis Permasalahan Keanggotaan dan Peminjaman Buku .....	34
Tabel 4.2 Analisis dari Rekomendasi Proses Bisnis Keanggotaan dan Peminjaman Buku.....	35
Tabel 4.3 Analisis Permasalahan Pengembalian Buku.....	35
Tabel 4.4 Analisis dari Rekomendasi Proses Bisnis Pengembalian Buku.....	36
Tabel 4.5 Pemetaan Proses Bisnis Baru Keanggotaan dan Peminjaman Buku.....	38
Tabel 4.6 Pemetaan Proses Bisnis Baru Pengembalian Buku .....	39
Tabel 4.7 Identifikasi Kebutuhan Pengguna .....	41
Tabel 4.8 Identifikasi Pengguna .....	42
Tabel 4.9 Identifikasi Fitur.....	43
Tabel 4.10 Kebutuhan Fungsional.....	44
Tabel 4.11 Kebutuhan Non-Fungsional.....	46
Tabel 4.12 <i>Use Case Scenario Login</i> Pada Sisi Petugas .....	48
Tabel 4.13 <i>Use Case Scenario Melihat Dashboard</i> .....	48
Tabel 4.14 <i>Use Case Scenario</i> Melihat Data Buku .....	49
Tabel 4.15 <i>Use Case Scenario</i> Menambah Data Buku .....	49
Tabel 4.16 <i>Use Case Scenario</i> Menghapus Data Buku.....	50
Tabel 4.17 <i>Use Case Scenario</i> Memperbarui Data Buku.....	50
Tabel 4.18 <i>Use Case Scenario</i> Melihat Data Anggota .....	51
Tabel 4.19 <i>Use Case Scenario</i> Menambah Data Anggota Pada Sisi Petugas .....	51
Tabel 4.20 <i>Use Case Scenario</i> Menghapus Data Anggota.....	52
Tabel 4.21 <i>Use Case Scenario</i> Memperbarui Data Anggota Pada Sisi Petugas ....	53
Tabel 4.22 <i>Use Case Scenario</i> Mencetak Kartu Anggota .....	54
Tabel 4.23 <i>Use Case Scenario</i> Mencetak Struk Peminjaman.....	54
Tabel 4.24 <i>Use Case Scenario</i> Melihat Data Transaksi Peminjaman .....	55

Tabel 4.25 <i>Use Case Scenario</i> Menambah Data Transaksi Peminjaman .....	55
Tabel 4.26 <i>Use Case Scenario</i> Menghapus Data Transaksi Peminjaman.....	56
Tabel 4.27 <i>Use Case Scenario</i> Memperbarui Data Transaksi Peminjaman .....	57
Tabel 4.28 <i>Use Case Scenario</i> Melihat Denda Peminjaman .....	57
Tabel 4.29 <i>Use Case Scenario</i> Melihat Data Tamu.....	58
Tabel 4.30 <i>Use Case Scenario</i> Menghapus Data Tamu .....	58
Tabel 4.31 <i>Use Case Scenario</i> Melihat Data Inventaris.....	59
Tabel 4.32 <i>Use Case Scenario</i> Menambah Data Inventaris .....	59
Tabel 4.33 <i>Use Case Scenario</i> Menghapus Data Inventaris .....	60
Tabel 4.34 <i>Use Case Scenario</i> Memperbarui Data Inventaris.....	60
Tabel 4.35 <i>Use Case Scenario Login</i> Pada Sisi Anggota .....	61
Tabel 4.36 <i>Use Case Scenario</i> Menambah Data Tamu Pada Sisi Anggota .....	61
Tabel 4.37 <i>Use Case Scenario</i> Mencari Buku Pada Sisi Anggota.....	62
Tabel 4.38 <i>Use Case Scenario</i> Memperbarui Data Anggota Pada Sisi Anggota....	63
Tabel 4.39 <i>Use Case Scenario</i> Melihat Riwayat Peminjaman .....	63
Tabel 4.40 <i>Use Case Scenario</i> Menambah Data Anggota Pada Sisi Pengunjung..	64
Tabel 4.41 <i>Use Case Scenario</i> Menambah Data Tamu Pada Sisi Pengunjung .....	64
Tabel 4.42 <i>Use Case Scenario</i> Mencari Buku Pada Sisi Pengunjung .....	65
Tabel 5.1 Kode Program Kelas Controller PetugasController .....	97
Tabel 5.2 Kode Program Kelas Model PetugasModel.....	98
Tabel 5.3 Kode Program Kelas Anggota .....	99
Tabel 6.1 Algoritma Fungsi editAnggota() .....	108
Tabel 6.2 Test Case editAnggota .....	110
Tabel 6.3 Hasil Pengujian Integration Testing Kelas PetugasController dengan Kelas PetugasModel .....	112
Tabel 6.4 Pengujian <i>Login</i> Pada Sisi Petugas .....	113
Tabel 6.5 Pengujian Menampilkan Data Buku .....	113
Tabel 6.6 Pengujian Menambahkan Data Buku .....	114
Tabel 6.7 Pengujian Menghapus Data Buku .....	114
Tabel 6.8 Pengujian Meperbarui Data Buku .....	115
Tabel 6.9 Pengujian Menampilkan <i>Dashboard</i> .....	115
Tabel 6.10 Pengujian Menampilkan Data Anggota.....	115

Tabel 6.11 Pengujian Menambahkan Data Anggota Pada Sisi Petugas.....	116
Tabel 6.12 Pengujian Menghapus Data Anggota.....	116
Tabel 6.13 Pengujian Meperbarui Data Anggota Pada Sisi Petugas.....	117
Tabel 6.14 Pengujian Mencetak Kartu Anggota.....	117
Tabel 6.15 Pengujian Menampilkan Data Transaksi Peminjaman.....	118
Tabel 6.16 Pengujian Menambahkan Data Transaksi Peminjaman.....	118
Tabel 6.17 Pengujian Menghapus Data Transaksi Peminjaman.....	119
Tabel 6.18 Pengujian Meperbarui Data Transaksi Peminjaman.....	119
Tabel 6.19 Pengujian Menampilkan Denda Peminjam.....	120
Tabel 6.20 Pengujian Menampilkan Data Tamu.....	120
Tabel 6.21 Pengujian Menghapus Data Tamu.....	121
Tabel 6.22 Pengujian Menampilkan Data Inventaris.....	121
Tabel 6.23 Pengujian Menambahkan Data Inventaris.....	122
Tabel 6.24 Pengujian Menghapus Data Inventaris.....	122
Tabel 6.25 Pengujian Meperbarui Data Inventaris.....	122
Tabel 6.26 Pengujian Mencetak Struk Peminjaman.....	123
Tabel 6.27 Pengujian <i>Login</i> Pada Sisi Anggota.....	124
Tabel 6.28 Pengujian Memperbarui Data Anggota Pada Sisi Anggota.....	124
Tabel 6.29 Pengujian Mencari Buku Pada Sisi Anggota.....	124
Tabel 6.30 Pengujian Menambahkan Data Tamu Pada Sisi Anggota.....	125
Tabel 6.31 Pengujian Menampilkan Riwayat Peminjaman.....	125
Tabel 6.32 Pengujian Menambahkan Data Anggota Pada Sisi Pengunjung.....	126
Tabel 6.33 Pengujian Mencari Buku Pada Sisi Pengunjung.....	126
Tabel 6.34 Pengujian Menambahkan Data Tamu Pada Sisi Pengunjung.....	127
Tabel 6.35 Hasil <i>Stress Testing</i> .....	128
Tabel 6.36 Hasil Uji <i>User Acceptance</i> yang Dilakukan <i>Stakeholder</i> .....	130
Tabel 6.37 Hasil Pengujian Unit.....	137

## DAFTAR GAMBAR

Gambar 2.1 <i>Events</i> .....	9
Gambar 2.2 <i>Activities</i> .....	9
Gambar 2.3 <i>Gateways</i> .....	10
Gambar 2.4 <i>Sequence flows</i> .....	10
Gambar 2.5 <i>Message flow</i> .....	10
Gambar 2.6 <i>Association</i> .....	10
Gambar 2.7 <i>Pool dan Lane</i> .....	11
Gambar 2.8 <i>Data object</i> .....	11
Gambar 2.9 <i>Group</i> .....	11
Gambar 2.10 <i>Annotation</i> .....	11
Gambar 2.11 <i>The Waterfall Model</i> .....	14
Gambar 2.12 <i>Use Case Diagram</i> .....	17
Gambar 2.13 <i>Sequence Diagram</i> .....	18
Gambar 2.14 <i>Class Diagram</i> .....	20
Gambar 2.15 Notasi Flow Graph.....	21
Gambar 2.16 Model MVC.....	22
Gambar 2.17 Antarmuka yang mendefinisikan pola MVC.....	23
Gambar 2.18 Cara Kerja PHP.....	26
Gambar 3.1 Diagram Metode Penelitian .....	28
Gambar 4.1 Proses Bisnis As-Is Keanggotaan dan Peminjaman Buku.....	33
Gambar 4.2 Proses Bisnis As-Is Pengembalian Buku .....	33
Gambar 4.3 Proses Bisnis To-Be Keanggotaan dan Peminjaman Buku .....	37
Gambar 4.4 Proses Bisnis To-Be Pengembalian Buku .....	38
Gambar 4.5 <i>Use Case Diagram</i> .....	47
Gambar 5.1 <i>Class Diagram</i> .....	68
Gambar 5.2 <i>Physical Data Model Diagram</i> .....	69
Gambar 5.3 <i>Sequence Diagram Login Pada Sisi Petugas</i> .....	70
Gambar 5.4 <i>Sequence Diagram Melihat Dashboard</i> .....	70
Gambar 5.5 <i>Sequence Diagram Melihat Data Buku</i> .....	71
Gambar 5.6 <i>Sequence Diagram Menambahkan Data Buku</i> .....	71

Gambar 5.7 <i>Sequence Diagram</i> Menghapus Data Buku .....	72
Gambar 5.8 <i>Sequence Diagram</i> Memperbarui Data Buku .....	72
Gambar 5.9 <i>Sequence Diagram</i> Melihat Data Anggota .....	73
Gambar 5.10 <i>Sequence Diagram</i> Menambah Data Anggota Pada Sisi Petugas ...	73
Gambar 5.11 <i>Sequence Diagram</i> Menghapus Data Anggota .....	74
Gambar 5.12 <i>Sequence Diagram</i> Memperbarui Data Anggota Pada Sisi Petugas	74
Gambar 5.13 <i>Sequence Diagram</i> Mencetak Kartu Anggota .....	75
Gambar 5.14 <i>Sequence Diagram</i> Mencetak Struk Peminjaman.....	76
Gambar 5.15 <i>Sequence Diagram</i> Melihat Data Transaksi Peminjaman .....	77
Gambar 5.16 <i>Sequence Diagram</i> Menambah Data Transaksi Peminjaman .....	77
Gambar 5.17 <i>Sequence Diagram</i> Menghapus Data Transaksi Peminjaman .....	78
Gambar 5.18 <i>Sequence Diagram</i> Memperbarui Data Transaksi Peminjaman .....	78
Gambar 5.19 <i>Sequence Diagram</i> Melihat Denda Peminjaman .....	79
Gambar 5.20 <i>Sequence Diagram</i> Melihat Data Tamu .....	79
Gambar 5.21 <i>Sequence Diagram</i> Menghapus Data Tamu.....	80
Gambar 5.22 <i>Sequence Diagram</i> Melihat Data Inventaris .....	80
Gambar 5.23 <i>Sequence Diagram</i> Menambah Data Inventaris .....	81
Gambar 5.24 <i>Sequence Diagram</i> Menghapus data Inventaris .....	81
Gambar 5.25 <i>Sequence Diagram</i> Memperbarui Data Inventaris .....	82
Gambar 5.26 <i>Sequence Diagram</i> Login Pada Sisi Anggota .....	82
Gambar 5.27 <i>Sequence Diagram</i> Menambah Data Tamu Pada Sisi Anggota.....	83
Gambar 5.28 <i>Sequence Diagram</i> Mencari Buku Pada Sisi Anggota .....	84
Gambar 5.29 <i>Sequence Diagram</i> Memperbarui Data Anggota Pada Sisi Anggota85	
Gambar 5.30 <i>Sequence Diagram</i> Melihat Riwayat Peminjaman.....	86
Gambar 5.31 <i>Sequence Diagram</i> Menambah Data Anggota Pada Sisi Pengunjung .....	86
Gambar 5.32 <i>Sequence Diagram</i> Menambah Data Tamu Pada Sisi Pengunjung .	87
Gambar 5.33 <i>Sequence Diagram</i> Mencari Buku Pada Sisi Pengunjung.....	88
Gambar 5.34 Rancangan Halaman <i>Login</i> Petugas .....	89
Gambar 5.35 Rancangan Halaman <i>Dashboard</i> .....	89
Gambar 5.36 Rancangan Halaman Data Buku .....	90
Gambar 5.37 Rancangan Halaman Data Anggota.....	90

Gambar 5.38 Rancangan Halaman Data Tamu .....	91
Gambar 5.39 Rancangan Halaman Transaksi.....	91
Gambar 5.40 Rancangan Halaman Data Inventaris .....	92
Gambar 5.41 Rancangan Halaman Pengaturan Akun Petugas .....	92
Gambar 5.42 Rancangan Halaman Utama Pengguna .....	93
Gambar 5.43 Rancangan Halaman Presensi Tamu .....	93
Gambar 5.44 Rancangan Halaman Cari Buku .....	94
Gambar 5.45 Rancangan Halaman Registrasi Anggota.....	94
Gambar 5.46 Rancangan Halaman Login Anggota.....	95
Gambar 5.47 Rancangan Halaman Daftar Riwayat Peminjaman .....	95
Gambar 5.48 Rancangan Halaman Pengaturan Akun Anggota .....	96
Gambar 5.49 <i>Screenshot</i> Halaman <i>Login</i> Petuga.....	100
Gambar 5.50 <i>Screenshot</i> Halaman <i>Dashboard</i> .....	101
Gambar 5.51 <i>Screenshot</i> Halaman Data Buku .....	101
Gambar 5.52 <i>Screenshot</i> Halaman Data Anggota.....	102
Gambar 5.53 <i>Screenshot</i> Halaman Data Tamu .....	102
Gambar 5.54 <i>Screenshot</i> Halaman Transaksi Peminjaman .....	103
Gambar 5.55 <i>Screenshot</i> Halaman Data Inventaris .....	103
Gambar 5.56 <i>Screenshot</i> Halaman Pengaturan Akun Petugas .....	104
Gambar 5.57 <i>Screenshot</i> Halaman Utama Pengguna .....	104
Gambar 5.58 <i>Screenshot</i> Halaman Presensi Tamu .....	105
Gambar 5.59 <i>Screenshot</i> Halaman Cari Buku .....	105
Gambar 5.60 <i>Screenshot</i> Halaman Registrasi Anggota Baru .....	106
Gambar 5.61 <i>Screenshot</i> Halaman <i>Login</i> Anggota.....	106
Gambar 5.62 <i>Screenshot</i> Halaman Daftar Riwayat Peminjaman .....	107
Gambar 5.63 <i>Screenshot</i> Halaman Pengaturan Akun Anggota .....	107
Gambar 6.1 Flow Graph editAnggota .....	109
Gambar 6.2 Hasil <i>Stress Testing</i> Menggunakan WAPT.....	128
Gambar 6.3 Hasil <i>Compability Testing</i> Menggunakan Sortsite.....	129

## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Perpustakaan adalah suatu kesatuan unit kerja yang terdiri dari beberapa bagian yaitu bagian pengembangan koleksi, bagian pengolahan koleksi, bagian pelayanan pengguna, dan bagian pemeliharaan sarana dan prasarana. Perpustakaan memiliki tujuan untuk menyediakan fasilitas dan sumber informasi serta menjadi pusat pembelajaran bagi seluruh lapisan masyarakat. Seiring perkembangan teknologi informasi saat ini, seharusnya dapat memberikan dampak dalam pengelolaan perpustakaan dan pelayanan pengguna. Perpustakaan sebagai pengelola informasi dan pengetahuan harus bisa menggunakan dan memanfaatkan teknologi informasi dengan optimal untuk memenuhi berbagai kebutuhannya seperti pada pengelolaan datanya dan kualitas pelayanan terhadap pengguna perpustakaan.

Sistem informasi merupakan salah satu teknologi informasi yang menyediakan informasi untuk membantu pengambilan keputusan manajemen dan memberikan informasi yang layak. Dalam upaya meningkatkan kinerja pelayanan perpustakaan, penggunaan sistem informasi merupakan alternatif atau solusi yang tepat. Alasan untuk menggunakan sistem informasi diantaranya adalah pengelolaan data dan informasi yang menyeluruh, terintegrasi, terpadu, menghasilkan informasi yang cepat dan akurat, mereduksi biaya, dan keamanan yang lebih baik. Perkembangan perpustakaan di era saat ini dapat dilihat dari tingkat penerapan teknologi informasi guna menunjang kegiatannya. Dengan adanya perkembangan teknologi informasi membuat manusia berfikir untuk dapat bekerja lebih efektif dan efisien. Salah satunya yaitu membuat sistem konvensional menjadi sistem yang terkomputerisasi.

Seperti pada Perpustakaan Kecamatan Bungah yang masih menggunakan sistem layanan manual. Perpustakaan Kecamatan Bungah merupakan perpustakaan kecamatan yang berada di Jalan Raya Bungah No. 02 Bungah Gresik bersebelahan dengan Kantor Kecamatan Bungah. Selain terdapat ruang baca, Perpustakaan Kecamatan Bungah juga memiliki fasilitas bermain anak. Menurut catatan tahun 2017 terdapat 3097 koleksi buku, 72 inventaris, dan 2366 anggota yang terdaftar di Perpustakaan Kecamatan Bungah.

Berdasarkan hasil wawancara dengan pengelola perpustakaan yaitu Rochimah, diketahui bahwa perpustakaan memiliki beberapa kendala diantaranya kesulitan dalam mencari buku yang diinginkan, seringnya terjadi kehilangan buku dikarenakan data pencatatan transaksi peminjaman tidak dikelola dengan baik, selain itu pendataan koleksi buku, pendataan inventaris perpustakaan, pendataan anggota perpustakaan, dan pelaporan peminjaman serta pengembalian buku yang masih dilakukan secara manual yaitu masih dituliskan pada buku. Antrian panjang kerap kali terjadi saat pengunjung melakukan sirkulasi peminjaman akibat layanan perpustakaan yang masih menerapkan sistem manual membuat pengunjung harus menunggu lama untuk

mendapatkan giliran transaksi. Pendaftaran keanggotaan juga ditemukan kendala karena calon anggota harus menunggu cukup lama untuk mendapatkan kartu anggota perpustakaan, waktu terbit kartu anggota pun terkadang masih belum dapat dipastikan sehingga mengakibatkan anggota yang baru terdaftar tidak dapat melakukan peminjaman buku sebelum mendapatkan kartu anggota perpustakaan. Selain itu di Perpustakaan Kecamatan Bungah terdapat seorang petugas tetap dalam melakukan transaksi sirkulasi dengan cara manual membuat pekerjaan petugas dirasa kurang efisien dalam melayani semua aktivitas pengunjung perpustakaan.

Sehubungan dengan penelitian ini, penulis mengusulkan sebuah sistem informasi yang dapat membantu meningkatkan pelayanan perpustakaan dan mempermudah dalam pengelolaan koleksi datanya. Peneliti merancang, mendesain, serta mengimplementasikan Sistem Informasi Perpustakaan dengan menggunakan *design pattern* Model View Controller (MVC), konsep MVC sendiri merupakan konsep pemisah antara tampilan dengan logika bisnis dan *database*. Arsitektur MVC akan diterapkan dalam rancang bangun Sistem Informasi Perpustakaan dalam mempermudah pengelolaan dan pemeliharaan sistem, karena adanya pemisahan antara *model*, *view*, dan *controller*. Namun juga perlu adanya sebuah acuan dalam pengembangan sistem informasi, dalam hal ini penulis akan menggunakan model *waterfall* untuk mengembangkan sistem informasi perpustakaan yang diusulkan. Alasan untuk menggunakan model *waterfall* adalah karena solusi dari masalah yang diangkat dan kebutuhan pengguna yang sudah terdefinisi dengan jelas.

Dengan dibangunnya sistem informasi perpustakaan berbasis website diharapkan akan mempermudah pelayanan dan pengelolaan data perpustakaan. Sehingga, akan diperoleh efisiensi pekerjaan petugas perpustakaan dalam pengelolaan data perpustakaan, penyajian informasi yang lebih mudah dan interaktif, dan memberikan layanan yang lebih baik kepada pengguna layanan perpustakaan. Oleh karena itu dalam penelitian ini penulis menyusun tugas akhir dengan judul **“Pengembangan Sistem Informasi Perpustakaan Berbasis Web Pada Perpustakaan Kecamatan Bungah”**.

## 1.2 Rumusan Masalah

1. Bagaimana hasil analisis kebutuhan pengguna Sistem Informasi Perpustakaan Kecamatan Bungah?
2. Bagaimana hasil perancangan dan implementasi Sistem Informasi Perpustakaan Kecamatan dengan pola perancangan MVC?
3. Bagaimana hasil dari pengujian Sistem Informasi Perpustakaan Kecamatan Bungah?

## 1.3 Tujuan

Berdasarkan rumusan masalah di atas, tujuan dari penelitian ini antara lain:

1. Menganalisis kebutuhan dari pengguna Sistem Informasi Perpustakaan Kecamatan Bungah.
2. Merancang dan membangun Sistem Informasi Perpustakaan Kecamatan Bungah dengan pola perancangan MVC.
3. Melakukan pengujian terhadap Sistem Informasi Perpustakaan Kecamatan Bungah untuk mendapatkan sistem yang sesuai dengan kebutuhan.

#### **1.4 Manfaat**

Manfaat dari penelitian ini adalah sebagai berikut:

1. Meningkatkan pelayanan perpustakaan kepada pengunjung Perpustakaan Kecamatan Bungah.
2. Mempermudah untuk melakukan pencarian, peminjaman, dan pemeliharaan data di perpustakaan.
3. Bagi penulis sendiri, dapat menerapkan dan mengembangkan ilmu yang telah didapat selama perkuliahan terkait pengembangan sistem informasi.

#### **1.5 Batasan Masalah**

Berdasarkan pokok permasalahan yang telah diuraikan di atas, dapat diidentifikasi batasan-batasan masalah sebagai berikut:

1. Pengelolaan transaksi peminjaman dan pengembalian buku yang masih dilakukan secara manual.
2. Pengelolaan data buku, data anggota, dan data inventaris perpustakaan yang masih dilakukan secara manual.
3. Pencarian koleksi buku yang masih dilakukan secara manual.
4. Sistem informasi akan dibangun dengan bahasa pemrograman PHP dan basis data MySQL.

#### **1.6 Sistematika Pembahasan**

Sistematika pembahasan pada penulisan laporan sebagai berikut:

##### **BAB I PENDAHULUAN**

Berisi tentang latar belakang masalah, rumusan masalah, tujuan dan manfaat penelitian, batasan masalah, dan sistematika pembahasan.

##### **BAB II LANDASAN KEPUSTAKAAN**

Berisi tentang tinjauan pustaka dan dasar-dasar teori yang berfungsi sebagai landasan teori mengenai pengembangan sistem informasi perpustakaan berbasis website.

##### **BAB III METODOLOGI**

Berisi tentang objek penelitian, teknik pengumpulan data, identifikasi dan analisis masalah, perancangan sistem dan kerangka pemecahan masalah.

#### **BAB IV ANALISIS KEBUTUHAN**

Berisi tentang analisis proses bisnis saat ini dan saran proses bisnis sistem yang dibangun, serta analisis kebutuhan sistem yang terdiri dari kebutuhan fungsional dan non-fungsional yang didapat dari hasil wawancara.

#### **BAB V PERANCANGAN DAN IMPLEMENTASI**

Berisi tentang perancangan sistem yang akan dibangun dan hasil dari implementasi sistem.

#### **BAB VI PENGUJIAN**

Berisi tentang hasil dari pengujian sistem informasi yang telah dibangun untuk mendapatkan sistem yang sesuai kebutuhan.

#### **BAB VII PENUTUP**

Berisi tentang kesimpulan dan saran yang dibutuhkan terhadap hasil akhir dari penelitian yang telah dibuat.



## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Dalam penelitian ini akan digunakan beberapa penelitian terdahulu yang juga membahas tentang pengembangan sistem informasi perpustakaan yang nantinya dapat mendukung kelangsungan pengembangan sistem. Kajian pustaka yang diambil yaitu penelitian yang dilakukan oleh Richki Hardi dan Hardianto (2015) pada Perpustakaan STITEK Bontang. Permasalahan yang diangkat dalam penelitian ini adalah pengelolaan koleksi buku pada perpustakaan STITEK yang cukup banyak dan rumit sehingga peneliti mengusulkan sebuah solusi untuk mempermudah pengelolaan dan pencarian buku dengan membuat aplikasi berbasis web.

Dari penelitian yang dilakukan Richki Hardi dan Hardianto (2015) pada Perpustakaan STITEK Bontang dapat ditarik kesimpulan bahwa pertama, dengan dibuatnya sistem informasi berbasis web, mahasiswa dengan mudah melakukan pencarian buku, mengetahui status buku yang tersedia, sehingga tidak perlu menggunakan sistem manual untuk mencari dan mengetahui status buku. Dan yang kedua, sistem informasi perpustakaan STITEK juga dapat mempermudah petugas perpustakaan dalam melakukan pelayanan atau transaksi peminjaman pada perpustakaan STITEK Bontang.

Kajian pustaka yang kedua diambil dari penelitian yang dilakukan oleh Barany Fachri (2017) pada Perpustakaan Kopertis Wilayah I Sumatera Utara. Permasalahan yang diangkat dalam penelitian ini adalah seringnya terjadi kesalahan pada perhitungan jumlah koleksi buku dan jurnal yang ada di perpustakaan Kopertis wilayah I. Hal tersebut dikarenakan tidak tepatnya pengolahan data yang terkait dalam mendapatkan informasi koleksi buku dan jurnal. Selain itu untuk memperoleh informasi yang berkaitan dengan koleksi dibutuhkan waktu yang lama. Oleh karena itu peneliti membangun sistem informasi perpustakaan berbasis web untuk memperbaiki dan membuat pekerjaan menjadi lebih cepat sehingga tidak terjadi hambatan dan kendala dalam pendataan koleksi buku dan jurnal oleh kelalaian sumber daya manusianya pada saat pemeliharaan data koleksi buku yang ada di perpustakaan.

Berdasarkan penelitian yang telah dilakukan Banary Fachri (2017) pada Perpustakaan Kopertis Wilayah I Sumatera Utara dapat ditarik kesimpulan bahwa pertama, dengan adanya sistem informasi perpustakaan berbasis web pemeliharaan data buku-buku dan jurnal-jurnal yang ada di perpustakaan dapat dilakukan dengan teratur. Kedua, sistem informasi perpustakaan menyimpan seluruh data buku dan jurnal pada *database* sehingga proses pencarian data menjadi lebih cepat dan efisien. Dan yang ketiga, sistem informasi perpustakaan berbasis web mengolah data dan informasi dari *database* yang diupdate oleh admin sehingga user atau pengguna umum maupun anggota mendapatkan data yang akurat dan selalu terupdate.

## 2.2 Landasan Teori

### 2.2.1 Perpustakaan

Perpustakaan adalah mencakup suatu ruangan, bagian dari gedung/bangunan atau gedung tersendiri yang berisi buku-buku koleksi, yang diatur dan disusun sedemikian rupa, sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pembaca (Sutarno NS, 2006:11).

Perpustakaan diartikan sebuah ruangan atau gedung yang digunakan untuk menyimpan buku dan terbitan lainnya yang biasanya disimpan menurut tata susunan tertentu yang digunakan pembaca bukan untuk dijual (Sulistyo Basuki, 1991)

Pada pasal 1 Undang-undang Perpustakaan No. 43 tahun 2007 disebutkan bahwa perpustakaan adalah institusi pengelola koleksi karya tulis, karya cetak, dan atau karya rekam secara profesional dengan sistem yang baku guna memenuhi kebutuhan pendidikan, penelitian, pelestarian, informasi, dan rekreasi para pemustaka (Wiji Suwarno, 2013:45).

Dari beberapa pengertian di atas dapat ditarik suatu kesimpulan pengertian perpustakaan secara umum adalah unit kerja yang berupa tempat mengumpulkan, menyimpan dan memelihara koleksi pustaka baik buku-buku ataupun bacaan lainnya yang diatur, diorganisasikan dan diadministrasikan dengan cara tertentu untuk memberi kemudahan dan digunakan secara kontinu oleh pemakainya sebagai informasi.

Menurut Sutarno NS (2006) sebuah perpustakaan dibentuk atau dibangun dengan maksud:

- a. Menjadi tempat mengumpulkan/menghimpun informasi, dalam arti aktif, perpustakaan tersebut mempunyai kegiatan yang terus menerus untuk menghimpun sebanyak mungkin sumber informasi untuk dikoleksi.
- b. Sebagai tempat mengolah atau memproses semua bahan pustaka dengan metode atau sistem tertentu seperti registrasi, klasifikasi, katalogisasi, dan kelengkapan lainnya, baik secara manual maupun menggunakan sarana teknologi informasi, pembuatan perlengkapan lain agar semua koleksi mudah digunakan.
- c. Menjadi tempat menyimpan dan memelihara. Artinya ada kegiatan untuk mengatur, menyusun, menata, memelihara, merawat, agar koleksi rapi, bersih, awet, utuh, lengkap, mudah diakses, tidak mudah rusak, hilang, dan berkurang.
- d. Sebagai salah satu pusat informasi, sumber belajar, penelitian, dan rekreasi, preservasi serta kegiatan ilmiah lainnya. Memberikan layanan kepada pemakai, seperti membaca, meminjam, meneliti, dengan cara cepat, tepat, mudah dan murah.

- e. Membangun tempat informasi yang lengkap dan “*up date*” bagi pengembangan pengetahuan (*knowledge*), keterampilan (*skill*), dan perilaku/sikap (*attitude*).
- f. Merupakan agen perubahan dan agen kebudayaan dari masa lalu, sekarang dan masa depan. Halaman konsep yang lebih hakiki eksistensi dan kemajuan perpustakaan menjadi kebanggaan, dan symbol peradaban kehidupan umat manusia.

### 2.2.2 Teknologi Informasi Perpustakaan

Menurut Supriyanto dan Muhsin (2008) perpustakaan adalah institusi yang menyediakan koleksi bahan pustaka tertulis, tercetak dan terekam sebagai pusat sumber informasi yang diatur menurut sistem aturan dan didayagunakan untuk keperluan pendidikan, penelitian serta rekreasi intelektual bagi masyarakat. Perpustakaan berperan melakukan layanan informasi literal kepada masyarakat. Karena tujuannya memberi layanan informasi literal kepada masyarakat maka tugas pokoknya adalah:

1. Menghimpun bahan pustaka yang meliputi buku dan nonbuku sebagai sumber informasi;
2. Mengolah dan merawat pustaka;
3. Memberikan layanan bahan pustaka.

Perpustakaan sebagai sentral pengetahuan tidak lepas dari fungsi untuk menyediakan sarana informasi dan ilmu pengetahuan. Kita dapat melihat ruangan perpustakaan begitu banyak koleksi-koleksi berbagai bidang ilmu yang tertata di rak. Bahkan banyak dari koleksi ini yang berwarna kuning karena sudah dimakan usia ataupun sebagian habis dimakan rayap.

Kebutuhan akan teknologi informasi sangat berhubungan dengan peran perpustakaan sebagai kekuatan dalam pelestarian dan penyebaran informasi ilmu pengetahuan, tempat rujukan bagi pencari ilmu, dan pengembangan karya-karya ilmiah (Supriyanto dan Muhsin, 2008). Dengan digunakannya teknologi informasi pergeseran kebudayaan berkembang seiring dengan meningkatnya minat untuk menulis, mencetak, mendidik dan kebutuhan manusia akan informasi. Tugas perpustakaan dalam menyebarkan informasi dengan jalan mengidentifikasi, mengumpulkan, mengelola dan menyediakannya untuk pendidikan maupun masyarakat luas.

Penerapan teknologi informasi di perpustakaan saat ini sudah menjadi ukuran untuk mengetahui tingkat kemajuan dari perpustakaan tersebut, bukan lagi pada besarnya gedung yang dipakai, banyak rak buku, ataupun berjubelnya pengguna. Semakin canggih dan otomatisasi kinerja perpustakaan maka semakin maju perpustakaan itu. Alasannya sederhana dengan teknologi informasi maka akan lebih banyak yang dikerjakan dan dilayani.

Penerapan teknologi informasi (TI) di perpustakaan merupakan wujud dari suatu perubahan layanan. Perubahan ini semakin besar ini sekan mejadi

“tantangan” bagi perpustakaan untuk berbenah dan selalu inovatif untuk dapat memberikan layanan yang terbaik melalui fasilitas TI. Hari demi hari perpustakaan terus berbenah agar fungsinya sebagai penyedia gerbang ilmu kemajuan dunia menjadi semakin berperan. Yang paling utama dari penerapan teknologi informasi adalah fungsi kemudahan yang diberikan untuk pustakawan dan pengguna perpustakaan.

### 2.2.3 Sistem Informasi

Sistem informasi bisa diartikan sebagai kesatuan elemen informasi, termasuk cara merancang, mengaktifkan, menangani, memelihara, dan memanfaatkan informasi. Sistem Informasi telah dipakai sebagai nama dari suatu cabang ilmu yang menangani informasi yang diperlukan dalam melaksanakan kegiatan tertentu. Di antara beragam kegiatan, kegiatan pengelolaan suatu usaha (*business management*) merupakan satu di antara kegiatan lain yang menonjolkan dan mengembangkan sistem informasi. Hal ini bisa dimaklumi, karena bagi organisasi berskala besar selalu diikuti dengan penanganan informasi yang kompleks bervolume besar (Taufiq Rochim, 2002).

Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Definisi sistem informasi juga bisa didefinisikan kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (*output*), guna mencapai sasaran-sasaran perusahaan. Sistem informasi didefinisikan sebagai sekumpulan komponen yang terdiri dari manusia atau orang, prosedur kerja, data, informasi dan teknologi informasi yang berguna untuk pengambilan keputusan dalam organisasi (Noviandi, Destiani, Partono, 2012).

### 2.2.4 Sistem Informasi Perpustakaan

Menurut Riki Nuryadin (2014) sistem informasi perpustakaan dapat didefinisikan sebagai sebuah sistem terintegrasi, sistem manusia mesin, untuk menyediakan informasi yang mendukung operasi, manajemen, dan fungsi pengambilan keputusan dalam sebuah perpustakaan. Sistem ini memanfaatkan perangkat keras dan perangkat lunak komputer, prosedur manual, model manajemen, dan pengambilan keputusan basis data.

Dalam suatu organisasi atau perusahaan bisnis, harus ada hubungan timbal balik dan keterkaitan yang erat antara setiap fungsi pelayanan dengan pengguna atau klien. Sistem informasi ini bertugas mengumpulkan, menyimpan dan mengolah data, hingga akhirnya menyajikan informasi kepada setiap orang yang membutuhkannya, dan berkaitan erat dengan fungsi organisasi atau perusahaan yang bersangkutan.

Adapun manfaat dari penerapan sistem informasi di perpustakaan adalah sebagai berikut:

1. Dapat mempermudah dan mempercepat layanan
2. Meningkatkan mutu layanan
3. Secara tidak langsung mengarsipkan atau memasukkan data teknis ketersediaan bahan koleksi maupun pengguna
4. Mempermudah pencarian arsip maupun penelusuran
5. Merupakan sebuah alternative pelayanan yang menarik dan interaktif antara petugas dan pengguna.

### 2.2.5 Bussines Process Model Notation

*Bussines Process Model Notation* (BPMN) adalah representasi grafis untuk menentukan proses bisnis dalam suatu pemodelan proses bisnis. Tujuan utama dari BPMN adalah menyediakan suatu notasi standar yang mudah dipahami oleh semua pemangku kepentingan bisnis. BPMN menyediakan kemampuan memahami prosedur internal bisnis dalam notasi grafis. BPMN mengikuti tradisi flowcharting notasi untuk dibaca dan fleksibilitas. Ada 4 kategori dasar dari elemen-elemen BPD adalah sebagai berikut:

#### a. *Flow Objects*

- *Events*: hal-hal yang terjadi di antara proses bisnis Event ini mempengaruhi aliran proses dan biasanya memiliki penyebab (pemicu) atau dampak (hasil). *Event* disimbolkan dengan lingkaran kecil dengan garis tipis (*start event*), lingkaran kecil dengan garis ganda (*intermediate event*), lingkaran kecil dengan garis tebal (*end event*).



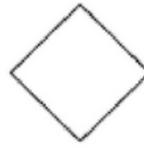
**Gambar 2.1 Events**

- *Activities*: Aktivitas artinya perusahaan melakukan Proses. Sebuah Kegiatan dapat berupa Sub-Proses dan Task, yang disimbolkan dengan bulat persegi panjang.



**Gambar 2.2 Activities**

- *Gateways*: *Gateway* digunakan untuk mengontrol perbedaan dan konvergensi dari urutan Arus dalam Proses. Dengan demikian, akan menentukan percabangan, forking, penggabungan, dan bergabung dengan jalur.



**Gambar 2.3 Gateways**

b. *Connecting Objects*

*Flow object* dihubungkan satu sama lain dalam suatu diagram untuk menciptakan struktur dasar proses bisnis. Ada tiga *connecting object* yang disediakan fungsi ini, yaitu:

- *Sequence flow*: Sebuah Arus urutan digunakan untuk menunjukkan urutan Kegiatan yang akan dilakukan dalam proses, disimbolkan dengan garis dengan panah solid.



**Gambar 2.4 Sequence flows**

- *Message flow*: Dipresentasikan oleh garis terputus-putus dengan kepala panah terbuka dan digunakan untuk menunjukkan aliran Pesan antara dua partisipan proses (entitas bisnis atau peran bisnis).



**Gambar 2.5 Message flow**

- *Association*: asosiasi direpresentasikan oleh garis titik-titik dengan kepala panah garis dan digunakan untuk asosiasi data teks dan artefak lainnya.

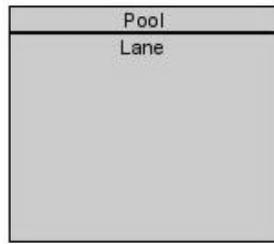


**Gambar 2.6 Association**

c. *Swimlanes*

*Swimlanes* adalah pengelompokan dari beberapa model elemen. *Swimlanes* digunakan untuk memisahkan dan mengatur kegiatan oleh peserta sehingga kita secara intuitif dapat memahami siapa yang bertanggung jawab untuk setiap event. *Swimlanes* ada 2 jenis yaitu:

- *Pools*: semua *Bisnis Flow Diagram* mengandung setidaknya satu *Pool*.
- *Lanes*: *Pool* yang dibagi lagi menjadi sub *Swimlanes*.



**Gambar 2.7 Pool dan Lane**

d. *Artifacts*

*Artifacts* adalah elemen yang digunakan untuk memberikan informasi tambahan dari sebuah proses. Bentuk dan penggunaan *artifacts* itu bermacam-macam dan bisa lebih luas tergantung dari standar pengertian BPMN yang digunakan. Saat ini terdapat 3 setingan pengaturan *artifact*, yaitu:

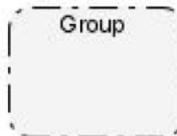
- *Data Object*: *data object* digunakan untuk menjelaskan data apa yang dibutuhkan dalam proses.



Data

**Gambar 2.8 Data object**

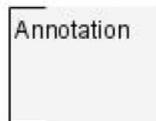
- *Group*: *group* digunakan untuk melakukan grouping aktivitas.



Group

**Gambar 2.9 Group**

- *Annotation*: *annotation* digunakan untuk menjelaskan model atau diagram.



Annotation

**Gambar 2.10 Annotation**

### 2.2.6 Physical Data Model (PDM)

Model Relasional atau *Physical Data Model* (PDM) adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data. Setiap tabel mempunyai sejumlah kolom di mana setiap kolom memiliki nama yang unik beserta tipe datanya. PDM merupakan konsep yang menerangkan detail dari bagaimana data disimpan di dalam basis data. Sasarannya adalah menciptakan perancangan untuk penyimpanan data yang



menyediakan kinerja yang baik dan memastikan integritas, keamanan, serta kemampuan untuk dipulihkan.

Berikut adalah simbol-simbol yang ada di PDM:

Simbol	Deskripsi
Tabel nama_tabel	Tabel yang menyimpan data dalam basis data
Relasi Id_tb11 = id_fk_tb12	Relasi antar tabel yang terdiri dari persamaan antara <i>primary key</i> (kunci primer) tabel yang diacu dengan kunci yang menjadi referensi acuan di tabel lain.

### 2.2.7 Software Development Life Cycle (SDLC)

Menurut Marimin, Tanjung, dan Prabowo (2006) *System Development Life Cycle* (SDLC) merupakan sebuah metodologi dalam pembangunan atau pengembangan sistem. *System Development Life Cycle* memberikan kerangka kerja yang konsisten terhadap tujuan yang diinginkan dalam pembangunan dan pengembangan sistem. Metodologi SDLC dimulai dengan ide-ide yang berasal dari pengguna, melalui studi kelayakan, analisis dan desain sistem, pemrograman, *pilot testing*, implementasi, dan analisis setelah diimplementasikan (evaluasi). Dokumentasi yang dibuat selama melakukan pembangunan atau pengembangan sistem digunakan untuk perubahan-perubahan di masa yang akan datang, misalnya melanjutkan pengembangan sistem, modifikasi atau penghilangan (*deletion*).

Beberapa ahli sistem informasi menyatakan bahwa SDLC merupakan pengembangan sistem secara tradisional dan memiliki beberapa tahapan. *Department of Justice USA* (2003) menyatakan bahwa SDLC memiliki 10 tahapan; O'Brien menyatakan bahwa SDLC memiliki 5 tahapan; dan menurut Turban, McLean, dan Weatherbe (2001) SDLC memiliki 8 tahapan. Pada intinya langkah-langkah dalam metodologi SDLC adalah:

1. Mengevaluasi sistem yang ada

Dengan evaluasi, akan diketahui kekurangan-kekurangan (difisiensi) yang ada di dalam sistem. Hal ini dapat dilakukan dengan cara melakukan *interview/wawancara* dengan pengguna yang menggunakan sistem tersebut dan melakukan konsultasi dengan orang-orang yang berkompeten di bidang tersebut.

2. Mendefinisikan kebutuhan sistem baru yang akan dibangun

Kekurangan-kekurangan yang ada pada sistem lama harus dijelaskan secara spesifik sehingga menjadi perhatian untuk perbaikan sistem yang akan

dibangun. Selain menganalisis dan mendefinisikan masalah, sistem informasi yang akan dibangun atau dikembangkan serta proses organisasinya.

3. Mendesain sistem yang diusulkan

Rencana-rencana yang akan dilakukan didasarkan/difokuskan pada konstruksi fisik, perangkat keras, perangkat lunak, sistem operasi, pemrograman, komunikasi, dan masalah keamanan sistem informasi. Merancang *input*, *output*, program, struktur file, prosedur, perangkat lunak dan perangkat keras yang digunakan untuk mendukung sistem informasi.

4. Pengembangan sistem yang baru

Komponen-komponen dan program harus tersedia dan *diinstall*. Membangun perangkat lunak yang diperlakukan untuk mendukung sistem dan melakukan pengujian secara akurat. Melakukan instalasi dan pengujian terhadap perangkat keras dan mengoperasikan perangkat lunak. Pengguna yang akan menggunakan sistem harus dilatih dan semua aspek yang terdapat dalam sistem informasi tersebut harus dicoba.

5. Penggunaan sistem yang baru

Hal ini dapat dilakukan dengan berbagai cara. Sistem baru dapat diimplementasikan untuk menggantikan sistem lama. Penerapan sistem baru sebagai pengganti sistem lama yang ada dapat dilakukan secara serentak ataupun bertahap. Hal ini tergantung dari kesiapan organisasi, teknis, operasional, dan biaya yang dimiliki oleh organisasi tersebut. Pada tahap ini, organisasi perlu melakukan pelatihan dan panduan seperlunya.

6. Evaluasi yang harus dilakukan terhadap sistem informasi baru yang telah/sedang berjalan.

Hal yang dilakukan adalah mengevaluasi sejauh mana sistem telah dibangun dan seberapa bagus sistem telah dioperasikan. Pemeliharaan sistem dilakukan dengan sungguh-sungguh dan teliti secara terus-menerus, sehingga sistem informasi yang dibangun dapat bermanfaat bagi organisasi tersebut.

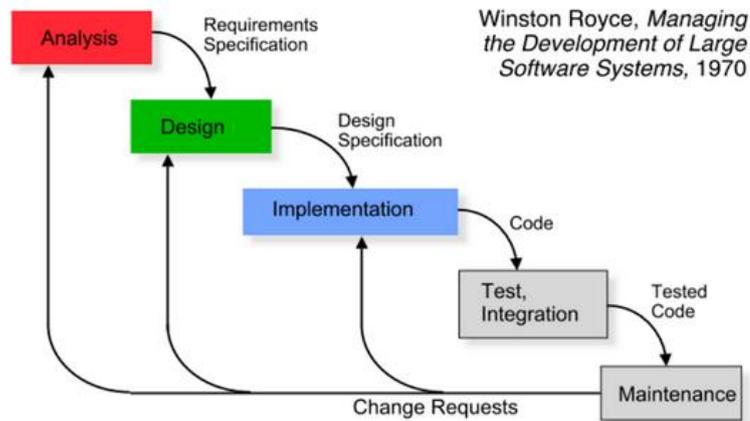
### 2.2.8 Model Waterfall

Model Waterfall sering dinamakan siklus hidup klasik, dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modelling*), konstruksi (*construction*), serta penyerahan sistem ke para pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Pressman, 2012).

Menurut Oka (2017) Waterfall merupakan model klasik yang sederhana dengan aliran sistem yang linier, keluaran dari tahap sebelumnya merupakan masukan untuk tahap berikutnya. Royce (1970) seperti dikutip oleh professor Hubmann (2009) dalam publikasi elektroniknya memperkenalkan model

waterfallnya Royce. Wiston Royce dari *managing the Development of large System* memperkenalkan model waterfall seperti pada gambar 2.11.

### The “Waterfall” Model - Textbook Version



Gambar 2.11 The Waterfall Model

Sumber diadaptasi dari Royce (1970) dalam Hubmann, 2009.

Menurut Sommerville (2003) Tahap-tahap utama dari model ini memetakan kegiatan-kegiatan pengembangan dasar yaitu:

1. Analisis dan definisi persyaratan  
 Pelayanan, batasan, dan tujuan sistem ditentukan melalui konsultasi dengan user sistem. Persyaratan ini kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.
2. Perancangan sistem dan perangkat lunak  
 Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan-hubungannya.
3. Implementasi dan pengujian unit  
 Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian unit melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.
4. Integrasi dan pengujian sistem  
 Unit program atau program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sistem telah dipenuhi. Setelah pengujian sistem, perangkat lunak dikirim kepada pelanggan.
5. Operasi dan Pemeliharaan

Biasanya (walaupun tidak seharusnya), ini merupakan fase siklus hidup yang paling lama. Sistem diinstall dan dipakai. Pemeliharaan mencakup koreksi dan berbagai error yang tidak ditemukan pada tahap-tahap terdahulu, perbaikan atas implementasi unit sistem dan pengembangan pelayanan sistem, sementara persyaratan-persyaratan baru ditambahkan.

#### **2.2.8.1 Kebutuhan Perangkat Lunak**

Kebutuhan perangkat lunak adalah kondisi, kriteria, syarat, atau kemampuan yang harus dimiliki oleh perangkat lunak untuk memenuhi apa yang disyaratkan atau diinginkan pengguna, dikelompokkan menjadi dua, yaitu:

##### **1. Kebutuhan Fungsional**

Kebutuhan fungsional merupakan sebuah pernyataan terhadap layanan yang harus disediakan oleh sistem, bagaimana sistem seharusnya bereaksi terhadap input tertentu, dan bagaimana sistem harus berperilaku dalam keadaan tertentu. Kebutuhan fungsional mencakup apa yang seharusnya dilakukan sistem dalam kebutuhan yang spesifik yang mencerminkan sistem yang ada pada organisasi, dalam hal ini kebutuhan fungsional sistem harus memenuhi ekspektasi dari pengguna. Kebutuhan fungsional biasanya dideskripsikan sebagai langkah abstrak yang dapat dimengerti oleh pengguna sistem. Selain itu, kebutuhan fungsional juga mendeskripsikan fungsi sistem, input, output, pengecualian, dan sebagainya (Sommerville, 2011).

##### **2. Kebutuhan Non-Fungsional**

Kebutuhan non-fungsional mendeskripsikan sekumpulan batasan, karakteristik dan property pada sistem, baik dalam lingkungan pengembangan maupun operasional, atau atribut kualitas yang harus dipenuhi oleh sistem. Kebutuhan non-fungsional sering diaplikasikan pada sistem secara keseluruhan dibandingkan sistem secara individu. Kebutuhan non-fungsional juga dapat dikatakan sebuah kebutuhan yang tidak berhubungan secara langsung terhadap layanan atau fungsi dari sistem ke penggunaannya. Kebutuhan ini meliputi kinerja, ketersediaan, ataupun keamanan sistem yang biasanya merupakan karakteristik sistem secara keseluruhan (Sommerville, 2011).

#### **2.2.8.2 Unified Modeling Language (UML)**

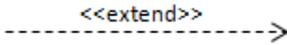
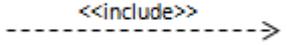
Menurut Nugroho (2010) UML (*Unified Modeling Language*) adalah 'bahasa' pemodelan untuk sistem atau perangkat lunak yang berparadigma 'berorientasi objek'. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Adapun tujuan pemodelan (dalam kerangka pengembangan sistem/perangkat lunak aplikasi) adalah sebagai sarana analisis, pemahaman, visualisasi, dan komunikasi antar anggota tim pengembang (saat seorang analisis/perancang perangkat lunak bekerja dalam tim yang beranggotakan beberapa/banyak anggota), serta sebagai sarana dokumentasi (yang bermanfaat untuk melakukan pengujian terhadap perangkat lunak yang telah selesai dikembangkan) (Nugroho, 2010).

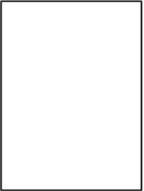
Sedangkan menurut Pressman (2010) UML (*Unified Modeling Language*) adalah “bahasa standar untuk penulisan cetak biru perangkat lunak. UML dapat digunakan untuk memvisualisasikan, menentukan, mengonstruksi, dan mendokumentasikan artefak-artefak suatu sistem *software-intensive*”. Dengan kata lain, sama seperti arsitek bangunan membuat cetak biru untuk digunakan oleh perusahaan konstruksi, arsitek perangkat lunak membuat diagram UML untuk membantu pengembang perangkat lunak membangun perangkat lunak.

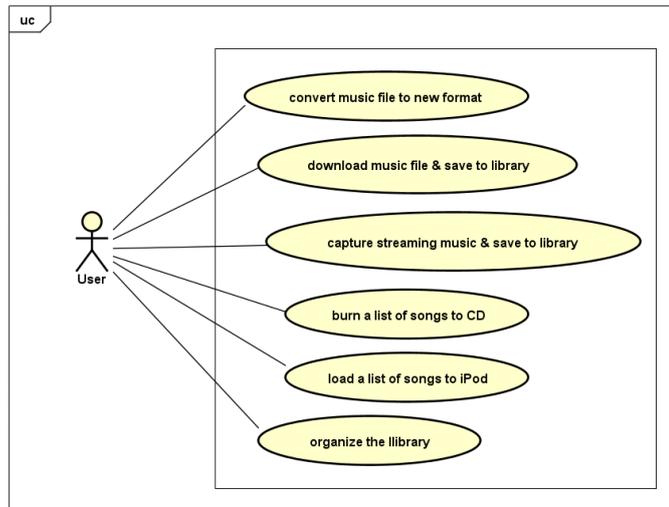
### 1. Use Case Diagram

*Use case* merupakan gambaran bagaimana interaksi antara pengguna atau *user* dengan sistem dengan mendefinisikan langkah-langkah yang dibutuhkan untuk mencapai suatu tujuan tertentu. Diagram *use case* merupakan ikhtisar dari semua *use case* dan bagaimana *use case-use case* tersebut berhubungan. Diagram *use case* memberikan suatu gambaran besar tentang fungsionalitas sistem (Pressman, 2010). Pada diagram *use case*, *use case-use case* ditampilkan sebagai oval. Aktor-aktor dihubungkan dengan garis-garis ke *use case* yang mereka bawa. Berikut merupakan beberapa notasi dalam *use case* antara lain:

**Tabel 2.1 Notasi Pada Use Case Diagram**

No	Notasi	Nama	Deskripsi
1.		<i>Actor</i>	Merupakan representasi dari pengguna atau <i>user</i> yang berinteraksi dengan sistem.
2.		<i>Use case</i>	Merupakan representasi dari kebutuhan fungsional berdasarkan perspektif pengguna terhadap sistem.
3.		<i>Association</i>	Merupakan representasi dari hubungan antara <i>actor</i> dan <i>use case</i> .
4.		<i>Generalization</i>	Merupakan representasi dari sebuah elemen yang menjadi spesialisasi dari elemen yang lain.
5.		<i>Extend</i>	Merupakan representasi dari situasi optional yang dijalankan pengguna.
6.		<i>Include</i>	Merupakan representasi dari kelakuan yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi, dimana pada kondisi ini

			sebuah <i>use case</i> adalah bagian dari <i>use case</i> lainnya.
7.		<i>System Boundary</i>	Merepresentasikan batasan antara internal dan eksternal sistem.



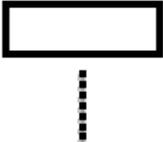
**Gambar 2.12 Use Case Diagram**

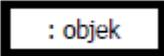
Sumber: Pressman (2010)

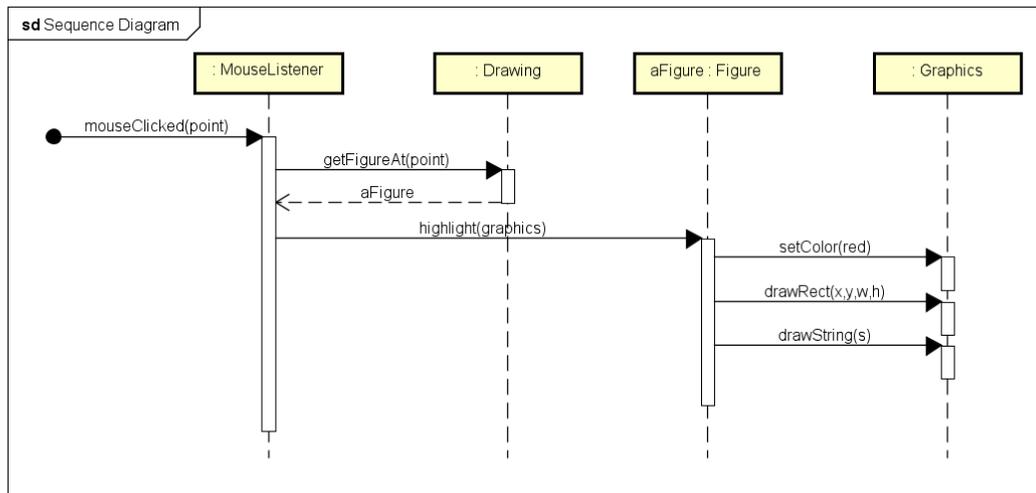
## 2. Sequence Diagram

*Sequence diagram* adalah sebuah diagram yang digunakan untuk menunjukkan komunikasi yang dinamis antara objek-objek selama eksekusi suatu pekerjaan berlangsung (Pressman, 2010). *Sequence diagram* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Berikut merupakan beberapa notasi pada *sequence diagram* antara lain:

**Tabel 2.2 Notasi Sequence Diagram**

No	Notasi	Nama	Deskripsi
1.		<i>Actor</i>	Merupakan representasi dari pengguna atau <i>user</i> yang berinteraksi dengan sistem.
2.		<i>Lifeline</i>	Merupakan representasi dari antarmuka yang saling berinteraksi.

3.		<i>Call Message</i>	Merupakan representasi dari alur <i>message</i> dari <i>lifeline</i> pengirim ke <i>lifeline</i> penerima.
4.		<i>Return Message</i>	Merupakan representasi dari alur pengambilan <i>message</i> dan akan mengembalikan nilai dari objek yang dipanggil.
5.		<i>Activation Bar</i>	Merupakan representasi dari hubungan antara <i>object</i> dan <i>message</i> yang menunjukkan bahwa objek sedang dalam keadaan aktif dan sedang berinteraksi.
6.		<i>Object</i>	Merupakan representasi dari apa saja yang berinteraksi pada satu kejadian. Digambarkan sebagai sebuah kelas dengan nama obyek yang diawali dengan tanda titik koma.



Gambar 2.13 Sequence Diagram

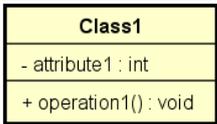
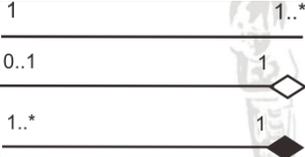
Sumber: Pressman (2010)

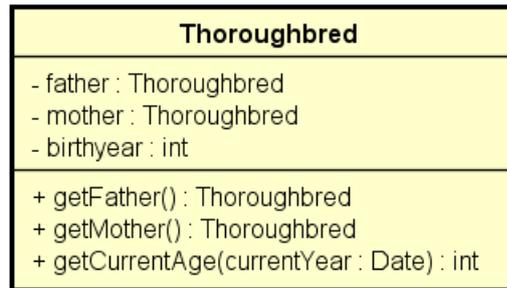
### 3. Class Diagram

*Class Diagram* adalah sebuah diagram yang menyediakan pandangan statis atau struktural dari sebuah sistem atau perangkat lunak dimana hal tersebut

tidak menunjukkan sifat dinamis dan komunikasi antar objek dari kelas pada diagram (Pressman, 2010). Berikut merupakan beberapa notasi pada *class diagram* antara lain:

**Tabel 2.3 Notasi Class Diagram**

No.	Notasi	Nama Notasi	Deskripsi Notasi
1.		<i>Class</i>	Merupakan representasi dari seperangkat objek yang terdiri dari atribut, operasi, dan relasi yang sama.
2.		<i>Association</i>	Merupakan representasi dari hubungan structural antar satu kelas dengan kelas yang lain yang saling berelasi.
3.		<i>Aggregation</i>	Merupakan representasi dari hubungan yang menspesifikasikan antara sebuah kumpulan ( <i>the whole</i> ) dengan sebuah bagian ( <i>the part</i> ).
4.		<i>Composition</i>	Merupakan representasi dari sebuah komposisi.
5.		<i>Multiplicity</i>	Merupakan representasi dari jumlah objek yang terlibat dalam hubungan antar kelas.
6.		<i>Generalization</i>	Merupakan representasi dari relasi spesialisasi / generalisasi dimana satu kelas dapat lebih spesifik dibandingkan dengan kelas lainnya.



**Gambar 2.14 Class Diagram**

Sumber: Pressman (2010)

### 2.2.8.3 Pengujian Perangkat Lunak

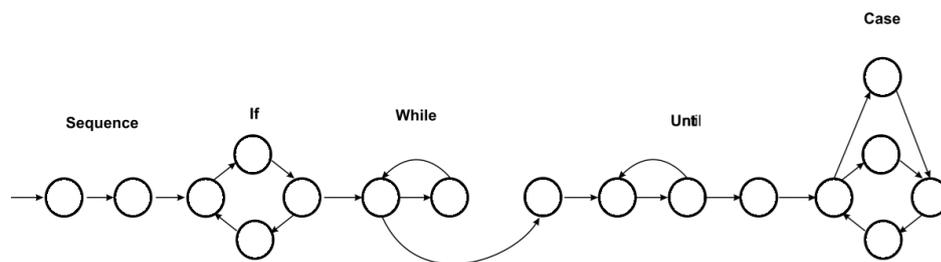
Pengujian perangkat merupakan suatu proses investigasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari suatu produk atau layanan yang sedang diuji, atau lebih spesifiknya *software testing* adalah proses mengeksekusi suatu program untuk menemukan bug (kesalahan atau cacat lainnya). Pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara obyektif dan independen, yang bermanfaat untuk memahami tingkat resiko pada implementasinya. Perakayasa menciptakan deret *test case* yang ditujukan untuk membongkar perangkat lunak. Sehingga, pengujian ini dapat dianggap sebagai hal destruktif daripada konstruktif (Pressman, 2010).

#### 1. Unit Testing

*Unit testing* atau pengujian komponen program adalah pengujian yang dilakukan pada bagian *basic* dari kode program atau unit terkecil dari sistem yang berupa kelas atau fungsi. *Basis path testing* merupakan salah satu metode dari *unit testing*. Metode ini memungkinkan penguji dapat mengukur kompleksitas logis dari desain *procedural* dan menggunakannya sebagai pedoman untuk menetapkan himpunan basis dari semua jalur eksekusi.

Notasi yang digunakan untuk menggambarkan jalur eksekusi adalah notasi diagram alir atau *flow graph* yang menggunakan notasi lingkaran dan anak panah. Notasi ini menggambarkan aliran control logika yang digunakan dalam suatu bahasa pemrograman. *Basis path* menggunakan notasi *graph* untuk menggambarkan aliran kontrolnya. Gambar 2.15 merupakan notasi *graph* untuk menggambarkan skema dasar pemrograman.

The structured constructs in flow graph form:



**Gambar 2.15 Notasi Flow Graph**

Sumber: Pressman (2010)

Selanjutnya melakukan perhitungan *cyclomatic complexity* untuk menunjukkan kompleksitas *logic* suatu program. *Cyclomatic complexity* dapat diperoleh dengan menghitung daerah yang dapat dibentuk oleh *graph*. *Cyclomatic complexity* atau  $V(G)$  dapat dihitung dengan rumus  $V(G) = E - N + 2$  dimana  $E$  adalah jumlah edge pada *flow graph* dan  $N$  adalah jumlah node pada *flow graph*. *Cyclomatic complexity* juga dapat dihitung dengan rumus  $V(G) = P + 1$  dimana  $P$  adalah jumlah predikat node pada *flow graph*. Nilai dari *cyclomatic complexity* mendefinisikan *independent path*. *Independent path* atau jalur independen adalah jalur yang melintas atau melalui program dimana sekurang-kurangnya dieksekusi satu kali (Pressman, 2010).

Dan tahap terakhir *basis path testing* adalah menurunkan *test case*. *Test case* merupakan data *input* yang digunakan untuk memeriksa alur logika atau kondisi pada masing-masing *independent path* yang didapatkan dari hasil perhitungan *cyclomatic complexity* pada tahap sebelumnya.

## 2. Integration Testing

*Integration testing* merupakan teknik untuk membangun arsitektur perangkat lunak yang mana pada saat yang bersamaan juga melakukan pengujian sistem untuk menemukan kesalahan terkait antarmuka sistem. Pengujian ini lebih kompleks daripada *unit testing* dimana tujuannya yaitu untuk menguji suatu unit dan membangun program sesuai desain yang telah dibuat (Pressman, 2010).

## 3. System Testing

*System testing* mencakup pengujian aplikasi yang telah selesai dikembangkan. Karena itu, aplikasi harus terlihat dan berfungsi sebagaimana mestinya terhadap *end-user* atau pengguna akhir. Pengujian ini berfokus untuk menemukan kesalahan terhadap *output* sistem dan menunjukkan bahwa sistem yang telah dibangun sesuai dengan kebutuhan fungsional dan non-fungsional sistem (Sommerville, 2011).

## 4. Acceptance Testing

*Acceptance testing* merupakan sebuah pengujian yang dilakukan sebelum sistem digunakan secara operasional. Pada tahap ini, sistem diuji oleh pengguna sistem untuk memastikan bahwa sistem telah berjalan dengan baik dan sesuai

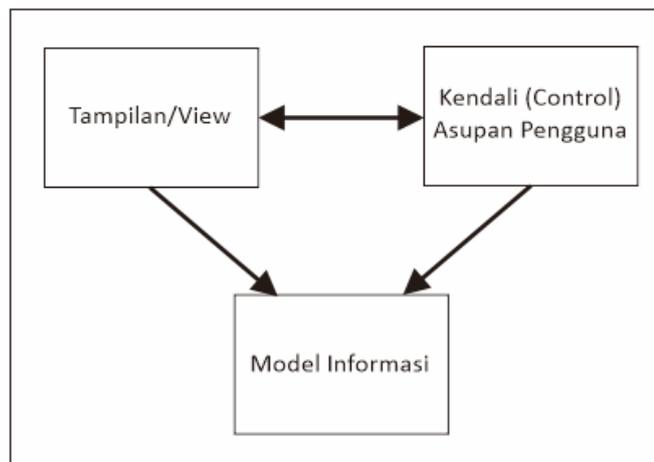
dengan fase elisitasi kebutuhan sistem (Sommerville, 2011). Proses ini merupakan salah satu tahap final sebelum pengguna menyetujui dan menerima penerapan sistem aplikasi yang baru.

### 2.2.9 Design Pattern

*Design Pattern* adalah sebuah istilah dalam Rekayasa Perangkat Lunak (Software Engineering) yang mengacu kepada solusi umum yang dapat digunakan secara berulang kali untuk menyelesaikan masalah-masalah umum yang ditemukan dalam desain perangkat lunak. Sebuah *design pattern* tidak berbentuk solusi akhir yang dapat langsung diterjemahkan menjadi kode program.

*Design Pattern* merupakan penjelasan atau template yang menunjukkan bagaimana cara menyelesaikan sebuah masalah yang kemudian dapat digunakan di berbagai situasi yang berbeda-beda. *Design Pattern* untuk *object-oriented* biasanya menunjukkan relasi dan interaksi antar kelas dan objek, tanpa menjelaskan kelas dan objek akhir yang terlibat dalam sebuah aplikasi. Algoritma biasanya tidak disebut sebagai *design pattern*, karena algoritma akan menjadi solusi masalah komputasi dan bukan masalah desain.

#### 2.2.7.1 MVC (Model-View-Controller)



**Gambar 2.16 Model MVC**

Sumber: Nugroho (2009)

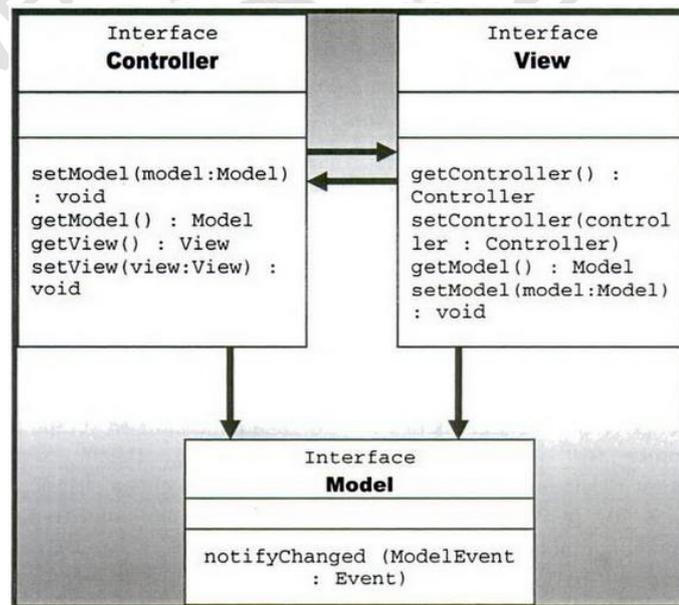
Menurut Nugroho (2009) MVC adalah sebuah metode pemisah tampilan pengguna (*interface*) dari kendali asupan pengguna (*controller*) dan model informasi yang mendasarinya (*model*), seperti yang tampak pada Gambar 2.16 Untuk hal tersebut, ada beberapa alasan pokok mengapa model MVC menjadi sangat bermanfaat, yaitu:

- Penggunaan ulang komponen-komponen antarmuka pengguna (*user interface reusable component*).
- Kemampuan untuk mengembangkan aplikasi dengan antarmuka pengguna secara terpisah.

- Kemampuan untuk melakukan pewarisan (*inheritance*) dari berbagai bagian yang berbeda pada suatu hierarki kelas.
- Kemampuan untuk mendefinisikan kelas-kelas pengaturan tampilan (*control style*) yang menyediakan fitur-fitur umum ditampilkan oleh aplikasi yang kita kembangkan.

Hal-hal yang disebutkan mempunyai arti bahwa berbagai antarmuka yang berbeda dapat digunakan dalam aplikasi yang sama, tanpa aplikasi yang bersangkutan perlu mengetahuinya. Hal-hal tersebut juga berarti bahwa setiap bagian dari sistem/perangkat lunak aplikasi dapat diubah tanpa menimbulkan imbas apapun pada bagian aplikasi lainnya.

Untuk menyediakan *framework* MVC, kita dapat mengadopsi sejumlah teknik. Meskipun demikian, pendekatan yang paling umum terlihat pada Gambar 2.17 memperlihatkan tiga antarmuka, yaitu *Controller*, *View*, dan *Model*, dan mereka bertindak sebagai penanda (*marker*) untuk konsep-konsep inti atau entitas-entitas dalam *framework* MVC. Perhatikan bahwa antarmuka *Controller* dan *View* memiliki metode *accessor* (setter dan getter) yang bermanfaat untuk mendapatkan model dan view.



Gambar 2.17 Antarmuka yang mendefinisikan pola MVC  
 Sumber: Nugroho (2009)

### 2.2.10 HTML

HTML (*Hyper Text Markup Language*) merupakan sekumpulan simbol-simbol atau tag-tag yang dituliskan dalam sebuah *file* untuk ditampilkan pada halaman *web browser* (Anhar, 2010: 40). Tag HTML akan terhubung dengan *browser* untuk menampilkan halaman web dengan lengkap kepada pengguna (Astarnal, 2006). HTML memiliki sintaks dan struktur tersendiri dalam menuliskan *script* atau kode yang biasa disebut Tag HTML. Aturan penulisan pada HTML ini diawali dengan lambang <tag dan diakhiri dengan lambang </tag. Setiap tag ini memiliki



fungsi-fungsi tertentu yang dapat membuat tampilan *website* menjadi lebih menarik. Terdapat beberapa macam tag HTML beserta fungsinya (Anhar, 2010).

**Tabel 2.4 Tag HTML**

Lambang	Keterangan
<!DOCTYPE>	Mendefinisikan informasi tipe dokumen
<!-- ... -->	Untuk menambahkan sebuah komentar, dimana kalimat yang terletak pada kolom komentar tidak akan ditampilkan pada <i>browser</i> .
<a href>	Untuk membuat sebuah <i>link</i> agar dapat berpindah dari satu halaman ke halaman lain.
<applet>	Untuk memasukkan <i>file</i> Java ke dalam sebuah dokumen HTML.
<b>	Untuk membuat teks tebal
<base />	Mendefinisikan URL dasar atau target untuk semua URL relatif dalam dokumen HTML.
<basefont />	Untuk membuat atribut teks <i>default</i> , seperti warna, ukuran, atau jenis huruf.
<blockquote>	Mendefinisikan sebuah kutipan panjang sehingga teks akan tampil lebih menjorok ke dalam pada <i>browser</i> .
<body>	Mendefinisikan <i>body</i> atau isi dokumen HTML, yang digunakan untuk menentukan isi dari suatu dokumen yang ditampilkan pada <i>web browser</i> .
 	Untuk memberikan baris baru.
<button>	Mendefinisikan tombol yang dapat di klik.
<caption>	Untuk menambahkan <i>caption</i> pada tabel.
<center>	Untuk membuat perataan tengah pada teks atau gambar.
<div>	Mendefinisikan sebuah <i>section</i> pada dokumen HTML.
<dl>	Mendefinisikan daftar definisi.
<dt>	Mendefinisikan <i>item</i> dalam daftar definisi.
<em>	Untuk membuat teks miring dimana fungsi <i> juga dapat digunakan untuk membuat teks miring.
<embed>	Untuk menambahkan <i>file</i> video atau <i>file</i> audio.
<font>	Mendefinisikan jenis, ukuran, dan warna huruf untuk teks.
<form>	Mendefinisikan sebuah formulir untuk input formulir pada HTML.

<h1> to <h6>	Untuk menunjukkan awal dari suatu <i>header</i> atau judul dari dokumen HTML.
<head>	Untuk memberikan informasi mengenai dokumen HTML.
<hr />	Untuk membuat garis horizontal.
<html>	Mendefinisikan <i>root</i> dari sebuah dokumen HTML.
<img />	Untuk menampilkan gambar pada dokumen HTML.
<input />	Mendefinisikan area input pada formulir HTML.
<label>	Mendefinisikan label untuk sebuah elemen <input>.
<li>	Untuk menampilkan informasi dalam bentuk daftar.
<link />	Mendefinisikan hubungan antara suatu dokumen dengan sumber eksternalnya.
<marquee>	Untuk membuat teks berjalan secara vertical atau horizontal.
<meta>	Mendefinisikan sebuah metadata dari sebuah dokumen HTML.
<nobr>	Untuk mencegah ganti baris pada teks atau gambar.
<noscript>	Digunakan jika <i>browser</i> tidak mendukung <i>client-side script</i> .
<ol>	Mendefinisikan daftar dalam format penomoran.
<option>	Untuk menampilkan beberapa pilihan dalam bentuk <i>drop-down</i> pada dokumen HTML.
<p>	Mendefinisikan sebuah paragraf.
<script>	Mendefinisikan <i>client-side script</i> .
<small>	Untuk memperkecil ukuran teks.
<span>	Mendefinisikan <i>section</i> dalam dokumen HTML.
<strong>	Untuk membuat teks tebal, fungsi tag ini sama dengan tag <b>.
<style>	Mendefinisikan informasi <i>style</i> dalam dokumen HTML.
<table>	Untuk membuat tabel.
<td>	Mendefinisikan kolom dalam sebuah tabel.
<th>	Mendefinisikan <i>header</i> dalam sebuah tabel.
<title>	Untuk memberikan judul pada dokumen HTML.
<tr>	Mendefinisikan baris dalam sebuah tabel.



<u>	Untuk membuat teks dengan garis bawah.
<ul>	Mendefinisikan daftar dalam format <i>bullet</i> .

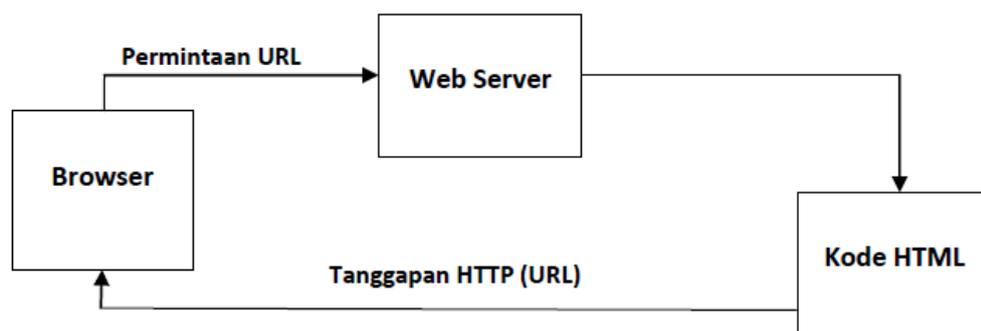
### 2.2.11 PHP

Menurut Anhar (2010) PHP atau *Hypertext Preprocessor* adalah sebuah bahasa pemrograman web *server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada server (*server-side HTML embedded scripting*). PHP juga merupakan sebuah *script* yang digunakan untuk membangun *website* yang dinamis. Dinamis artinya halaman yang ditampilkan dibuat pada saat halaman tersebut diminta oleh *client*. Hal ini menyebabkan informasi yang disampaikan kepada pengguna selalu yang terbaru.

PHP adalah bahasa *server side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis, karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi ke server yang kemudian hasilnya dikirimkan ke browser. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. PHP bernama FI (Form Interpreted), pada saat tersebut PHP adalah sekumpulan *script* yang digunakan untuk mengolah data form dari web (Saputra, 2012).

PHP merupakan bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memprogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. PHP merupakan bahasa *Server Side Scripting*, dimana PHP selalu membutuhkan web server dalam menjalankan aksinya. Secara prinsip, server akan bekerja apabila ada permintaan dari clien, yaitu kode-kode PHP. Client tersebut akan dikirimkan ke server, kemudian server akan mengembalikan pada halaman sesuai instruksi yang diminta (Saputra, 2012).

Cara kerja PHP menurut Saputra (2012) adalah sebagai berikut:



Gambar 2.18 Cara Kerja PHP

Sumber: Saputra (2012)

### 2.2.12 MySQL

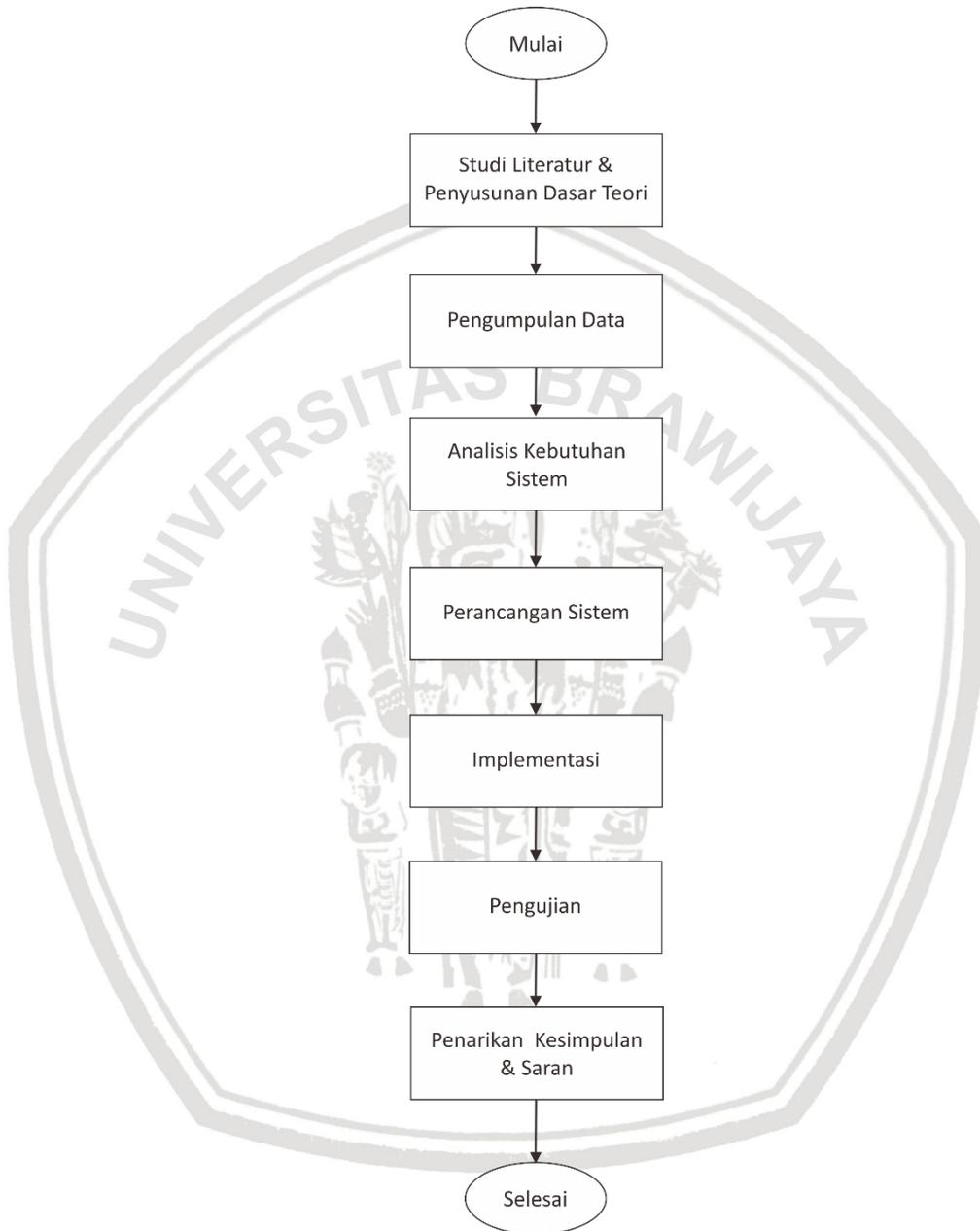
MySQL (*My Structure Query Language*) adalah sebuah perangkat lunak sistem manajemen basis data SQL *Database Management System* seperti Oracle, MS SQL, Postgre SQL, dan lainnya. MySQL merupakan DBMS yang multithread, multi-user yang bersifat gratis di bawah lisensi GNU General Public Licence (GPL) (Anhar, 2010).

MySQL memiliki beberapa kelebihan, diantaranya:

- MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah.
- MySQL memiliki kecepatan yang bagus dalam menangani *query* sederhana.
- MySQL memiliki operator dan fungsi secara penuh dan mendukung perintah *select* dan *where* dalam perintah *query*.
- MySQL memiliki keamanan yang bagus karena beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perijinan yang mendetail secara sandi terenskripsi.
- MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman lebih dari 50 juta dan 60 ribu tabel serta kurang lebih 5 milyar baris. Selain itu batas index yang dapat ditampung mencapai 32 index pada tiap tabelnya.
- MySQL dapat melakukan koneksi dengan *client* menggunakan *protocol* TCP/IP, *unix* soket (UNIX), atau *names pipes* (NT).
- MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari 20 bahasa.
- MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows*, *Linux*, *FreeBSD*, *Mac Os X Server*, *Solaris*, *Amiga*, dan masih banyak lagi.

## BAB 3 METODOLOGI

Pada bab ini akan dijabarkan langkah-langkah yang dilakukan dalam penyusunan skripsi seperti yang ditunjukkan pada Gambar 3.1, dan penjelasan mengenai prosesnya akan dibahas pada sub bab metodologi penelitian.



**Gambar 3.1 Diagram Metode Penelitian**

### 3.1 Studi Pustaka dan Penyusunan Dasar Teori

Studi Literatur adalah suatu teknik untuk mendapatkan teori guna memperoleh pendapat dari para ahli dan teorinya melalui bacaan. Teknik ini penulis gunakan untuk memperoleh data dari berbagai buku, jurnal, dan

halaman *web* yang berhubungan dengan masalah dalam penelitian untuk menunjang penelitian dan digunakan sebagai pedoman pembanding atau untuk memperkuat informasi yang berkaitan dengan masalah dan analisis dalam penelitian, yang meliputi teori, prinsip, konsep, hukum-hukum. Teori pendukung untuk penelitian ini meliputi:

1. Perpustakaan
2. Teknologi informasi perpustakaan
3. Sistem informasi perpustakaan
4. *System development life cycle* (SDLC)
5. Model Waterfall
6. *Design pattern* Model View Controller (MVC)

### 3.2 Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam pelaksanaan penelitian, data dibutuhkan untuk proses analisis dan untuk melakukan perincian mengenai apa saja yang dibutuhkan dalam proyeksi sistem. Berikut merupakan metode dalam melakukan pengumpulan data diantaranya:

a. Observasi

Pada metode observasi ini dilakukan peninjauan dan penelitian langsung di lapangan untuk memperoleh dan mengumpulkan data yang dibutuhkan. Dalam hal ini peneliti datang langsung untuk mengamati aktivitas layanan sirkulasi pada Perpustakaan Kecamatan Bungah seperti aktivitas peminjaman dan pengembalian buku, pengunjung dalam melakukan pencarian buku, pengisian buku kehadiran pengunjung, serta pendaftaran anggota perpustakaan yang dilakukan petugas.

b. Wawancara

Teknik pengumpulan data ini dilakukan melalui tatap muka dan tanya jawab langsung dengan pihak yang terlibat dalam aktivitas sirkulasi perpustakaan yaitu petugas Perpustakaan Kecamatan Bungah tentang bagaimana proses dan masalah yang petugas dapatkan dalam melakukan sirkulasi peminjaman, pencarian buku yang diinginkan pengunjung, serta pengelolaan data perpustakaan.

c. Studi Dokumen

Pada metode studi dokumen ini dilakukan dengan cara mempelajari dokumen untuk mendapatkan data atau informasi yang berhubungan dengan masalah yang diteliti. Dalam hal ini peneliti mengumpulkan dan mempelajari sejumlah dokumen perpustakaan seperti dokumen pembukuan data transaksi peminjaman, data koleksi buku, data anggota, data inventaris, data tamu/pengunjung serta dokumen prosedur layanan perpustakaan.

### 3.3 Analisis Kebutuhan Sistem

Proses analisis dilakukan terhadap pengumpulan data untuk mendapatkan spesifikasi kebutuhan sistem yang akan dibangun. Maka yang akan dilakukan oleh peneliti yaitu:

#### 1. Menggunakan Metode *Waterfall*

Metode yang digunakan dalam pengembangan sistem ini, terdiri dari mengikuti alur model waterfall mulai dari tahap analisis dan definisi persyaratan, hingga tahap pengujian.

**Tabel 3.1 Tahapan metodologi pengembangan sistem waterfall**

No	Tahapan	Keterangann
1	Analisis dan definisi persyaratan	Tahap pengumpulan kebutuhan sistem seperti data perpustakaan.
2	Perancangan sistem	Tahap menganalisis gambaran sistem yang akan dibuat, bagaimana antarmuka setiap kegiatannya, kebutuhan <i>hardware</i> , serta mendefinisikan arsitektur sistem secara keseluruhan.
3	Impelementasi dan pengujian unit	Tahap melakukan <i>coding</i> dan pemeriksaan secara teknis terhadap keseluruhan fungsi.
4	Integrasi dan pengujian sistem	Tahap penggabungan modul-modul yang telah dibuat dan pengujian dengan maksud mengetahui apakah <i>software</i> yang dikembangkan telah atau belum sesuai dengan perancangan.

#### 2. Analisis sistem yang sedang berjalan

Menganalisis sistem yang sedang berjalan untuk mengetahui permasalahan-permasalahan yang muncul pada sistem yang saat ini sedang berjalan seperti sulitnya melakukan pencarian buku oleh pengunjung dan membutuhkan waktu cukup lama dalam proses keanggotaan, dan sering terjadi antrian dalam melakukan sirkulasi peminjaman akibat dari pelaporan data yang masih dilakukan secara manual.

#### 3. Analisis sistem usulan

Dari hasil analisis sitem yang sedang berjalan, secara tidak langsung akan terlihat kelemahan yang ada pada sistem tersebut. Sehingga bisa dilakukan analisis kebutuhan sistem yang bertujuan untuk mengidentifikasi kekurangan dari sistem tersebut yang kemudian dilakukan langkah-langkah

pengembangan kedepannya seperti membuat sistem informasi perpustakaan yang dapat mempermudah petugas dan pengunjung dalam kegiatan sirkulasi perpustakaan dan pencarian buku yang diinginkan.

### 3.4 Perancangan Sistem

Perancangan sistem merupakan gambaran dari analisis kebutuhan. Perancangan sistem dalam penelitian ini meliputi: *Unified Modelling Language* (UML), basis data, dan tampilan. UML merupakan pemodelan mengenai sebuah sistem dengan menggunakan diagram dan teks-teks sebagai penjelas diagram. Pemodelan UML sistem ini menggunakan pola perancangan Model View Controller (MVC) untuk mempermudah dalam pengelolaan dan pengembangan sistem karena adanya pemisahan antara model, view, dan controller.

### 3.5 Implementasi

Implementasi kode dari desain yang telah dibuat secara nyata ke dalam bahasa pemrograman. Pada sistem ini implementasi kode dibuat ke dalam bahasa pemrograman PHP dan menggunakan basis data MySQL. Implementasi kode untuk membentuk fungsi-fungsi yang dibutuhkan oleh program dibuat dengan bantuan *framework* Bootstrap. Tahap ini diharapkan agar semua fungsi-fungsi berjalan dengan semestinya sesuai dengan desain yang telah dibuat.

### 3.6 Pengujian

Sistem yang telah dibuat dan memenuhi fungsi-fungsi yang dibutuhkan akan diuji menggunakan salah satu metode *white box testing* yaitu *basis path testing* untuk melakukan *unit testing*. Kemudian dilakukan *integration testing* antar fungsi sistem dan dilanjutkan dengan *system testing* untuk memastikan bahwa sistem telah diimplementasi dengan baik diantaranya pengujian sistem akan dilakukan dengan *validation testing*, *reliability testing*, dan *compability testing*. Dan terakhir adalah pengujian *acceptance testing* dengan menggunakan metode *black box* terhadap *stakeholder* untuk mengetahui apakah sistem telah sesuai dengan kebutuhan pengguna atau tidak.

### 3.7 Penarikan Kesimpulan dan Saran

Pada tahap ini merupakan tahap penarikan kesimpulan dari semua data yang telah diperoleh sebagai hasil dari penelitian. Penarikan kesimpulan merupakan tahap akhir dari kegiatan penelitian yang tak lupa dilengkapi dengan saran yang berisi kekurangan yang ditemukan dalam penelitian, kemungkinan penelitian lanjut, maupun potensi yang dimiliki oleh metode penelitian yang digunakan.

## BAB 4 ANALISIS KEBUTUHAN

Bab ini menjelaskan tentang analisis kebutuhan dan pemodelan proses bisnis yang didapatkan dari hasil wawancara dan observasi di perpustakaan Kecamatan Bungah. Pada pemodelan proses bisnis dilakukan penjabaran proses bisnis as-is dan proses bisnis to-be yang digambarkan dengan BPMN. Kemudian dilakukan analisis permasalahan dengan memetakan informasi hasil wawancara dan observasi ke dalam tabel analisis permasalahan. Dari hasil analisis permasalahan akan didapatkan saran berupa sistem yang akan dikembangkan beserta spesifikasi dan aktor yang terlibat di dalamnya. Setelah itu dilakukan analisis kebutuhan fungsional dan non-fungsional yang dimodelkan ke dalam *use case diagram* dan *use case scenario*.

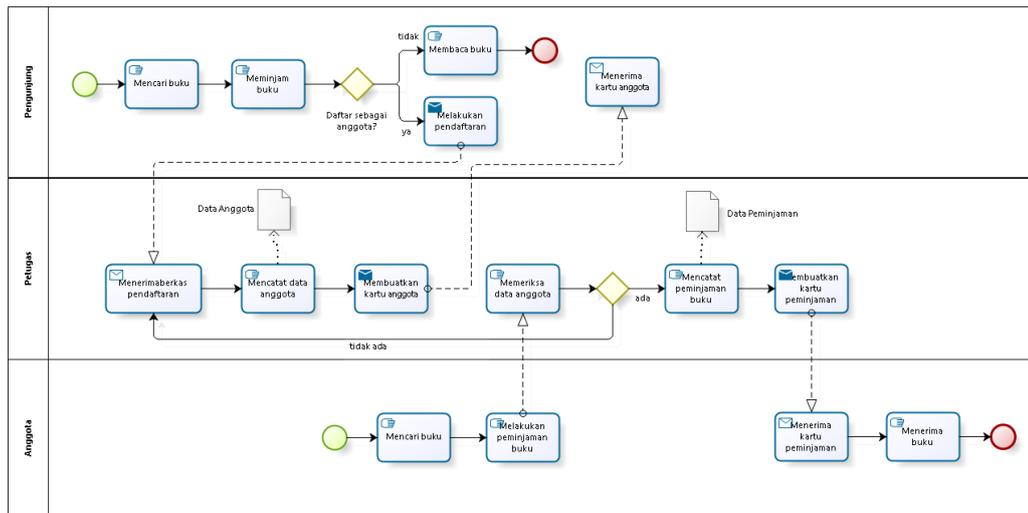
### 4.1 Business Perspective

Pada sub bab ini dilakukan analisis proses bisnis sebelum sistem informasi perpustakaan berbasis web diterapkan berupa diagram BPMN, dan juga dilakukan analisis permasalahan atau kendala yang dihadapi oleh *stakeholder*, serta siapa saja aktor yang terlibat dalam alur proses bisnis. Kemudian dari hasil analisis nantinya akan didapatkan saran atau solusi berupa proses bisnis yang dapat memaksimalkan pelayanan perpustakaan dengan bantuan sistem informasi yang akan dibangun.

#### 4.1.1 Identifikasi Proses Bisnis As-Is

##### 1. Proses Bisnis As-Is Keanggotaan dan Peminjaman Buku

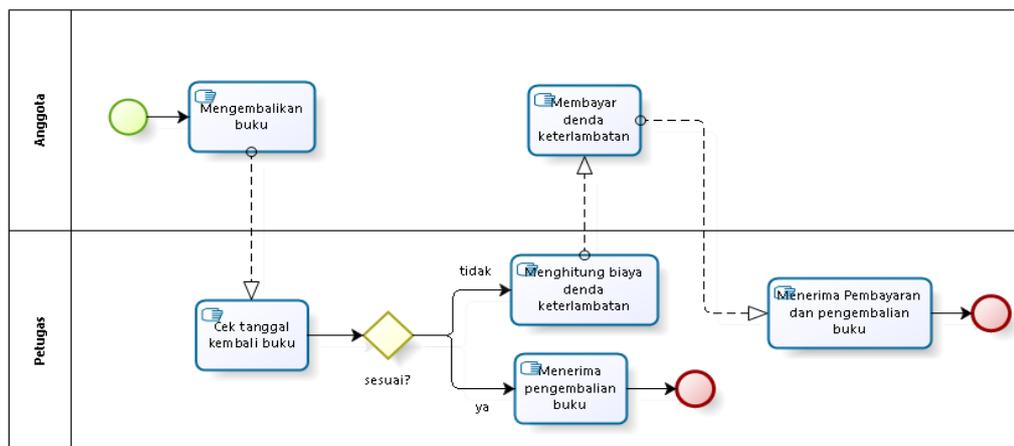
Dari hasil wawancara dengan petugas perpustakaan, didapatkan bagaimana proses untuk menangani peminjaman buku dan pendaftaran keanggotaan perpustakaan. Proses ini diawali setelah pengunjung mendapatkan buku yang diinginkan kemudian apabila pengunjung ingin meminjam buku, pengunjung harus terdaftar sebagai anggota perpustakaan terlebih dahulu. Pengunjung mendaftar dengan mengisi form pendaftaran yang sudah disediakan petugas, setelah itu berkas pendaftaran diterima petugas untuk dicatat di buku daftar keanggotaan dan membuat kartu anggota. Namun untuk mendapatkan kartu anggota, calon anggota harus menunggu beberapa hari sampai kartu anggota selesai dibuat. Tapi bukan berarti calon anggota tidak bisa melakukan peminjaman buku, calon anggota diperbolehkan meminjam buku meskipun belum mendapatkan kartu anggota, namun hal ini hanya diperbolehkan pada awal keanggotaan pengunjung saja, setelahnya wajib membawa kartu anggota untuk melakukan peminjaman buku. Setelah proses pendaftaran selesai, petugas mencatat peminjaman buku pada buku daftar peminjaman dan membuatkan kartu peminjaman untuk peminjam. Dan proses berakhir setelah peminjam mendapatkan buku yang diinginkan. Dari penjabaran alur proses bisnis di atas, berikut Gambar 4.1 merupakan penggambaran notasi BPMN untuk menggambarkan bagaimana proses bisnis untuk menangani peminjaman buku dan pendaftaran keanggotaan di perpustakaan.



Gambar 4.1 Proses Bisnis As-Is Keanggotaan dan Peminjaman Buku

## 2. Proses Bisnis As-Is Pengembalian Buku

Dari hasil wawancara dengan petugas perpustakaan, juga didapatkan bagaimana proses untuk menangani pengembalian buku. Proses ini diawali ketika peminjam buku mengembalikan buku dengan menyerahkan buku ke petugas secara langsung, selanjutnya petugas memeriksa tanggal kembali buku dari kartu peminjaman. Apabila waktu pengembalian dan batas waktu peminjaman buku sesuai, maka petugas dapat menerima pengembalian buku dari peminjam. Tapi apabila waktu pengembalian melebihi batas waktu peminjaman, maka petugas menghitung biaya denda yang dikenakan peminjam secara manual. Setelah itu peminjam membayarkan denda sesuai hasil perhitungan dari petugas dengan begitu buku akhirnya dapat diterima oleh petugas. Berikut Gambar 4.2 merupakan penggambaran notasi BPMN untuk menggambarkan bagaimana proses bisnis untuk menangani pengembalian buku di perpustakaan.



Gambar 4.2 Proses Bisnis As-Is Pengembalian Buku

#### 4.1.2 Analisis Permasalahan

##### 1. Analisis Permasalahan Keanggotaan dan Peminjaman Buku

Berikut merupakan penjelasan tentang analisis permasalahan saat ini pada proses keanggotaan dan peminjaman buku yang dijabarkan dalam Tabel 4.1.

**Tabel 4.1 Analisis Permasalahan Keanggotaan dan Peminjaman Buku**

Masalah	1. Mencari buku yang dilakukan secara manual
	2. Membuat kartu anggota membutuhkan waktu sehari-hari
	3. Memeriksa ketersediaan data keanggotaan peminjam secara manual dengan mengecek di buku daftar keanggotaan
	4. Harus membuat kartu peminjaman untuk mencatat daftar peminjaman setiap peminjam
Mempengaruhi	1. Kecepatan dalam mencari buku yang diinginkan
	2. Kecepatan mendapatkan kartu anggota untuk calon anggota
	3. Kecepatan dalam melakukan proses peminjaman buku
Dampak	1. Lama dalam melakukan pencarian buku yang diinginkan
	2. Membutuhkan waktu yang cukup lama untuk mendapatkan kartu anggota
	3. Membutuhkan waktu yang cukup lama dalam melakukan proses peminjaman buku sehingga menimbulkan antrian yang cukup panjang
	4. Seringnya terjadi kehilangan kartu peminjaman milik peminjam sehingga harus menuliskan ulang daftar peminjaman yang pernah dilakukan
Solusi	1. Menyediakan sistem yang dapat membantu mencarikan buku yang diinginkan
	2. Menyediakan sistem yang dapat mencetak kartu anggota secara langsung
	3. Menyediakan sistem yang dapat memberikan informasi keanggotaan secara cepat
	4. Menyediakan sistem yang dapat menyimpan data daftar peminjaman secara otomatis

Setelah dilakukan analisis pemasalahan pada Tabel 4.1, beberapa proses yang digambarkan pada diagram As-Is Gambar 4.1 dapat dipetakan dengan melakukan analisis rekomendasi proses bisnis yang baru untuk menyelesaikan masalah yang sudah dijabarkan pada analisis pemasalahan. Tabel 4.2 merupakan penjelasan dari hasil pemetaan dan analisis rekomendasi proses bisnis baru dari proses keanggotaan dan peminjaman buku.

**Tabel 4.2 Analisis dari Rekomendasi Proses Bisnis Keanggotaan dan Peminjaman Buku**

No	Deskripsi Masalah	Proses	Rekomendasi
1.	Mencari buku yang dilakukan secara manual	 Mencari buku	Proses mencari buku yang diinginkan dibantu oleh sistem
2.	Membuatkan kartu anggota membutuhkan waktu sehari-hari	 Membuatkan kartu anggota	Proses pembuatan kartu anggota dilakukan dengan bantuan sistem agar bisa langsung dicetak
3.	Memeriksa ketersediaan data keanggotaan peminjam secara manual dengan mengecek di buku daftar keanggotaan	 Memeriksa data anggota	Proses untuk melakukan pengecekan keanggotaan dapat diperiksa melalui sistem
4.	Harus membuatkan kartu peminjaman untuk mencatat daftar peminjaman setiap peminjam	 Membuatkan kartu peminjaman	Proses pencatatan daftar peminjaman setiap peminjam dilakukan dan disimpan dalam sistem

**2. Analisis Permasalahan Pengembalian Buku**

Berikut merupakan penjelasan tentang analisis permasalahan saat ini pada proses pengembalian buku yang dijabarkan dalam Tabel 4.3.

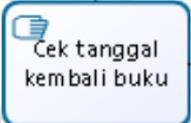
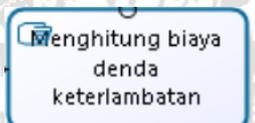
**Tabel 4.3 Analisis Permasalahan Pengembalian Buku**

Masalah	1. Memeriksa tanggal kembali buku secara manual 2. Menghitung biaya denda keterlambatan secara manual
Mempengaruhi	1. Kesesuaian pencatatan tanggal kembali 2. Ketepatan perhitungan total denda keterlambatan
Dampak	1. Kesesuaian pencatatan tanggal kembali di kartu

	peminjaman dengan buku daftar peminjaman
	2. Ketepatan perhitungan total denda keterlambatan yang dikenakan peminjam dari batas akhir tanggal peminjaman
Solusi	1. Menyediakan sistem yang dapat memberi informasi tanggal pengembalian buku secara akurat
	2. Menyediakan sistem yang dapat menghitung total denda keterlambatan secara otomatis

Setelah dilakukan analisis pemasalahan pada Tabel 4.3, beberapa proses yang digambarkan pada diagram As-Is Gambar 4.3 dapat dipetakan dengan melakukan analisis rekomendasi proses bisnis yang baru untuk menyelesaikan masalah yang sudah dijabarkan pada analisis pemasalahan. Tabel 4.4 merupakan penjelasan dari hasil pemetaan dan analisis rekomendasi proses bisnis baru dari proses pengembalian buku.

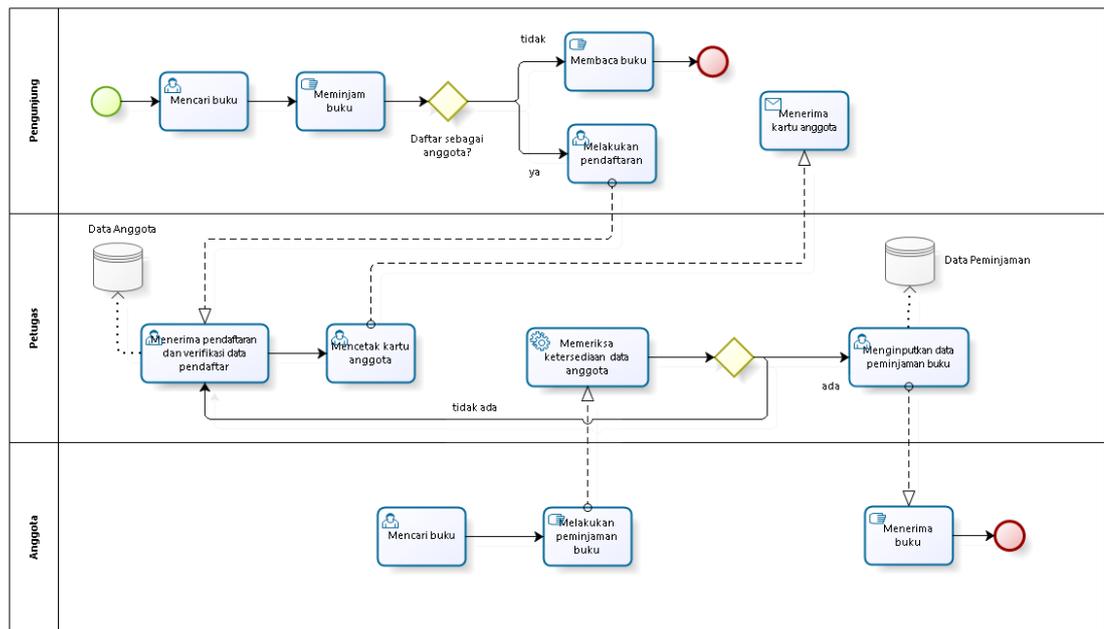
**Tabel 4.4 Analisis dari Rekomendasi Proses Bisnis Pengembalian Buku**

No	Deskripsi Masalah	Proses	Rekomendasi
1.	Memeriksa tanggal kembali buku secara manual		Proses pengecekan tanggal pengembalian buku dapat diperiksa melalui sistem
2.	Menghitung biaya denda keterlambatan secara manual		Proses perhitungan biaya denda keterlambatan dilakukan sistem

#### 4.1.3 Identifikasi Proses Bisnis To-Be

##### 1. Proses Bisnis To-Be Keanggotaan dan Peminjaman Buku

Berdasarkan hasil dari analisis permasalahan yang sudah dijelaskan sebelumnya, maka proses bisnis untuk menangani pendaftaran keanggotaan dan peminjaman buku akan diubah seperti pada Gambar 4.3.



**Gambar 4.3 Proses Bisnis To-Be Keanggotaan dan Peminjaman Buku**

Gambar 4.3 merupakan penggambaran proses bisnis baru keanggotaan dan peminjaman buku dengan diagram BPMN. Berikut merupakan urutan proses bisnis baru untuk menangani pendaftaran keanggotaan dan peminjaman buku.

- Pengunjung mencari buku dengan bantuan sistem.
- Jika pengunjung ingin meminjam buku, maka pengunjung harus mendaftar sebagai anggota perpustakaan terlebih dahulu.
- Jika pengunjung tidak ingin mendaftar sebagai anggota, maka pengunjung hanya diperbolehkan membaca buku di perpustakaan.
- Pendaftaran dapat dilakukan pengunjung sendiri atau petugas melalui sistem.
- Petugas menerima pendaftaran yang masuk dan selanjutnya diverifikasi melalui sistem.
- Petugas dapat langsung mencetak kartu anggota setelah anggota sudah terdaftar, dengan begitu calon anggota tidak perlu menunggu lama lagi untuk mendapatkan kartu anggota.
- Setelah pengunjung sudah menjadi anggota, petugas dapat memproses peminjaman buku yang diinginkan diawali dengan memeriksa ketersediaan data anggota melalui sistem.
- Jika data anggota ditemukan, maka petugas dapat memasukkan data peminjaman buku ke dalam sistem.
- Jika data anggota tidak ditemukan, maka pengunjung perpustakaan dianjurkan untuk melakukan pendaftaran.

- j. Setelah pencatatan peminjaman buku melalui sistem selesai, peminjam buku/anggota dapat menerima buku yang dipinjamnya.

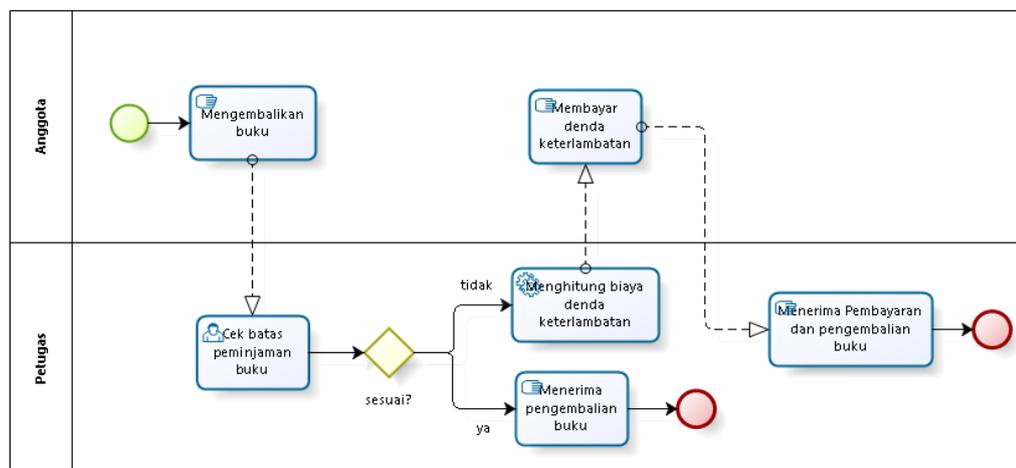
Tabel 4.5 merupakan pemetaan proses bisnis baru dari Gambar 4.3 disertai hasil rekomendasi dari analisis permasalahan pada Tabel 4.2.

**Tabel 4.5 Pemetaan Proses Bisnis Baru Kenggotaan dan Peminjaman Buku**

No.	Deskripsi Masalah	Rekomendasi	Proses
1.	Mencari buku yang dilakukan secara manual	Proses mencari buku yang diinginkan dibantu oleh sistem	Mencari buku
2.	Membuatkan kartu anggota membutuhkan waktu sehari-hari	Proses pembuatan kartu anggota dilakukan dengan bantuan sistem agar bisa langsung dicetak	Mencetak kartu anggota
3.	Memeriksa ketersediaan data keanggotaan peminjam secara manual dengan mengecek di buku daftar keanggotaan	Proses untuk melakukan pengecekan keanggotaan dapat diperiksa melalui sistem	Memeriksa ketersediaan data anggota
4.	Harus membuat kartu peminjaman untuk mencatat daftar peminjaman setiap peminjam	Proses pencatatan daftar peminjaman setiap peminjam dilakukan dan disimpan dalam sistem	Menginputkan data peminjaman buku

**2. Proses Bisnis To-Be Pengembalian Buku**

Proses bisnis untuk menangani pengembalian buku akan diubah seperti pada Gambar 4.4.



**Gambar 4.4 Proses Bisnis To-Be Pengembalian Buku**



Gambar 4.4 merupakan penggambaran proses bisnis baru pengembalian buku dengan diagram BPMN. Berikut merupakan urutan proses bisnis baru untuk menangani pengembalian buku.

- a. Peminjam buku mengembalikan buku dengan menyerahkan buku secara langsung ke petugas.
- b. Petugas melakukan pengecekan tanggal kembali buku melalui sistem.
- c. Apabila waktu penyerahan buku dan batas peminjaman buku sesuai, maka petugas menerima pengembalian buku dari peminjam.
- d. Apabila waktu penyerahan buku dan batas peminjaman buku tidak sesuai, maka sistem akan menunjukkan biaya denda yang dikenakan peminjam secara otomatis.
- e. Jika peminjam terkena denda, peminjam harus membayarkan denda sesuai yang dikalkulasikan oleh sistem.
- f. Terakhir petugas menerima uang denda dan pengembalian buku.

Tabel 4.6 merupakan pemetaan proses bisnis baru dari Gambar 4.4 disertai hasil rekomendasi dari analisis permasalahan pada Tabel 4.4.

**Tabel 4.6 Pemetaan Proses Bisnis Baru Pengembalian Buku**

No.	Deskripsi Masalah	Rekomendasi	Proses
1.	Memeriksa tanggal kembali buku secara manual	Proses pengecekan tanggal pengembalian buku dapat diperiksa melalui sistem	 Cek batas peminjaman buku
2.	Menghitung biaya denda keterlambatan secara manual	Proses perhitungan biaya denda keterlambatan dilakukan sistem	 Menghitung biaya denda keterlambatan

## 4.2 Deskripsi Perangkat Lunak

Perangkat lunak ini merupakan sistem informasi perpustakaan berbasis web yang dapat digunakan untuk mengolah data sirkulasi peminjaman buku, mengolah data inventaris perpustakaan, melakukan pencarian buku yang diinginkan dan melakukan pendaftaran anggota perpustakaan. Pengembangan sistem informasi perpustakaan ini menggunakan metode waterfall yaitu mulai dari analisis kebutuhan sistem, perancangan, implementasi dan pengujian yang dilakukan secara bertahap sampai menghasilkan produk layak pakai.

### 4.2.1 Fungsi Perangkat Lunak

Fungsi utama yang dimiliki perangkat lunak ini diantaranya:

1. Informasi yang dapat ditampilkan perangkat lunak sebagai berikut:
  - Data buku-buku yang terdapat di perpustakaan yaitu informasi mengenai judul, pengarang, penerbit, tahun terbit buku serta dapat menampilkan informasi letak koleksi buku yang ingin dicari.
  - Data anggota yang terdaftar di perpustakaan.
  - Data tamu atau pengunjung perpustakaan.
  - Data transaksi sirkulasi peminjaman buku yaitu informasi mengenai anggota yang melakukan peminjaman dan pengembalian buku.
  - Data inventaris yang terdapat di perpustakaan seperti fasilitas di ruang baca dan tempat bermain anak.
2. Melakukan pencarian koleksi buku dengan cepat dan mudah.
3. Melakukan perhitungan biaya denda bagi peminjam yang telat mengembalikan buku secara otomatis.
4. Memiliki fitur cetak kartu untuk mencetak kartu anggota perpustakaan tanpa harus menunggu waktu yang lama.

### 4.2.2 Batasan Perangkat Lunak

Batasan-batasan pada perangkat lunak ini adalah sebagai berikut:

- Sistem informasi dibangun berbasis web.
- Pemegang utama perangkat lunak ini adalah petugas perpustakaan yang dapat mengelola semua data perpustakaan
- Hanya beberapa fitur tertentu yang dapat digunakan oleh anggota dan pengunjung perpustakaan.
- Terdapat tiga aktor yang dapat menggunakan perangkat lunak ini dan aktor tidak dapat menambahkan pengguna lain.

### 4.2.3 Lingkungan Operasi

Sistem informasi ini dapat dijalankan pada sistem operasi manapun dan tidak memerlukan browser khusus untuk mengaksesnya, segala jenis browser dapat digunakan seperti Mozilla Firefox, Google Chrome, Opera, Microsoft Edge dan lain lain.

### 4.3 Identifikasi Kebutuhan

Identifikasi kebutuhan dilakukan untuk memperoleh informasi mengenai kebutuhan pengguna yang berisi pernyataan-pernyataan yang berhubungan dengan masalah yang telah diidentifikasi pada tahap analisis masalah. Pada penelitian ini, identifikasi kebutuhan pengguna dilakukan melalui wawancara dengan petugas perpustakaan serta melalui analisis permasalahan yang dilakukan sebelumnya. Hasil dari identifikasi kebutuhan pengguna ini dijabarkan pada tabel 4.7.

**Tabel 4.7 Identifikasi Kebutuhan Pengguna**

Kebutuhan Pengguna	Pengguna	Situasi saat ini	Solusi
Sistem dapat menyediakan layanan untuk mempermudah melakukan pencarian buku.	Pengunjung, Anggota	Pengunjung atau Anggota mencari buku yang diinginkan dengan cara manual.	Sistem yang menyediakan layanan pencarian buku secara cepat disertai informasi yang lengkap.
Sistem dapat mempermudah melakukan pengelolaan data perpustakaan.	Petugas	Petugas melakukan pelaporan data perpustakaan secara manual dengan mencatatkan pada buku.	Sistem yang membantu melakukan pengelolaan semua data perpustakaan.
Sistem dapat menyediakan layanan untuk pencatatan daftar riwayat peminjaman buku setiap peminjam.	Petugas	Petugas melakukan pencatatan daftar peminjaman tiap pengunjung pada selebar kertas yang dapat rentan hilang.	Sistem yang menyediakan layanan untuk menyimpan daftar peminjaman dan pengembalian buku.
Sistem dapat menyediakan layanan untuk mencetak kartu anggota.	Petugas	Petugas melakukan penerbitan kartu anggota	Sistem menyediakan layanan untuk mencetak kartu

		perpustakaan membutuhkan waktu yang lama.	anggota perpustakaan setelah pengunjung mendaftar sebagai anggota.
Sistem dapat menyediakan layanan untuk memvisualisasikan laporan peningkatan atau penurunan transaksi peminjaman dan pengunjung perpustakaan dari tahun ke tahun.	Petugas	Tidak ada	Sistem menyediakan layanan untuk memvisualisasikan data rekap transaksi peminjaman dan pengunjung perpustakaan dari tahun ke tahun.
Informasi di dalam sistem harus dapat diakses sesuai dengan identitas pengguna.	Petugas, Anggota, Pengunjung	Tidak ada	Sistem informasi menyediakan layanan untuk membatasi hak akses pengguna terhadap informasi yang dapat diakses.

#### 4.4 Identifikasi Pengguna

Identifikasi pengguna merupakan salah satu analisis kebutuhan untuk mengidentifikasi aktor-aktor yang terlibat dalam perangkat lunak ini. Identifikasi pengguna didapat dari hasil analisis proses bisnis yang sudah dijelaskan sebelumnya. Informasi dari identifikasi pengguna ini berguna untuk identifikasi aktor yang akan dimodelkan ke dalam diagram *use case*. Terdapat 3 aktor yang dapat berinteraksi dengan sistem yang dijelaskan pada Tabel 4.8.

**Tabel 4.8 Identifikasi Pengguna**

Pengguna	Deskripsi
Petugas	Petugas merupakan aktor utama dalam sistem yang dapat mengatur seluruh manajemen data dan penanganan anggota secara langsung.
Anggota	Anggota merupakan pengguna sistem setelah melakukan login dan <i>register</i> .
Pengunjung	Pengunjung merupakan aktor yang hanya dapat mengakses fitur-fitur tertentu dari sistem perpustakaan diantaranya mengisi data tamu

	perpustakaan dan mencari buku yang diinginkan.
--	--

#### 4.5 Identifikasi Fitur

Identifikasi fitur merupakan representasi solusi yang ditawarkan untuk memenuhi kebutuhan pengguna sehingga masalah yang dihadapi oleh *stakeholder* dapat diselesaikan. Hasil dari identifikasi fitur ini nantinya akan digunakan sebagai informasi untuk menentukan kebutuhan fungsional sistem. Identifikasi fitur yang ada pada sistem informasi perpustakaan kecamatan bungah dijabarkan pada Tabel 4.9.

**Tabel 4.9 Identifikasi Fitur**

Kode Fitur	Fitur	Deskripsi
FSIP-01	Visualisasi <i>dashboard</i>	Sistem dapat menampilkan <i>dashboard</i> berupa grafik pengunjung, serta grafik transaksi peminjaman dan pengembalian.
FSIP-02	Kelola data buku	Sistem dapat digunakan untuk melakukan pengelolaan data koleksi buku.
FSIP-03	Kelola data transaksi	Sistem dapat digunakan untuk melakukan pengelolaan data transaksi peminjaman dan pengembalian.
FSIP-04	Kelola data anggota	Sistem dapat digunakan untuk melakukan pengelolaan data anggota.
FSIP-05	Kelola data inventaris	Sistem dapat digunakan untuk melakukan pengelolaan data inventaris.
FSIP-06	Kelola data tamu	Sistem dapat digunakan untuk melakukan pengelolaan data tamu.
FSIP-07	Cari buku	Sistem dapat membantu melakukan pencarian buku yang diinginkan.
FSIP-08	Informasi denda peminjam	Sistem dapat menyajikan informasi denda yang dikenakan peminjam buku.
FSIP-09	Riwayat peminjaman	Sistem dapat menyajikan informasi mengenai daftar riwayat peminjaman yang telah dilakukan peminjam.
FSIP-10	Cetak kartu anggota	Sistem menyediakan fitur untuk mencetak kartu anggota perpustakaan bagi anggota yang baru terdaftar.
FSIP-11	Cetak struk peminjaman	Sistem menyediakan fitur untuk mencetak struk peminjaman buku

		perpustakaan.
FSIP-012	Autentifikasi pengguna	Sistem dapat mengenali pengguna sebelum menggunakan sistem serta dapat membatasi pengguna terhadap informasi dan layanan yang diberikan sesuai dengan hak akses yang dimiliki.

#### 4.6 Analisis Kebutuhan Fungsional

Pada sub bab ini akan dijabarkan apa saja kebutuhan fungsional sistem yang dapat dilakukan oleh pengguna terhadap sistem dan proses-proses apa saja yang dapat dilakukan sistem. Berikut Tabel 4.10 merupakan daftar kebutuhan fungsional sistem yang terdiri dari spesifikasinya yaitu kode fitur, kode kebutuhan, nama fungsi, dan deskripsi.

**Tabel 4.10 Kebutuhan Fungsional**

Kode Fitur	Kode Kebutuhan	Nama Fungsi	Deskripsi
FSIP-01	SP-F-001	Melihat <i>dashboard</i>	Pengguna dapat melihat halaman <i>dashboard</i>
FSIP-02	SP-F-002	Melihat data buku	Pengguna dapat melihat seluruh data koleksi buku yang telah terdaftar
	SP-F-003	Menambah data buku	Pengguna dapat menambahkan data buku meliputi kode buku, judul, pengarang, penerbit
	SP-F-004	Memperbarui data buku	Pengguna dapat melakukan perbaruan data buku
	SP-F-005	Menghapus data buku	Pengguna dapat menghapus data buku
FSIP-03	SP-F-006	Melihat data transaksi peminjaman	Pengguna dapat melihat data transaksi peminjaman
	SP-F-007	Menambah data transaksi peminjaman	Pengguna dapat menambah data transaksi peminjaman
	SP-F-008	Memperbarui data transaksi peminjaman	Pengguna dapat melakukan perbaruan data transaksi peminjaman
	SP-F-009	Menghapus data transaksi	Pengguna dapat menghapus

		peminjaman	data transaksi peminjaman
FSIP-04	SP-F-010	Melihat data anggota	Pengguna dapat melihat seluruh data anggota yang telah terdaftar
	SP-F-011	Menambah data anggota	Pengguna dapat menambahkan data anggota
	SP-F-012	Memperbarui data anggota	Pengguna dapat melakukan perbaruan data anggota
	SP-F-013	Menghapus data anggota	Pengguna dapat menghapus data anggota
FSIP-05	SP-F-014	Melihat data inventaris	Pengguna dapat melihat data inventaris
	SP-F-015	Menambah data inventaris	Pengguna dapat menambahkan data inventaris
	SP-F-016	Memperbarui data inventaris	Pengguna dapat melakukan perbaruan data inventaris
	SP-F-017	Menghapus data inventaris	Pengguna dapat menghapus data inventaris
FSIP-06	SP-F-018	Melihat data tamu	Pengguna dapat melihat data tamu
	SP-F-019	Menambahkan data tamu	Pengguna dapat menambahkan data tamu
	SP-F-020	Menghapus data tamu	Pengguna dapat menghapus data tamu
FSIP-07	SP-F-021	Mecari buku	Pengguna dapat mencari letak keberadaan buku
FSIP-08	SP-F-022	Melihat denda peminjam	Pengguna dapat melihat denda peminjam yang telat mengembalikan buku
FSIP-09	SP-F-023	Melihat riwayat peminjaman	Pengguna dapat melihat daftar data riwayat peminjaman
FSIP-10	SP-F-024	Mencetak kartu anggota	Pengguna dapat mencetak kartu anggota
FSIP-11	SP-F-025	Mencetak struk peminjaman	Pengguna dapat mencetak struk peminjaman
FSIP-12	SP-F-026	Login	Sistem menyediakan layanan untuk masuk ke sistem sesuai

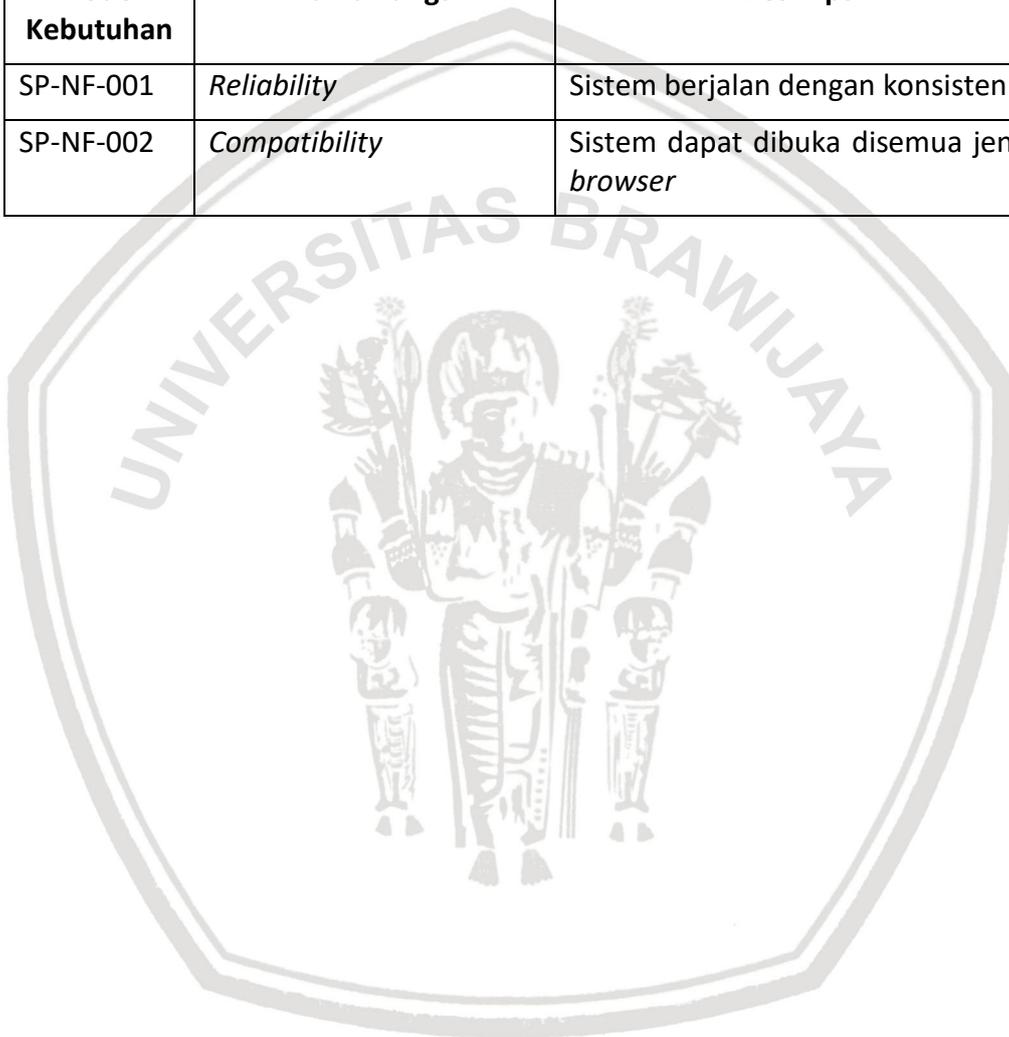
			dengan hak akses yang dimiliki
--	--	--	--------------------------------

#### 4.7 Analisis Kebutuhan Non-Fungsional

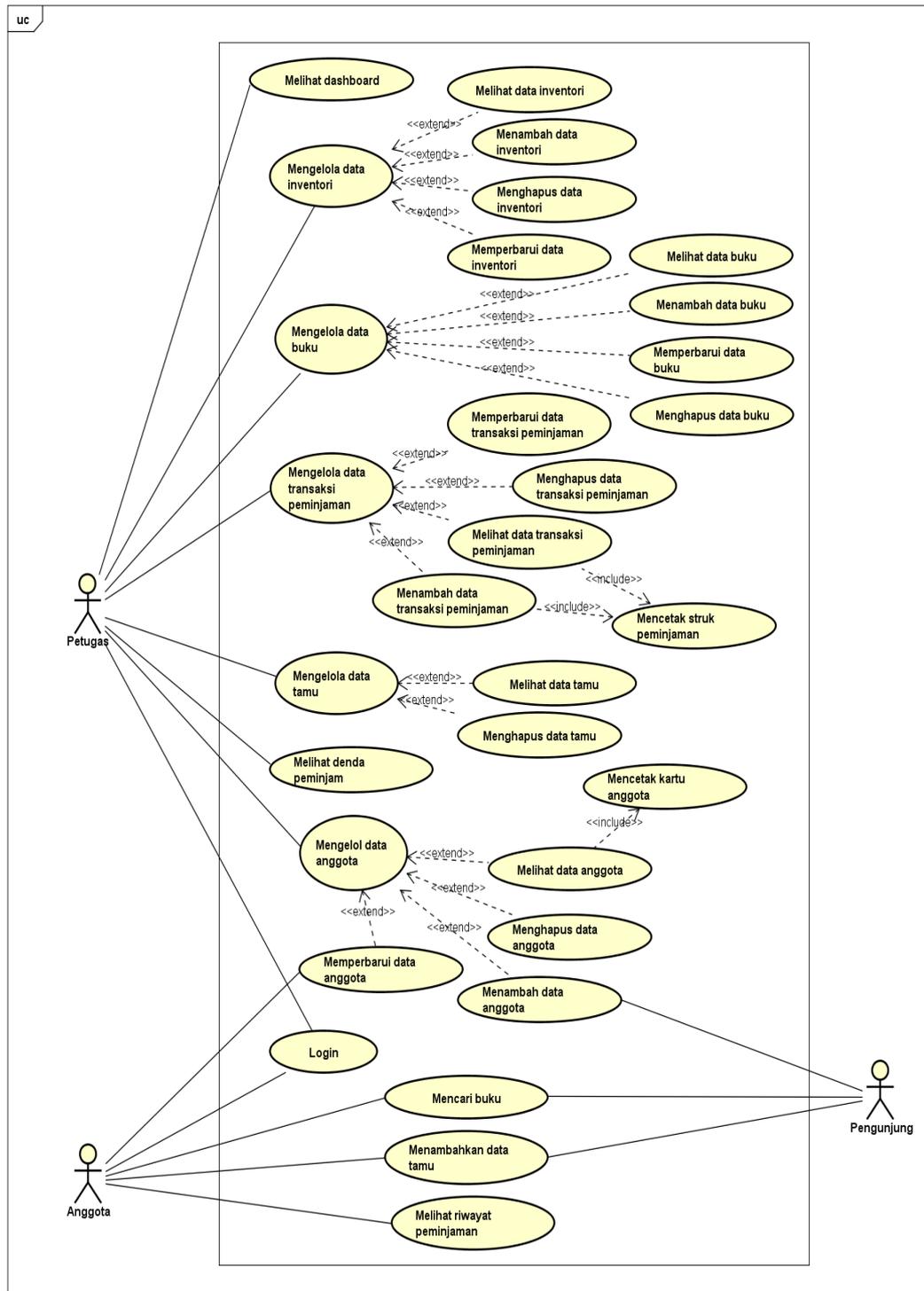
Kebutuhan non-fungsional sistem merupakan fitur-fitur penunjang kerja sebuah sistem yang menentukan atribut atau kualitas layanan secara keseluruhan. Tabel 4.11 merupakan daftar kebutuhan non-fungsional sistem yang terdiri dari spesifikasinya yaitu kode kebutuhan, nama fungsi, dan deskripsi.

**Tabel 4.11 Kebutuhan Non-Fungsional**

Kode Kebutuhan	Nama Fungsi	Deskripsi
SP-NF-001	<i>Reliability</i>	Sistem berjalan dengan konsisten
SP-NF-002	<i>Compatibility</i>	Sistem dapat dibuka disemua jenis <i>browser</i>



### 4.8 Use Case Diagram



**Gambar 4.5 Use Case Diagram**

Gambar 4.5 merupakan gambaran interaksi atau fungsi-fungsi yang dapat dilakukan aktor terhadap sistem. Fungsi-fungsi tersebut diantaranya, fungsi login untuk autentifikasi pengguna ketika masuk ke dalam sistem, kemudian terdapat fungsi melihat data buku, menambah data buku, menghapus data buku, dan memperbarui data buku yang merupakan *extend* dari fungsi mengelola data



buku, kemudian terdapat fungsi melihat data transaksi peminjaman, menambah data transaksi peminjaman, menghapus data transaksi peminjaman, dan memperbarui data transaksi peminjaman yang merupakan *extend* dari fungsi mengelola data transaksi peminjaman, kemudian terdapat fungsi melihat data anggota, menambah data anggota, menghapus data anggota, dan memperbarui data anggota yang merupakan *extend* dari fungsi mengelola data anggota, kemudian terdapat fungsi melihat data inventaris, menambah data inventaris, menghapus data inventaris, dan memperbarui data inventaris yang merupakan *extend* dari fungsi mengelola data inventaris. Selain itu juga terdapat fungsi mencetak kartu anggota, mencetak struk peminjaman, melihat denda peminjam, mencari buku, melihat data tamu, menambahkan data tamu, dan menghapus data tamu. Pada gambar tersebut juga terdapat 3 pengguna sistem yaitu Petugas, Anggota, dan Pengunjung.

#### 4.9 Use Case Scenario

Dari use case yang telah digambarkan di atas, maka dapat dijabarkan use case scenario dari setiap pengguna sebagai berikut.

##### 4.9.1 Use Case Scenario Petugas

**Tabel 4.12 Use Case Scenario Login Pada Sisi Petugas**

<b>Use Case</b>	<i>Login</i>
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas melakukan <i>login</i> untuk dapat menggunakan sistem
<b>Kondisi Awal</b>	Petugas ingin menggunakan sistem dan belum melakukan <i>login</i>
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memasukkan <i>username</i> dan <i>password</i></li> <li>2. Sistem memeriksa <i>valid</i> atau tidaknya data input</li> <li>3. Sistem menampilkan halaman utama dari sistem</li> </ol>
<b>Kondisi Akhir</b>	Petugas telah <i>login</i> dan siap berinteraksi dengan sistem
<b>Skenario Alternatif</b>	<ol style="list-style-type: none"> <li>1. Petugas memasukkan <i>username</i> dan <i>password</i></li> <li>2. Sistem memeriksa <i>valid</i> atau tidaknya data input</li> <li>3. Jika tidak <i>valid</i>, sistem menampilkan pesan data tidak valid</li> <li>4. Sistem kembali menampilkan halaman <i>login</i></li> </ol>

**Tabel 4.13 Use Case Scenario Melihat Dashboard**

<b>Use Case</b>	Melihat <i>dashboard</i>
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat melihat halaman <i>dashboard</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke dalam sistem

<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu halaman utama</li> <li>2. Sistem melakukan permintaan halaman utama</li> <li>3. Sistem membaca detail setiap data peminjaman dan pengunjung perpustakaan dari <i>database</i></li> <li>4. Sistem menampilkan halaman <i>dashboard</i></li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman <i>dashboard</i>
<b>Skenario Alternatif</b>	-

**Tabel 4.14 Use Case Scenario Melihat Data Buku**

<b>Use Case</b>	Melihat data buku
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat melihat seluruh data buku yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah login ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Koleksi Buku</li> <li>2. Sistem melakukan permintaan halaman koleksi buku</li> <li>3. Sistem membaca data koleksi buku dari <i>database</i></li> <li>4. Sistem menampilkan halaman data koleksi buku</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman daftar koleksi buku
<b>Skenario Alternatif</b>	-

**Tabel 4.15 Use Case Scenario Menambah Data Buku**

<b>Use Case</b>	Menambah data buku
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat menambah data buku yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Koleksi Buku</li> <li>2. Sistem melakukan permintaan halaman koleksi buku</li> <li>3. Sistem menampilkan halaman data koleksi buku</li> <li>4. Petugas memilih menu tambah buku</li> <li>5. Sistem melakukan permintaan ke formulir tambah buku</li> <li>6. Petugas mengisi formulir detail informasi buku</li> <li>7. Petugas mensubmit tombol tambah buku</li> <li>8. Sistem melakukan permintaan untuk menyimpan data koleksi buku baru ke dalam <i>database</i></li> <li>9. Sistem menampilkan halaman data koleksi buku yang</li> </ol>

	baru ditambahkan
<b>Kondisi Akhir</b>	Sistem menampilkan halaman data koleksi buku yang baru ditambahkan
<b>Skenario Alternatif</b>	-

**Tabel 4.16 Use Case Scenario Menghapus Data Buku**

<b>Use Case</b>	Menghapus data buku
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat menghapus data buku yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Koleksi Buku</li> <li>2. Sistem melakukan permintaan halaman koleksi buku</li> <li>3. Sistem menampilkan halaman data koleksi buku</li> <li>4. Petugas memilih tombol hapus pada data koleksi buku yang dipilih</li> <li>5. Sistem menampilkan kotak dialog konfirmasi hapus data koleksi buku yang dipilih</li> <li>6. Petugas menyetujui dengan memilih tombol OK</li> <li>7. Sistem menghapus data buku yang dipilih pengguna dari <i>database</i></li> </ol>
<b>Kondisi Akhir</b>	Sistem menghapus data buku yang dipilih pengguna dari <i>database</i>
<b>Skenario Alternatif</b>	-

**Tabel 4.17 Use Case Scenario Memperbarui Data Buku**

<b>Use Case</b>	Memperbarui data buku
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat memperbarui data buku yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Koleksi Buku</li> <li>2. Sistem melakukan permintaan halaman koleksi buku</li> <li>3. Sistem menampilkan halaman data koleksi buku</li> <li>4. Petugas memilih tombol edit pada data koleksi buku yang ingin diubah</li> <li>5. Sistem menampilkan formulir edit data buku</li> </ol>

	<ol style="list-style-type: none"> <li>6. Petugas mengubah informasi data buku yang ingin diubah</li> <li>7. Petugas memilih tombol simpan untuk menyimpan perubahan</li> <li>8. Sistem melakukan permintaan untuk menyimpan data buku yang baru diubah</li> <li>9. Sistem menampilkan halaman data koleksi buku setelah diperbarui</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman data koleksi buku setelah diperbarui
<b>Skenario Alternatif</b>	-

**Tabel 4.18 Use Case Scenario Melihat Data Anggota**

<b>Use Case</b>	Melihat data anggota
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat melihat seluruh data anggota yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Anggota Perpustakaan</li> <li>2. Sistem melakukan permintaan halaman data anggota perpustakaan</li> <li>3. Sistem membaca data anggota yang terdapat pada <i>database</i></li> <li>4. Sistem menampilkan halaman data anggota perpustakaan yang telah terdaftar</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman data anggota perpustakaan yang telah terdaftar
<b>Skenario Alternatif</b>	-

**Tabel 4.19 Use Case Scenario Menambah Data Anggota Pada Sisi Petugas**

<b>Use Case</b>	Menambah data anggota
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat menambah data anggota yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Anggota Perpustakaan</li> <li>2. Sistem melakukan permintaan halaman data anggota</li> </ol>

	<p>perpustakaan</p> <ol style="list-style-type: none"> <li>3. Sistem menampilkan halaman data anggota yang terdaftar</li> <li>4. Petugas memilih menu tambah anggota</li> <li>5. Sistem melakukan permintaan ke formulir tambah anggota</li> <li>6. Petugas mengisi formulir detail data anggota perpustakaan</li> <li>7. Petugas mensubmit tombol tambah anggota</li> <li>8. Sistem melakukan permintaan untuk menyimpan data anggota baru ke dalam <i>database</i></li> <li>9. Sistem menampilkan halaman data anggota perpustakaan yang baru ditambahkan</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman data anggota perpustakaan yang baru ditambahkan
<b>Skenario Alternatif</b>	-

**Tabel 4.20 Use Case Scenario Menghapus Data Anggota**

<b>Use Case</b>	Menghapus data anggota
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat menghapus data anggota yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Anggota Perpustakaan</li> <li>2. Sistem melakukan permintaan halaman data anggota perpustakaan</li> <li>3. Sistem menampilkan halaman data anggota yang terdaftar</li> <li>4. Petugas memilih tombol hapus pada data anggota yang dipilih untuk dihapus</li> <li>5. Sistem menampilkan kotak dialog konfirmasi hapus data anggota yang dipilih</li> <li>6. Petugas menyetujui dengan memilih tombol OK</li> <li>7. Sistem menghapus data anggota yang dipilih pengguna dari <i>database</i></li> </ol>
<b>Kondisi Akhir</b>	Sistem menghapus data anggota yang dipilih pengguna dari <i>database</i>
<b>Skenario Alternatif</b>	-

**Tabel 4.21 Use Case Scenario Memperbarui Data Anggota Pada Sisi Petugas**

<b>Use Case</b>	Memperbarui data anggota
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat memperbarui data anggota yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Anggota Perpustakaan</li> <li>2. Sistem melakukan permintaan halaman data anggota perpustakaan</li> <li>3. Sistem menampilkan halaman data anggota yang terdaftar</li> <li>4. Petugas memilih tombol edit pada data anggota yang ingin diubah</li> <li>5. Sistem menampilkan formulir edit data anggota</li> <li>6. Petugas mengubah informasi data anggota yang ingin diubah dengan informasi yang berbeda</li> <li>7. Petugas memilih tombol simpan untuk menyimpan perubahan</li> <li>8. Sistem melakukan permintaan untuk menyimpan data anggota yang baru diubah</li> <li>9. Sistem menampilkan halaman data anggota setelah diperbarui</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman data anggota setelah diperbarui
<b>Skenario Alternatif</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Anggota Perpustakaan</li> <li>2. Sistem melakukan permintaan halaman data anggota perpustakaan</li> <li>3. Sistem menampilkan halaman data anggota yang terdaftar</li> <li>4. Petugas memilih tombol edit pada data anggota yang ingin diubah</li> <li>5. Sistem menampilkan formulir edit data anggota</li> <li>6. Petugas mengganti <i>username</i> anggota dengan <i>username</i> yang sudah terdaftar di <i>database</i></li> <li>7. Petugas memilih tombol simpan untuk menyimpan perubahan</li> <li>8. Sistem memeriksa ketersediaan <i>username</i> di dalam <i>database</i></li> <li>9. Jika <i>username</i> sudah terdaftar di dalam <i>database</i>, maka sistem akan memberikan peringatan untuk mengubah <i>username</i> lain yang berbeda</li> </ol>

**Tabel 4.22 Use Case Scenario Mencetak Kartu Anggota**

<b>Use Case</b>	Mencetak kartu anggota
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat mencetak kartu anggota
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Anggota Perpustakaan</li> <li>2. Sistem melakukan permintaan halaman data anggota perpustakaan</li> <li>3. Sistem menampilkan halaman data anggota yang terdaftar</li> <li>4. Petugas memilih salah satu data anggota yang ingin diproses</li> <li>5. Sistem melakukan permintaan halaman detail data anggota dari <i>database</i></li> <li>6. Sistem menampilkan detail informasi data anggota</li> <li>7. Petugas memilih tombol cetak kartu</li> <li>8. Sistem menampilkan jendela konfirmasi untuk melakukan pencetakan kartu</li> <li>9. Petugas memilih tombol Print untuk menyetujui</li> <li>10. Kartu anggota perpustakaan dicetak</li> </ol>
<b>Kondisi Akhir</b>	Kartu anggota perpustakaan dicetak
<b>Skenario Alternatif</b>	-

**Tabel 4.23 Use Case Scenario Mencetak Struk Peminjaman**

<b>Use Case</b>	Mencetak struk peminjaman
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat mencetak struk peminjaman
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Transaksi</li> <li>2. Sistem melakukan permintaan halaman transaksi peminjaman dan pengembalian buku</li> <li>3. Sistem menampilkan halaman transaksi peminjaman dan pengembalian buku. Petugas mengisi form peminjaman buku meliputi nomor ID anggota dan kode buku</li> <li>4. Petugas memilih tombol simpan</li> <li>5. Sistem melakukan permintaan untuk menyimpan data transaksi yang baru ditambahkan</li> <li>6. Sistem menampilkan halaman informasi struk peminjaman buku yang baru ditambahkan dan</li> </ol>

	<p>menampilkan jendela konfirmasi untuk mencetak struk</p> <ol style="list-style-type: none"> <li>7. Petugas memilih tombol Print untuk menyetujui</li> <li>8. Struk peminjaman buku dicetak</li> </ol>
<b>Kondisi Akhir</b>	Struk peminjaman buku dicetak
<b>Skenario Alternatif</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Transaksi</li> <li>2. Sistem melakukan permintaan halaman transaksi peminjaman dan pengembalian buku</li> <li>3. Sistem menampilkan halaman transaksi peminjaman dan pengembalian buku</li> <li>4. Petugas memilih salah satu data transaksi peminjaman untuk dicetakkan struk peminjaman dengan memilih tombol cetak struk pada ujung paling kanan kolom tabel data transaksi</li> <li>5. Sistem menampilkan halaman informasi struk peminjaman buku yang ingin dicetak dan menampilkan jendela konfirmasi untuk mencetak struk</li> <li>6. Petugas memilih tombol Print untuk menyetujui</li> <li>7. Struk peminjaman buku dicetak</li> </ol>

**Tabel 4.24 Use Case Scenario Melihat Data Transaksi Peminjaman**

<b>Use Case</b>	Melihat data transaksi peminjaman
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat melihat seluruh data transaksi peminjaman yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Peminjaman dan Pengembalian</li> <li>2. Sistem melakukan permintaan halaman data transaksi peminjaman</li> <li>3. Sistem membaca data transaksi peminjaman yang terdapat pada <i>database</i></li> <li>4. Sistem menampilkan halaman data transaksi peminjaman</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman data transaksi peminjaman
<b>Skenario Alternatif</b>	-

**Tabel 4.25 Use Case Scenario Menambah Data Transaksi Peminjaman**

<b>Use Case</b>	Menambah data transaksi peminjaman
<b>Aktor</b>	Petugas

<b>Deskripsi</b>	Petugas dapat menambah data transaksi peminjaman yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Peminjaman dan Pengembalian</li> <li>2. Sistem melakukan permintaan halaman data transaksi peminjaman</li> <li>3. Sistem menampilkan halaman data transaksi peminjaman</li> <li>4. Petugas memilih menu tambah transaksi</li> <li>5. Sistem melakukan permintaan ke formulir tambah transaksi</li> <li>6. Petugas mengisi formulir detail data transaksi peminjaman</li> <li>7. Petugas mensubmit tombol tambah transaksi</li> <li>8. Sistem melakukan permintaan untuk menyimpan data transaksi baru ke dalam <i>database</i></li> <li>9. Sistem menampilkan halaman data transaksi peminjaman yang baru ditambahkan</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman data transaksi peminjaman yang baru ditambahkan
<b>Skenario Alternatif</b>	-

**Tabel 4.26 Use Case Scenario Menghapus Data Transaksi Peminjaman**

<b>Use Case</b>	Menghapus data transaksi peminjaman
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat menghapus data transaksi peminjaman yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Peminjaman dan Pengembalian</li> <li>2. Sistem melakukan permintaan halaman data transaksi peminjaman</li> <li>3. Sistem menampilkan halaman data transaksi peminjaman</li> <li>4. Petugas memilih tombol hapus pada data transaksi peminjaman yang dipilih untuk dihapus</li> <li>5. Sistem menampilkan kotak dialog konfirmasi hapus data transaksi peminjaman yang dipilih</li> <li>6. Petugas menyetujui dengan memilih tombol OK</li> <li>7. Sistem menghapus data transaksi peminjaman yang dipilih pengguna dari <i>database</i></li> </ol>

<b>Kondisi Akhir</b>	Sistem menghapus data transaksi peminjaman yang dipilih pengguna dari <i>database</i>
<b>Skenario Alternatif</b>	-

**Tabel 4.27 Use Case Scenario Memperbarui Data Transaksi Peminjaman**

<b>Use Case</b>	Memperbarui data transaksi peminjaman
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat memperbarui data transaksi peminjaman yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Peminjaman dan Pengembalian</li> <li>2. Sistem melakukan permintaan halaman data transaksi peminjaman</li> <li>3. Sistem menampilkan halaman data transaksi peminjaman</li> <li>4. Petugas memilih tombol edit pada data transaksi peminjaman yang ingin diubah</li> <li>5. Sistem menampilkan formulir edit data transaksi peminjaman</li> <li>6. Petugas mengubah informasi data transaksi peminjaman yang ingin diubah</li> <li>7. Petugas memilih tombol simpan untuk menyimpan perubahan</li> <li>8. Sistem melakukan permintaan untuk menyimpan data transaksi peminjaman yang baru diubah</li> <li>9. Sistem menampilkan halaman data transaksi peminjaman setelah diperbarui</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman data transaksi peminjaman setelah diperbarui
<b>Skenario Alternatif</b>	-

**Tabel 4.28 Use Case Scenario Melihat Denda Peminjaman**

<b>Use Case</b>	Melihat denda peminjaman
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat melihat denda peminjaman yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem

<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Transaksi</li> <li>2. Sistem melakukan permintaan halaman data transaksi peminjaman</li> <li>3. Sistem menampilkan halaman data transaksi peminjaman</li> <li>4. Petugas mengisi form pengembalian dengan kode pinjam</li> <li>5. Petugas pilih tombol submit</li> <li>6. Sistem melakukan permintaan data transaksi peminjaman sesuai kode pinjam</li> <li>7. Sistem menampilkan halaman detail informasi peminjaman termasuk denda yang dikenakan peminjam</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman detail informasi peminjaman
<b>Skenario Alternatif</b>	-

**Tabel 4.29 Use Case Scenario Melihat Data Tamu**

<b>Use Case</b>	Melihat data tamu
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat melihat seluruh data tamu yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Tamu</li> <li>2. Sistem melakukan permintaan halaman data tamu</li> <li>3. Sistem membaca data tamu yang terdapat pada <i>database</i></li> <li>4. Sistem menampilkan halaman data tamu</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman data tamu
<b>Skenario Alternatif</b>	-

**Tabel 4.30 Use Case Scenario Menghapus Data Tamu**

<b>Use Case</b>	Menghapus data tamu
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat menghapus data tamu yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Tamu</li> <li>2. Sistem melakukan permintaan halaman data Tamu</li> </ol>

	<ol style="list-style-type: none"> <li>3. Sistem menampilkan halaman data tamu</li> <li>4. Petugas memilih tombol hapus pada data tamu yang dipilih untuk dihapus</li> <li>5. Sistem menampilkan kotak dialog konfirmasi hapus data tamu yang dipilih</li> <li>6. Petugas menyetujui dengan memilih tombol OK</li> <li>7. Sistem menghapus data tamu yang dipilih pengguna dari <i>database</i></li> </ol>
<b>Kondisi Akhir</b>	Sistem menghapus data tamu yang dipilih pengguna dari <i>database</i>
<b>Skenario Alternatif</b>	-

**Tabel 4.31 Use Case Scenario Melihat Data Inventaris**

<b>Use Case</b>	Melihat data inventaris
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat melihat seluruh data inventaris yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Inventaris</li> <li>2. Sistem melakukan permintaan halaman data inventaris</li> <li>3. Sistem membaca data tamu yang terdapat pada <i>database</i></li> <li>4. Sistem menampilkan halaman data inventaris</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman data inventaris
<b>Skenario Alternatif</b>	-

**Tabel 4.32 Use Case Scenario Menambah Data Inventaris**

<b>Use Case</b>	Menambah data inventaris
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat menambah data transaksi peminjaman yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Inventaris</li> <li>2. Sistem melakukan permintaan halaman data inventaris</li> <li>3. Sistem menampilkan halaman data inventaris</li> <li>4. Petugas memilih menu tambah inventaris</li> <li>5. Sistem melakukan permintaan ke formulir tambah</li> </ol>

	<p>inventaris</p> <ol style="list-style-type: none"> <li>6. Petugas mengisi formulir detail data inventaris perpustakaan</li> <li>7. Petugas mensubmit tombol tambah inventaris</li> <li>8. Sistem melakukan permintaan untuk menyimpan data inventaris baru ke dalam <i>database</i></li> <li>9. Sistem menampilkan halaman data inventaris yang baru ditambahkan</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman data inventaris yang baru ditambahkan
<b>Skenario Alternatif</b>	-

**Tabel 4.33 Use Case Scenario Menghapus Data Inventaris**

<b>Use Case</b>	Menghapus data inventaris
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat menghapus data inventaris yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Inventaris</li> <li>2. Sistem melakukan permintaan halaman data inventaris</li> <li>3. Sistem menampilkan halaman data inventaris</li> <li>4. Petugas memilih tombol hapus pada data inventaris yang dipilih untuk dihapus</li> <li>5. Sistem menampilkan kotak dialog konfirmasi hapus data inventaris yang dipilih</li> <li>6. Petugas menyetujui dengan memilih tombol OK</li> <li>7. Sistem menghapus data inventaris yang dipilih pengguna dari <i>database</i></li> </ol>
<b>Kondisi Akhir</b>	Sistem menghapus data inventaris yang dipilih pengguna dari <i>database</i>
<b>Skenario Alternatif</b>	-

**Tabel 4.34 Use Case Scenario Memperbarui Data Inventaris**

<b>Use Case</b>	Memperbarui data transaksi peminjaman
<b>Aktor</b>	Petugas
<b>Deskripsi</b>	Petugas dapat memperbarui data transaksi peminjaman yang ada pada <i>database</i>

<b>Kondisi Awal</b>	Petugas telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Inventaris</li> <li>2. Sistem melakukan permintaan halaman data inventaris</li> <li>3. Sistem menampilkan halaman data inventaris</li> <li>4. Petugas memilih tombol edit pada data inventaris yang ingin diubah</li> <li>5. Sistem menampilkan formulir edit data inventaris</li> <li>6. Petugas mengubah informasi data inventaris yang ingin diubah</li> <li>7. Petugas memilih tombol simpan untuk menyimpan perubahan</li> <li>8. Sistem melakukan permintaan untuk menyimpan data inventaris yang baru diubah</li> <li>9. Sistem menampilkan halaman data inventaris setelah diperbarui</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman data inventaris setelah diperbarui
<b>Skenario Alternatif</b>	-

#### 4.9.2 Use Case Scenario Anggota

Tabel 4.35 Use Case Scenario Login Pada Sisi Anggota

<b>Use Case</b>	<i>Login</i>
<b>Aktor</b>	Anggota
<b>Deskripsi</b>	Anggota melakukan login untuk dapat menggunakan sistem
<b>Kondisi Awal</b>	Anggota ingin menggunakan sistem dan belum melakukan login
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Anggota memasukkan <i>username</i> dan <i>password</i></li> <li>2. Sistem memeriksa <i>valid</i> atau tidaknya data input</li> <li>3. Sistem menampilkan halaman utama dari sistem</li> </ol>
<b>Kondisi Akhir</b>	Anggota telah <i>login</i> dan siap berinteraksi dengan sistem
<b>Skenario Alternatif</b>	<ol style="list-style-type: none"> <li>1. Anggota memasukkan <i>username</i> dan <i>password</i></li> <li>2. Sistem memeriksa <i>valid</i> atau tidaknya data input</li> <li>3. Jika tidak <i>valid</i>, sistem menampilkan pesan data tidak valid</li> <li>4. Sistem menampilkan halaman <i>login</i></li> </ol>

Tabel 4.36 Use Case Scenario Menambah Data Tamu Pada Sisi Anggota

<b>Use Case</b>	Menambah data tamu
-----------------	--------------------

<b>Aktor</b>	Anggota
<b>Deskripsi</b>	Anggota dapat menambah data tamu yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Anggota telah mengakses halaman menu absensi tamu
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Anggota memilih menu Tamu</li> <li>2. Sistem melakukan permintaan halaman absensi tamu</li> <li>3. Sistem menampilkan formulir tambah data tamu</li> <li>4. Anggota mengisi formulir data tamu</li> <li>5. Anggota mensubmit tombol simpan</li> <li>6. Sistem melakukan permintaan untuk menyimpan data tamu ke dalam <i>database</i></li> <li>7. Sistem menampilkan halaman absensi tamu</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman absensi tamu
<b>Skenario Alternatif</b>	-

**Tabel 4.37 Use Case Scenario Mencari Buku Pada Sisi Anggota**

<b>Use Case</b>	Mencari buku
<b>Aktor</b>	Anggota
<b>Deskripsi</b>	Anggota dapat mencari buku yang diinginkan
<b>Kondisi Awal</b>	Anggota telah mengakses halaman menu pencarian buku
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Anggota memilih menu Cari Buku</li> <li>2. Sistem melakukan permintaan halaman pencarian buku</li> <li>3. Sistem menampilkan formulir cari buku</li> <li>4. Anggota mengisi formulir informasi buku yang diinginkan</li> <li>5. Anggota memilih tombol cari buku</li> <li>6. Sistem melakukan permintaan untuk mencari data koleksi buku pada <i>database</i></li> <li>7. Sistem menampilkan halaman informasi data buku yang diinginkan</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman informasi data buku yang diinginkan
<b>Skenario Alternatif</b>	<ol style="list-style-type: none"> <li>1. Anggota memilih menu Cari Buku</li> <li>2. Sistem melakukan permintaan halaman pencarian buku</li> <li>3. Sistem menampilkan formulir cari buku</li> <li>4. Anggota mengisi formulir informasi buku yang diinginkan</li> <li>5. Anggota memilih tombol cari buku</li> <li>6. Sistem melakukan permintaan untuk mencari data</li> </ol>

	<p>koleksi buku pada <i>database</i></p> <p>7. Jika data tidak ditemukan, sistem menampilkan pesan data tidak ditemukan</p>
--	---

**Tabel 4.38 Use Case Scenario Memperbarui Data Anggota Pada Sisi Anggota**

<b>Use Case</b>	Memperbarui data anggota
<b>Aktor</b>	Anggota
<b>Deskripsi</b>	Anggota dapat memperbarui data anggota yang ada pada <i>database</i>
<b>Kondisi Awal</b>	Anggota telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Anggota memilih menu Pengaturan Akun</li> <li>2. Sistem melakukan permintaan halaman pengaturan akun anggota perpustakaan</li> <li>3. Sistem menampilkan halaman pengaturan akun anggota</li> <li>4. Anggota mengubah informasi data anggota yang ingin diubah dengan informasi yang berbeda</li> <li>5. Anggota memilih tombol simpan untuk menyimpan perubahan</li> <li>6. Sistem melakukan permintaan untuk menyimpan data anggota yang baru diubah</li> <li>7. Sistem menampilkan data anggota setelah diperbarui pada halaman pengaturan akun anggota</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan data anggota setelah diperbarui pada halaman pengaturan akun anggota
<b>Skenario Alternatif</b>	<ol style="list-style-type: none"> <li>1. Anggota memilih menu Pengaturan Akun</li> <li>2. Sistem melakukan permintaan halaman pengaturan akun anggota perpustakaan</li> <li>3. Sistem menampilkan halaman pengaturan akun anggota</li> <li>4. Anggota mengubah <i>username</i> anggota dengan <i>username</i> yang sudah terdaftar di dalam <i>database</i></li> <li>5. Anggota memilih tombol simpan untuk menyimpan perubahan</li> <li>6. Sistem memeriksa ketersediaan <i>username</i> di dalam <i>database</i></li> <li>7. Jika <i>username</i> sudah terdaftar di dalam <i>database</i>, maka sistem akan memberikan peringatan untuk mengubah <i>username</i> lain yang berbeda</li> </ol>

**Tabel 4.39 Use Case Scenario Melihat Riwayat Peminjaman**

<b>Use Case</b>	Melihat data riwayat peminjaman
-----------------	---------------------------------

<b>Aktor</b>	Anggota
<b>Deskripsi</b>	Anggota dapat melihat data riwayat peminjaman buku yang pernah dilakukan
<b>Kondisi Awal</b>	Anggota telah <i>login</i> ke sistem
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Anggota memilih menu Daftar Peminjaman</li> <li>2. Sistem melakukan permintaan halaman data riwayat peminjaman</li> <li>3. Sistem membaca data riwayat peminjaman yang terdapat pada <i>database</i></li> <li>4. Sistem menampilkan halaman data riwayat peminjaman</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman data riwayat peminjaman
<b>Skenario Alternatif</b>	-

#### 4.9.3 Use Case Scenario Pengunjung

Tabel 4.40 Use Case Scenario Menambah Data Anggota Pada Sisi Pengunjung

<b>Use Case</b>	Menambah data anggota
<b>Aktor</b>	Pengunjung
<b>Deskripsi</b>	Pengunjung dapat melakukan pendaftaran melalui sistem dengan menambahkan data anggota pada <i>database</i>
<b>Kondisi Awal</b>	Pengunjung telah mengakses halaman menu daftar anggota
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Anggota memilih menu daftar anggota</li> <li>2. Sistem melakukan permintaan halaman daftar anggota</li> <li>3. Sistem menampilkan formulir pendaftaran anggota</li> <li>4. Anggota mengisi formulir pendaftaran</li> <li>5. Anggota mensubmit tombol daftar anggota</li> <li>6. Sistem melakukan permintaan untuk menyimpan data anggota baru ke dalam <i>database</i></li> <li>7. Sistem menampilkan halaman pendaftaran anggota</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman pendaftaran anggota
<b>Skenario Alternatif</b>	-

Tabel 4.41 Use Case Scenario Menambah Data Tamu Pada Sisi Pengunjung

<b>Use Case</b>	Menambah data tamu
<b>Aktor</b>	Pengunjung
<b>Deskripsi</b>	Pengunjung dapat menambah data tamu yang ada pada

	<i>database</i>
<b>Kondisi Awal</b>	Pengunjung telah mengakses halaman menu absensi tamu
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Pengunjung memilih menu Tamu</li> <li>2. Sistem melakukan permintaan halaman absensi tamu</li> <li>3. Sistem menampilkan formulir tambah data tamu</li> <li>4. Pengunjung mengisi formulir data tamu</li> <li>5. Pengunjung mensubmit tombol simpan</li> <li>6. Sistem melakukan permintaan untuk menyimpan data tamu ke dalam <i>database</i></li> <li>7. Sistem menampilkan halaman absensi tamu</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman absensi tamu
<b>Skenario Alternatif</b>	-

**Tabel 4.42 Use Case Scenario Mencari Buku Pada Sisi Pengunjung**

<b>Use Case</b>	Mencari buku
<b>Aktor</b>	Pengunjung
<b>Deskripsi</b>	Pengunjung dapat mencari buku yang diinginkan
<b>Kondisi Awal</b>	Pengunjung telah mengakses halaman menu pencarian buku
<b>Skenario Utama</b>	<ol style="list-style-type: none"> <li>1. Pengunjung memilih menu Cari Buku</li> <li>2. Sistem melakukan permintaan halaman pencarian buku</li> <li>3. Sistem menampilkan formulir cari buku</li> <li>4. Pengunjung mengisi formulir informasi buku yang diinginkan</li> <li>5. Pengunjung memilih tombol cari buku</li> <li>6. Sistem melakukan permintaan untuk mencari data koleksi buku pada <i>database</i></li> <li>7. Sistem menampilkan halaman informasi data buku yang diinginkan</li> </ol>
<b>Kondisi Akhir</b>	Sistem menampilkan halaman informasi data buku yang diinginkan
<b>Skenario Alternatif</b>	<ol style="list-style-type: none"> <li>1. Pengunjung memilih menu Cari Buku</li> <li>2. Sistem melakukan permintaan halaman pencarian buku</li> <li>3. Sistem menampilkan formulir cari buku</li> <li>4. Pengunjung mengisi formulir informasi buku yang diinginkan</li> <li>5. Pengunjung memilih tombol cari buku</li> <li>6. Sistem melakukan permintaan untuk mencari data koleksi buku pada <i>database</i></li> </ol>

	7. Jika data tidak ditemukan, sistem menampilkan pesan data tidak ditemukan
--	---

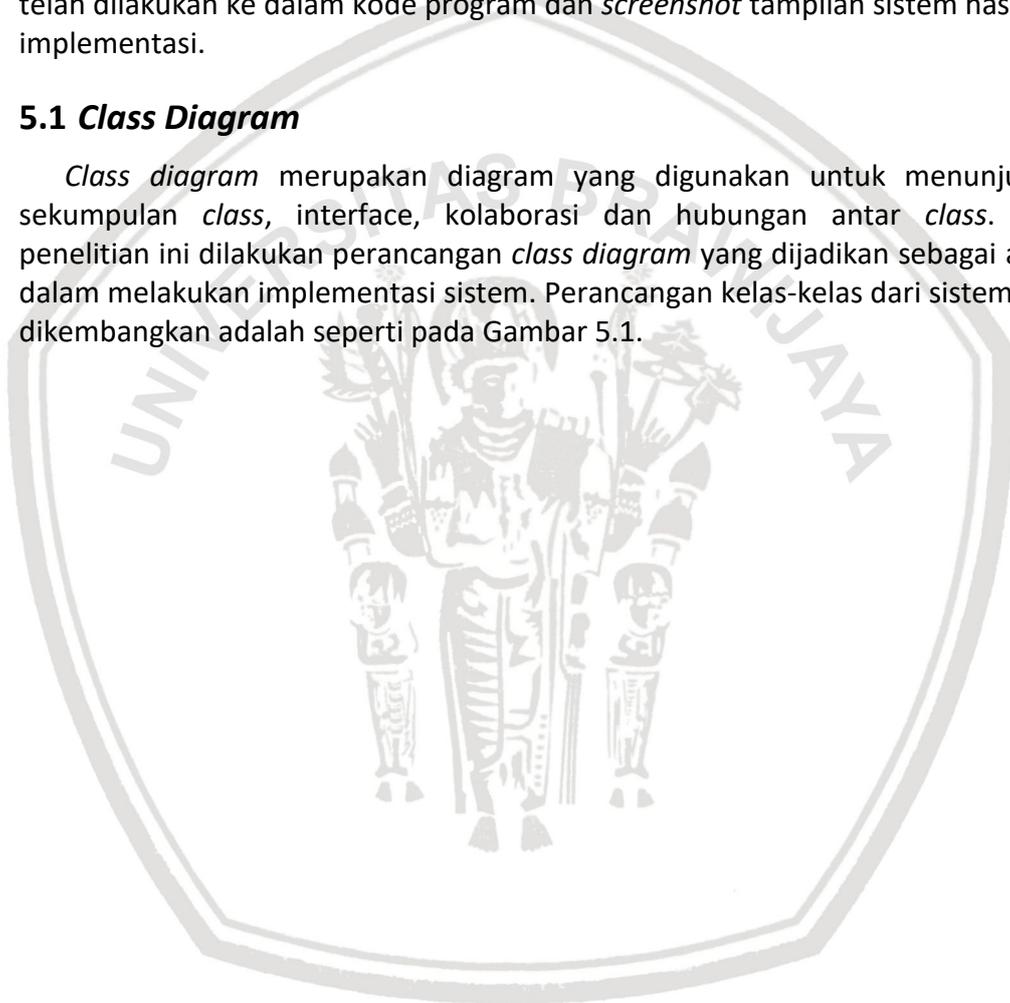


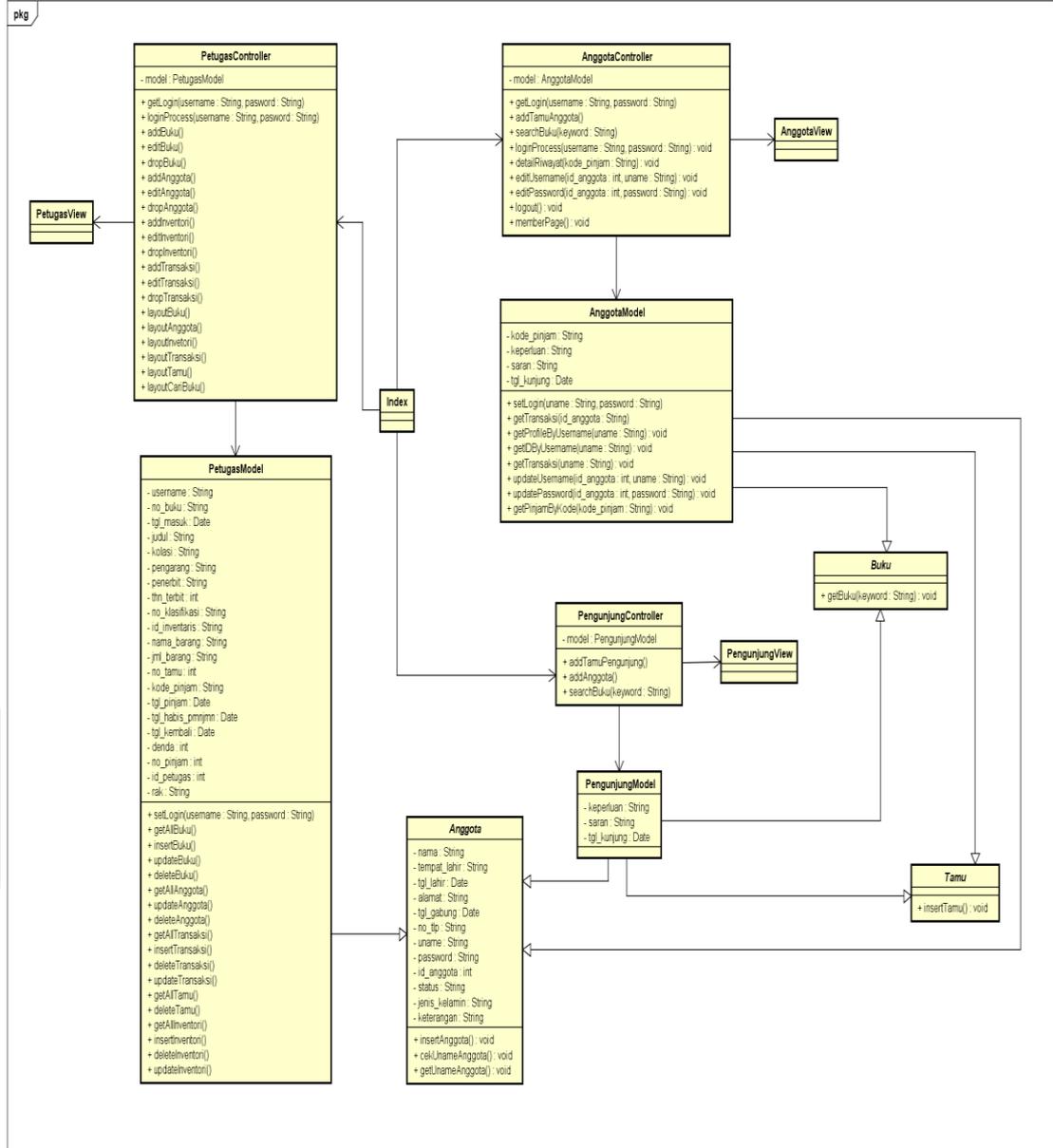
## BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini menjelaskan mengenai perancangan dan implementasi dari sistem informasi perpustakaan pada Perpustakaan Kecamatan Bungah yang akan dikembangkan sesuai dari hasil analisis kebutuhan pada bab sebelumnya. Pembahasan pertama menjelaskan perancangan sistem dengan pola MVC (Model View Controller) dan digambarkan dengan diagram UML diantaranya *Class Diagram* dan *Sequence Diagram*, serta perancangan model data yang digambarkan dengan menggunakan diagram PDM (*Physical Data Model*). Dan pembahasan selanjutnya adalah implementasi sistem sesuai perancangan yang telah dilakukan ke dalam kode program dan *screenshot* tampilan sistem hasil dari implementasi.

### 5.1 Class Diagram

*Class diagram* merupakan diagram yang digunakan untuk menunjukkan sekumpulan *class*, *interface*, kolaborasi dan hubungan antar *class*. Pada penelitian ini dilakukan perancangan *class diagram* yang dijadikan sebagai acuan dalam melakukan implementasi sistem. Perancangan kelas-kelas dari sistem yang dikembangkan adalah seperti pada Gambar 5.1.





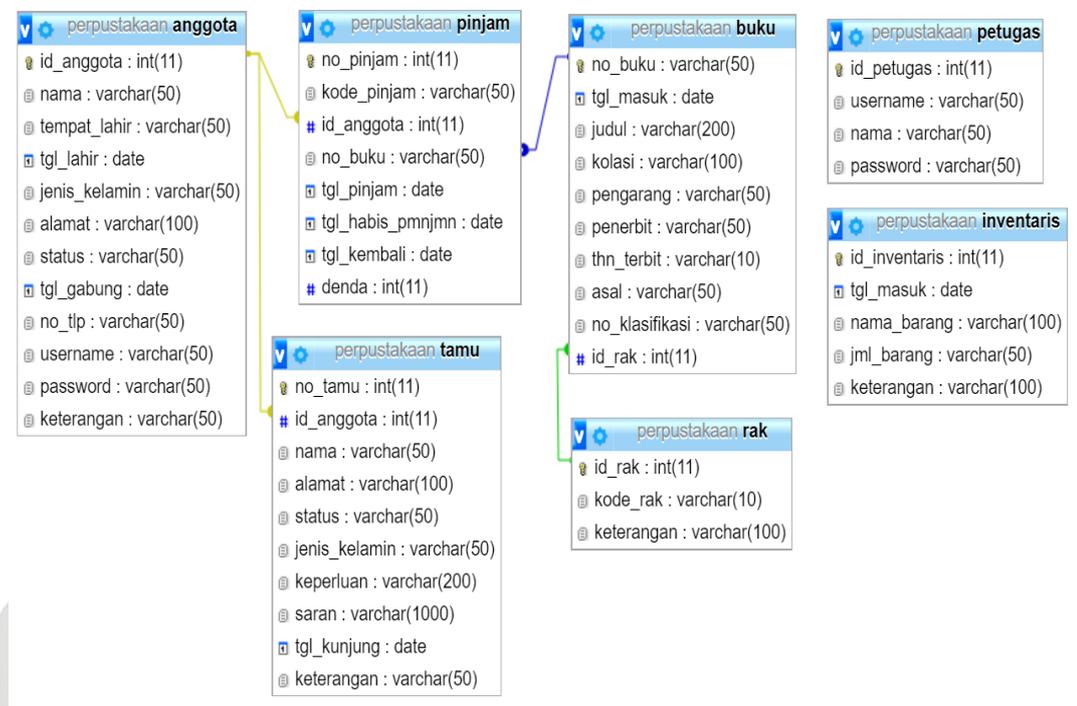
Gambar 5.1 Class Diagram

Gambar 5.1 merupakan perancangan *class diagram* sistem yang digambarkan dengan konsep MVC (Model Controller View) yang terdiri dari 3 kelas model, 3 kelas controller, dan view masing-masing aktor yaitu Petugas, Anggota, dan Pengunjung. Pada gambar di atas menjelaskan bagaimana masing-masing komponen MVC berkomunikasi yaitu antara PetugasController, PetugasModel, dan view Petugas, kemudian antara AnggotaController, AnggotaModel, dan view Anggota, dan yang terakhir antara PengunjungController, PengunjungModel, dan view Pengunjung. Selain itu terdapat juga 1 kelas *abstract* yang diturunkan pada kelas model yaitu kelas Anggota, dan terdapat 2 *interface* yaitu kelas Tamu dan Buku yang *diimplements* pada kelas model PengunjungModel dan AnggotaModel.



## 5.2 Physical Data Model Diagram

Perancangan *database* pada sistem yang dibangun ditunjukkan seperti Gambar 5.2.



**Gambar 5.2 Physical Data Model Diagram**

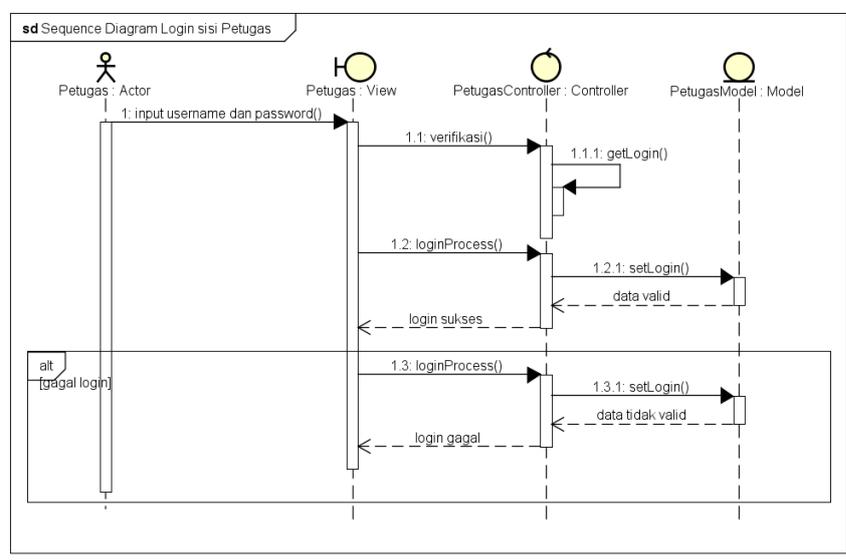
Gambar 5.2 merupakan perancangan model data berupa PDM (*Physical Data Model*) atau gambaran fisik *database* dari sistem yang akan dibuat. Terdapat lima dari tujuh tabel yang saling berhubungan yang membentuk empat relasi *one to many* antar dua tabel. Yang pertama hubungan antara tabel anggota dan tabel tamu dimana *primary key* di tabel anggota menjadi *foreign key* di tabel tamu. Yang kedua hubungan antara tabel buku dan tabel pinjam dimana *primary key* di tabel buku menjadi *foreign key* di tabel pinjam. Yang ketiga hubungan antara tabel anggota dan tabel pinjam dimana *primary key* di tabel anggota menjadi *foreign key* di tabel pinjam. Dan yang terakhir hubungan antara tabel rak dan tabel buku dimana *primary key* di tabel rak menjadi *foreign key* di tabel buku.



### 5.3 Sequence Diagram

#### 5.3.1 Sequence Diagram Petugas

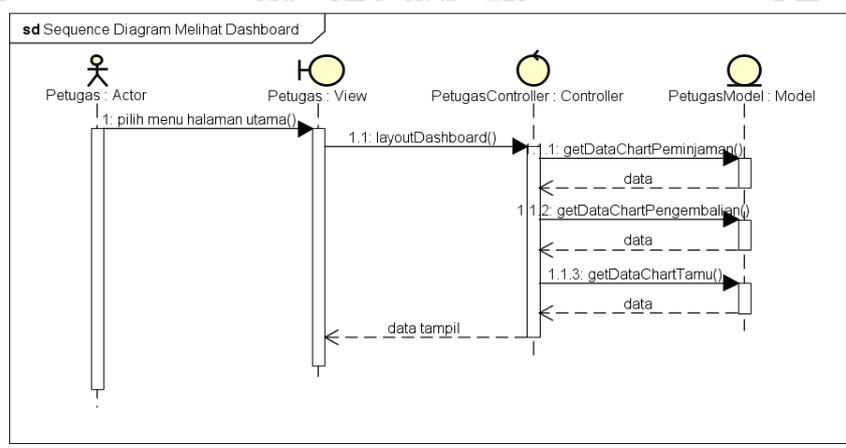
##### 1. Login Petugas



**Gambar 5.3 Sequence Diagram Login Pada Sisi Petugas**

Gambar 5.3 merupakan *sequence diagram* yang menjelaskan urutan proses untuk melakukan *login* ke sistem oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memasukkan *username* dan *password* pada *form* yang ditampilkan di *view*. Kemudian *controller* menjalankan fungsi *getLogin* untuk melakukan verifikasi data pengguna yang didalamnya memanggil fungsi *setLogin* pada *model*. Apabila data valid maka fungsi *loginProcess* pada *controller* dijalankan dan login berhasil, namun jika data tidak valid maka sistem akan menampilkan pesan bahwa data tidak valid dan login tidak berhasil.

##### 2. Melihat Dashboard



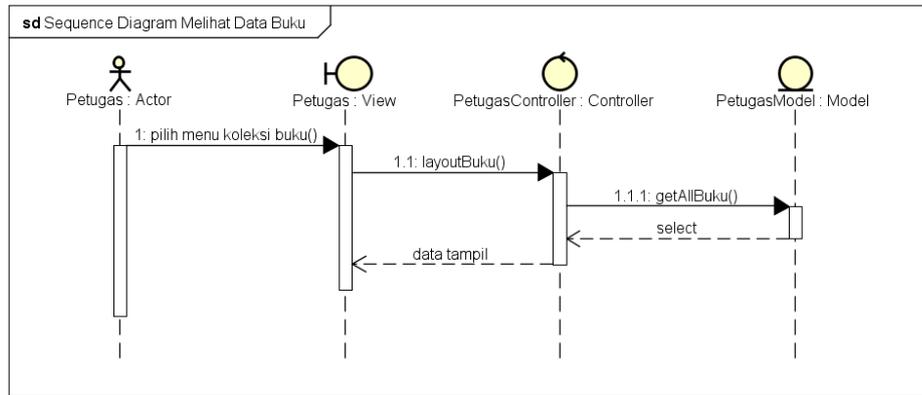
**Gambar 5.4 Sequence Diagram Melihat Dashboard**

Gambar 5.4 merupakan *sequence diagram* yang menjelaskan urutan proses untuk melihat *dashboard* oleh petugas. Proses dimulai dari petugas sebagai aktor



sistem memilih menu halaman utama untuk menampilkan halaman *dashboard* pada view. Selanjutnya controller memanggil fungsi `layoutDashboard` dan diteruskan dengan memanggil fungsi `getDataChartPeminjaman`, `getDataChartPengembalian` dan `getDataChartTamu` pada model untuk mendapatkan data *dashboard*. Setelah model membaca seluruh data *dashboard* pada *database*, data dikirim kembali ke fungsi `layoutDashboard` pada controller untuk ditampilkan di view.

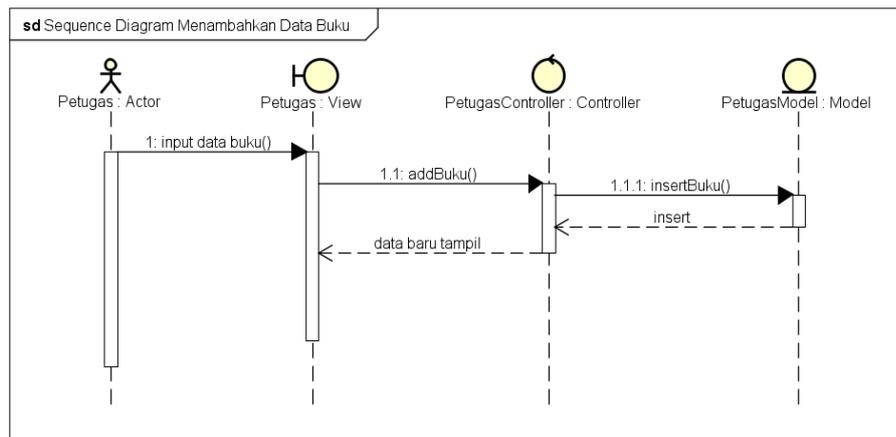
### 3. Melihat Data Buku



**Gambar 5.5 Sequence Diagram Melihat Data Buku**

Gambar 5.5 merupakan *sequence diagram* yang menjelaskan urutan proses untuk melihat data buku oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih menu koleksi buku untuk meminta halaman data koleksi buku pada view. Selanjutnya controller memanggil fungsi `layoutBuku` dan diteruskan dengan memanggil fungsi `getAllBuku` pada model untuk menampilkan seluruh data koleksi buku. Setelah model membaca seluruh data koleksi buku pada *database*, data dikirim kembali ke fungsi `layoutBuku` pada controller untuk ditampilkan di view.

### 4. Menambah Data Buku

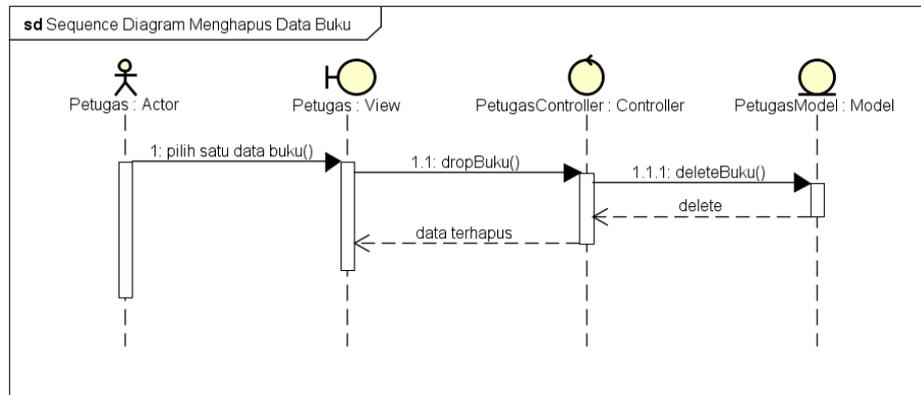


**Gambar 5.6 Sequence Diagram Menambahkan Data Buku**

Gambar 5.6 merupakan *sequence diagram* yang menjelaskan urutan proses untuk melihat data buku oleh petugas. Proses dimulai dari petugas sebagai aktor

sistem menginputkan data koleksi buku baru pada view. Selanjutnya controller memanggil fungsi `addBuku` dan diteruskan dengan memanggil fungsi `insertBuku` pada model untuk menyimpan data buku yang ditambahkan. Setelah model menyimpan data buku yang baru ditambahkan di *database*, selanjutnya controller mengirimkan kembali data untuk ditampilkan di view.

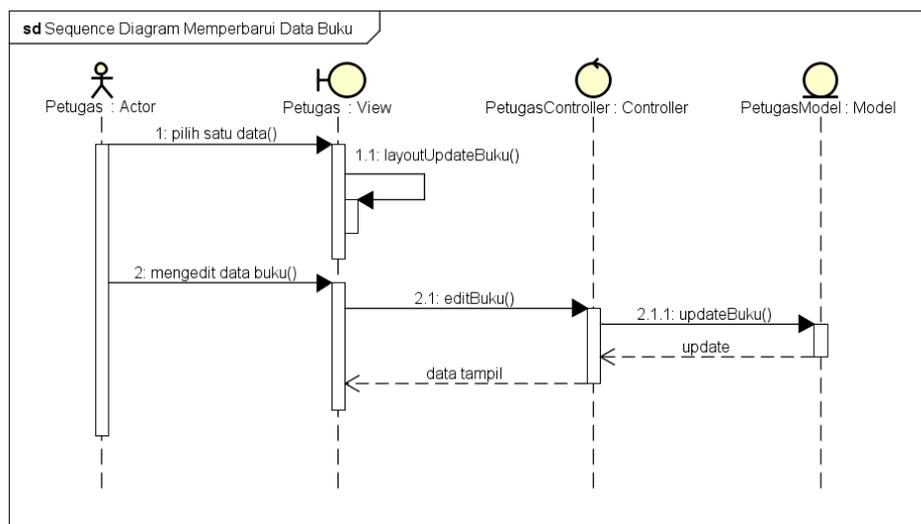
### 5. Menghapus Data Buku



**Gambar 5.7 Sequence Diagram Menghapus Data Buku**

Gambar 5.7 merupakan *sequence diagram* yang menjelaskan urutan proses untuk menghapus data buku oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih salah satu data koleksi buku yang ingin dihapus pada view. Selanjutnya controller memanggil fungsi `dropBuku` dan diteruskan dengan memanggil fungsi `deleteBuku` pada model untuk menghapus data buku yang dipilih. Setelah model menghapus data buku pada *database*, controller mengirimkan pesan untuk ditampilkan pada view bahwa data telah terhapus.

### 6. Memperbarui Data Buku



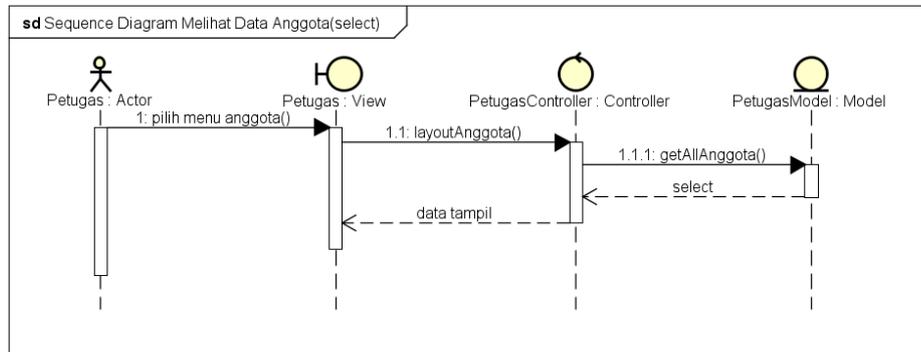
**Gambar 5.8 Sequence Diagram Memperbarui Data Buku**

Gambar 5.8 merupakan *sequence diagram* yang menjelaskan urutan proses untuk memperbarui data buku oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih salah satu data koleksi buku yang ingin diperbarui,



kemudian tampil layout form edit data buku. Petugas mengisi form edit untuk mengubah data buku. Selanjutnya controller memanggil fungsi `editBuku` dan diteruskan dengan memanggil fungsi `updateBuku` pada model untuk memperbarui data buku. Setelah model menyimpan data buku yang baru diubah pada *database*, controller mengirimkan data kembali untuk ditampilkan di view.

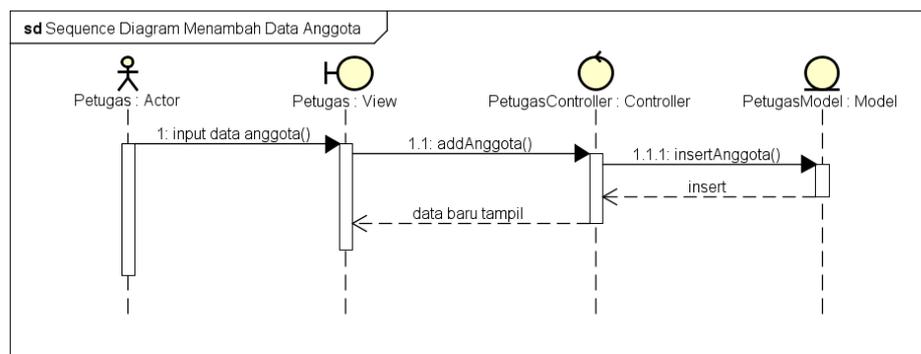
### 7. Melihat Data Anggota



**Gambar 5.9 Sequence Diagram Melihat Data Anggota**

Gambar 5.9 merupakan *sequence diagram* yang menjelaskan urutan proses untuk melihat data anggota oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih menu anggota untuk meminta halaman data anggota pada view. Selanjutnya controller memanggil fungsi `layoutAnggota` dan diteruskan dengan memanggil fungsi `getAllAnggota` pada model untuk menampilkan seluruh data anggota. Setelah model membaca seluruh data koleksi buku pada *database*, data dikirim kembali ke fungsi `layoutAnggota` pada controller untuk ditampilkan di view.

### 8. Menambah Data Anggota

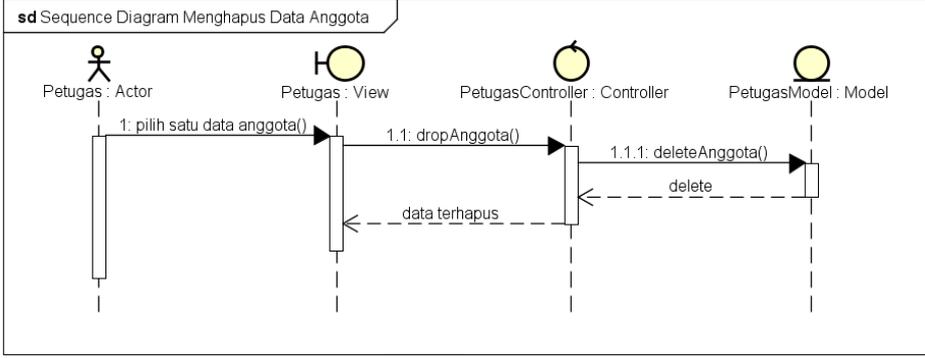


**Gambar 5.10 Sequence Diagram Menambah Data Anggota Pada Sisi Petugas**

Gambar 5.10 merupakan *sequence diagram* yang menjelaskan urutan proses untuk menambah data anggota oleh petugas. Proses dimulai dari petugas sebagai aktor sistem menginputkan data anggota baru pada view. Selanjutnya controller memanggil fungsi `addAnggota` dan diteruskan dengan memanggil fungsi `insertAnggota` pada model untuk menyimpan data buku yang baru ditambahkan. Setelah model menyimpan data buku yang baru ditambahkan di *database*, selanjutnya controller mengirimkan data kembali untuk ditampilkan di view.



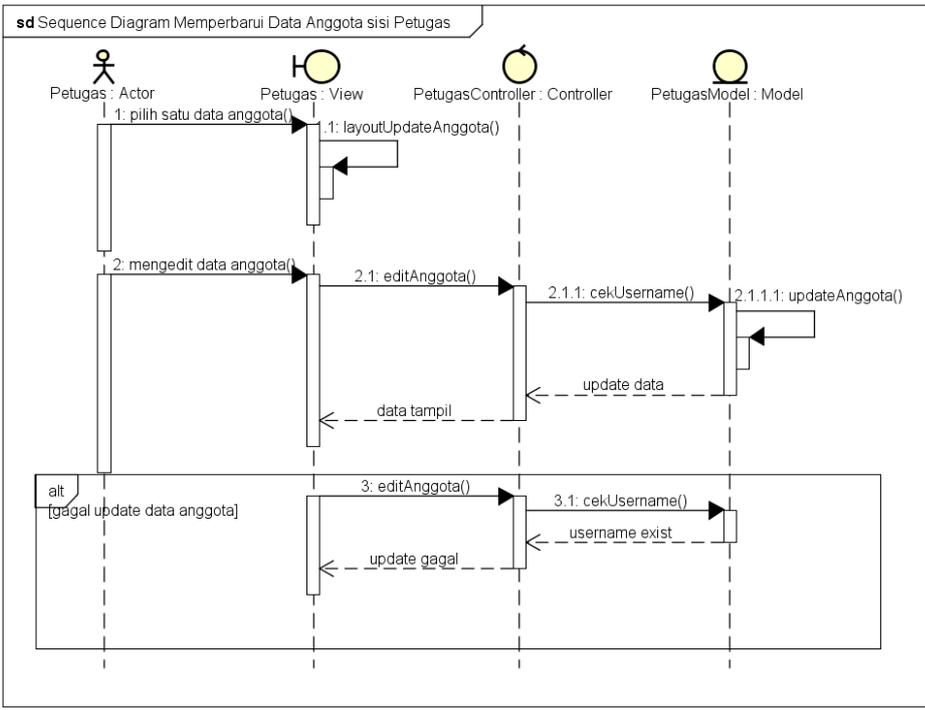
### 9. Menghapus Data Anggota



**Gambar 5.11 Sequence Diagram Menghapus Data Anggota**

Gambar 5.11 merupakan *sequence diagram* yang menjelaskan urutan proses untuk menghapus data anggota oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih salah satu data anggota yang ingin dihapus pada view. Selanjutnya controller memanggil fungsi `dropAnggota` dan diteruskan dengan memanggil fungsi `deleteAnggota` pada model untuk menghapus data anggota yang dipilih. Setelah model menghapus data anggota pada *database*, controller mengirimkan pesan untuk ditampilkan pada view bahwa data telah terhapus.

### 10. Memperbarui Data Anggota



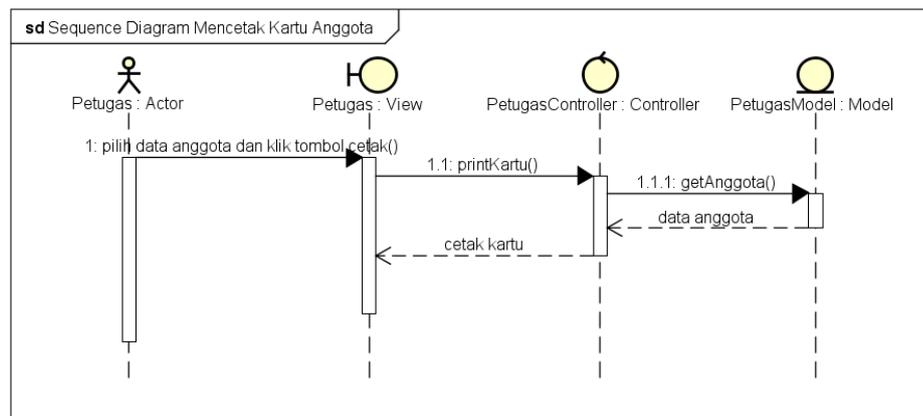
**Gambar 5.12 Sequence Diagram Memperbarui Data Anggota Pada Sisi Petugas**

Gambar 5.12 merupakan *sequence diagram* yang menjelaskan urutan proses untuk memperbarui data anggota oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih salah satu data anggota yang ingin diperbarui, kemudian tampil layout form edit data anggota. Aktor mengisi form edit untuk



mengubah data anggota. Selanjutnya controller memanggil fungsi `editAnggota` dan diteruskan dengan memanggil fungsi `cekUsername` untuk memeriksa apakah `username` yang dimasukkan sudah ada di dalam `database` atau belum, apabila `username` tidak ada dalam `database` maka fungsi `updateAnggota` pada model dipanggil untuk memperbarui data anggota. Setelah model menyimpan data anggota yang baru diubah pada `database`, maka controller mengirimkan data tersebut pada view untuk ditampilkan. Namun apabila `username` sudah ada di dalam `database`, maka data anggota tidak berhasil diubah dan sistem akan memberikan peringatan kepada aktor bahwa data anggota tidak berhasil diubah.

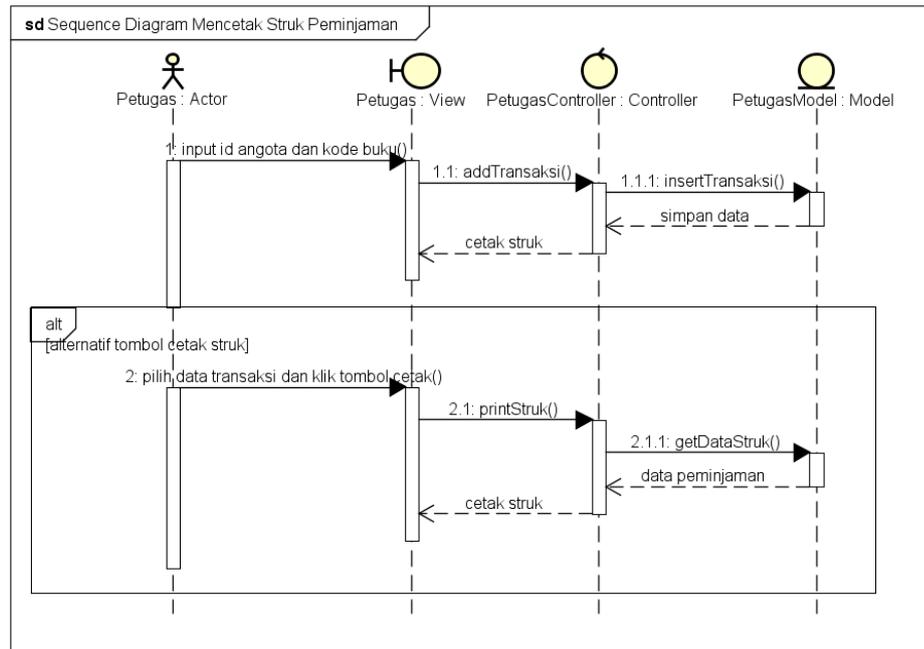
### 11. Mencetak Kartu Anggota



**Gambar 5.13 Sequence Diagram Mencetak Kartu Anggota**

Gambar 5.13 merupakan *sequence diagram* yang menjelaskan urutan proses untuk mencetak kartu anggota oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih salah satu data anggota yang ingin dicetak kartu lalu petugas memilih tombol cetak kartu untuk memulai proses percetakan kartu anggota. Selanjutnya controller memanggil fungsi `printKartu` dan diteruskan dengan memanggil fungsi `getAnggota` pada model untuk mendapatkan data anggota yang ingin dicetak. Setelah model mendapatkan data anggota pada `database`, controller mengirimkan data tersebut pada view untuk menampilkan detail data anggota yang akan dicetak dan dilakukan pencetakan kartu anggota.

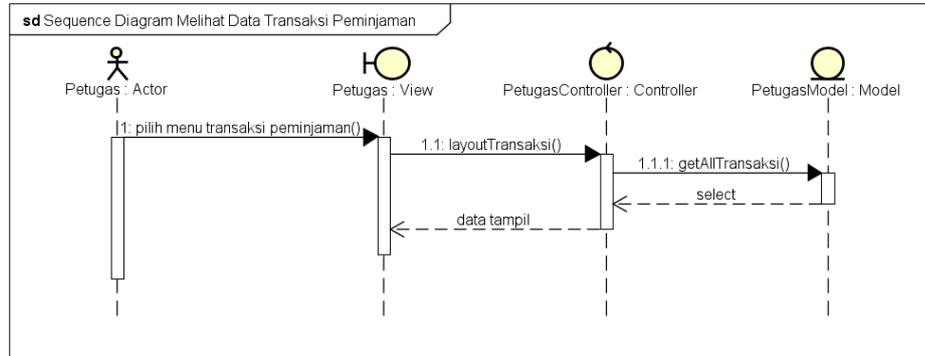
## 12. Mencetak Struk Peminjaman



**Gambar 5.14 Sequence Diagram Mencetak Struk Peminjaman**

Gambar 5.14 merupakan *sequence diagram* yang menjelaskan urutan proses untuk mencetak struk peminjaman oleh petugas. Proses dimulai dari petugas sebagai aktor sistem menginputkan data transaksi baru pada view. Selanjutnya controller memanggil fungsi `addTransaksi` dan diteruskan dengan memanggil fungsi `insertTransaksi` pada model untuk menyimpan data buku yang baru ditambahkan. Setelah model menyimpan data buku yang baru ditambahkan di *database*, selanjutnya controller mengirimkan data tersebut ke view untuk ditampilkan dan struk peminjaman siap untuk dicetak. Alternatif untuk mencetak struk peminjaman lainnya yaitu pertama, memilih salah satu data transaksi peminjaman yang ingin dicetak struk lalu petugas memilih tombol cetak struk untuk memulai proses percetakan struk peminjaman. Kemudian controller memanggil fungsi `printStruk` dan diteruskan dengan memanggil fungsi `getDataStruk` pada model untuk mendapatkan data peminjaman yang akan dicetak. Setelah model mendapatkan data peminjaman pada *database*, maka controller mengirimkan data tersebut pada view untuk menampilkan detail data struk yang akan dicetak dan dilakukan pencetakan struk peminjaman.

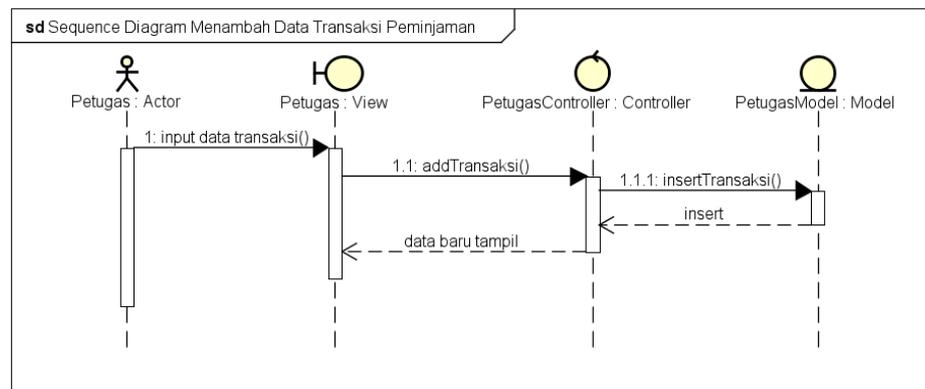
### 13. Melihat Data Transaksi Peminjaman



**Gambar 5.15 Sequence Diagram Melihat Data Transaksi Peminjaman**

Gambar 5.15 merupakan *sequence diagram* yang menjelaskan urutan proses untuk melihat data transaksi oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih menu transaksi peminjaman untuk meminta halaman data transaksi pada view. Selanjutnya controller memanggil fungsi `layoutTransaksi` dan diteruskan dengan memanggil fungsi `getAllTransaksi` pada model untuk menampilkan seluruh data transaksi peminjaman. Setelah model membaca seluruh data koleksi buku pada *database*, data dikirim kembali ke fungsi `layoutTransaksi` pada controller untuk ditampilkan di view.

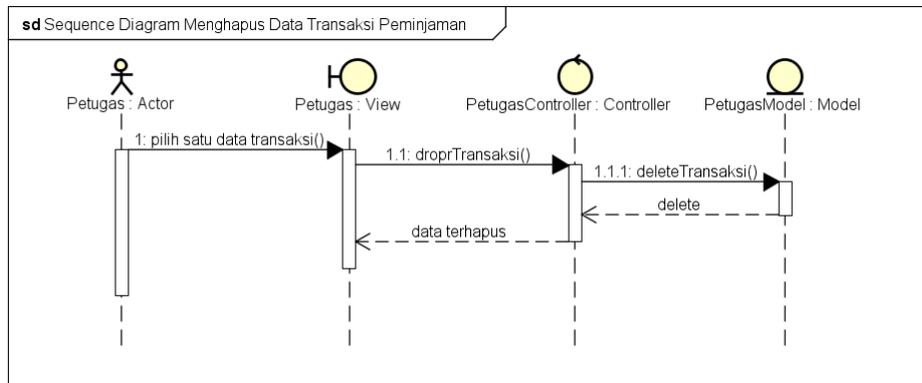
### 14. Menambah Data Transaksi Peminjaman



**Gambar 5.16 Sequence Diagram Menambah Data Transaksi Peminjaman**

Gambar 5.16 merupakan *sequence diagram* yang menjelaskan urutan proses untuk menambah data transaksi peminjaman oleh petugas. Proses dimulai dari petugas sebagai aktor sistem menginputkan data transaksi baru pada view. Selanjutnya controller memanggil fungsi `addTransaksi` dan diteruskan dengan memanggil fungsi `insertTransaksi` pada model untuk menyimpan data buku yang baru ditambahkan. Setelah model menyimpan data buku yang baru ditambahkan di *database*, selanjutnya controller mengirimkan data kembali untuk ditampilkan di view.

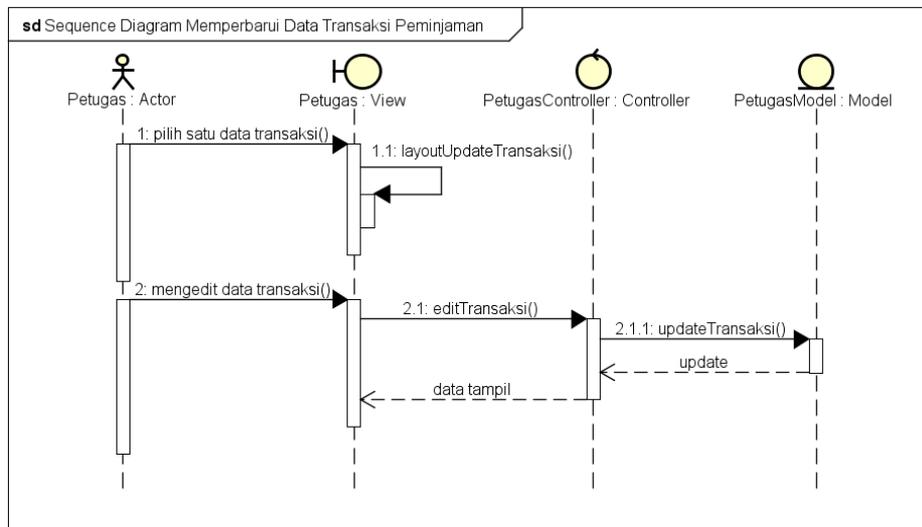
### 15. Menghapus Data Transaksi Peminjaman



**Gambar 5.17 Sequence Diagram Menghapus Data Transaksi Peminjaman**

Gambar 5.17 merupakan *sequence diagram* yang menjelaskan urutan proses untuk menghapus data transaksi peminjaman oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih salah satu data transaksi yang ingin dihapus pada view. Selanjutnya controller memanggil fungsi `dropTransaksi` dan diteruskan dengan memanggil fungsi `deleteTransaksi` pada model untuk menghapus data transaksi yang dipilih. Setelah model menghapus data transaksi pada *database*, controller mengirimkan pesan untuk ditampilkan pada view bahwa data telah terhapus.

### 16. Memperbarui Data Transaksi Peminjaman



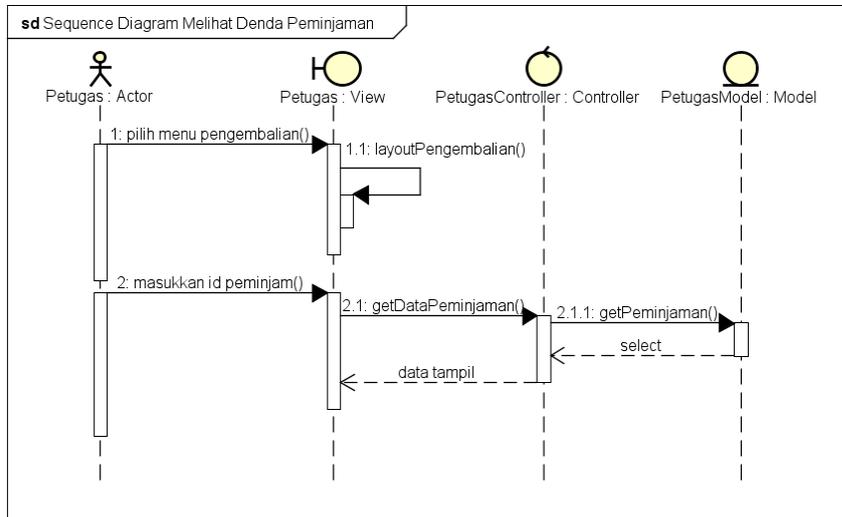
**Gambar 5.18 Sequence Diagram Memperbarui Data Transaksi Peminjaman**

Gambar 5.18 merupakan *sequence diagram* yang menjelaskan urutan proses untuk memperbarui data transaksi peminjaman oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih salah satu data transaksi peminjaman yang ingin diperbarui, kemudian tampil layout form edit data transaksi. Petugas mengisi form edit untuk mengubah data transaksi. Selanjutnya controller memanggil fungsi `editTransaksi` dan diteruskan dengan memanggil fungsi `updateTransaksi` pada model untuk memperbarui data transaksi peminjaman.



Setelah model menyimpan data transaksi yang baru diubah pada *database*, controller mengirimkan data kembali untuk ditampilkan di view.

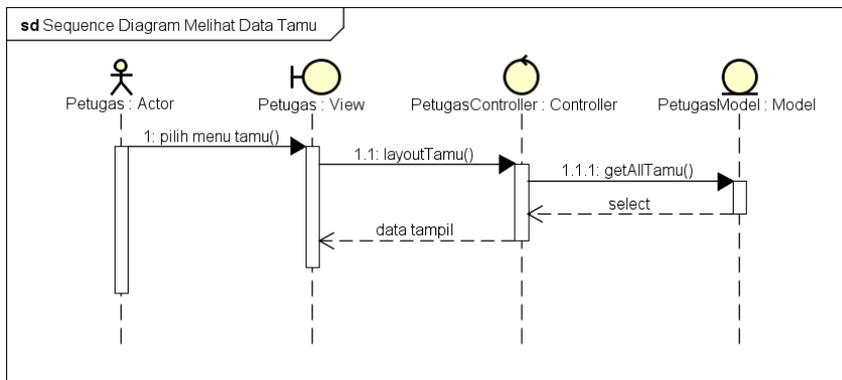
### 17. Melihat Denda Peminjaman



**Gambar 5.19 Sequence Diagram Melihat Denda Peminjaman**

Gambar 5.19 merupakan *sequence diagram* yang menjelaskan urutan proses untuk melihat denda peminjaman oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih menu pengembalian untuk meminta halaman menu pengembalian buku pada view. Selanjutnya controller memanggil fungsi `layoutPengembalian` yang menampilkan form untuk mengisi id peminjam pada view. Setelah aktor memasukkan id peminjam, controller menjalankan fungsi `getDataPeminjaman` dan diteruskan dengan memanggil fungsi `getPeminjaman` pada model untuk menampilkan informasi peminjaman berikut denda peminjaman sesuai id peminjam yang telah dimasukkan. Setelah model membaca data peminjaman pada *database*, controller akan mengirimkan data kembali untuk ditampilkan di view.

### 18. Melihat Data Tamu

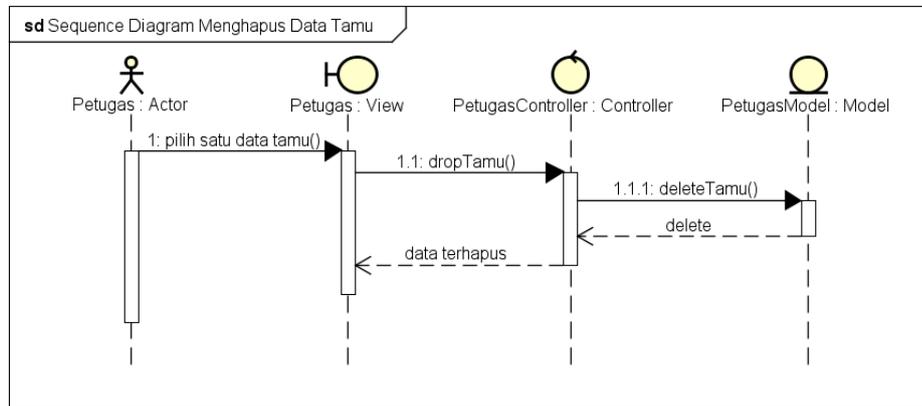


**Gambar 5.20 Sequence Diagram Melihat Data Tamu**

Gambar 5.20 merupakan *sequence diagram* yang menjelaskan urutan proses untuk melihat data tamu oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih menu tamu untuk meminta halaman data tamu pada view.

Selanjutnya controller memanggil fungsi `layoutTamu` dan diteruskan dengan memanggil fungsi `getAllTamu` pada model untuk menampilkan seluruh data tamu. Setelah itu data dikirim kembali ke fungsi `layoutTamu` pada controller untuk ditampilkan di view.

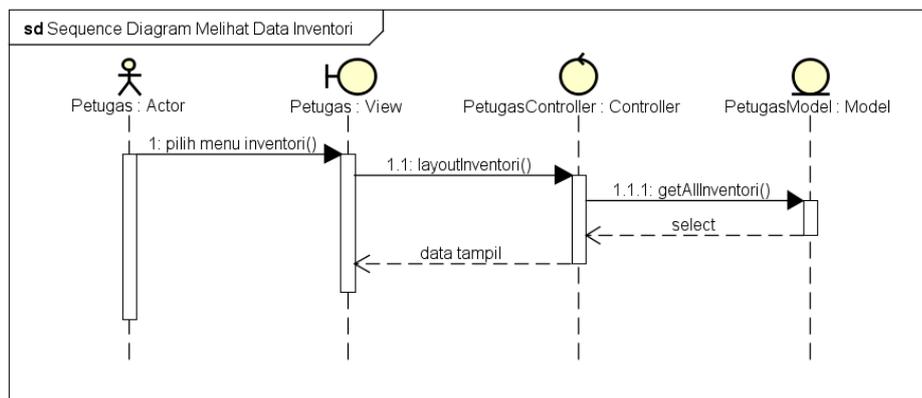
### 19. Menghapus Data Tamu



**Gambar 5.21 Sequence Diagram Menghapus Data Tamu**

Gambar 5.21 merupakan *sequence diagram* yang menjelaskan urutan proses untuk menghapus data tamu oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih salah satu data tamu yang ingin dihapus pada view. Selanjutnya controller memanggil fungsi `dropTamu` dan diteruskan dengan memanggil fungsi `deleteTamu` pada model untuk menghapus data tamu yang dipilih. Setelah model menghapus data tamu pada *database*, controller mengirimkan pesan untuk ditampilkan pada view bahwa data telah terhapus.

### 20. Melihat Data Inventaris



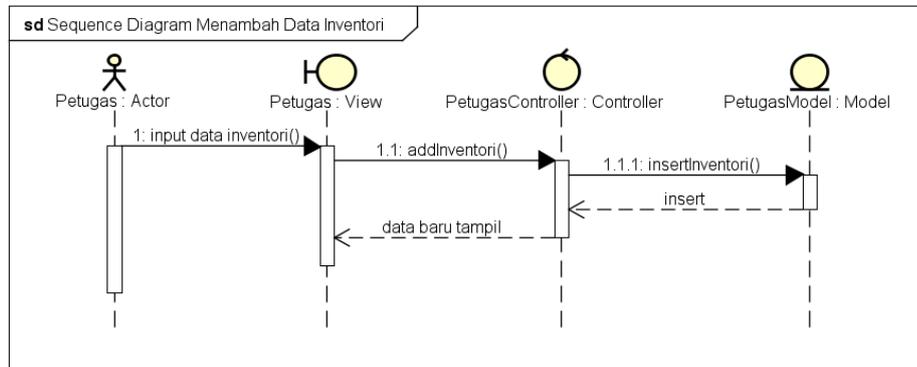
**Gambar 5.22 Sequence Diagram Melihat Data Inventaris**

Gambar 5.22 merupakan *sequence diagram* yang menjelaskan urutan proses untuk melihat data inventaris oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih menu inventaris untuk meminta halaman data inventaris pada view. Selanjutnya controller memanggil fungsi `layoutInventori` dan diteruskan dengan memanggil fungsi `getAllInventori` pada model untuk menampilkan seluruh data inventaris. Setelah model membaca seluruh data



inventaris pada *database*, data dikirim kembali ke fungsi `layoutInventori` pada controller untuk ditampilkan di view.

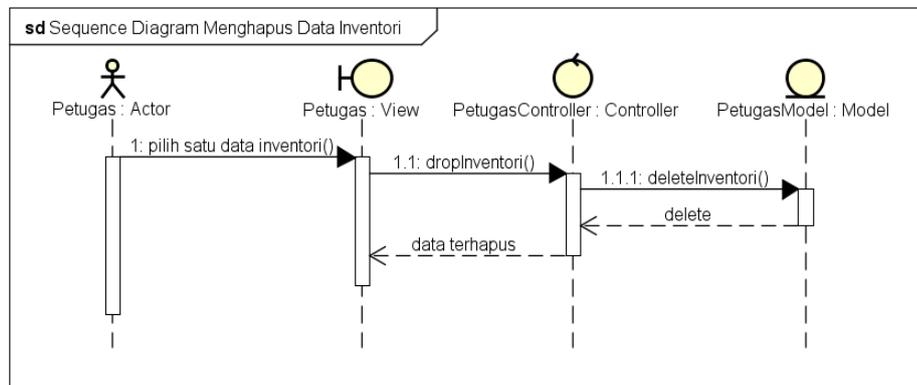
### 21. Menambah Data Inventaris



**Gambar 5.23 Sequence Diagram Menambah Data Inventaris**

Gambar 5.23 merupakan *sequence diagram* yang menjelaskan urutan proses untuk menambah data inventaris oleh petugas. Proses dimulai dari petugas sebagai aktor sistem menginputkan data inventaris baru pada view. Selanjutnya controller memanggil fungsi `addInventori` dan diteruskan dengan memanggil fungsi `insertInventori` pada model untuk menyimpan data buku yang ditambahkan ke dalam *database*. Setelah itu controller mengirimkan data kembali untuk ditampilkan di view.

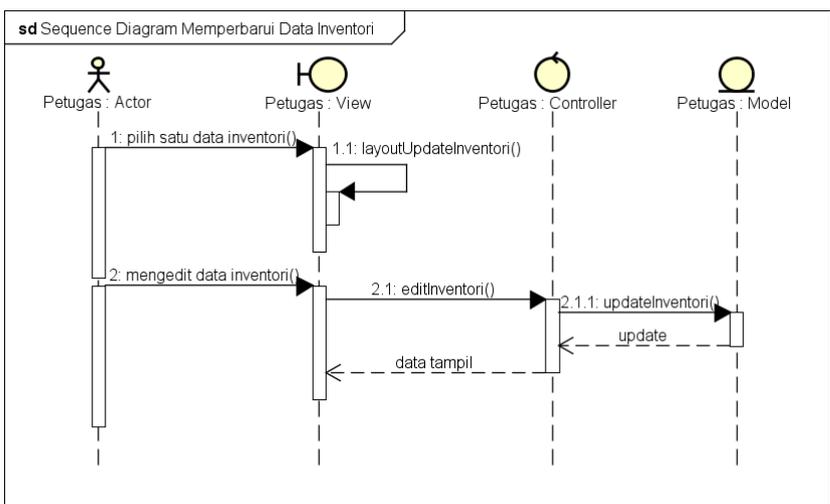
### 22. Menghapus Data Inventaris



**Gambar 5.24 Sequence Diagram Menghapus data Inventaris**

Gambar 5.24 merupakan *sequence diagram* yang menjelaskan urutan proses untuk menghapus data inventaris oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih salah satu data inventaris yang ingin dihapus pada view. Selanjutnya controller memanggil fungsi `dropInventori` dan diteruskan dengan memanggil fungsi `deleteInventori` pada model untuk menghapus data inventaris yang dipilih. Setelah itu controller mengirimkan pesan untuk ditampilkan pada view bahwa data telah terhapus.

23. Memperbarui Data Inventaris

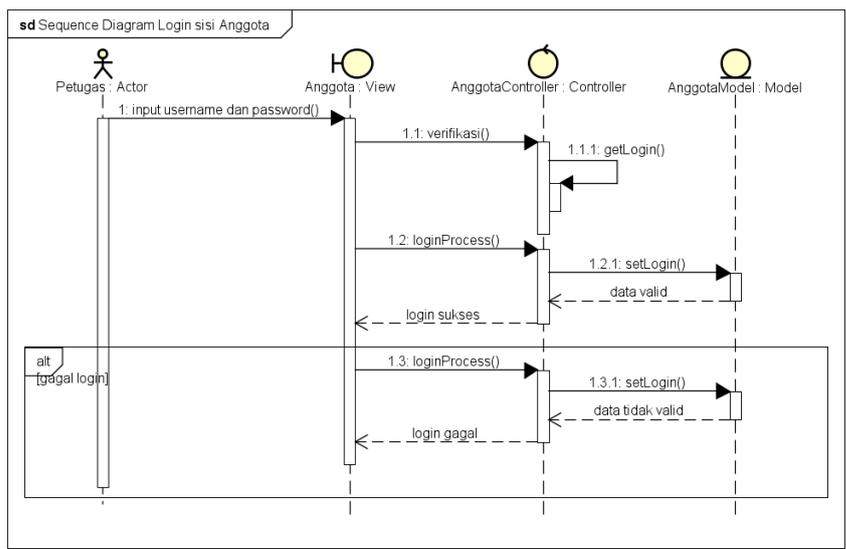


Gambar 5.25 Sequence Diagram Memperbarui Data Inventaris

Gambar 5.25 merupakan *sequence diagram* yang menjelaskan urutan proses untuk memperbarui data inventaris oleh petugas. Proses dimulai dari petugas sebagai aktor sistem memilih salah satu data inventaris yang ingin diperbarui, kemudian tampil layout form edit data inventaris. Petugas mengisi form edit untuk mengubah data inventaris. Selanjutnya controller memanggil fungsi `changeInventori` dan diteruskan dengan memanggil fungsi `updateInventori` pada model untuk memperbarui data inventaris. Setelah model menyimpan data inventaris yang baru diubah pada *database*, controller mengirimkan data kembali untuk ditampilkan di view.

5.3.2 Sequence Diagram Anggota

1. Login



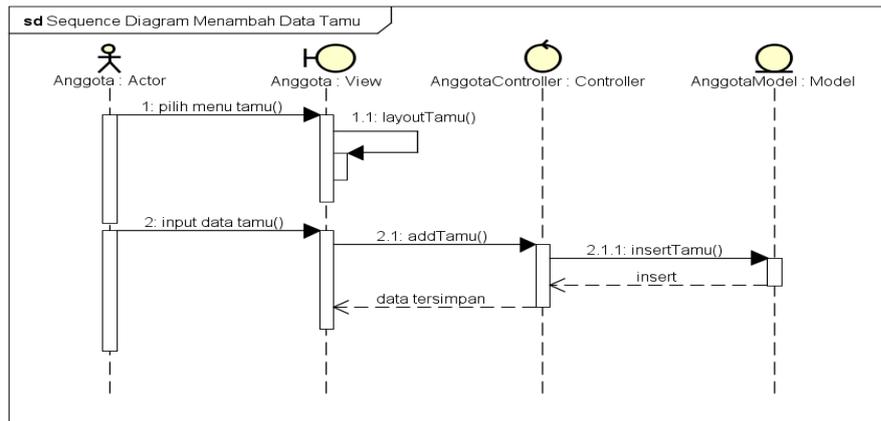
Gambar 5.26 Sequence Diagram Login Pada Sisi Anggota

Gambar 5.26 merupakan *sequence diagram* yang menjelaskan urutan proses untuk melakukan *login* ke sistem oleh anggota. Proses dimulai dari petugas



sebagai aktor sistem memasukkan username dan password pada *form* yang ditampilkan di view. Selanjutnya controller memanggil fungsi `getLogin` untuk melakukan verifikasi data pengguna yang didalamnya memanggil fungsi `setLogin` pada model. Apabila data valid maka fungsi `loginProcess` pada controller dijalankan dan login berhasil, namun jika data tidak valid maka sistem akan menampilkan pesan bahwa data tidak valid dan login tidak berhasil.

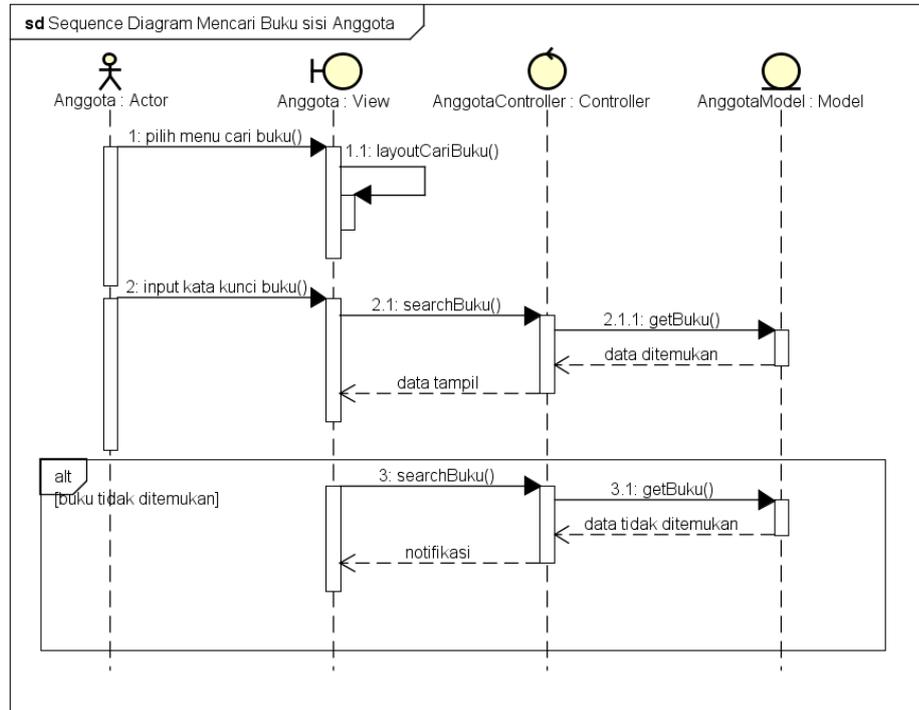
## 2. Menambah Data Tamu



**Gambar 5.27 Sequence Diagram Menambah Data Tamu Pada Sisi Anggota**

Gambar 5.27 merupakan *sequence diagram* yang menjelaskan urutan proses untuk menambah data tamu oleh anggota. Proses dimulai dari anggota sebagai aktor sistem memilih menu tamu untuk menampilkan halaman form tamu. Selanjutnya aktor menginputkan data tamu pada view. Selanjutnya controller memanggil fungsi `addTamu` dan diteruskan dengan memanggil fungsi `insertTamu` pada model untuk menyimpan data tamu yang baru ditambahkan. Setelah itu controller mengirimkan informasi ke view bahwa data baru telah tersimpan.

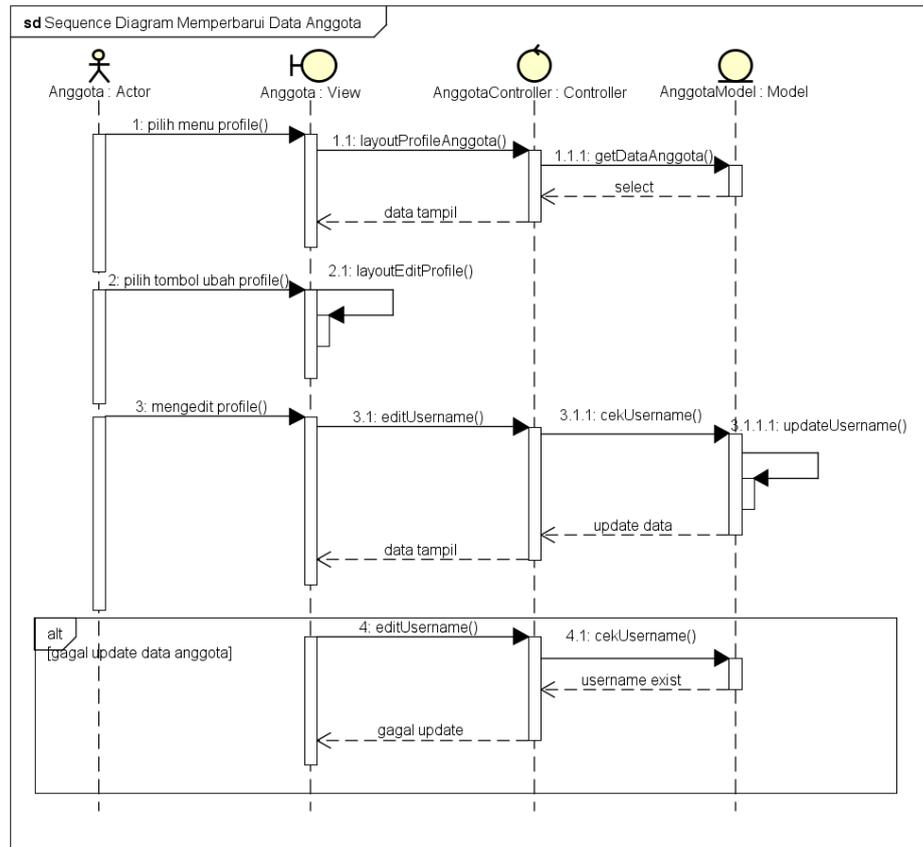
### 3. Mencari Buku



**Gambar 5.28 Sequence Diagram Mencari Buku Pada Sisi Anggota**

Gambar 5.28 merupakan *sequence diagram* yang menjelaskan urutan proses untuk mencari buku oleh anggota. Proses dimulai dari anggota sebagai aktor sistem memilih menu cari buku, kemudian tampil layout form cari buku. Anggota mengisi form cari buku dengan memasukkan kata kunci buku yang diinginkan. Selanjutnya controller memanggil fungsi `searchBuku` dan diteruskan dengan memanggil fungsi `getBuku` pada model untuk mendapatkan data buku yang diinginkan. Setelah model mendapatkan data buku yang diinginkan pada *database*, maka controller mengirimkan data tersebut pada view untuk ditampilkan. Namun apabila fungsi pada `getBuku` pada model tidak ditemukan, maka pesan notifikasi akan ditampilkan pada view bahwa buku tidak ditemukan.

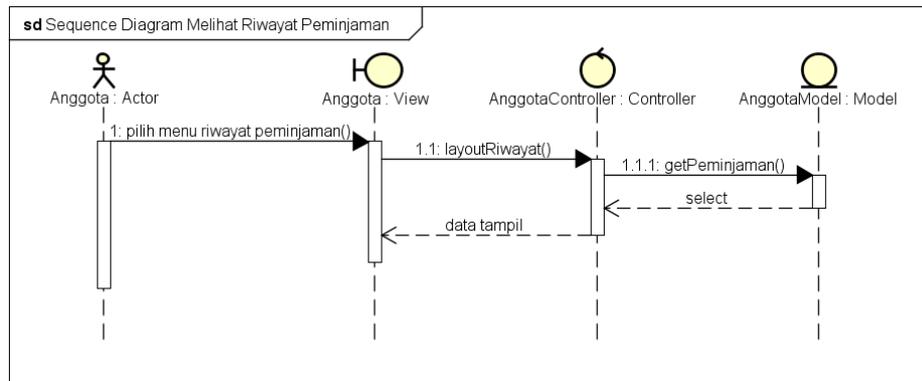
#### 4. Memperbarui Data Anggota



**Gambar 5.29 Sequence Diagram Memperbarui Data Anggota Pada Sisi Anggota**

Gambar 5.29 merupakan *sequence diagram* yang menjelaskan urutan proses untuk memperbarui data anggota oleh anggota. Proses dimulai dari anggota sebagai aktor sistem memilih menu profile untuk menampilkan halaman profile anggota. Kemudian aktor memilih tombol ubah profile untuk menampilkan halaman form edit profile anggota. Aktor mengisi form edit untuk mengubah data anggota. Selanjutnya controller memanggil fungsi `editUsername` yang didalamnya memanggil fungsi `cekUsername` untuk memeriksa apakah `username` yang dimasukkan sudah ada di dalam `database` atau belum, apabila `username` tidak ada dalam `database` maka fungsi `updateUsername` pada model dipanggil untuk memperbarui data anggota. Setelah model menyimpan data anggota yang baru diubah pada `database`, maka controller mengirimkan data tersebut pada view untuk ditampilkan. Namun apabila `username` sudah ada di dalam `database`, maka data anggota tidak berhasil diubah dan sistem akan memberikan peringatan kepada aktor bahwa data anggota tidak berhasil diubah.

## 5. Melihat Riwayat Peminjaman

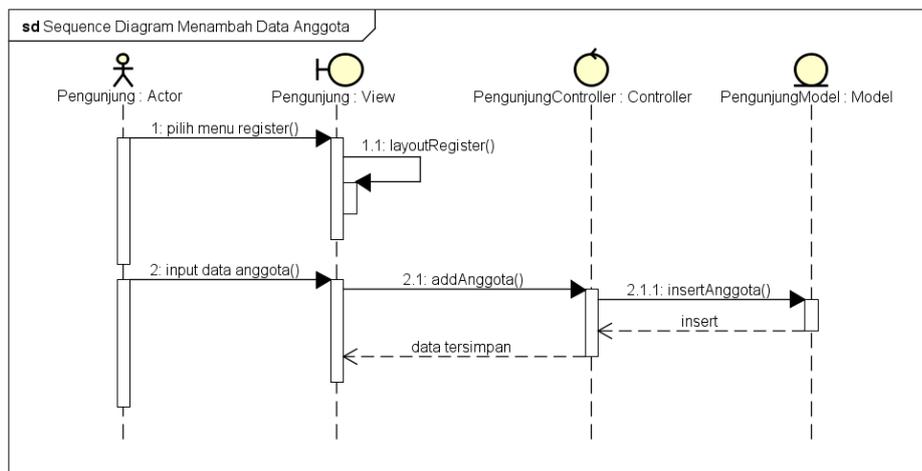


**Gambar 5.30 Sequence Diagram Melihat Riwayat Peminjaman**

Gambar 5.30 merupakan *sequence diagram* yang menjelaskan urutan proses untuk melihat data riwayat peminjaman oleh anggota. Proses dimulai dari anggota sebagai aktor sistem memilih menu riwayat peminjaman untuk meminta halaman data riwayat peminjaman pada view. Selanjutnya controller memanggil fungsi `layoutRiwayat` dan diteruskan dengan memanggil fungsi `getPeminjaman` pada model untuk menampilkan data riwayat peminjaman yang dilakukan anggota. Setelah model membaca data riwayat peminjaman pada *database*, data dikirim kembali ke fungsi `layoutRiwayat` pada controller untuk ditampilkan di view.

### 5.3.3 Sequence Diagram Pengunjung

#### 1. Menambah Data Anggota

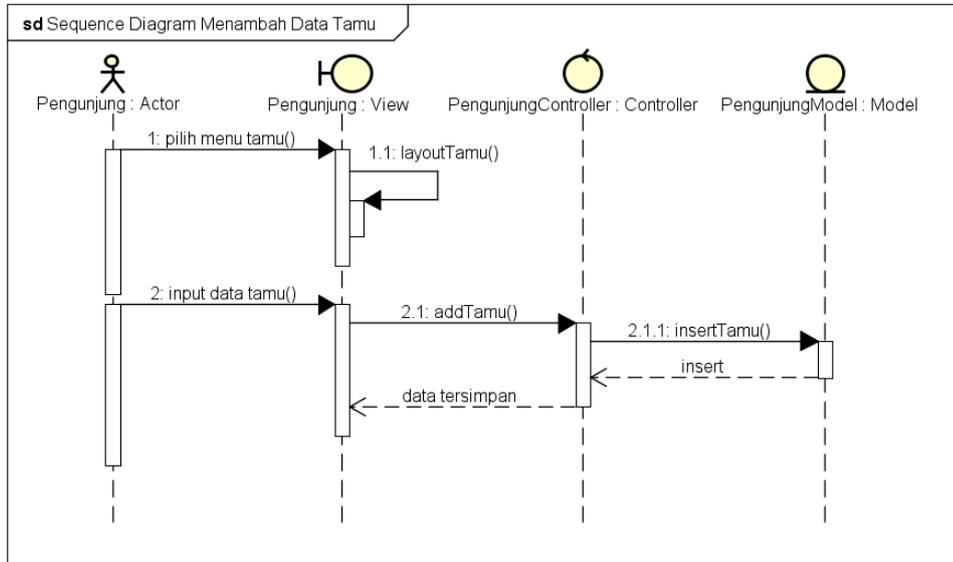


**Gambar 5.31 Sequence Diagram Menambah Data Anggota Pada Sisi Pengunjung**

Gambar 5.31 merupakan *sequence diagram* yang menjelaskan urutan proses untuk menambah data anggota oleh pengunjung. Proses dimulai dari pengunjung sebagai aktor sistem memilih menu register kemudian tampil halaman register berisi form pendaftaran sebagai anggota perpustakaan. Aktor menginputkan data anggota pada form tersebut. Selanjutnya controller

menjalankan fungsi `addAnggota` dan diteruskan dengan memanggil fungsi `insertAnggota` pada model untuk menyimpan data buku yang baru ditambahkan. Setelah model menyimpan data baru pada *database*, selanjutnya controller mengirimkan data kembali untuk ditampilkan di view.

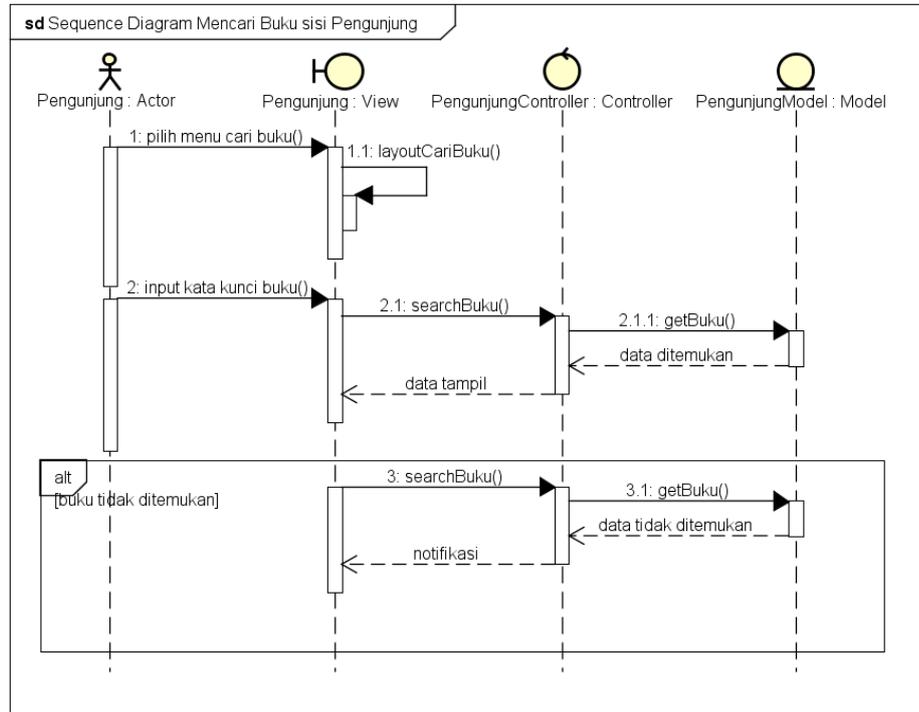
## 2. Menambah Data Tamu



**Gambar 5.32 Sequence Diagram Menambah Data Tamu Pada Sisi Pengunjung**

Gambar 5.32 merupakan *sequence diagram* yang menjelaskan urutan proses untuk menambah data tamu oleh pengunjung. Proses dimulai dari pengunjung sebagai aktor sistem memilih menu tamu untuk menampilkan halaman form tamu. Kemudian aktor menginputkan data tamu pada view. Selanjutnya controller memanggil fungsi `addTamu` dan diteruskan dengan memanggil fungsi `insertTamu` pada model untuk menyimpan data tamu yang ditambahkan. Setelah model menyimpan data tamu yang baru ditambahkan pada *database*, controller mengirimkan informasi ke view bahwa data baru telah tersimpan.

### 3. Mencari Buku

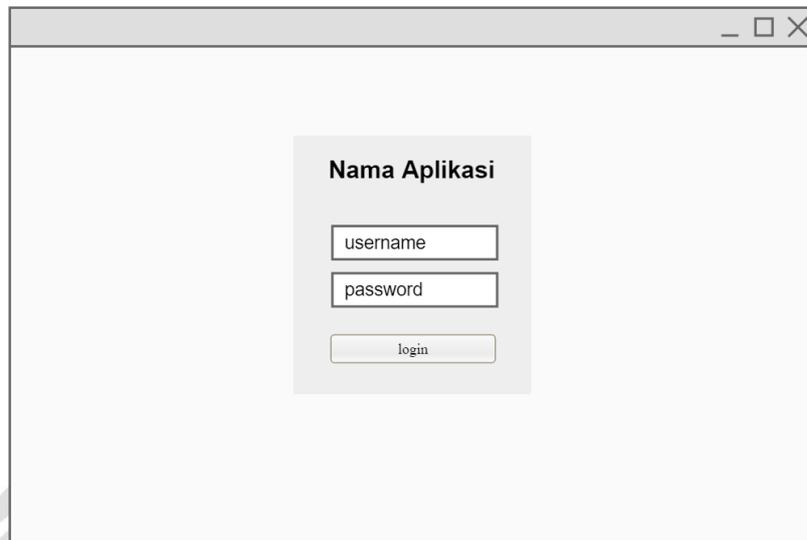


**Gambar 5.33 Sequence Diagram Mencari Buku Pada Sisi Pengunjung**

Gambar 5.33 merupakan *sequence diagram* yang menjelaskan urutan proses untuk mencari buku oleh pengunjung. Proses dimulai dari pengunjung sebagai aktor sistem memilih menu cari buku, kemudian tampil layout form cari buku. Pengunjung mengisi form cari buku dengan memasukkan kata kunci buku yang diinginkan. Selanjutnya controller memanggil fungsi `searchBuku` dan diteruskan dengan memanggil fungsi `getBuku` pada model untuk mendapatkan data buku yang diinginkan. Setelah model mendapatkan data buku yang diinginkan pada *database*, controller mengirimkan data kembali untuk ditampilkan di view. Namun apabila fungsi pada `getBuku` pada model tidak ditemukan, maka pesan notifikasi akan ditampilkan pada view bahwa buku tidak ditemukan.

## 5.4 Perancangan Antarmuka

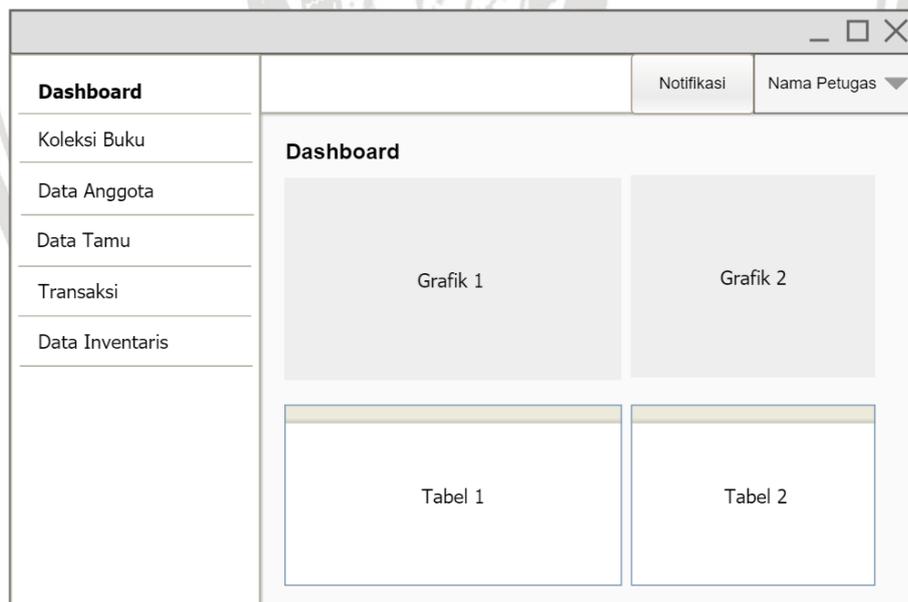
### 5.4.1 Halaman *Login* Petugas



**Gambar 5.34 Rancangan Halaman *Login* Petugas**

Gambar 5.34 merupakan rancangan untuk halaman *login*. Halaman ini berfungsi untuk keamanan sistem agar aplikasi hanya dapat diakses oleh petugas dan anggota perpustakaan. Halaman ini terdapat *text field* untuk menginputkan username, *password field* untuk menginputkan password, dan tombol *login*.

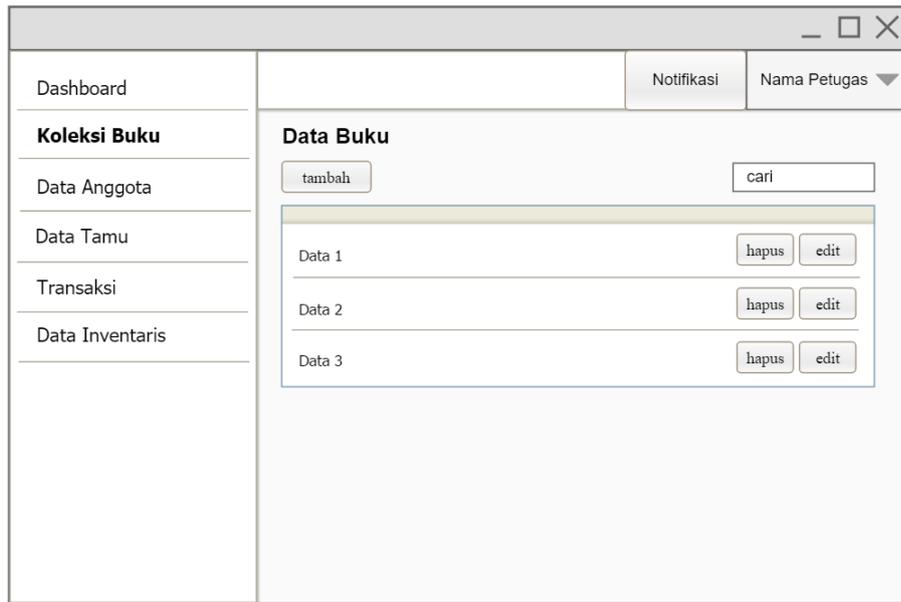
### 5.4.2 Halaman *Dashboard*



**Gambar 5.35 Rancangan Halaman *Dashboard***

Gambar 5.35 merupakan rancangan halaman *dashboard* yang terdapat grafik dan tabel-tabel.

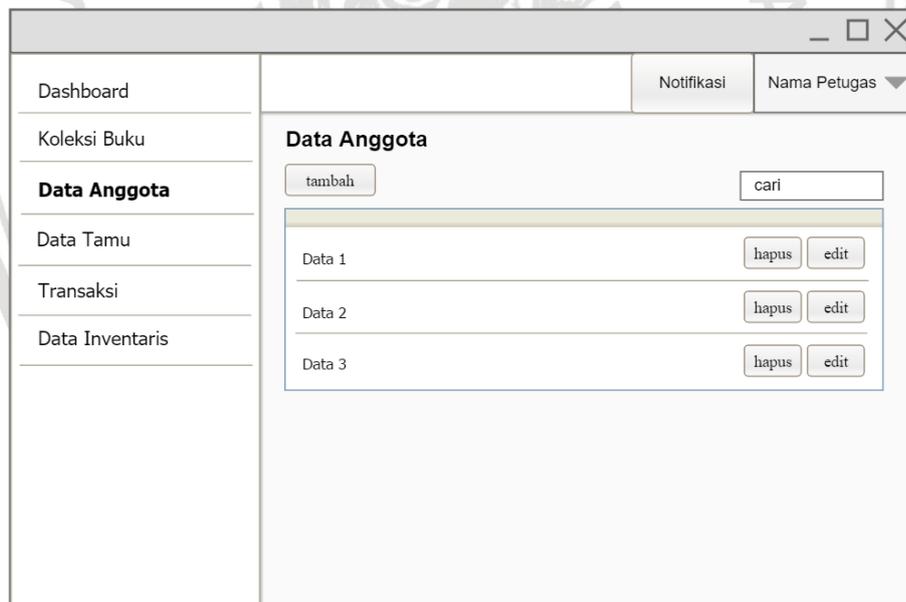
### 5.4.3 Halaman Data Buku



**Gambar 5.36 Rancangan Halaman Data Buku**

Gambar 5.36 merupakan rancangan halaman data koleksi buku yang menampilkan seluruh data buku yang terdapat pada *database*.

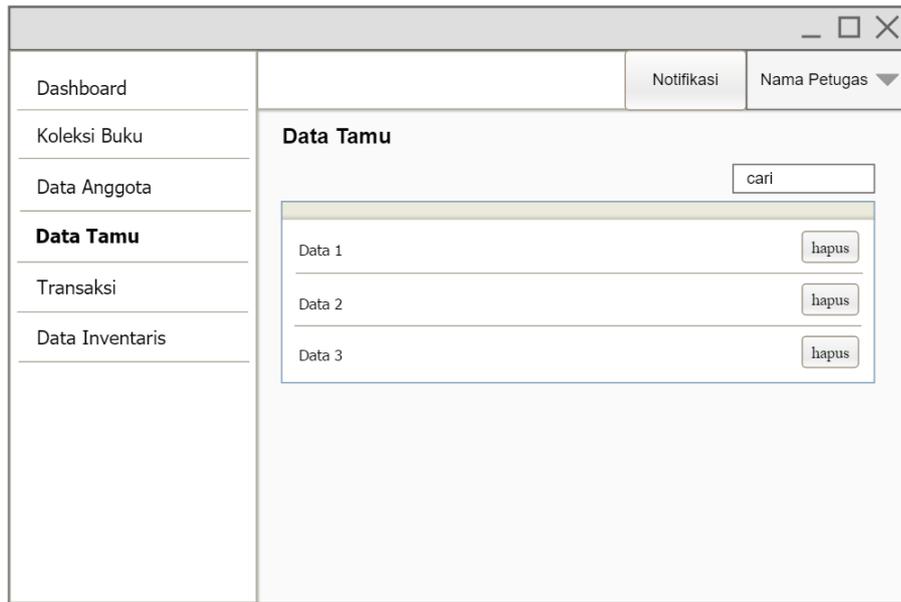
### 5.4.4 Halaman Data Anggota



**Gambar 5.37 Rancangan Halaman Data Anggota**

Gambar 5.37 merupakan rancangan halaman data anggota yang menampilkan seluruh data anggota yang terdapat pada *database*.

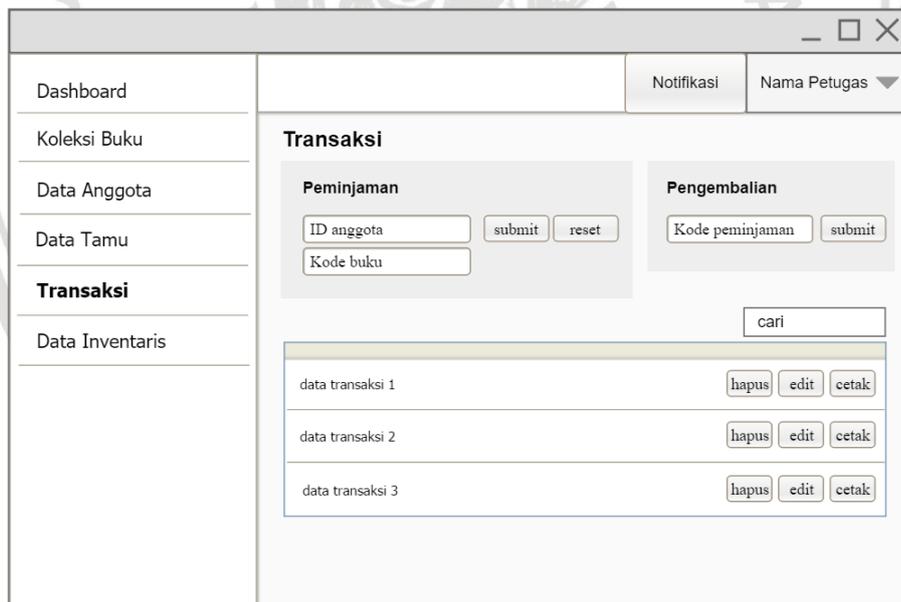
### 5.4.5 Halaman Data Tamu



**Gambar 5.38 Rancangan Halaman Data Tamu**

Gambar 5.38 merupakan rancangan halaman data tamu yang menampilkan seluruh data tamu yang terdapat pada *database*.

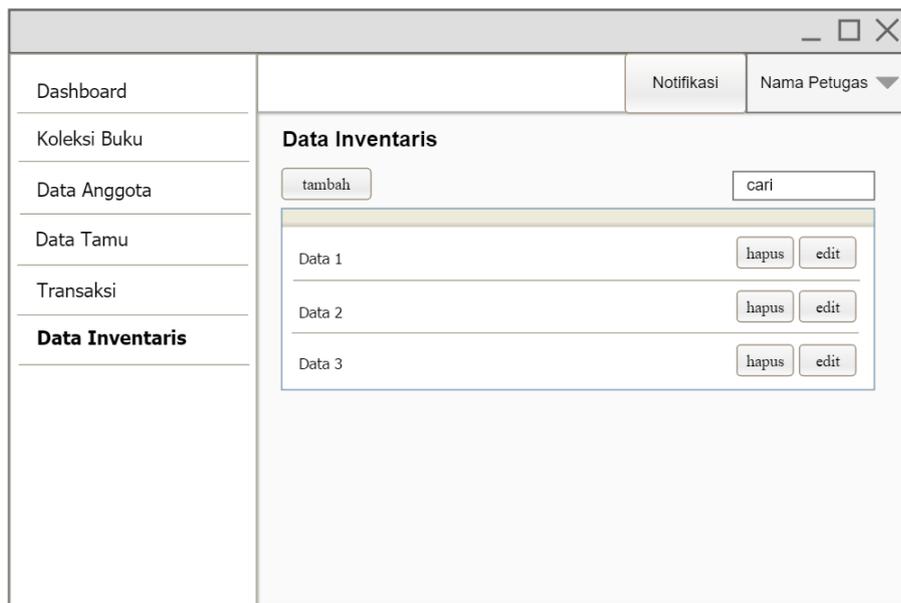
### 5.4.6 Halaman Transaksi



**Gambar 5.39 Rancangan Halaman Transaksi**

Gambar 5.39 merupakan rancangan halaman data transaksi dimana pengguna dapat melayani transaksi peminjaman dan pengembalian buku serta halaman tersebut terdapat tabel yang berisi seluruh data transaksi yang ada pada *database*.

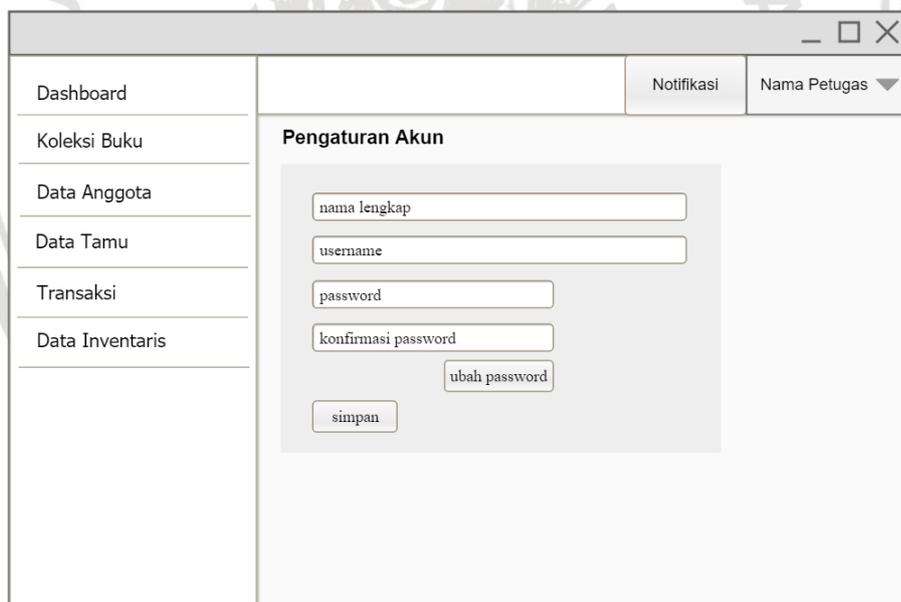
### 5.4.7 Halaman Data Inventaris



**Gambar 5.40 Rancangan Halaman Data Inventaris**

Gambar 5.40 merupakan rancangan halaman data inventaris yang menampilkan tabel berisi seluruh data inventaris yang ada pada *database*.

### 5.4.8 Halaman Pengaturan Akun Petugas



**Gambar 5.41 Rancangan Halaman Pengaturan Akun Petugas**

Gambar 5.41 merupakan rancangan halaman pengaturan akun petugas dimana pengguna dapat memperbarui data petugas pada *database*.

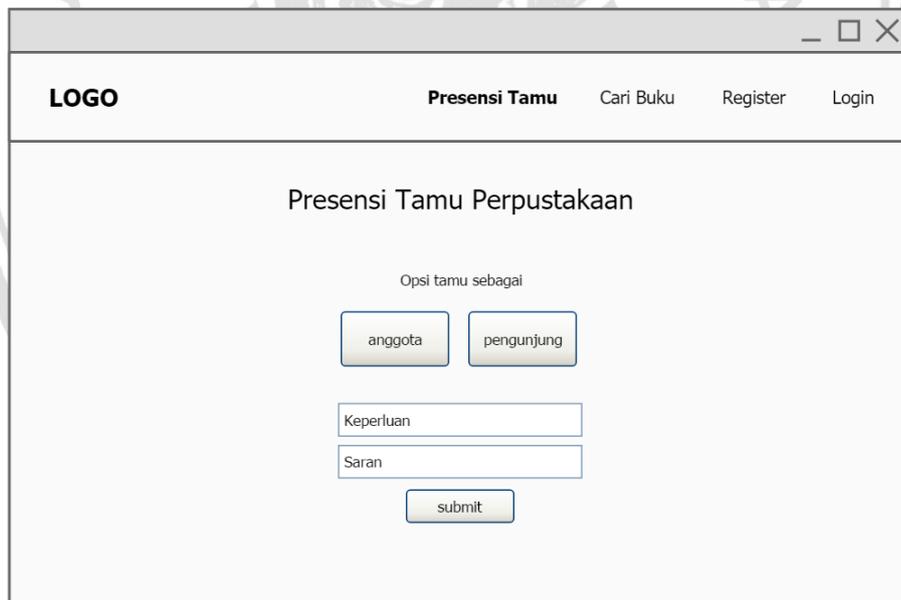
### 5.4.9 Halaman Utama Pengguna



**Gambar 5.42 Rancangan Halaman Utama Pengguna**

Gambar 5.42 merupakan rancangan halaman utama website pengunjung dan anggota perpustakaan.

### 5.4.10 Halaman Presensi Tamu

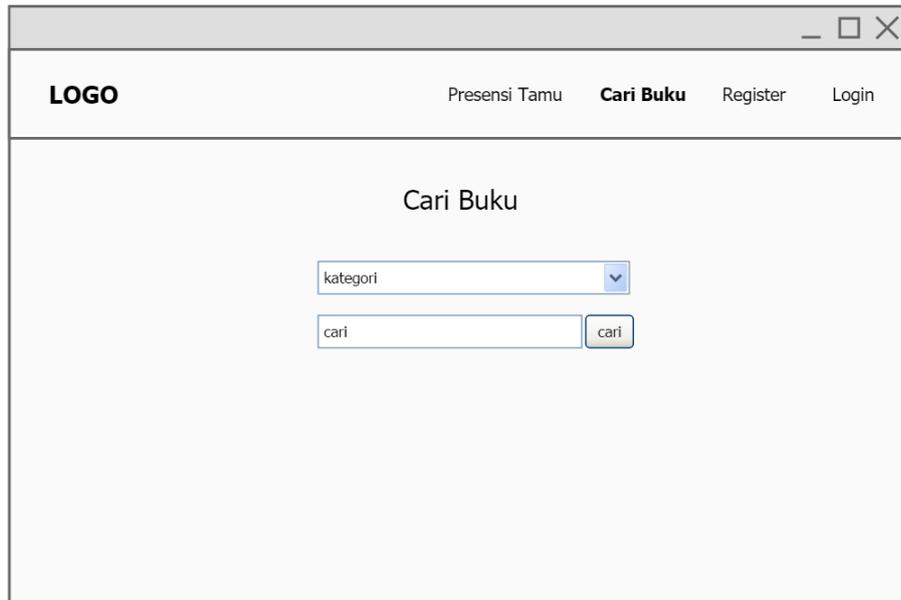


**Gambar 5.43 Rancangan Halaman Presensi Tamu**

Gambar 5.43 merupakan rancangan halaman presensi tamu yang berkunjung ke perpustakaan.



#### 5.4.11 Halaman Cari Buku



LOGO Presensi Tamu **Cari Buku** Register Login

Cari Buku

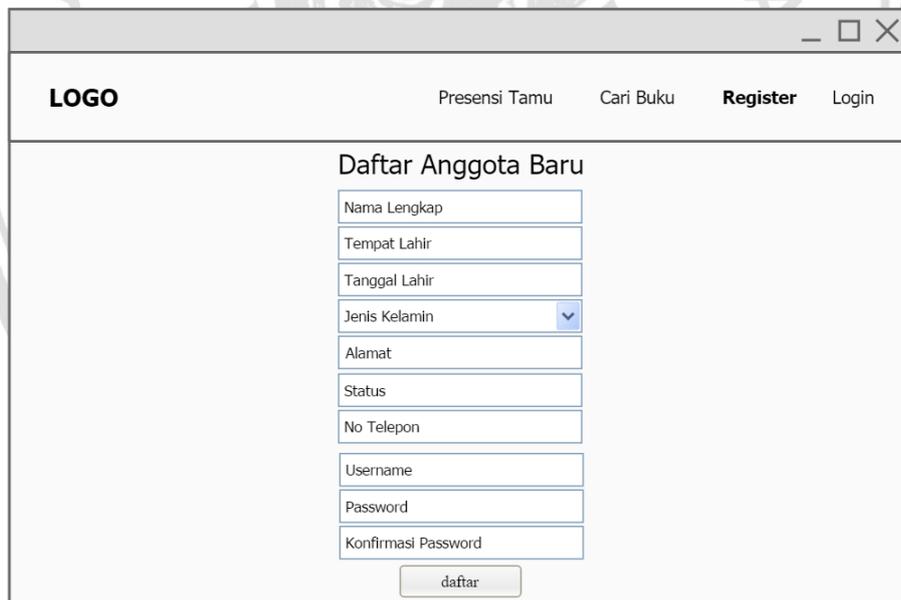
kategori

cari cari

**Gambar 5.44 Rancangan Halaman Cari Buku**

Gambar 5.44 merupakan rancangan halaman pencarian buku yang dilakukan oleh anggota atau pengunjung perpustakaan.

#### 5.4.12 Halaman Registrasi Anggota



LOGO Presensi Tamu Cari Buku **Register** Login

Daftar Anggota Baru

Nama Lengkap

Tempat Lahir

Tanggal Lahir

Jenis Kelamin

Alamat

Status

No Telepon

Username

Password

Konfirmasi Password

daftar

**Gambar 5.45 Rancangan Halaman Registrasi Anggota**

Gambar 5.45 merupakan rancangan halaman pendaftaran anggota perpustakaan yang terdiri dari beberapa *input field* diantaranya nama lengkap, tempat lahir, tanggal lahir, alamat, status, tanggal bergabung, nomor telepon, password, konfirmasi password dan tombol daftar.

### 5.4.13 Halaman Login Anggota

**Gambar 5.46 Rancangan Halaman Login Anggota**

Gambar 5.46 merupakan rancangan halaman login untuk anggota perpustakaan yang sudah terdaftar.

### 5.4.14 Halaman Daftar Riwayat Peminjaman

Tanggal Pinjam	Kode Pinjam	Tanggal Kembali
	Data 1	
	Data 2	
	Data 3	

**Gambar 5.47 Rancangan Halaman Daftar Riwayat Peminjaman**

Gambar 5.47 merupakan rancangan halaman daftar riwayat peminjaman buku yang dilakukan oleh anggota perpustakaan.

#### 5.4.15 Halaman Pengaturan Akun Anggota

**Gambar 5.48 Rancangan Halaman Pengaturan Akun Anggota**

Gambar 5.48 merupakan rancangan halaman pengaturan akun anggota perpustakaan, diantaranya pengguna dapat mengubah *username* dan *password*.

#### 5.5 Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam proses implementasi sistem informasi sebagai berikut:

1. *System manufacturer* : DELL
2. *System model* : Inspiron 13-5368
3. *Processor* : Intel® Core™ i3-6100U CPU @ 2.30GHz 2.30 GHz
4. *Memory* : 4.00 GB RAM
5. *Harddisk* : 500 GB

#### 5.6 Perangkat Lunak

Perangkat lunak yang digunakan dalam proses implementasi sistem informasi yaitu:

1. Sistem Operasi Windows 10 Home Single Language 64-bit
2. Google Chrome
3. Mozilla Firefox
4. Xampp versi 3.2.2
5. Sublime Text Editor 3.0

#### 5.7 Implementasi Algoritma

Pada sub bab implementasi algoritma hanya mencantumkan satu fungsi yaitu fungsi memperbarui anggota yang terdapat pada kelas controller PetugasController dan kelas model PetugasModel.

### 5.7.1 Kelas Controller PetugasController

Fungsi pada kelas *controller* `PetugasController` yang memiliki fungsi untuk memperbarui data anggota perpustakaan adalah fungsi `editAnggota`. Pada fungsi `editAnggota` terdapat parameter `id anggota`, `nama`, `tempat lahir`, `tanggal lahir`, `alamat`, `status`, `tanggal gabung`, `nomor telepon`, `username`, `password` yang merupakan variabel untuk menampung masukan dari pengguna. Di dalam fungsi `editAnggota` terdapat variabel `exist_username` yang diinisialisasikan dengan nilai hasil pengembalian (*return*) dari fungsi `cekUnameAnggota` yang dipanggil dari kelas model, nilai dari variabel ini digunakan untuk melakukan pengecekan terhadap *username* anggota perpustakaan yang terdaftar pada *database*. Jika ada *username* yang terdeteksi sama, maka akan dinilai 1 dan sistem akan memberi peringatan kepada pengguna apabila *username* telah terpakai dan harus memasukkan *username* lain yang berbeda. Sebaliknya jika tidak ada *username* yang terdeteksi sama, maka akan dinilai 0 dan sistem akan memanggil fungsi `updateAnggota` dari kelas model untuk menyimpan data yang telah diubah ke dalam *database*. Pada Tabel 5.1 merupakan algoritma pada kelas *controller* `PetugasController`.

**Tabel 5.1 Kode Program Kelas Controller PetugasController**

PetugasController.php
<pre> &lt;?php require_once 'model\PetugasModel.php'; class PetugasController {     private \$model;     function __construct() {         \$this-&gt;model = new PetugasModel();     }     public function editAnggota(\$id_anggota, \$nama,     \$tempat_lahir, \$tgl_lahir, \$alamat, \$status, \$no_tlp,     \$username, \$password) {         \$exist_username=\$this-&gt;model-&gt;cekUnameAnggota(\$username);         \$username_member=\$this-&gt;model-&gt;getUnameAnggota(\$id_anggota);         if (\$username==\$username_member['username']) {             \$this-&gt;model-&gt;updateAnggota(\$id_anggota, \$nama,             \$tempat_lahir, \$tgl_lahir, \$alamat, \$status,             \$no_tlp, \$username, \$password);         } elseif (\$username!=\$username_member['username']) {             if (\$exist_username==0) {                 \$this-&gt;model-&gt;updateAnggota(\$id_anggota,                 \$nama, \$tempat_lahir, \$tgl_lahir, \$alamat,                 \$status, \$no_tlp, \$username, \$password);             } elseif (\$exist_username==1) {                 echo "&lt;script&gt;alert('Username sudah </pre>

```

                terpakai!')</script>";
            }
        }
    }
}
?>

```

### 5.7.2 Kelas Model PetugasModel dan Kelas Anggota

Dalam kelas model terdapat beberapa fungsi yang mendukung proses perbaruan data anggota perpustakaan pada *database*. Kelas *PetugasModel* melakukan pewarisan kelas dari kelas *Anggota*, diantaranya fungsi yang diwariskan dari kelas *Anggota* adalah fungsi *cekUnameAnggota* dengan parameter *uname* atau *username* merupakan fungsi yang digunakan untuk mengecek ketersediaan *username* dengan menyeleksi *username* yang terdapat pada *database* agar terhindar dari redudansi data. Fungsi yang kedua yang diwariskan dari kelas *Anggota* adalah fungsi *getUnameAnggota* dengan parameter *id\_anggota* merupakan fungsi yang digunakan untuk menyeleksi *username* sesuai *id\_anggota* yang terpilih untuk diperbarui datanya. *Username* yang terseleksi akan digunakan sebagai kondisi untuk proses pengecekan ketersediaan *username* pada *database*. Kemudian fungsi *updateAnggota* pada kelas *PetugasModel* dengan parameter *id anggota*, *nama*, *tempat lahir*, *tanggal lahir*, *alamat*, *status*, *tanggal gabung*, *nomor telepon*, *username*, dan *password* merupakan fungsi yang berisikan *query* untuk melakukan perbaruan data anggota pada *database*. Pada Tabel 5.2 menunjukkan kode program fungsi-fungsi yang mendukung proses perbaruan data anggota pada kelas *PetugasModel*, dan pada Tabel 5.3 menunjukkan kode program fungsi-fungsi yang mendukung proses perbaruan data anggota pada kelas *Anggota*.

**Tabel 5.2 Kode Program Kelas Model PetugasModel**

PetugasModel.php
<pre> &lt;?php class PetugasModel extends Anggota {     private \$mysqli, \$no_buku, \$tgl_masuk, \$judul, \$kolasi,         \$pengarang, \$penerbit, \$thn_terbit, \$asal,     \$no_klasifikasi, \$status, \$id_inventaris, \$nama_barang,     \$jml_barang, \$no_tamu, \$kode_pinjam,     \$tgl_pinjam, \$tgl_habis_pmnjmn, \$tgl_kembali, \$denda,     \$no_pinjam, \$id_petugas, \$rak;      function __construct() {         \$this-&gt;mysqli = new         mysqli('localhost', 'root', '', 'perpustakaan');     }      public function updateAnggota(\$id_anggota, \$nama,     \$tempat_lahir, \$tgl_lahir, \$alamat, \$status, \$no_tlp,     \$uname, \$password) { </pre>



```

$this->id_anggota = $id_anggota;
$this->nama = $nama;
$this->tempat_lahir = $tempat_lahir;
$this->tgl_lahir = $tgl_lahir;
$this->alamat = $alamat;
$this->status = $status;
$this->no_tlp = $no_tlp;
$this->uname = $uname;
$this->password = $password;

$query = $this->mysql->query("UPDATE anggota SET
nama='$this->nama', tempat_lahir='$this-
>tempat_lahir', tgl_lahir='$this->tgl_lahir',
alamat='$this->alamat', status='$this->status',
no_tlp='$this->no_tlp', username='$this->uname',
password='$this->password' WHERE id_anggota='$this-
>id_anggota'");
return $query;
}
}
?>

```

**Tabel 5.3 Kode Program Kelas Anggota**

<p><b>Anggota.php</b></p> <pre> &lt;?php class Anggota {     private \$mysql, \$id_anggota, \$uname, \$password, \$nama,     \$alamat, \$status, \$jenis_kelamin, \$tempat_lahir,     \$tgl_lahir, \$tgl_gabung, \$no_tlp, \$keterangan;      public function cekUnameAnggota(\$uname) {         \$this-&gt;mysql=new         mysql('localhost','root','','perpustakaan');          \$this-&gt;uname=\$uname;          \$member = \$this-&gt;mysql-&gt;query("SELECT username FROM         anggota WHERE username='\$this-&gt;uname'");         \$exist_uname = mysql_num_rows(\$member);         return \$exist_uname;     }      public function getUnameAnggota(\$id_anggota) {         \$this-&gt;mysql=new         mysql('localhost','root','','perpustakaan');          \$this-&gt;id_anggota=\$id_anggota;          \$member = \$this-&gt;mysql-&gt;query("SELECT username FROM </pre>
---

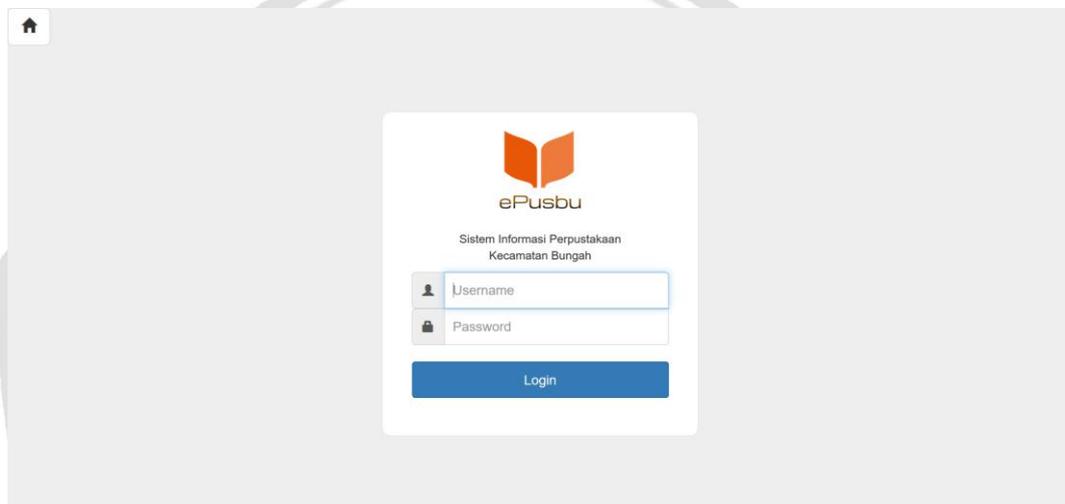


```
        anggota WHERE id_anggota='$this->id_anggota');  
        return $member->fetch_array();  
    }  
}  
?>
```

## 5.8 Screenshot Program

Setelah sistem diimplementasi ke dalam bentuk kode program, maka didapatkan tampilan sistem berupa *screenshot* dari hasil implementasi sebagai berikut.

### Screenshot Halaman Login Petugas



**Gambar 5.49 Screenshot Halaman Login Petuga**

Gambar 5.49 merupakan hasil implementasi halaman *login* petugas sistem informasi perpustakaan.

**Screenshot Halaman Dashboard**



**Gambar 5.50 Screenshot Halaman Dashboard**

Gambar 5.50 merupakan hasil implementasi halaman *dashboard* sistem informasi perpustakaan.

**Screenshot Halaman Data Buku**

The screenshot shows the 'Koleksi Buku' page with a table of book collection data. The table includes columns for book number, date, title, classification, author, publisher, year, source, and classification number.

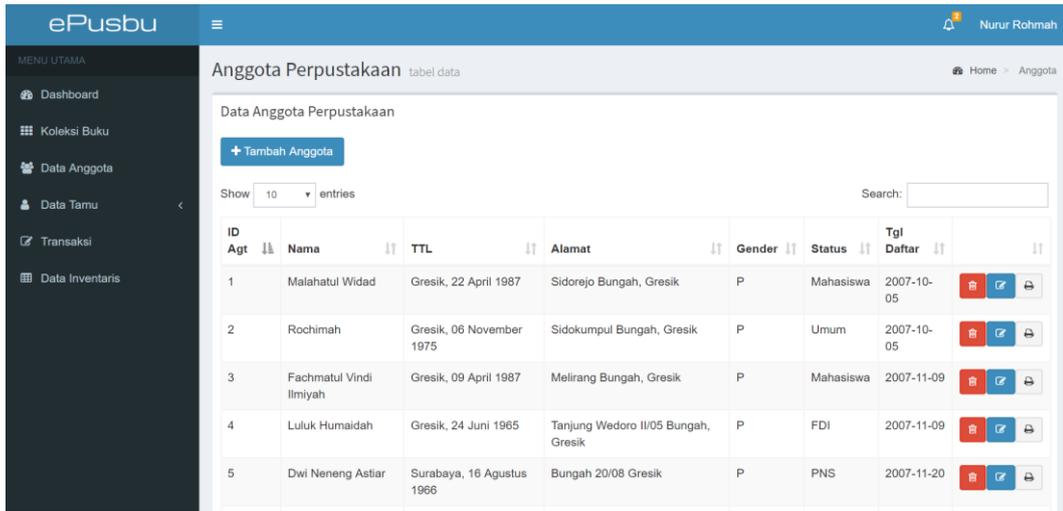
No Buku	Tanggal Masuk	Judul Buku	Kolasi	Pengarang	Penerbit	Tahun Terbit	Asal	No Klasifikasi
00001	2007-10-24	Metode Penelitian		Drs. Mardalis	Bumi Aksara	2006	beli	001.42
00002	2007-10-24	Pengelolaan Perpustakaan		Drs. Ibrahim Bafadal	Bumi Aksara	2005	beli	027.8
00003	2007-10-24	The Young Scientist Vol. 1 No. 12		Young Scientist Enterprise	Grasindo	2006	beli	050.1
00004	2007-10-24	The Young Scientist Vol. 1 No. 13		Young Scientist Enterprise	Grasindo	2006	beli	050.1

**Gambar 5.51 Screenshot Halaman Data Buku**

Gambar 5.51 merupakan hasil implementasi halaman data koleksi buku sistem informasi perpustakaan.



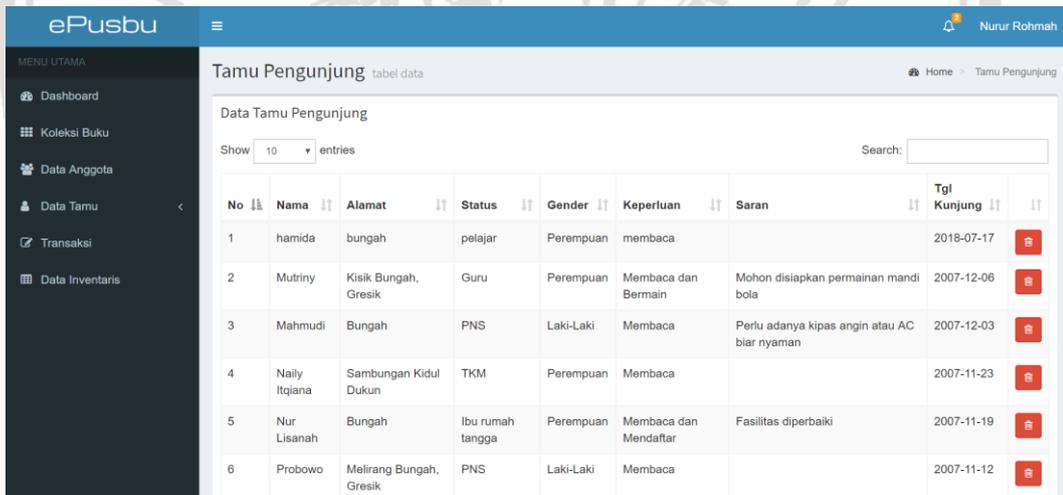
### Screenshot Halaman Data Anggota



**Gambar 5.52 Screenshot Halaman Data Anggota**

Gambar 5.52 merupakan hasil implementasi halaman data anggota sistem informasi perpustakaan.

### Screenshot Halaman Data Tamu

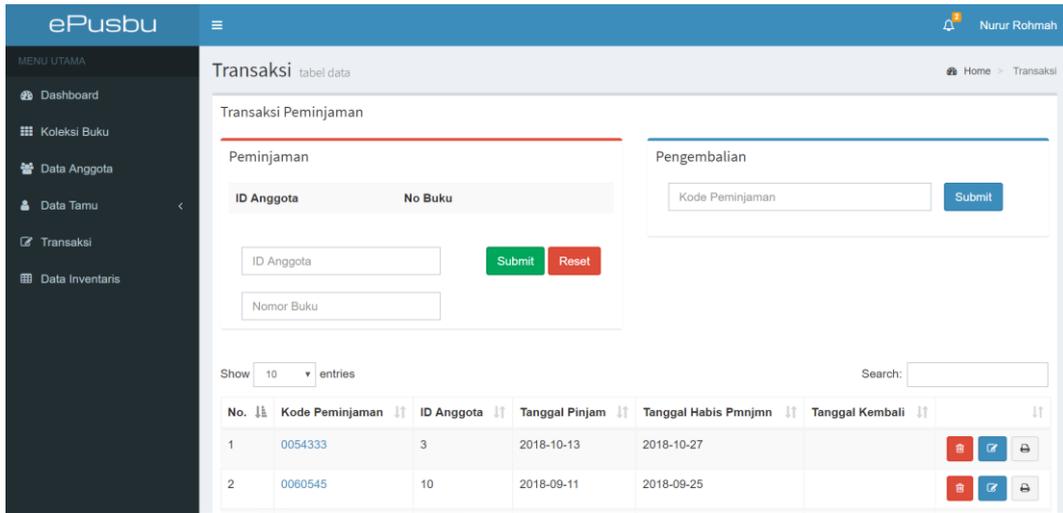


**Gambar 5.53 Screenshot Halaman Data Tamu**

Gambar 5.53 merupakan hasil implementasi halaman data tamu sistem informasi perpustakaan.



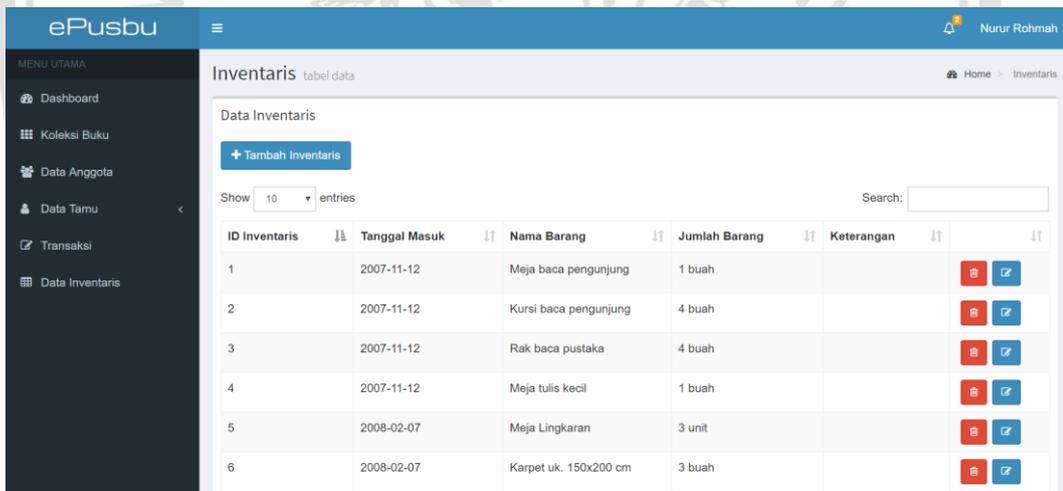
**Screenshot Halaman Transaksi Peminjaman**



**Gambar 5.54 Screenshot Halaman Transaksi Peminjaman**

Gambar 5.54 merupakan hasil implementasi halaman data transaksi peminjaman sistem informasi perpustakaan.

**Screenshot Halaman Data Inventaris**

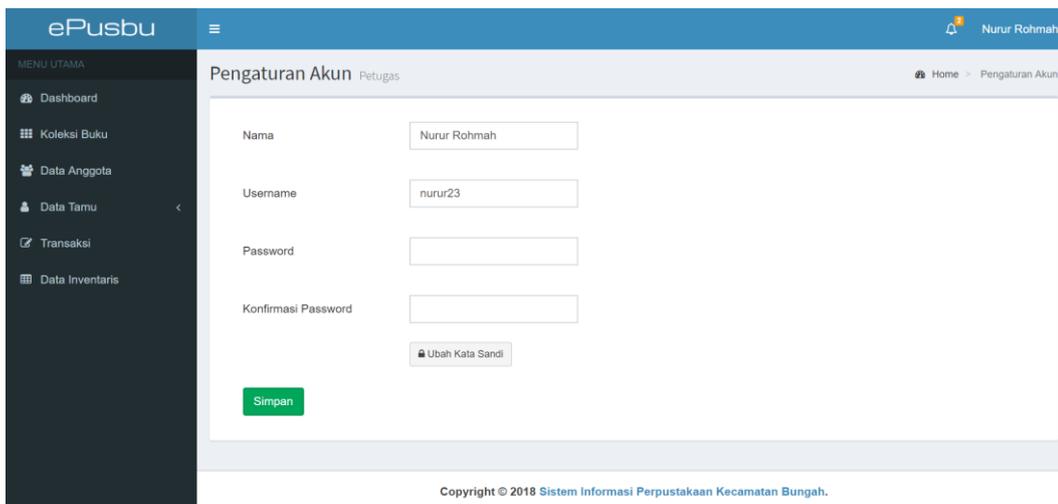


**Gambar 5.55 Screenshot Halaman Data Inventaris**

Gambar 5.55 merupakan hasil implementasi halaman data inventaris sistem informasi perpustakaan.



### Screenshot Halaman Pengaturan Akun Petugas



**Gambar 5.56 Screenshot Halaman Pengaturan Akun Petugas**

Gambar 5.56 merupakan hasil implementasi halaman pengaturan akun petugas sistem informasi perpustakaan.

### Screenshot Halaman Utama Pengguna



**Gambar 5.57 Screenshot Halaman Utama Pengguna**

Gambar 5.57 merupakan hasil implementasi halaman utama sistem informasi perpustakaan.

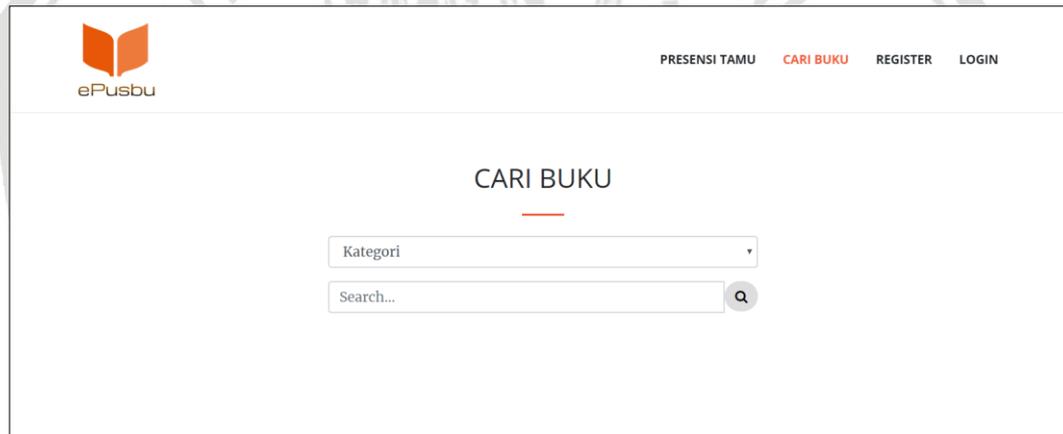
**Screenshot Halaman Presensi Tamu**



**Gambar 5.58 Screenshot Halaman Presensi Tamu**

Gambar 5.58 merupakan hasil implementasi halaman presensi tamu sistem informasi perpustakaan.

**Screenshot Halaman Cari Buku**



**Gambar 5.59 Screenshot Halaman Cari Buku**

Gambar 5.59 merupakan hasil implementasi halaman pencarian buku sistem informasi perpustakaan.

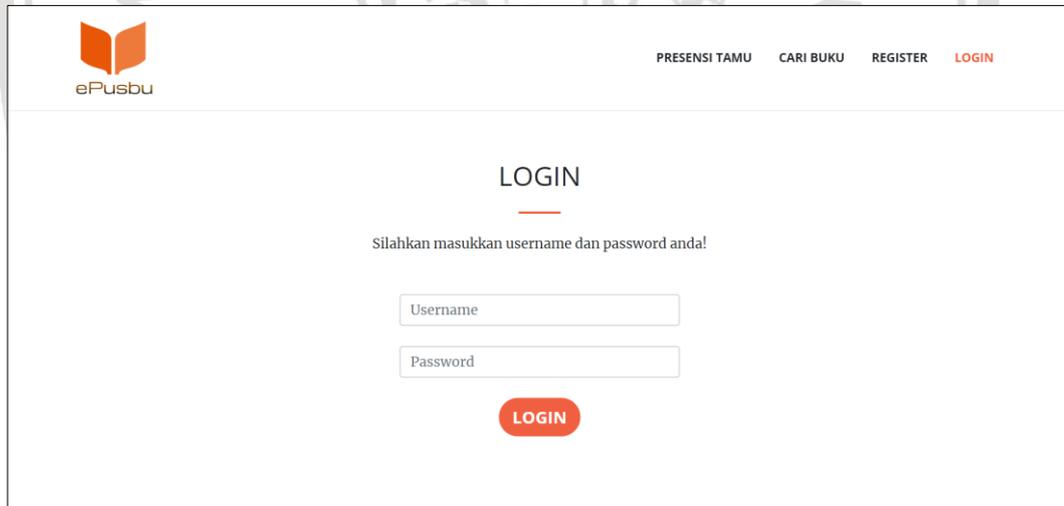
**Screenshot Halaman Registrasi Anggota Baru**



**Gambar 5.60 Screenshot Halaman Registrasi Anggota Baru**

Gambar 5.60 merupakan hasil implementasi halaman registrasi anggota baru sistem informasi perpustakaan.

**Screenshot Halaman Login Anggota**



**Gambar 5.61 Screenshot Halaman Login Anggota**

Gambar 5.61 merupakan hasil implementasi halaman login anggota sistem informasi perpustakaan.

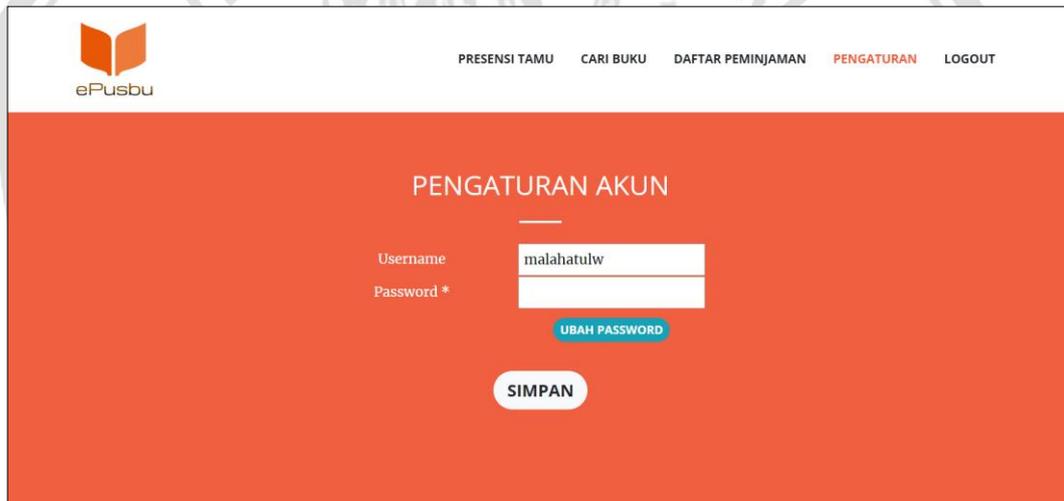
**Screenshot Halaman Daftar Riwayat Peminjaman**



**Gambar 5.62 Screenshot Halaman Daftar Riwayat Peminjaman**

Gambar 5.62 merupakan hasil implementasi halaman daftar riwayat peminjaman sistem informasi perpustakaan.

**Screenshot Halaman Pengaturan Akun Anggota**



**Gambar 5.63 Screenshot Halaman Pengaturan Akun Anggota**

Gambar 5.63 merupakan hasil implementasi halaman pengaturan akun anggota sistem informasi perpustakaan.

## BAB 6 PENGUJIAN

Pada bab ini menjelaskan proses pengujian sistem yang dilakukan secara bertahap dari *unit testing* hingga *acceptance testing*. Tahapan pertama dilakukan *unit testing* menggunakan salah satu metode *white box testing* yaitu *basis path testing*. Kemudian dilakukan *integration testing* antar fungsi sistem, dan dilanjutkan *validation testing*. Selanjutnya dilakukan pengujian non-fungsional yaitu *reliability testing* dan *compatibility testing* dengan menggunakan tools. Kemudian pengujian *acceptance* dengan menggunakan metode black box yaitu *user acceptance testing* yang dilakukan oleh *stakeholder* untuk mengetahui apakah sistem telah sesuai dengan kebutuhan pengguna atau tidak.

### 6.1 Unit Testing

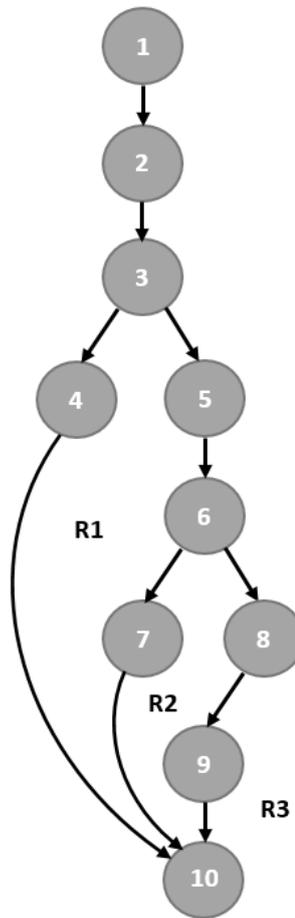
#### 6.1.1 Basis Path Testing

*Basis path testing* dilakukan terhadap fungsi `editAnggota` pada kelas controller `PetugasController`. Tabel 6.1 berisi algoritma fungsi `editAnggota` yang terdapat pada kelas `PetugasController` untuk mengubah data anggota perpustakaan pada *database*.

Tabel 6.1 Algoritma Fungsi `editAnggota()`

Pseudocode	Node
Mulai	1
Menyimpan hasil pengecekan username anggota pada database ke dalam variabel <code>exist_uname</code>	2
Menyimpan hasil seleksi username anggota sesuai <code>id_anggota</code> pada database ke dalam variabel <code>uname_member</code>	
Jika parameter <code>uname</code> sama dengan nilai variabel <code>uname_member</code>	3
Memperbaharui data anggota ke dalam database dengan method <code>setUpdateAnggota()</code>	4
Lainnya jika parameter <code>uname</code> tidak sama dengan nilai variabel <code>uname_member</code>	5
Jika nilai variabel <code>exist_uname</code> sama dengan 0	6
Memperbaharui data anggota ke dalam database dengan method <code>setUpdateAnggota()</code>	7
Lainnya jika nilai variabel <code>exist_uname</code> lebih besar daripada 0	8
Menampilkan alert "Username sudah terpakai!"	9
Selesai	10

Dari pseudocode yang dijabarkan pada Tabel 6.1 dapat digambarkan *flow graph* sebagai berikut.



**Gambar 6.1 Flow Graph editAnggota**

Gambar 6.1 merupakan *flow graph* fungsi `editAnggota` pada kelas `PetugasController` yang selanjutnya dilakukan perhitungan *cyclomatic complexity* seperti dibawah ini:

Region (R)	= 3
Node (N)	= 10
Edge (E)	= 11
V(G) Region	= 3
V(G) dari N dan E	= (E - N) + 2
	= (11 - 10) + 2
	= 1 + 2
	= 3

Dari perhitungan *cyclomatic complexity* pada fungsi `editAnggota` pada kelas `PetugasController` dihasilkan 3 *independent path*, diantaranya:

1. *Independent path* 1: 1 – 2 – 3 – 4 – 10
2. *Independent path* 2: 1 – 2 – 3 – 5 – 6 – 7 – 10
3. *Independent path* 3: 1 – 2 – 3 – 5 – 6 – 8 – 9 – 10

Kemudian untuk memeriksa alur logika, dilakukan *test case* pada masing-masing *independent path*. Hasil pengujian *basis path* dikatakan *valid* apabila *expected result* sesuai dengan hasil dari pengujian yang dilakukan. Berikut *test case* fungsi `editAnggota` pada Tabel 6.2.

**Tabel 6.2 Test Case editAnggota**

No.	Independent Path	Test Case	Expected Result	Result	Status
1.	1 – 2 – 3 – 4 – 10	Parameter <code>id_anggota</code> dengan <i>value</i> "12", parameter <code>nama</code> dengan <i>value</i> "Nurur Rohmah", parameter <code>tempat_lahir</code> dengan <i>value</i> "Gresik", parameter <code>tgl_lahir</code> dengan <i>value</i> "1996-07-23", parameter <code>gender</code> dengan <i>value</i> "Perempuan", parameter <code>alamat</code> dengan <i>value</i> "Manyar Gresik", parameter <code>status</code> dengan <i>value</i> "Mahasiswa", parameter <code>no_tlp</code> dengan <i>value</i> "088123333211", parameter <code>uname</code> dengan <i>value</i> "nururrohmah23", parameter <code>password</code> dengan <i>value</i> "passwordku". Dimana variabel <code>uname_member</code> sama dengan <i>value</i>	Sistem akan menyimpan perbaruan data anggota ke dalam <i>database</i> .	Sistem menyimpan perbaruan data anggota ke dalam <i>database</i> .	Valid

		parameter uname.			
2.	1 – 2 – 3 – 5 – 6 – 7 – 10	Parameter id_anggota dengan value "12", parameter nama dengan value "Nurur Rohmah", parameter tempat_lahir dengan value "Malang", parameter tgl_lahir dengan value "1996-07-23", parameter gender dengan value "Perempuan", parameter alamat dengan value "Lowokwaru Dinoyo Malang", parameter status dengan value "Mahasiswa", parameter no_tlp dengan value "088123343211", parameter uname dengan value "nurur23", parameter password dengan value "mypassword". Dimana variabel uname_member tidak sama dengan value parameter uname dan variabel exist_uname sama dengan 0.	Sistem Sistem akan menyimpan perbaruan data anggota ke dalam database.	Sistem menyimpan perbaruan data anggota ke dalam database.	Valid
3.	1 – 2 – 3 – 5 – 6 – 8 – 9 – 10	Parameter id_anggota dengan value "12", parameter nama dengan value "Nurur", parameter tempat_lahir dengan value "Malang",	Sistem menampilkan alert "Username sudah terpakai!".	Sistem menampilkan alert "Username sudah terpakai!".	Valid

		parameter tgl_lahir dengan value "1995-06-01", parameter gender dengan value "Perempuan", parameter alamat dengan value "Lowokwaru Dinoyo Malang", parameter status dengan value "Mahasiswa", parameter no_tlp dengan value "08822334311", parameter uname dengan value "restmk", parameter password dengan value "mypassword". Dimana variabel uname_member tidak sama dengan value parameter uname dan nilai variabel exist_uname lebih dari 0.			
--	--	---	--	--	--

### 6.2 Integration Testing

Integration Testing dilakukan dengan menggabungkan dua atau lebih unit untuk memastikan kesesuaian output yang diharapkan dari perangkat lunak. Tabel 6.3 merupakan tabel pengujian integrasi dengan menggunakan fungsi kelas controller *PetugasController* dan kelas model *PetugasModel* dimana kedua fungsi tersebut saling berhubungan. Pengujian ini dikatakan valid apabila nilai *expected result* dan nilai *result* dari pengujian yang dilakukan sesuai.

**Tabel 6.3 Hasil Pengujian Integration Testing Kelas *PetugasController* dengan Kelas *PetugasModel***

Input	Fungsi dari <i>PetugasController</i>	Fungsi dari <i>PetugasModel</i>	Expected Result	Result	Status
Data Anggota	editAnggota(\$id_anggota, \$nama, \$tempat_lahir,	updateAnggota(\$id_anggota, \$nama, \$tempat_lahir,	Sistem menyimpan pembaharuan data anggota ke	Sistem menyimpan pembaharuan data anggota ke	Valid

	\$tgl_lahir, \$gender, \$alamat, \$status, \$no_tlp, \$uname, \$password)	\$tgl_lahir, \$gender, \$alamat, \$status, \$no_tlp, \$uname, \$password)	database dan menampilkan data anggota setelah proses pembaharuan data berhasil	database dan menampilkan data anggota setelah proses pembaharuan data berhasil	
--	---	---	--	--	--

### 6.3 Validation Testing

Validasi perangkat lunak diperoleh melalui sederetan pengujian *black box* yang memperlihatkan kesesuaian dengan kebutuhan perangkat lunak. Validasi dikatakan berhasil jika perangkat lunak berfungsi sesuai kebutuhan pengguna.

#### 6.3.1 Pengujian Pada Sisi Petugas

Pada Tabel 6.4 menunjukkan hasil pengujian validasi pada fitur *login* yang dilakukan oleh petugas perpustakaan.

**Tabel 6.4 Pengujian Login Pada Sisi Petugas**

<b>Nomor Kebutuhan</b>	SP-F-026
<b>Kasus Uji</b>	Melakukan <i>login</i>
<b>Tujuan</b>	Memastikan sistem dapat memvalidasi pengguna yang memiliki hak akses untuk masuk ke sistem
<b>Prosedur</b>	1. Petugas memilih menu <i>Login</i> Petugas 2. Sistem menampilkan form <i>login</i> petugas 3. Petugas mengisi form <i>login</i> dengan <i>username</i> dan <i>password</i> 4. Petugas memilih tombol Login
<b>Hasil Yang Diharapkan</b>	Sistem akan memvalidasi pengguna dan menampilkan halaman <i>dashboard</i>
<b>Hasil</b>	Sistem berhasil memvalidasi pengguna dan menampilkan halaman <i>dashboard</i>
<b>Status</b>	Valid

Pada Tabel 6.5 menunjukkan hasil pengujian validasi pada menu koleksi buku yang dilakukan oleh petugas perpustakaan.

**Tabel 6.5 Pengujian Menampilkan Data Buku**

<b>Nomor Kebutuhan</b>	SP-F-002
<b>Kasus Uji</b>	Menampilkan data buku

<b>Tujuan</b>	Memastikan sistem dapat menampilkan data koleksi buku
<b>Prosedur</b>	1. Petugas memilih menu Koleksi Buku
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan data koleksi buku
<b>Hasil</b>	Sistem menampilkan data koleksi buku
<b>Status</b>	Valid

Pada Tabel 6.6 menunjukkan hasil pengujian validasi pada fitur tambah koleksi buku yang dilakukan oleh petugas perpustakaan.

**Tabel 6.6 Pengujian Menambahkan Data Buku**

<b>Nomor Kebutuhan</b>	SP-F-003
<b>Kasus Uji</b>	Menambah data buku
<b>Tujuan</b>	Memastikan sistem dapat menambahkan data buku baru
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Koleksi Buku</li> <li>2. Petugas memilih tombol tambah buku</li> <li>3. Petugas memasukkan data lengkap buku baru</li> <li>4. Petugas memilih tombol Simpan untuk menyimpan data baru</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan menambahkan data buku baru dan menampilkan kembali data koleksi buku
<b>Hasil</b>	Sistem menambahkan data buku baru dan menampilkan kembali data koleksi buku
<b>Status</b>	Valid

Pada Tabel 6.7 menunjukkan hasil pengujian validasi pada fitur hapus data buku yang dilakukan oleh petugas perpustakaan.

**Tabel 6.7 Pengujian Menghapus Data Buku**

<b>Nomor Kebutuhan</b>	SP-F-005
<b>Kasus Uji</b>	Menghapus data buku
<b>Tujuan</b>	Memastikan sistem dapat menghapus data buku
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Koleksi Buku</li> <li>2. Petugas memilih tombol hapus pada salah satu data buku yang ingin dihapus</li> <li>3. Sistem menampilkan kotak dialog konfirmasi hapus data buku</li> <li>4. Petugas memilih tombol OK pada kotak dialog</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan menghapus data buku yang dipilih dan menampilkan kembali data koleksi buku

<b>Hasil</b>	Sistem menghapus data buku yang dipilih dan menampilkan kembali data koleksi buku
<b>Status</b>	Valid

Pada Tabel 6.8 menunjukkan hasil pengujian validasi pada fitur perbarui data buku yang dilakukan oleh petugas perpustakaan.

**Tabel 6.8 Pengujian Meperbarui Data Buku**

<b>Kasus Uji</b>	Memperbarui data buku
<b>Nomor Kebutuhan</b>	SP-F-004
<b>Tujuan</b>	Memastikan sistem dapat memperbarui data buku
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Koleksi Buku</li> <li>2. Petugas memilih tombol edit pada salah satu data buku yang ingin diperbarui</li> <li>3. Sistem menampilkan form edit data buku</li> <li>4. Petugas mengisi form edit dengan informasi data buku yang baru</li> <li>5. Petugas memilih tombol Simpan</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan memperbarui data buku yang dipilih dan menampilkan kembali data koleksi buku
<b>Hasil</b>	Sistem memperbarui data buku yang dipilih dan menampilkan kembali data koleksi buku
<b>Status</b>	Valid

Pada Tabel 6.9 menunjukkan hasil pengujian validasi pada menu *Dashboard* yang dilakukan oleh petugas perpustakaan.

**Tabel 6.9 Pengujian Menampilkan Dashboard**

<b>Nomor Kebutuhan</b>	SP-F-001
<b>Kasus Uji</b>	Menampilkan <i>Dashboard</i>
<b>Tujuan</b>	Memastikan sistem dapat menampilkan <i>dashboard</i>
<b>Prosedur</b>	1. Petugas memilih menu <i>Dashboard</i>
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan <i>dashboard</i>
<b>Hasil</b>	Sistem menampilkan <i>dashboard</i>
<b>Status</b>	Valid

Pada Tabel 6.10 menunjukkan hasil pengujian validasi pada menu untuk menampilkan data anggota yang dilakukan oleh petugas perpustakaan.

**Tabel 6.10 Pengujian Menampilkan Data Anggota**

<b>Nomor Kebutuhan</b>	SP-F-010
<b>Kasus Uji</b>	Menampilkan data anggota

<b>Tujuan</b>	Memastikan sistem dapat menampilkan data anggota perpustakaan
<b>Prosedur</b>	1. Petugas memilih menu Data Anggota
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan data anggota perpustakaan
<b>Hasil</b>	Sistem menampilkan data anggota perpustakaan
<b>Status</b>	Valid

Pada Tabel 6.11 menunjukkan hasil pengujian validasi pada fitur tambah data anggota yang dilakukan oleh petugas perpustakaan.

**Tabel 6.11 Pengujian Menambahkan Data Anggota Pada Sisi Petugas**

<b>Nomor Kebutuhan</b>	SP-F-011
<b>Kasus Uji</b>	Menambah data anggota
<b>Tujuan</b>	Memastikan sistem dapat menambahkan data anggota baru
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Data Anggota</li> <li>2. Petugas memilih tombol tambah anggota</li> <li>3. Petugas memasukkan data lengkap anggota</li> <li>4. Petugas memilih tombol Tambah untuk menyimpan data anggota baru</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan menambahkan data anggota baru dan menampilkan kembali data anggota
<b>Hasil</b>	Sistem menambahkan data anggota baru dan menampilkan kembali data anggota
<b>Status</b>	Valid

Pada Tabel 6.12 menunjukkan hasil pengujian validasi pada fitur hapus data anggota yang dilakukan oleh petugas perpustakaan.

**Tabel 6.12 Pengujian Menghapus Data Anggota**

<b>Nomor Kebutuhan</b>	SP-F-013
<b>Kasus Uji</b>	Menghapus data anggota
<b>Tujuan</b>	Memastikan sistem dapat menghapus data anggota
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Data Anggota</li> <li>2. Petugas memilih tombol hapus pada salah satu data anggota yang ingin dihapus</li> <li>3. Sistem menampilkan kotak dialog konfirmasi hapus data anggota</li> <li>4. Petugas memilih tombol OK pada kotak dialog</li> </ol>

<b>Hasil Yang Diharapkan</b>	Sistem akan menghapus data anggota yang dipilih dan menampilkan kembali data anggota
<b>Hasil</b>	Sistem menghapus data anggota yang dipilih dan menampilkan kembali data anggota
<b>Status</b>	Valid

Pada Tabel 6.13 menunjukkan hasil pengujian validasi pada fitur perbarui data anggota yang dilakukan oleh petugas perpustakaan.

**Tabel 6.13 Pengujian Meperbarui Data Anggota Pada Sisi Petugas**

<b>Nomor Kebutuhan</b>	SP-F-012
<b>Kasus Uji</b>	Memperbarui data anggota
<b>Tujuan</b>	Memastikan sistem dapat memperbarui data anggota
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Data Anggota</li> <li>2. Petugas memilih tombol edit pada salah satu data anggota yang ingin diperbarui</li> <li>3. Sistem menampilkan form edit data anggota</li> <li>4. Petugas mengisi form edit dengan informasi data anggota yang baru</li> <li>5. Petugas memilih tombol Simpan</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan memperbarui data anggota yang dipilih dan menampilkan kembali data anggota
<b>Hasil</b>	Sistem memperbarui data anggota yang dipilih dan menampilkan kembali data anggota
<b>Status</b>	Valid

Pada Tabel 6.14 menunjukkan hasil pengujian validasi pada fitur cetak kartu anggota yang dilakukan oleh petugas perpustakaan.

**Tabel 6.14 Pengujian Mencetak Kartu Anggota**

<b>Nomor Kebutuhan</b>	SP-F-024
<b>Kasus Uji</b>	Mencetak kartu anggota
<b>Tujuan</b>	Memastikan sistem dapat mencetak kartu anggota
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Data Anggota</li> <li>2. Petugas memilih tombol cetak pada salah satu data anggota</li> <li>3. Sistem menampilkan halaman informasi kartu anggota yang akan dicetak dan menampilkan jendela konfirmasi cetak kartu anggota</li> <li>4. Petugas memilih tombol Print untuk mencetak kartu anggota</li> </ol>

<b>Hasil Yang Diharapkan</b>	Sistem akan mencetak kartu anggota dari salah satu data anggota yang dipilih
<b>Hasil</b>	Sistem mencetak kartu anggota dari salah satu data anggota yang dipilih
<b>Status</b>	Valid

Pada Tabel 6.15 menunjukkan hasil pengujian validasi pada menu transaksi peminjaman yang dilakukan oleh petugas perpustakaan.

**Tabel 6.15 Pengujian Menampilkan Data Transaksi Peminjaman**

<b>Nomor Kebutuhan</b>	SP-F-006
<b>Kasus Uji</b>	Menampilkan data transaksi peminjaman
<b>Tujuan</b>	Memastikan sistem dapat menampilkan data transaksi peminjaman
<b>Prosedur</b>	1. Petugas memilih menu Transaksi
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan data transaksi peminjaman
<b>Hasil</b>	Sistem menampilkan data transaksi peminjaman
<b>Status</b>	Valid

Pada Tabel 6.16 menunjukkan hasil pengujian validasi pada fitur tambah transaksi peminjaman yang dilakukan oleh petugas perpustakaan.

**Tabel 6.16 Pengujian Menambahkan Data Transaksi Peminjaman**

<b>Nomor Kebutuhan</b>	SP-F-007
<b>Kasus Uji</b>	Menambah data transaksi peminjaman
<b>Tujuan</b>	Memastikan sistem dapat menambahkan data transaksi peminjaman baru
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Transaksi</li> <li>2. Petugas mengisi form peminjaman meliputi ID anggota dan nomor buku</li> <li>3. Petugas memilih tombol Submit</li> <li>4. Petugas mengisi lagi form nomor buku apabila buku yang dipinjam lebih dari 1</li> <li>5. Petugas memilih tombol Simpan</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan menambahkan data transaksi peminjaman baru dan menampilkan halaman struk peminjaman untuk dicetak
<b>Hasil</b>	Sistem menambahkan data transaksi peminjaman baru dan menampilkan halaman struk peminjaman untuk dicetak

<b>Status</b>	Valid
---------------	-------

Pada Tabel 6.17 menunjukkan hasil pengujian validasi pada fitur hapus data transaksi peminjaman yang dilakukan oleh petugas perpustakaan.

**Tabel 6.17 Pengujian Menghapus Data Transaksi Peminjaman**

<b>Nomor Kebutuhan</b>	SP-F-009
<b>Kasus Uji</b>	Menghapus data transaksi peminjaman
<b>Tujuan</b>	Memastikan sistem dapat menghapus data transaksi peminjaman
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Transaksi</li> <li>2. Petugas memilih tombol hapus pada salah satu data peminjaman yang ingin dihapus</li> <li>3. Sistem menampilkan kotak dialog konfirmasi hapus data peminjaman</li> <li>4. Petugas memilih tombol OK pada kotak dialog</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan menghapus data transaksi peminjaman yang dipilih dan menampilkan kembali halaman transaksi peminjaman
<b>Hasil</b>	Sistem menghapus data transaksi peminjaman yang dipilih dan menampilkan kembali halaman transaksi peminjaman
<b>Status</b>	Valid

Pada Tabel 6.18 menunjukkan hasil pengujian validasi pada fitur perbarui data transaksi peminjaman yang dilakukan oleh petugas perpustakaan.

**Tabel 6.18 Pengujian Meperbarui Data Transaksi Peminjaman**

<b>Nomor Kebutuhan</b>	SP-F-008
<b>Kasus Uji</b>	Memperbarui data transaksi peminjaman
<b>Tujuan</b>	Memastikan sistem dapat memperbarui data transaksi peminjaman
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Transaksi</li> <li>2. Petugas memilih tombol edit pada salah satu data peminjaman yang ingin diperbarui</li> <li>3. Sistem menampilkan form edit data peminjaman</li> <li>4. Petugas mengisi form edit dengan informasi data peminjaman yang baru</li> <li>5. Petugas memilih tombol Simpan</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan memperbarui data transaksi peminjaman yang dipilih dan menampilkan kembali halaman data transaksi peminjaman

<b>Hasil</b>	Sistem memperbarui data transaksi peminjaman yang dipilih dan menampilkan kembali halaman data transaksi peminjaman
<b>Status</b>	Valid

Pada Tabel 6.19 menunjukkan hasil pengujian validasi pada fitur untuk menampilkan denda peminjam yang dilakukan oleh petugas perpustakaan.

**Tabel 6.19 Pengujian Menampilkan Denda Peminjam**

<b>Nomor Kebutuhan</b>	SP-F-022
<b>Kasus Uji</b>	Menampilkan denda peminjaman
<b>Tujuan</b>	Memastikan sistem dapat menampilkan denda peminjam yang telat mengembalikan buku
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Transaksi</li> <li>2. Petugas masukkan kode peminjaman pada kolom pencarian tabel data transaksi peminjaman</li> <li>3. Petugas mengklik kode peminjaman yang telah ditemukan</li> <li>4. Sistem menampilkan detail informasi peminjaman termasuk denda peminjam yang belum mengembalikan buku</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan detail informasi peminjaman termasuk denda peminjam yang telat mengembalikan buku
<b>Hasil</b>	Sistem menampilkan detail informasi peminjaman termasuk denda peminjam yang telat mengembalikan buku
<b>Status</b>	Valid

Pada Tabel 6.20 menunjukkan hasil pengujian validasi pada menu data tamu yang dilakukan oleh petugas perpustakaan.

**Tabel 6.20 Pengujian Menampilkan Data Tamu**

<b>Nomor Kebutuhan</b>	SP-F-018
<b>Kasus Uji</b>	Menampilkan data tamu
<b>Tujuan</b>	Memastikan sistem dapat menampilkan data tamu perpustakaan
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Data Tamu</li> <li>2. Petugas memilih Anggota untuk melihat data tamu yang sudah terdaftar menjadi anggota</li> <li>3. Petugas memilih Penguunjung untuk melihat data tamu yang belum terdaftar menjadi anggota</li> <li>4. Sistem menampilkan halaman data tamu</li> </ol>

	perpustakaan
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan data tamu
<b>Hasil</b>	Sistem menampilkan data tamu
<b>Status</b>	Valid

Pada Tabel 6.21 menunjukkan hasil pengujian validasi pada fitur hapus data tamu yang dilakukan oleh petugas perpustakaan.

**Tabel 6.21 Pengujian Menghapus Data Tamu**

<b>Nomor Kebutuhan</b>	SP-F-020
<b>Kasus Uji</b>	Menghapus data tamu
<b>Tujuan</b>	Memastikan sistem dapat menghapus data tamu
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Data Tamu</li> <li>2. Petugas memilih tombol hapus pada salah satu data tamu yang ingin dihapus</li> <li>3. Sistem menampilkan kotak dialog konfirmasi hapus data tamu</li> <li>4. Petugas memilih tombol OK pada kotak dialog</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan menghapus data tamu yang dipilih dan menampilkan kembali data tamu
<b>Hasil</b>	Sistem menghapus data tamu yang dipilih dan menampilkan kembali data tamu
<b>Status</b>	Valid

Pada Tabel 6.22 menunjukkan hasil pengujian validasi pada menu Data Inventaris yang dilakukan oleh petugas perpustakaan.

**Tabel 6.22 Pengujian Menampilkan Data Inventaris**

<b>Nomor Kebutuhan</b>	SP-F-014
<b>Kasus Uji</b>	Menampilkan data inventaris
<b>Tujuan</b>	Memastikan sistem dapat menampilkan data inventaris perpustakaan
<b>Prosedur</b>	1. Petugas memilih menu Data Inventaris
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan data inventaris perpustakaan
<b>Hasil</b>	Sistem menampilkan data inventaris perpustakaan
<b>Status</b>	Valid

Pada Tabel 6.23 menunjukkan hasil pengujian validasi pada fitur tambah data inventaris yang dilakukan oleh petugas perpustakaan.

**Tabel 6.23 Pengujian Menambahkan Data Inventaris**

<b>Nomor Kebutuhan</b>	SP-F-015
<b>Kasus Uji</b>	Menambah data inventaris
<b>Tujuan</b>	Memastikan sistem dapat menambahkan data inventaris perpustakaan
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Data Inventaris</li> <li>2. Petugas memilih tombol tambah inventaris</li> <li>3. Petugas memasukkan data lengkap inventaris baru</li> <li>4. Petugas memilih tombol Simpan untuk menyimpan data baru</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan menambahkan data inventaris baru dan menampilkan kembali data inventaris
<b>Hasil</b>	Sistem menambahkan data inventaris baru dan menampilkan kembali data inventaris
<b>Status</b>	Valid

Pada Tabel 6.24 menunjukkan hasil pengujian validasi pada fitur hapus data inventaris yang dilakukan oleh petugas perpustakaan.

**Tabel 6.24 Pengujian Menghapus Data Inventaris**

<b>Nomor Kebutuhan</b>	SP-F-017
<b>Kasus Uji</b>	Menghapus data inventaris
<b>Tujuan</b>	Memastikan sistem dapat menghapus data inventaris perpustakaan
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Data Inventaris</li> <li>2. Petugas memilih tombol hapus pada salah satu data inventaris yang ingin dihapus</li> <li>3. Sistem menampilkan kotak dialog konfirmasi hapus data inventaris</li> <li>4. Petugas memilih tombol OK pada kotak dialog</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan menghapus data inventaris yang dipilih dan menampilkan kembali data inventaris
<b>Hasil</b>	Sistem menghapus data inventaris yang dipilih dan menampilkan kembali data inventaris
<b>Status</b>	Valid

Pada Tabel 6.25 menunjukkan hasil pengujian validasi pada fitur perbarui data inventaris yang dilakukan oleh petugas perpustakaan.

**Tabel 6.25 Pengujian Meperbarui Data Inventaris**

<b>Nomor Kebutuhan</b>	SP-F-016
------------------------	----------

<b>Kasus Uji</b>	Memperbarui data inventaris
<b>Tujuan</b>	Memastikan sistem dapat memperbarui data inventaris perpustakaan
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Data Inventaris</li> <li>2. Petugas memilih tombol edit pada salah satu data inventaris yang ingin diperbarui</li> <li>3. Sistem menampilkan form edit data inventaris</li> <li>4. Petugas mengisi form edit dengan informasi data inventaris yang baru</li> <li>5. Petugas memilih tombol Simpan</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan memperbarui data inventaris yang dipilih dan menampilkan kembali data inventaris
<b>Hasil</b>	Sistem memperbarui data inventaris yang dipilih dan menampilkan kembali data inventaris
<b>Status</b>	Valid

Pada Tabel 6.26 menunjukkan hasil pengujian validasi pada fitur cetak struk peminjaman yang dilakukan oleh petugas perpustakaan.

**Tabel 6.26 Pengujian Mencetak Struk Peminjaman**

<b>Nomor Kebutuhan</b>	SP-F-025
<b>Kasus Uji</b>	Mencetak struk peminjaman
<b>Tujuan</b>	Memastikan sistem dapat mencetak struk peminjaman
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Petugas memilih menu Transaksi</li> <li>2. Petugas memilih tombol cetak pada salah satu data peminjaman</li> <li>3. Sistem menampilkan halaman informasi struk peminjaman yang akan dicetak dan menampilkan jendela konfirmasi cetak struk peminjaman</li> <li>4. Petugas memilih tombol Print untuk mencetak struk peminjaman</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan mencetak struk peminjaman dari salah satu data peminjaman yang dipilih
<b>Hasil</b>	Sistem mencetak struk peminjaman dari salah satu data peminjaman yang dipilih
<b>Status</b>	Valid

### 6.3.2 Pengujian Pada Sisi Anggota

Pada Tabel 6.27 menunjukkan hasil pengujian validasi pada fitur *login* yang dilakukan oleh anggota perpustakaan.

**Tabel 6.27 Pengujian *Login* Pada Sisi Anggota**

<b>Nomor Kebutuhan</b>	SP-F-026
<b>Kasus Uji</b>	Melakukan <i>login</i>
<b>Tujuan</b>	Memastikan sistem dapat memvalidasi pengguna yang memiliki hak akses untuk masuk ke sistem
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Anggota memilih menu <i>Login</i> untuk anggota</li> <li>2. Sistem menampilkan form <i>login</i> anggota</li> <li>3. Anggota mengisi form <i>login</i> dengan <i>username</i> dan <i>password</i></li> <li>4. Anggota memilih tombol Login</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan memvalidasi pengguna dan menampilkan halaman <i>dashboard</i>
<b>Hasil</b>	Sistem berhasil memvalidasi pengguna dan menampilkan halaman <i>dashboard</i>
<b>Status</b>	Valid

Pada Tabel 6.28 menunjukkan hasil pengujian validasi pada fitur perbarui data anggota yang dilakukan oleh anggota perpustakaan.

**Tabel 6.28 Pengujian Memperbarui Data Anggota Pada Sisi Anggota**

<b>Nomor Kebutuhan</b>	SP-F-012
<b>Kasus Uji</b>	Memperbarui data anggota
<b>Tujuan</b>	Memastikan sistem dapat memperbarui data anggota
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Anggota memilih menu Pengaturan</li> <li>2. Anggota mengisi form pengaturan yang perlu diperbarui</li> <li>3. Anggota memilih tombol Simpan</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan memperbarui data anggota
<b>Hasil</b>	Sistem memperbarui data anggota
<b>Status</b>	Valid

Pada Tabel 6.29 menunjukkan hasil pengujian validasi pada fitur pencarian buku yang dilakukan oleh anggota perpustakaan.

**Tabel 6.29 Pengujian Mencari Buku Pada Sisi Anggota**

<b>Nomor Kebutuhan</b>	SP-F-021
------------------------	----------

<b>Kasus Uji</b>	Mencari Buku
<b>Tujuan</b>	Memastikan sistem dapat menampilkan hasil pencarian buku
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Anggota memilih menu Cari Buku</li> <li>2. Anggota memilih kategori buku dan mengisi kata kunci buku yang ingin dicari</li> <li>3. Anggota mengklik tombol cari</li> <li>4. Sistem menampilkan detail informasi buku yang telah ditemukan sesuai pencarian</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan detail informasi buku yang diinginkan
<b>Hasil</b>	Sistem menampilkan detail informasi buku yang diinginkan
<b>Status</b>	Valid

Pada Tabel 6.30 menunjukkan hasil pengujian validasi pada fitur tambah data tamu yang dilakukan oleh anggota perpustakaan.

**Tabel 6.30 Pengujian Menambahkan Data Tamu Pada Sisi Anggota**

<b>Nomor Kebutuhan</b>	SP-F-019
<b>Kasus Uji</b>	Menambah data tamu
<b>Tujuan</b>	Memastikan sistem dapat menambahkan data tamu
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Anggota memilih menu Absensi Tamu</li> <li>2. Anggota memilih tombol Anggota karena pengguna sudah terdaftar sebagai anggota</li> <li>3. Anggota mengisi form absensi tamu perpustakaan</li> <li>4. Anggota memilih tombol Submit untuk menyimpan data tamu baru</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan menambahkan data tamu baru
<b>Hasil</b>	Sistem menambahkan data tamu baru
<b>Status</b>	Valid

Pada Tabel 6.31 menunjukkan hasil pengujian validasi pada menu riwayat peminjaman yang dilakukan oleh anggota perpustakaan.

**Tabel 6.31 Pengujian Menampilkan Riwayat Peminjaman**

<b>Nomor Kebutuhan</b>	SP-F-023
<b>Kasus Uji</b>	Menampilkan Riwayat Peminjaman
<b>Tujuan</b>	Memastikan sistem dapat menampilkan riwayat peminjaman

<b>Prosedur</b>	1. Pengunjung memilih menu Daftar Peminjaman
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan daftar riwayat peminjaman
<b>Hasil</b>	Sistem menampilkan daftar riwayat peminjaman
<b>Status</b>	Valid

### 6.3.3 Pengujian Pada Sisi Pengunjung

Pada Tabel 6.32 menunjukkan hasil pengujian validasi pada fitur tambah data anggota yang dilakukan oleh pengunjung perpustakaan.

**Tabel 6.32 Pengujian Menambahkan Data Anggota Pada Sisi Pengunjung**

<b>Nomor Kebutuhan</b>	SP-F-011
<b>Kasus Uji</b>	Menambah data anggota
<b>Tujuan</b>	Memastikan sistem dapat menambahkan data anggota baru
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Pengunjung memilih menu Register Anggota</li> <li>2. Pengunjung memasukkan data diri lengkap pada form pendaftaran anggota baru</li> <li>3. Pengunjung memilih tombol Daftar untuk menyimpan data anggota baru</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan menambahkan data anggota baru
<b>Hasil</b>	Sistem menambahkan data anggota baru
<b>Status</b>	Valid

Pada Tabel 6.33 menunjukkan hasil pengujian validasi pada fitur pencarian buku yang dilakukan oleh pengunjung perpustakaan.

**Tabel 6.33 Pengujian Mencari Buku Pada Sisi Pengunjung**

<b>Nomor Kebutuhan</b>	SP-F-021
<b>Kasus Uji</b>	Mencari Buku
<b>Tujuan</b>	Memastikan sistem dapat menampilkan hasil pencarian buku
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Pengunjung memilih menu Cari Buku</li> <li>2. Pengunjung memilih kategori buku dan mengisi kata kunci buku yang ingin dicari</li> <li>3. Pengunjung mengklik tombol cari</li> <li>4. Sistem menampilkan detail informasi buku yang telah ditemukan sesuai pencarian</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan detail informasi buku yang diinginkan

<b>Hasil</b>	Sistem menampilkan detail informasi buku yang diinginkan
<b>Status</b>	Valid

Pada Tabel 6.34 menunjukkan hasil pengujian validasi pada fitur tambah data tamu yang dilakukan oleh pengunjung perpustakaan.

**Tabel 6.34 Pengujian Menambahkan Data Tamu Pada Sisi Pengunjung**

<b>Nomor Kebutuhan</b>	SP-F-019
<b>Kasus Uji</b>	Menambah data tamu
<b>Tujuan</b>	Memastikan sistem dapat menambahkan data tamu
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Pengunjung memilih menu Absensi Tamu</li> <li>2. Pengunjung memilih tombol Anggota karena pengguna sudah terdaftar sebagai anggota</li> <li>3. Pengunjung mengisi form absensi tamu perpustakaan</li> <li>4. Pengunjung memilih tombol Submit untuk menyimpan data tamu baru</li> </ol>
<b>Hasil Yang Diharapkan</b>	Sistem akan menambahkan data tamu baru
<b>Hasil</b>	Sistem menambahkan data tamu baru
<b>Status</b>	Valid

## 6.4 Reliability Testing

Pengujian reliabilitas dilakukan dengan menganalisa hasil pengujian stress menggunakan *tool* WAPT. Pengujian akan menghasilkan nilai *success rate* dan *failed rate*.

Test execution parameters:  
 Test status: finished  
 Test started at: 12/6/2018 4:40:02 PM  
 Scenario name:  
 Test run comment:  
 Test executed by: DELL (DESKTOP-2A70018)  
 Test executed on: DESKTOP-2A70018

Response codes

Code	Request	Pages	Hits
Profile1	-	50	110
200 OK	Profile1.All	50	108
200 OK	Profile1.page_1: http://localhost/epusbu/index.php	1	14
	Profile1.page_2: http://localhost/epusbu/index.php	1	12
	Profile1.page_3: http://localhost/epusbu/index.php	1	1
	Profile1.page_4: http://localhost/epusbu/index.php	1	1
	Profile1.page_5: http://localhost/epusbu/index.php	1	29
	Profile1.page_6: http://localhost/epusbu/pages/admin/detail-data-entry.php	1	1
	Profile1.page_7: http://localhost/epusbu/pages/admin/detail-data-entry.php	1	1
	Profile1.page_8: http://localhost/epusbu/index.php	1	1
	Profile1.page_9: http://localhost/epusbu/index.php	1	1
	Profile1.page_10: http://localhost/epusbu/index.php	1	4
	Profile1.page_11: http://localhost/epusbu/index.php	1	1
	Profile1.page_12: http://localhost/epusbu/index.php	1	1
	Profile1.page_13: http://localhost/epusbu/pages/admin/edit-buku.php	1	1
	Profile1.page_14: http://localhost/epusbu/index.php	1	1
	Profile1.page_15: http://localhost/epusbu/index.php	1	1
	Profile1.page_16: http://localhost/epusbu/index.php	1	1
	Profile1.page_17: http://localhost/epusbu/index.php	1	1
	Profile1.page_18: http://localhost/epusbu/index.php	1	1
	Profile1.page_19: http://localhost/epusbu/index.php	1	1
	Profile1.page_20: http://localhost/epusbu/index.php	1	1
	Profile1.page_21: http://localhost/epusbu/pages/admin/edit-anggota.php	1	1
	Profile1.page_22: http://localhost/epusbu/index.php	1	1
	Profile1.page_23: http://localhost/epusbu/index.php	1	1
	Profile1.page_24: http://localhost/epusbu/index.php	1	3
	Profile1.page_25: http://localhost/epusbu/index.php	1	1
	Profile1.page_26: http://localhost/epusbu/index.php	1	1
	Profile1.page_27: http://localhost/epusbu/index.php	1	1
	Profile1.page_28: http://localhost/epusbu/index.php	1	1
	Profile1.page_29: http://localhost/epusbu/index.php	1	1
	Profile1.page_30: http://localhost/epusbu/pages/admin/edit-inventaris.php	1	1
	Profile1.page_31: http://localhost/epusbu/index.php	1	1
	Profile1.page_32: http://localhost/epusbu/index.php	1	1
	Profile1.page_33: http://localhost/epusbu/index.php	1	1
	Profile1.page_34: http://localhost/epusbu/index.php	1	1
	Profile1.page_35: http://localhost/epusbu/index.php	1	2
	Profile1.page_36: http://localhost/epusbu/index.php	1	1
	Profile1.page_37: http://localhost/epusbu/index.php	1	1
	Profile1.page_38: http://localhost/epusbu/index.php	1	1
	Profile1.page_39: http://localhost/epusbu/index.php	1	1
	Profile1.page_40: http://localhost/epusbu/index.php	1	1
	Profile1.page_41: http://localhost/epusbu/index.php	1	1
	Profile1.page_42: http://localhost/epusbu/pages/client/cari-buku.php	1	1
	Profile1.page_43: http://localhost/epusbu/index.php	1	1
	Profile1.page_44: http://localhost/epusbu/index.php	1	1
	Profile1.page_45: http://localhost/epusbu/index.php	1	1
	Profile1.page_46: http://localhost/epusbu/pages/client/detail-rwayat.php	1	1
	Profile1.page_47: http://localhost/epusbu/pages/client/cari-buku.php	1	1
	Profile1.page_48: http://localhost/epusbu/index.php	1	1
	Profile1.page_49: http://localhost/epusbu/index.php	1	1
	Profile1.page_50: http://localhost/epusbu/index.php	1	1
302 Found	Profile1.page_3: http://localhost/epusbu/index.php	1	1
	Profile1.page_38: http://localhost/epusbu/index.php	1	1

Report generated at 06.12.2018 16:40:06 by WAPT 10.0 © SoftLogics 2018

**Gambar 6.2 Hasil Stress Testing Menggunakan WAPT**

Gambar 6.2 merupakan hasil pengujian stress dengan menggunakan *tool* WAPT. Diketahui bahwa jumlah *successful pages* sebanyak 50, *successful hits* sebanyak 108, dan *failed hits* sebanyak 2. Hasil tersebut dapat disajikan dalam bentuk tabel seperti pada tabel 6.35.

**Tabel 6.35 Hasil Stress Testing**

Metrik	Sukses	Gagal
Pages	50	0
Hits	110	2
Total	160	2

Kemudian dilakukan perhitungan nilai *reliability* dengan rumus nelson, sebagai berikut:

Diketahui:  $ne = 2$  (jumlah input yang gagal),  $n = 160$  (jumlah seluruh input)

Perhitungan:

$$R1 = 1 - \frac{ne}{n}$$

$$R1 = 1 - \frac{2}{160}$$

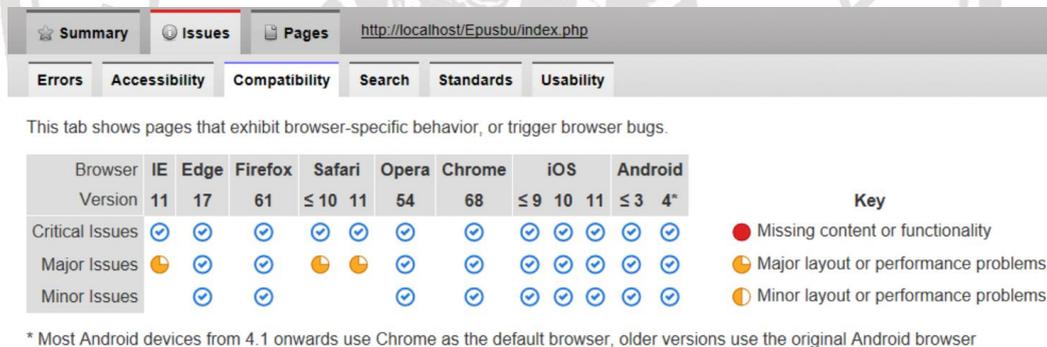
$$R1 = 1 - 0.0125$$

$$R1 = 0.9875$$

Dari hasil perhitungan, menunjukkan bahwa skor kelayakan sistem EPUSBU sebesar 0,9875. jika dinyatakan dalam bentuk presentase maka sebesar 98,75%. Menurut standar Telcordia, reliabilitas perangkat lunak dinyatakan lolos bila mencapai minimal 95% keberhasilan jika diuji, hal ini berarti bahwa sistem ini telah memenuhi aspek *reliability*.

### 6.5 Compatibility Testing

*Compatibility testing* memiliki fokus pada persyaratan pengujian non-fungsional yang diterapkan pada sisi browser. Pengujian ini dilakukan dengan menggunakan *tool* Sortsite dari PowerMapper. Dengan *tool* ini, kita bisa mengetahui apakah sistem EPUSBU dapat berjalan di berbagai *browser* atau tidak.



**Gambar 6.3 Hasil *Compatibility Testing* Menggunakan Sortsite**

Hasil pengujian kompatibilitas dengan menggunakan *tool* Sortsite ditunjukkan pada Gambar 6.3 terlihat bahwa sistem informasi perpustakaan yang dibangun dalam penelitian ini dapat digunakan tanpa ada masalah konten atau fungsi yang tidak didukung pada beberapa *browser* maupun *mobile*. Namun, masih terdapat masalah tampilan atau performa pada *browser* Internet Explorer versi 11 dan Safari versi 10 dan 11. Kesimpulannya *browser* Internet Explorer versi 11 dan Safari versi 10 dan 11 tidak disarankan untuk menjalankan sistem yang dibangun pada penelitian ini.



### 1.6 Acceptance Testing

Pengujian penerimaan pengguna dilakukan terhadap *stakeholder* atau penanggung jawab sistem dari pihak perpustakaan untuk memverifikasi bahwa solusi yang dibuat dalam sistem sudah sesuai dengan kebutuhan pengguna. Tabel 6.36 merupakan hasil uji *user acceptance testing* yang sudah dilakukan oleh *stakeholder*.

**Tabel 6.36 Hasil Uji *User Acceptance* yang Dilakukan *Stakeholder***

No	Proses	Berhasil/Gagal
1	<p>Nama Uji : Melakukan <i>Login</i></p> <p>Deskripsi Pengujian : Verifikasi melakukan autentifikasi pengguna masuk ke sistem</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Username : nurur23</li> <li>- Password : adm12345</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan memverifikasi pengguna sesuai hak aksesnya terhadap sistem</li> </ul>	Berhasil
2	<p>Nama Uji : Menampilkan <i>dashboard</i></p> <p>Deskripsi Pengujian : Verifikasi menampilkan <i>dashboard</i></p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Memilih menu <i>Dashboard</i></li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menampilkan halaman <i>dashboard</i></li> </ul>	Berhasil
3	<p>Nama Uji : Menampilkan data buku</p> <p>Deskripsi Pengujian : Verifikasi menampilkan data koleksi buku</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Memilih menu Koleksi Buku</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menampilkan halaman data koleksi buku</li> </ul>	Berhasil
4	<p>Nama Uji : Menambah data buku</p> <p>Deskripsi Pengujian : Verifikasi memasukkan data buku</p>	Berhasil

	<p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Nomor Buku : 00481</li> <li>- Tanggal Masuk : 13-12-2007</li> <li>- Judul Buku : Intisari Ilmu: Hewan Merayap</li> <li>- Kolasi : 120 hlm: 12x18 cm</li> <li>- Pengarang : Steve Sefford</li> <li>- Penerbit : Erlangga</li> <li>- Tahun Terbit : 2005</li> <li>- Asal : beli</li> <li>- No Klasifikasi : 597.9.0</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menambahkan data buku baru</li> </ul>	
5	<p>Nama Uji : Menghapus data buku</p> <p>Deskripsi Pengujian : Verifikasi menghapus data buku</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Memilih tombol hapus pada kolom tabel data buku</li> <li>- Memilih tombol 'OK' pada jendela konfirmasi</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menghapus data buku yang dipilih dan menampilkan kembali data koleksi</li> </ul>	Berhasil
6	<p>Nama Uji : Memperbarui data buku</p> <p>Deskripsi Pengujian : Verifikasi memperbarui data buku</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Judul Buku : Intisari Ilmu: Mamalia</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan memperbarui data buku yang dipilih dan menampilkan kembali data koleksi</li> </ul>	Berhasil
7	<p>Nama Uji : Menampilkan data anggota</p> <p>Deskripsi Pengujian : Verifikasi menampilkan data anggota</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Memilih menu Data Anggota</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menampilkan data anggota</li> </ul>	Berhasil

	perpustakaan	
8	<p>Nama Uji : Menambah data anggota</p> <p>Deskripsi Pengujian : Verifikasi memasukkan data anggota</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Nama Lengkap : Retno Dewi Anggraini</li> <li>- Tempat Lahir : Gresik</li> <li>- Tanggal Lahir : 27-07-1996</li> <li>- Jenis Kelamin : Perempuan</li> <li>- Alamat : Karangrejo</li> <li>- Status : Mahasiswi</li> <li>- No Telepon : 0822333311</li> <li>- Username : rain</li> <li>- Password : retildetilatil</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menambahkan data anggota baru</li> </ul>	Berhasil
9	<p>Nama Uji : Menghapus data anggota</p> <p>Deskripsi Pengujian : Verifikasi menghapus data anggota</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Memilih tombol hapus pada kolom tabel data anggota</li> <li>- Memilih tombol 'OK' pada jendela konfirmasi</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menghapus data anggota yang dipilih dan menampilkan kembali data anggota</li> </ul>	Berhasil
10	<p>Nama Uji : Memperbarui data anggota</p> <p>Deskripsi Pengujian : Verifikasi memperbarui data anggota</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Nama Lengkap : Retno Dewi</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan memperbarui data anggota yang dipilih dan menampilkan kembali data anggota</li> </ul>	Berhasil
11	<p>Nama Uji : Mencetak kartu anggota</p> <p>Deskripsi Pengujian : Verifikasi mencetak kartu anggota</p>	Berhasil

	<p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Memilih tombol cetak pada kolom tabel data anggota</li> <li>- Memilih tombol print pada jendela perintah untuk mencetak</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan mencetak kartu anggota dari salah satu data anggota yang dipilih</li> </ul>	
12	<p>Nama Uji : Menampilkan data transaksi peminjaman</p> <p>Deskripsi Pengujian : Verifikasi menampilkan data transaksi peminjaman</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Memilih menu Transaksi</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menampilkan data transaksi peminjaman</li> </ul>	Berhasil
13	<p>Nama Uji : Menambah data transaksi peminjaman</p> <p>Deskripsi Pengujian : Verifikasi memasukkan data transaksi peminjaman</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- ID Anggota : 18</li> <li>- Nomor Buku : 00019</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menambahkan data transaksi peminjaman baru dan menampilkan halaman struk peminjaman untuk dicetak</li> </ul>	Berhasil
14	<p>Nama Uji : Menghapus data transaksi peminjaman</p> <p>Deskripsi Pengujian : Verifikasi menghapus data transaksi peminjaman</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Memilih tombol hapus pada kolom tabel data transaksi peminjaman</li> <li>- Memilih tombol 'OK' pada jendela konfirmasi</li> </ul>	Berhasil

	<p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menghapus data transaksi peminjaman yang dipilih dan menampilkan kembali data transaksi peminjaman</li> </ul>	
15	<p>Nama Uji : Memperbarui data transaksi peminjaman</p> <p>Deskripsi Pengujian : Verifikasi memperbarui data transaksi peminjaman</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Nomor Buku : 00017</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan memperbarui data buku yang dipilih dan menampilkan kembali data transaksi peminjaman</li> </ul>	Berhasil
16	<p>Nama Uji : Menampilkan denda peminjaman</p> <p>Deskripsi Pengujian : Verifikasi menampilkan denda peminjam yang telat mengembalikan buku</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Kode Peminjaman : 00021</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menampilkan detail informasi peminjaman termasuk denda peminjam yang telat mengembalikan buku</li> </ul>	Berhasil
17	<p>Nama Uji : Mencetak struk peminjaman</p> <p>Deskripsi Pengujian : Verifikasi mencetak struk peminjaman</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Memilih tombol cetak pada kolom tabel data transaksi</li> <li>- Memilih tombol print pada jendela perintah untuk mencetak</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan mencetak struk peminjaman dari salah satu data peminjaman yang dipilih</li> </ul>	Berhasil
18	<p>Nama Uji : Menampilkan data tamu</p> <p>Deskripsi Pengujian : Verifikasi menampilkan data</p>	Berhasil

	<p>tamu perpustakaan</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Memilih menu Data Tamu</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menampilkan halaman data tamu</li> </ul>	
19	<p>Nama Uji : Menghapus data tamu</p> <p>Deskripsi Pengujian : Verifikasi menghapus data tamu</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Memilih tombol hapus pada kolom tabel data tamu</li> <li>- Memilih tombol 'OK' pada jendela konfirmasi</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menghapus data tamu yang dipilih dan menampilkan kembali data tamu</li> </ul>	Berhasil
20	<p>Nama Uji : Menambah data tamu</p> <p>Deskripsi Pengujian : Verifikasi memasukkan data tamu perpustakaan</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Nama Lengkap : Khoirul</li> <li>- Alamat : Simorejo</li> <li>- Status : Pelajar</li> <li>- Jenis Kelamin : Laki-laki</li> <li>- Keperluan : Membaca</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menambahkan data tamu baru</li> </ul>	Berhasil
21	<p>Nama Uji : Menampilkan data inventaris</p> <p>Deskripsi Pengujian : Verifikasi menampilkan data inventaris</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Memilih menu Data Inventaris</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menampilkan data inventaris perpustakaan</li> </ul>	Berhasil
22	<p>Nama Uji : Menambah data inventaris</p>	Berhasil

	<p>Deskripsi Pengujian : Verifikasi memasukkan data inventaris</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Nama Barang : Kursi Putar</li> <li>- Tanggal Masuk : 02-04-2010</li> <li>- Jumlah Barang : 2 unit</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menambahkan data inventaris baru</li> </ul>	
23	<p>Nama Uji : Menghapus data inventaris</p> <p>Deskripsi Pengujian : Verifikasi menghapus data inventaris</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Memilih tombol hapus pada kolom tabel data inventaris</li> <li>- Memilih tombol 'OK' pada jendela konfirmasi</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menghapus data inventaris yang dipilih dan menampilkan kembali data inventaris</li> </ul>	Berhasil
24	<p>Nama Uji : Memperbarui data inventaris</p> <p>Deskripsi Pengujian : Verifikasi memperbarui data inventaris</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Nama Barang : Kursi Lipat</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan memperbarui data inventaris yang dipilih dan menampilkan kembali data inventaris</li> </ul>	Berhasil
25	<p>Nama Uji : Mencari Buku</p> <p>Deskripsi Pengujian : Verifikasi mencari buku yang diinginkan pengguna</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Kategori : Judul Buku</li> <li>- Judul : Metode Penelitian</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Jika judul buku ditemukan, maka sistem akan menampilkan informasi buku yang diinginkan</li> <li>- Jika judul buku tidak ditemukan, maka sistem akan</li> </ul>	Berhasil

	memberi informasi apabila buku tidak tersedia	
26	<p>Nama Uji : Menampilkan Riwayat Peminjaman</p> <p>Deskripsi Pengujian : Verifikasi menampilkan data daftar riwayat peminjaman oleh anggota perpustakaan</p> <p>Kasus Uji :</p> <ul style="list-style-type: none"> <li>- Memilih menu Daftar Peminjaman</li> </ul> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Sistem akan menampilkan daftar riwayat peminjaman</li> </ul>	Berhasil

### 1.7 Analisis Pengujian

Berdasarkan hasil pengujian yang telah dilakukan secara bertahap yaitu pengujian unit, pengujian integrasi, pengujian validasi, pengujian reliabilitas, pengujian kompatibilitas, dan pengujian penerimaan pengguna diperoleh analisis pengujian sebagai berikut:

1. *Unit testing* atau uji unit dilakukan dengan metode *white box* yaitu *basis path testing* yang menghasilkan sebanyak 3 *cyclomatic complexity* dari hasil pengujian pada fungsi `editAnggota` yang terdapat dalam kelas `PetugasController`. Maka dapat disimpulkan bahwa fungsi `editAnggota` memiliki struktur program yang sederhana, mudah dipahami dan diimplementasi.

Tabel 6.37 Hasil Pengujian Unit

<b>Class</b>	PetugasController
<b>Fungsi</b>	editAnggota
<b>Region</b>	3
<b>Independent Path</b>	3
<b>Cyclomatic Complexity</b>	3

2. *Integration testing* atau uji integrasi dilakukan pada fungsi dari kelas `PetugasController` dan kelas `PetugasModel` dimana kedua unit saling berhubungan. Hasil dari pengujian integrasi, fungsi `editAnggota($id_anggota, $nama, $tempat_lahir, $gender, $tgl_lahir, $alamat, $status, $no_tlp, $uname, $password)` pada kelas `PetugasController` mengirimkan data input berisi data anggota yang kemudian dikirimkan ke fungsi `updateAnggota($id_anggota, $nama, $tempat_lahir, $tgl_lahir, $gender, $alamat, $status,`

`$no_tlp, $uname, $password)` pada kelas `PetugasModel` dan menghasilkan *result* yang sesuai dengan hasil yang diharapkan. Sehingga dapat disimpulkan bahwa pengujian integrasi fungsi `editAnggota` pada kelas `PetugasController` dengan fungsi `updateAnggota` pada kelas `PetugasModel` adalah *valid*.

3. *Validation testing* atau uji validasi dilakukan pada seluruh kebutuhan fungsional dalam sistem dari berbagai sisi, diantaranya dari sisi petugas, anggota, dan pengunjung. Hasil dari pengujian validasi sistem menunjukkan bahwa seluruh kebutuhan fungsional dalam sistem dapat dijalankan dengan baik dan menghasilkan *result* yang sesuai dengan hasil yang diharapkan. Maka dapat disimpulkan bahwa kebutuhan fungsional pada sistem perpustakaan yang dibangun adalah *valid*.
4. *Reliability testing* atau uji reliabilitas dilakukan dengan menganalisa hasil *stress testing* menggunakan *tool* WAPT dan perhitungan nelson menghasilkan skor reliabilitas sebesar 98,75% yang berarti bahwa sistem telah memenuhi aspek *reliability* dan dapat berjalan dengan konsisten.
5. *Compability testing* atau uji komabilitas dilakukan dengan menggunakan *tool* Sortsite. Dari hasil pengujian menunjukkan bahwa sistem dapat dijalankan di berbagai *browser* tanpa ada masalah konten atau fungsi yang tidak didukung pada *browser*. Namun, masih terdapat masalah *layout* atau performa pada *browser* Internet Explorer versi 11 dan Safari versi 10 dan 11.
6. *Acceptance testing* atau uji penerimaan merupakan tahap pengujian akhir dari proses pengujian sistem informasi perpustakaan pada penelitian ini. Pengujian penerimaan dilakukan dengan metode *black box* oleh *stakeholder* dengan hasil menunjukkan bahwa semua fitur pada sistem berhasil dijalankan dan telah diterima oleh *stakeholder* karena sudah sesuai dengan kebutuhan pengguna.

## BAB 7 PENUTUP

Pada bab ini akan diuraikan mengenai kesimpulan dan saran dari hasil penelitian yang sudah dilakukan.

### 7.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, maka dapat disimpulkan bahwa:

1. Proses analisis kebutuhan dimulai dengan melakukan wawancara terhadap *stakeholder* untuk mendapatkan informasi mengenai permasalahan pada perpustakaan yang kemudian dipetakan ke dalam tabel analisis permasalahan dan proses bisnis yang digambarkan menggunakan BPMN. Didapatkan pula informasi mengenai aktor yang terlibat yang dapat menggunakan sistem informasi perpustakaan, ada tiga aktor yang terdefinisi diantaranya yaitu petugas perpustakaan sebagai admin sistem, anggota dan pengunjung perpustakaan sebagai pengguna sistem. Setelah dilakukan analisis permasalahan, didapatkan solusi mengenai rincian sistem yang dibangun untuk menyelesaikan masalah pada analisis permasalahan berupa identifikasi kebutuhan, identifikasi fitur, analisis kebutuhan fungsional dan non-fungsional, dimana masing-masing terdefinisi 26 kebutuhan fungsional dan 2 kebutuhan non-fungsional. Hasil analisis kebutuhan fungsional digunakan untuk membuat diagram *use case* dan *use case scenario* yang dapat dimanfaatkan untuk membuat perancangan sistem.
2. Perancangan sistem menggunakan *class diagram* dan *sequence diagram* dengan pola perancangan MVC (Model View Controller) dan digambarkan dengan diagram UML. *Class diagram* terdiri dari 3 kelas model, 3 kelas controller, dan beberapa view masing-masing aktor yaitu view petugas, anggota, dan pengunjung, serta terdapat 1 kelas *abstract* dan 2 *interface*. *Sequence Diagram* digambarkan sebanyak 32 diagram masing-masing 24 *sequence diagram* petugas, 5 *sequence diagram* anggota, dan 3 *sequence diagram* pengunjung. Kemudian perancangan model data digambarkan dengan menggunakan diagram PDM (*Physical Data Model*) dimana terdapat 7 tabel diantaranya tabel anggota, tabel tamu, tabel pinjam, tabel buku, tabel rak, tabel petugas, dan tabel inventaris. Perancangan antarmuka sistem digambarkan dengan mockup sebanyak 15 perancangan halaman antarmuka sistem. Implementasi sistem menggunakan bahasa pemrograman PHP dan MySQL sebagai manajemen basis data. Implementasi sistem menghasilkan penjelasan diskriptif mengenai fungsi dan potongan program beserta *screenshot* tampilan dari setiap halaman pada sistem yang menggambarkan kode program yang telah diimplementasikan.
3. Proses pengujian sistem dilakukan secara bertahap meliputi pengujian unit, pengujian integrasi, pengujian validasi, pengujian reliabilitas, pengujian kompatibilitas, dan pengujian penerimaan pengguna. Tahap pengujian

pertama yaitu pengujian *basis path* digunakan untuk melakukan *unit testing* dengan perhitungan *cyclomatic complexity* yang menghasilkan 3 *independent path* pada kasus uji unit editAnggota menunjukkan bahwa sistem memiliki struktur yang sederhana, mudah dipahami dan diimplementasi. Selanjutnya dilakukan *integration testing* pada kasus uji fungsi editAnggota dari kelas PetugasController dengan fungsi updateAnggota dari kelas PetugasModel menunjukkan hasil yang *valid*. Kemudian *validation testing* dilakukan dengan metode *black box* pada seluruh fitur dalam sistem yaitu sebanyak 26 fungsi menunjukkan hasil yang *valid*. *Reliability testing* dilakukan dengan menganalisis hasil pengujian stress dan perhitungan nelson menghasilkan skor reliabilitas sebesar 98,75% yang berarti bahwa sistem telah memenuhi aspek *reliability* dan dapat berjalan dengan konsisten. *Compatibility testing* dilakukan dengan *tool* Sortsite menunjukkan bahwa sistem dapat dijalankan di berbagai *browser* tanpa ada masalah konten atau fungsi yang tidak didukung pada *browser*, namun masih terdapat masalah *layout* atau performa *browser* pada Internet Explorer versi 11 dan Safari versi 10 dan 11. *Acceptance testing* dilakukan dengan metode *black box* oleh *stakeholder* dengan hasil menunjukkan bahwa semua fitur pada sistem berhasil dijalankan dan telah diterima oleh *stakeholder*.

## 7.2 Saran

1. Diharapkan pada penelitian selanjutnya sistem informasi perpustakaan ini dapat dikembangkan lebih lanjut dengan menambahkan fitur-fitur lain yang lebih beragam seperti *upload* dan *download ebook*.
2. Diharapkan pada penelitian selanjutnya dapat ditambahkan sistem *barcode* atau *QR code* pada kartu anggota sehingga bisa lebih meningkatkan pelayanan perpustakaan.

## DAFTAR PUSTAKA

- A.S, Rosa dan Shalahuddin, M., 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak*. Bandung: Modula.
- A.S, Rosa dan Shalahuddin, M., 2016. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Al Fatta, Hanif., 2007. *Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*. Yogyakarta: Penerbit Andi.
- Anhar, ST., 2010. *Panduan Menguasai PHP dan MySQL Secara Otodidak*. Jakarta: Mediakita.
- Basuki, Sulisty., 1993. *Pengantar Ilmu Perpustakaan*. Jakarta: Gramedia Pustaka Utama.
- Dewanto, I. Joko., 2004. System Development Life Cycle Dengan Beberapa Pendekatan. *Jurnal Ilmu Komputer*.
- Fachri, Barany., 2017. Perancangan Sistem Informasi Berbasis Web (Studi Kasus Perpustakaan Kopertis Wilayah I Sumut. *Jurnal Manajemen Informatika dan Teknik Komputer*, vol. 2, no 1.
- Hardi, Richki dan Hardianto., 2015. Pengembangan Sistem Informasi Perpustakaan Menggunakan Kerangka PIECES (Studi Kasus Perpustakaan STITEK Bontang. *Jurnal Ilmiah Teknologi Informasi Terapan*, vol. 1, no 3.
- Hidayat, Arief., dan Surarso, Bayu., 2012. Penerapan Arsitektur Model View Controller (MVC) dalam Rancang Bangun Sistem Kuis Online Adaptif. *Seminar Nasional Teknologi Informasi dan Komunikasi (SENTIKA 2012)*, Yogyakarta, 10 Maret 2012.
- Maniah., Hamidin, Dini., 2017. *Analisis dan Perancangan Sistem Informasi Pembahasan Secara Praktis dengan Contoh Kasus*. Yogyakarta: Deepublish.
- Marimin., Tanjung, Hendri., dan Prabowo, Haryo., 2006. *Sistem Informasi Manajemen Sumber Daya Manusia*. Jakarta: Grasindo.
- Noviandi, B. M., Destiani, D., dan Partono., 2012. Perancangan Sistem Informasi Inventori Barang Di Bank Sampah Garut. *Jurnal Algoritma* , vol. 9, no 32.
- NS, Sutarno., 2006. *Perpustakaan dan Masyarakat*. Jakarta: Sagung Seto.
- Nugroho, Adi., 2009. *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. Yogyakarta: Penerbit Andi.
- Nugroho, Adi., 2010. *Rekaya Perangkat Lunak Berorientasi Objek dengan Metode USDP*. Yogyakarta: Penerbit Andi.
- Nuryadin, Riki., 2014. Sistem Informasi Perpustakaan. [e-journal]. Tersedia melalui: Academia <<http://www.academia.edu>> [Diakses 17 Februari 2018].

- Oka, Gde Putu Arya., 2017. *Model Konseptual Pengembangan Produk Pembelajaran Beserta Teknik Evaluasi*. Yogyakarta: Deepublish.
- Pressman, Roger S., 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Dua)*. Yogyakarta: Andi.
- Pressman, Roger S., 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*. Yogyakarta: Andi.
- Pressman, Roger S., 2010. *Software Engineering A Practitioner's Approach, Seventh Edition*. New York: McGraw-Hill.
- Rochim, Taufiq., 2002. *Sistem Informasi*. Bandung: Penerbit ITB.
- Saputra, Agus., 2012. *Web Tips PHP, HTML5 dan CSS3*. Jakarta: Jasakom.
- Sommerville, Ian., 2011. *Software Engineering Ninth Edition*. Massachusetts: Pearson.
- Sommerville, Ian., 2003. *Software Engineering (Rekayasa Perangkat Lunak)/Edisi 6/Jilid 1*. Jakarta: Erlangga.
- Supriyanto, Wahyu dan Muhsin, Ahmad., 2008. *Teknologi Informasi Perpustakaan: Strategi Perancangan Perpustakaan Digital*. Yogyakarta: Kanisius.
- Suwarno, Wiji., 2013. *Ilmu Perpustakaan dan Kode Etik Pustakawan*. Yogyakarta: Ar-Ruz.
- Whitten, Jeffery L., Bentley, Lonnie D., dan Dittman, Kevin C., 2004. *Metode Desain dan Analisis Sistem*. Yogyakarta: Andi.