



**LAPORAN PRAKTEK KERJA LAPANGAN**

**RANCANG BANGUN SISTEM INFORMASI  
MANAJEMEN PERPUSTAKAAN DI JURUSAN  
KIMIA FAKULTAS MATEMATIKA DAN ILMU  
PENGETAHUAN ALAM UNIVERSITAS  
UDAYANA**

Oleh:

MADE DINDA PRADNYA PRAMITA

NIM : 1308605021

Pembimbing:

I MADE WIDIARTHA S.SI,M.KOM

Program Studi Teknik Informatika

Jurusan Ilmu Komputer

Fakultas Matematika Dan Ilmu Pengetahuan  
Alam

Universitas Udayana

2016

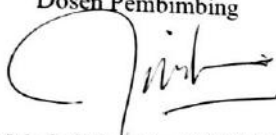
## HALAMAN PENGESAHAN

RANCANG BANGUN SISTEM INFORMASI PERPUSTAKAAN  
JURUSAN KIMIA FAKULTAS MATEMATIKA DAN ILMU  
PENGETAHUAN ALAM UNIVERSITAS UDAYANA

Oleh :  
Made Dinda Pradnya Pramita  
NIM : 1308605021

Denpasar, 10 Oktober 2016  
Menyetujui,

Dosen Pembimbing



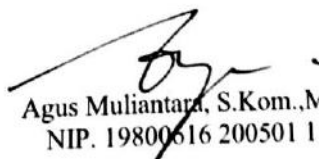
I Made Wisartha, S.Si., M.Kom  
NIP. 19821220 200801 1 008

Pembimbing Lapangan



Dr. I Nengah Wirajana, S.Si., M.Si  
NIP. 19710219 199702 1 001

Mengetahui,  
Ketua Jurusan Ilmu Komputer  
FMIPA Universitas Udayana



Agus Muliantara, S.Kom., M.Kom  
NIP. 19800616 200501 1 001

## **KATA PENGANTAR**

Puji syukur penulis panjatkan kehadirat Ida Sang Hyang Widhi Wasa atau Tuhan Yang Maha Esa karena atas berkat dan rahmatNya penulis dapat menyelesaikan laporan praktek kerja lapangan yang berjudul “Rancang Bangun Sistem Informasi Perpustakaan Jurusan Kimia Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana” secara tepat waktu.

Selama melaksanakan Praktek Kerja Lapangan (PKL) dan dalam menyelesaikan laporan ini, penulis telah banyak menerima bimbingan, pengarahan, petunjuk, saran, serta fasilitas yang mendukung hingga akhir penulisan laporan ini. Sehubungan dengan telah terselesaikannya laporan PKL ini, maka penulis mengucapkan terima kasih dan penghargaan kepada berbagai pihak yang telah membantu, antara lain :

1. Bapak Agus Muliantara, S.Kom.,M.Kom sebagai Ketua Jurusan Ilmu Komputer FMIPA UNUD yang telah memberikan dukungan dalam pelaksanaan Praktek Kerja Lapangan ini;
2. Bapak I Made Widiartha,S.Si.,M.Kom sebagai Pembimbing yang telah bersedia meluangkan waktunya untuk mengkritisi, mengkoreksi, dan membantu penulisan laporan ini;
3. Bapak I Nengah Wirajana selaku Ketua Jurusan Kimia yang telah memberikan kesempatan untuk melaksanakan Praktek Kerja Lapangan dan membimbing serta memberikan arahan selama melaksanakan Praktek Kerja Lapangan di Jurusan Kimia.

Penulis menyadari bahwa laporan ini masih jauh dari kesempurnaan karena masih mengandung kelemahan dan kekurangan. Maka dari itu dengan segala kerendahan hati, penulis mengharapkan adanya masukan, kritik yang membangun dan saran – saran penyempurnaan sehingga nantinya dapat memperbaiki laporan ini dan melakukan pengembangan di kemudian hari.

Denpasar, 9 Oktober 2016  
Penulis

## DAFTAR ISI

HALAMAN PENGESAHAN .....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR .....	vi
DAFTAR TABEL .....	viii
DAFTAR LAMPIRAN .....	ix
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Tujuan.....	2
1.3 Manfaat .....	3
1.4 Waktu dan Pelaksanaan.....	4
BAB II GAMBARAN UMUM.....	5
2.1 Sejarah Jurusan Kimia .....	5
2.2 Kegiatan Instansi Tempat PKL .....	9
2.3 Struktur Instansi Tempat PKL.....	9
2.4 Visi, Misi dan Tujuan Instansi Tempat PKL .....	10
2.1.1 Visi Jurusan Kimia .....	10
2.1.2 Misi Jurusan Kimia .....	10
2.1.3 Tujuan Jurusan Kimia .....	10
BAB III KAJIAN PUSTAKA.....	12
3.1 Sistem Informasi Manajemen .....	12
3.1.1 Pengertian Sistem Informasi Manajemen .....	12
3.1.2 Komponen Sistem Informasi Manajemen .....	13
3.1.3 Elemen Sistem Informasi Manajemen .....	15
3.2 Metode Pengembangan Perangkat Lunak.....	16
3.3 Pemrograman Berorientasi Object .....	18
3.3.1 Ekapsulasi .....	19
3.3.2 Inheritansi.....	19
3.3.3 Polimorfisme .....	19
3.3.4 Overriding .....	20
3.4 Unified Modelling Language.....	20
3.4.1 <i>Use Case Diagram</i> .....	21
3.4.2 Activity Diagram.....	22

3.4.3	Class Diagram .....	23
3.5	Entity Relationship Diagram (ERD) .....	23
3.6	Normalisasi Database .....	25
3.7	MySQL.....	26
3.8	Structured Query Language (SQL) .....	27
3.8.1	Kelompok Pernyataan SQL .....	28
3.8.2	Kelebihan MySQL .....	28
BAB IV PELAKSANAAN PKL.....		30
4.1	Perancangan Sistem Informasi Perpustakaan.....	30
4.1.1	Metode Pengembangan Sistem .....	30
4.2	<i>Unified Modelling Language(UML)</i> .....	34
4.2.1	<i>Use Case Diagram</i> .....	34
4.2.2	Activity Diagram.....	37
4.2.2.1	<i>Activity Diagram</i> untuk Admin .....	37
4.2.2.2	<i>Activity Diagram</i> untuk User .....	54
4.2.3	Class Diagram .....	56
4.2.4	<i>Squence Diagram</i> .....	59
4.3	<i>Entity Relationship Diagram (ERD)</i> .....	72
4.4	Implementasi Sistem.....	73
4.5	Pengujian <i>Black Box</i> .....	79
BAB V KESIMPULAN DAN SARAN .....		82
5.1	Kesimpulan.....	82
5.2	Saran.....	82
DAFTAR PUSTAKA.....		83

## DAFTAR GAMBAR

Gambar 2. 1 Struktur Organisasi Jurusan Kimia .....	9
Gambar 3. 1 SDLC Model Proses Waterfall .....	17
Gambar 4. 1 Use Case Diagram Sistem Informasi Perpustakaan ..	35
Gambar 4. 2 Activity Diagram Autentifikasi untuk Admin .....	38
Gambar 4.3 Activity Diagram Tambah Data Buku .....	39
Gambar 4.4 Activity Diagram Edit Data Buku .....	40
Gambar 4.5 Activity Diagram Hapus Data Buku .....	41
Gambar 4. 6 Activity Diagram Tambah Data Skripsi .....	42
Gambar 4. 7 Activity Diagram Edit Data Skripsi .....	43
Gambar 4.8 Activity Diagram Hapus Data Skripsi.....	44
Gambar 4. 9 Activity Diagram Tambah Data Anggota .....	45
Gambar 4. 10 Activity Diagram Edit Data Anggota.....	46
Gambar 4. 11 Activity Diagram Hapus Data Anggota .....	47
Gambar 4.12 Activity Diagram Tambah Data Peminjaman .....	48
Gambar 4. 13 Activity Diagram Edit Data Peminjaman.....	49
Gambar 4. 14 Activity Diagram Hapus Data Peminjaman .....	50
Gambar 4. 15 Activity Diagram Tambah Data Pengembalian.....	51
Gambar 4. 16 Activity Diagram Edit Data Pengembalian .....	52
Gambar 4. 17 Activity Diagram Hapus Data Pengembalian.....	53
Gambar 4. 18 Activity Diagram Lihat Data Buku .....	54
Gambar 4. 19 Activity Diagram Lihat Data Skripsi .....	55
Gambar 4. 20 Activity Diagram Pencarian Data Buku .....	55
Gambar 4. 21 Activity Diagram Pencarian Data .....	56
Gambar 4. 22 Class Diagram Sistem Informasi Perpustakaan.....	57
Gambar 4. 23 Squence Diagram Autentifikasi .....	59
Gambar 4. 24 Squence Diagram Tambah Data Buku .....	60
Gambar 4. 25 Squence Diagram Edit Data Buku .....	61
Gambar 4. 26 Squence Diagram Hapus Data Buku .....	62
Gambar 4. 27 Squence Diagram Tambah Data Skripsi .....	62
Gambar 4. 28 Squence Diagram Edit Data Skripsi.....	63
Gambar 4. 29 Squence Diagram Hapus Data Skripsi .....	64
Gambar 4. 30 Squence Diagram Tambah Data Anggota .....	64

Gambar 4. 31 Squence Diagram Edit Data Anggota .....	65
Gambar 4. 32 Squence Diagram Hapus Data Anggota .....	66
Gambar 4. 33 Squence Diagram Tambah Data Peminjaman .....	67
Gambar 4. 34 Squence Diagram Edit Data Peminjaman Buku .....	68
Gambar 4. 35 Squence Diagram Hapus Data Peminjaman Buku ..	69
Gambar 4. 36 Squence Diagram Tambah Data Pengembalian .....	69
Gambar 4. 37 Squence Diagram Hapus Data Pengembalian .....	70
Gambar 4. 38 Squence Diagram Lihat Data Buku .....	71
Gambar 4. 39 Squence Diagram Lihat Data Skripsi .....	71
Gambar 4. 40 ERD Sistem Perpustakaan .....	72
Gambar 4. 41 Skema Database Sistem Perpustakaan .....	73
Gambar 4. 42 Form Login Sistem Perpustakaan .....	74
Gambar 4. 43 Form Menu Skripsi Sistem Perpustakaan .....	74
Gambar 4. 44 Form Tambah Data Skripsi .....	75
Gambar 4. 45 Form Edit Data Skripsi .....	75
Gambar 4. 46 Form Tampil Data Anggota .....	76
Gambar 4. 47 Form Tampil Data Buku .....	76
Gambar 4. 48 Form Peminjaman Buku .....	77
Gambar 4. 49 Form Pengembalian Buku .....	77
Gambar 4. 50 Form Menampilkan Data Skripsi .....	78
Gambar 4. 51 Form Menampilkan Data Buku .....	78

## **DAFTAR TABEL**

Tabel 3. 1 Simbol Use Case Diagram.....	21
Tabel 4. 1 Analisis Kebutuhan Sistem Perpustakaan .....	31
Tabel 4. 2 Definisi Aktor.....	35
Tabel 4. 3 Definisi Use Case .....	36
Tabel 4. 4 Hubungan Class dan Use Case Diagram .....	58
Tabel 4. 5 Pengujian dengan Black Box Testing .....	79



## **DAFTAR LAMPIRAN**

Lampiran 1. Aktivitas Harian.....	A-1
-----------------------------------	-----

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi pada masa sekarang ini mengalami kemajuan yang pesat, terutama teknologi informasi komputer. Komputer pada masa sekarang bukan merupakan barang asing lagi, hampir semua bidang pekerjaan membutuhkan komputer sebagai alat bantu untuk menyelesaikan pekerjaan dengan kelebihan diantaranya kecepatan, keakuratan serta keefisienan dalam pengelolaan data bila dibandingkan dengan pengelolaan data secara manual. Pengelolaan data yang efisien dan akurat sangat dibutuhkan dalam suatu organisasi dan instansi misalnya dalam pengelolaan data perpustakaan disuatu perguruan tinggi.

Menurut C. Larasati Milburga, dkk., perpustakaan adalah suatu unit kerja yang berupa tempat menyimpan koleksi bahan pustaka yang diatur secara sistematis dengan cara tertentu untuk dipergunakan secara berkesinambungan oleh pemakainya sebagai sumber informasi.

Pelayanan yang diberikan pada mahasiswa dalam hal melakukan registrasi, administrasi, peminjaman buku dan lain-lain yang dilakukan dengan cara manual sudah tidak zamannya lagi diterapkan pada zaman yang kita rasakan sekarang ini, karena menghabiskan waktu, tenaga, pikiran dan sulit mendapatkan informasi maupun data yang akurat.

Perpustakaan akan berkembang dengan baik apabila sistem teknologi yang digunakan mengikuti perkembangan teknologi saat ini. Namun apabila perpustakaan masih menggunakan sistem manual maka perpustakaan akan tertinggal dengan seiringnya waktu dan perkembangan teknologi yang telah menggunakan sistem terkomputerisasi. Dengan menggunakan sistem terkomputerisasi, maka sangat memungkinkan untuk melakukan pendapatan atau pengelolaan data agar dapat mengorganisir data lebih terstruktur, mudah dipahami dan dapat membantu pengguna mengakses data.

Perpustakaan dalam suatu perguruan tinggi seharusnya memiliki suatu aplikasi *database* yang lebih baik secara terkomputerisasi yang, sehingga mahasiswa dapat melihat data buku yang tersedia pada perpustakaan melalui website yang telah ada dan petugas perpustakaan tidak perlu mencatat secara manual mengenai data buku dan data peminjaman buku.

Perpustakaan yang beroperasi pada Jurusan Kimia Fakultas MIPA Universitas Udayana masih menggunakan cara manual yang masih terasa cukup rumit bagi petugas perpustakaan dalam mengolah data buku karena sistem yang digunakan belum menggunakan sistem komputerisasi yang berbasis *database* namun mencatat data buku pada lembaran kertas setiap data yang masuk. Data – data buku, dosen, mahasiswa dan pengunjung yang terdaftar sebagai anggota perpustakaan serta transaksi peminjaman dan pengembalian yang diproses dengan menggunakan komputer akan sangat membantu perpustakaan dalam mengolah data. Data yang disimpan dari waktu ke waktu yang semakin bertambah membuat petugas perpustakaan rumit untuk mencari informasi bahkan dalam pembuatan laporan yang dibutuhkan, karena data yang disimpan belum berbasis komputer.

Berdasarkan permasalahan tersebut, maka penulis diberikan tugas selama melaksanakan praktek kerja lapangan di Jurusan Kimia Fakultas MIPA Universitas Udayana untuk merancang dan mengimplementasikan sebuah sistem informasi perpustakaan yang berfungsi untuk mencatat data buku dan mengelola transaksi peminjaman serta pengembalian buku.

Pengambilan judul “*Rancang Bangun Sistem Informasi Perpustakaan di Jurusan Kimia Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana*” ini sebagai laporan dalam praktek kerja lapangan yang telah dilaksanakan, sehingga penulis dapat mempraktekan ilmu yang didapat untuk diterapkan di dunia kerja.

## **1.2 Tujuan**

Adapun tujuan dari pengembangan sistem informasi perpustakaan ini dapat dikategorikan sebagai berikut.

a. Tujuan Umum

Menghasilkan sistem informasi perpustakaan yang dapat digunakan untuk mengelola data data buku dan sistem registrasi dalam peminjaman dan pengembalian buku di perpustakaan Jurusan Kimia Fakultas MIPA Universitas Udayana.

b. Tujuan Khusus

1. Menganalisis kebutuhan yang diperlukan pada sistem informasi perpustakaan berdasarkan permasalahan dari sistem yang sedang berjalan misalnya dari segi pencatatan buku dan registrasi yang masih dilakukan secara manual.
2. Memperbaiki sistem manual yang sedang berjalan menjadi sistem yang terkomputerisasi sehingga dapat memberikan satu nilai tambah dalam membantu petugas perpustakaan.
3. Uji coba sistem informasi perpustakaan untuk mendukung evaluasi pelayanan di perpustakaan Jurusan Kimia.
4. Mengevaluasi kualitas sistem informasi beserta sistem berdasarkan hasil uji coba dan pendapat user atau pengguna.

### 1.3 Manfaat

Beberapa manfaat yang dapat diperoleh atau diambil dari penelitian ini adalah sebagai berikut.

a. Bagi penulis

1. Mampu mengidentifikasi masalah yang terjadi pada perpustakaan yang belum berjalan secara maksimal karena masih menggunakan sistem secara manual.
2. Dapat mengembangkan ilmu pengetahuan yang diperoleh selama mengikuti perkuliahan dengan menerapkannya di lapangan melalui praktek kerja lapangan terkait pemanfaatan teknologi informasi dalam kehidupan nyata, khususnya pengguna sistem informasi.

3. Dapat menambah pengetahuan tentang pemanfaatan teknologi informasi dalam dunia nyata, serta menambah pengalaman penulis terkait proses membangun suatu sistem di lapangan atau lingkungan nyata.
- b. Bagi Jurusan Kimia
1. Membantu dalam pengelolaan perpustakaan khususnya dalam permasalahan registrasi peminjaman dan pengembalian buku serta membantu pencatatan data buku yang lebih efisien.
  2. Memudahkan petugas perpustakaan dalam membuat laporan data buku dan data peminjaman serta pengembalian buku.
  3. Memudahkan mahasiswa Jurusan Kimia untuk melihat data – data buku yang tersedia di perpustakaan dengan melakukan pengecekan data buku melalui website yang telah dibuat.
- c. Bagi Institusi Pendidikan
- Dapat digunakan sebagai referensi untuk menambah wawasan bagi mahasiswa khususnya mahasiswa Jurusan Ilmu Komputer Fakultas MIPA Universitas Udayana ataupun mahasiswa lainnya yang membutuhkan informasi terkait rancang bangun Sistem Informasi Perpustakaan.

#### **1.4 Waktu dan Pelaksanaan**

Waktu pelaksanaan Praktek Kerja Lapangan (PKL) ini berlangsung selama dua bulan tepatnya dimulai dari 1 Maret 2016 sampai dengan 20 Mei 2016. Pelaksanaan Praktek Kerja Lapangan (PKL) yang penulis laksanakan yaitu berlokasi di Jurusan Kimia Fakultas MIPA Universitas Udayana yang beralamat di Bukit Jimbaran, Badung.

## **BAB II**

### **GAMBARAN UMUM**

#### **2.1 Sejarah Jurusan Kimia**

Jurusan Kimia FMIPA Universitas Udayana didirikan atas dasar Surat Keputusan Rektor UNUD no. 613/PT.17/I.12/1984 tertanggal 1 Juli 1984 dan bernaung di bawah Program Studi Matematika dan Ilmu Pengetahuan Alam (PS MIPA). Dengan keluarnya Surat Keputusan Dirjen Pendidikan Tinggi No. 66/Dikti/Kep.1988 tertanggal 1 Nopember 1988 Jurusan Kimia PS MIPA UNUD berubah statusnya menjadi Program Studi Kimia setingkat Jurusan di bawah Fakultas Peternakan. Selanjutnya dengan keluarga Surat Keputusan Dirjen Pendidikan Tinggi No. 81/Dikti/Kep./1989 tertanggal 20 September 1989 statusnya berubah lagi menjadi Program Studi Kimia Antar Fakultas langsung di bawah Rektor. Pada tanggal 22 Oktober 1993 keluarlah surat Keputusan Mendikbud RI No. 0382/0/1993 tentang terbentuknya Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Universitas Udayana yang menggabungkan Program Studi Fisika, Kimia, dan Biologi. Dengan keluarnya Surat Keputusan Direktur Jenderal Pendidikan Tinggi Departemen Pendidikan dan Kebudayaan Republik Indonesia No. 071/Dikti/Kep/1994, maka Jurusan Kimia bersama-sama dengan jurusan yang lain dibentuk sebagai unsur pelaksana akademik pada Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana.

Semua aktivitas Program Studi Kimia (1984 – 1986) berlangsung di kampus Universitas Udayana Jl. PB Sudirman dan Jl. Dr. Goris No. 7 Denpasar. Selanjutnya pada permulaan tahun 1986 beberapa perkuliahan dan praktikum dilaksanakan di kampus UNUD Bukit Jimbaran. Kemudian seluruh kegiatan akademik maupun administrasi PS Kimia dipindahkan dari Kampus Denpasar ke Kampus Bukit Jimbaran secara penuh pada tahun 1987. Hingga kini Jurusan Kimia FMIPA UNUD berkantor di Kampus Bukit Jimbaran dengan menempati tiga buah gedung FG, FH, FI, dan sebuah gedung penyimpanan bahan/zat kimia (Gedung FM).

Pada awal perjalanan Jurusan Kimia PS. MIPA UNUD memakai kurikulum Jurusan Kimia FMIPA Universitas Airlangga (UNAIR) Surabaya. Hal ini didasarkan atas pertimbangan dosen pemberi kuliah sangat terbatas, kepangkatan para dosen relatif junior, serta kegiatan praktikum sebagian besar tidak dapat dilaksanakan karena fasilitas, termasuk laboratorium, alat-alat dan bahan kimia sangat terbatas. Demikian juga dijalin hubungan kerja sama antara Universitas Udayana dengan Universitas Airlangga untuk penyelesaian tugas akhir mahasiswa Jurusan Kimia PS MIPA UNUD.

Selama perjalanan sampai dengan tahun 1990, jumlah dosen bertambah banyak dan memenuhi syarat sebagai penguji untuk S1, maka atas dasar rapat dosen diputuskan tidak lagi mengirim mahasiswa ke UNAIR. Namun kurikulum Jurusan Kimia UNAIR masih dipakai di Jurusan Kimia UNUD karena kurikulum baru belum disusun. Selanjutnya bertitik tolak pada kurikulum nasional, Jurusan Kimia FMIPA UNUD menyusun dan mengeluarkan kurikulum baru pada tanggal 5 Mei 1995. Kurikulum ini diberlakukan mulai tahun ajaran 1995/1996 terhadap mahasiswa baru.

Perjalanan selanjutnya menyongsong tahun 2000 Jurusan Kimia terus berbenah diri untuk meningkatkan kualitas diri sehingga dapat berperan aktif dalam pembangunan nasional. Mengantisipasi hal tersebut Jurusan Kimia selanjutnya mengevaluasi kurikulum yang telah diberlakukan sejak 5 tahun yang telah lewat. Evaluasi yang telah dilakukan sudah barang tentu juga mengacu pada kurikulum nasional dengan menambahkan mata kuliah yang menjadi penciri lokal yang bertujuan untuk mengoptimalkan potensi lokal. Kurikulum 2000 ini diberlakukan sejak awal tahun ajaran 2000 dan diberlakukan untuk semua mahasiswa di Jurusan Kimia.

Pada tahun 2002 diterbitkan SK Mendiknas No. 045/DIKTI/2002 tentang kurikulum berbasis kompetensi. Untuk merespons hal tersebut, maka dilakukan evaluasi terhadap kurikulum 2000. Kompetensi utama yang dikedepankan, disesuaikan dengan potensi lokal dengan harapan lulusan jurusan Kimia UNUD dapat berkiprah di masyarakat. Kurikulum 2005 baru diberlakukan pada tahun ajaran 2006.

Melalui SK Mendiknas No 045/U/2002, kurikulum mengalami perubahan yang awalnya berbasis isi menjadi berbasis kompetensi. Kurikulum berbasis kompetensi (KBK) merupakan suatu konsep kurikulum yang menekankan pada pengembangan kemampuan melakukan (kompetensi) tugas-tugas dengan standar performan tertentu, sehingga hasilnya dapat dirasakan oleh peserta didik, berupa penguasaan terhadap seperangkat kompetensi tertentu. Kurikulum disusun berdasarkan atas elemen-elemen kompetensi yang dapat mengantarkan peserta didik untuk mencapai kompetensi utama, kompetensi pendukung, dan kompetensi lainnya. Ciri-ciri rancangan kurikulum berbasis kompetensi di antaranya adalah sebagai berikut: (1) Menyatakan secara jelas rincian kompetensi peserta didik sebagai luaran proses pembelajaran; (2) Materi ajar dan proses pembelajaran didesain dengan orientasi pada pencapaian kompetensi dan berfokus pada minat peserta didik; (3) Lebih mensinergikan dan mengintegrasikan penguasaan ranah kognitif, psikomotorik dan afektif; (4) Proses penilaian hasil belajar lebih ditekankan pada kemampuan untuk berkreasi secara prosedural atas dasar pemahaman penerapan, analisis, dan evaluasi yang benar pula; (5) Disusun oleh penyelenggara pendidikan tinggi dan pihak-pihak berkepentingan terhadap lulusan pendidikan tinggi (masyarakat profesi dan pengguna lulusan).

Luaran (outcome) perguruan tinggi dinilai dari kompetensi yang dicapai oleh anak didiknya. Penilaian bukan lagi dilakukan oleh perguruan tinggi semata, melainkan oleh masyarakat luas pemangku kepentingan. Jadi, orientasi hasil pembelajaran bukan terletak pada output saja melainkan bergeser ke outcome. Mahasiswa bukan saja memiliki nilai yang bagus, melainkan apakah mereka dapat berkehidupan bermasyarakat sesuai ilmu pengetahuan dan teknologi yang diperoleh selama kuliah dan menerapkan/mengimplementasikannya dengan sikap dan berperilaku dalam berkarya.

Tujuan revisi kurikulum di Jurusan Kimia FMIPA Unud adalah mengembangkan KBK di Program Studi Kimia dalam rangka meningkatkan mutu pendidikan tinggi dan lulusan sesuai dengan Kerangka Kualifikasi Nasional Indonesia (KKNI) atau Indonesian Qualification Framework. Selain itu, revisi kurikulum dilaksanakan atas dasar hasil Tracer Study UPMF MIPA Unud tahun 2010



terhadap tiga jurusan di FMIPA Unud (Jurusan Kimia, Biologi, dan Fisika), yang di antaranya menyimpulkan bahwa masih terdapat kesenjangan yang signifikan beberapa komponen-komponen yang terkait dengan soft skills antara yang diperoleh di bangku kuliah dengan yang dibutuhkan di dunia kerja. Hasil ini merekomendasikan agar pihak yang berwenang dan pengambil keputusan/kebijakan perlu segera melakukan langkah-langkah evaluasi dan perbaikan terhadap kurikulum secara periodik, dan secara simultan melakukan peningkatan kualitas sumberdaya, terutama sarana dan sarana pendukung proses pembelajaran berbasis KBK serta suasana yang kondusif.

Berdasarkan Peraturan Presiden Republik Indonesia nomor 8 tahun 2012 tentang KKNI, pada pasal 5 menyatakan bahwa penyetaraan capaian pembelajaran yang dihasilkan melalui pendidikan dengan jenjang kualifikasi pada KKNI lulusan Sarjana Terapan dan Sarjana paling rendah setara dengan jenjang ke-6. KKNI terdiri dari 9 jenjang kualifikasi, dimulai dari jenjang 1 sebagai jenjang terendah sampai jenjang 9 sebagai jenjang tertinggi. Deskripsi umum jenjang KKNI pada setiap jenjang/level kualifikasi mencakup proses yang menumbuhkembangkan afeksi sebagai berikut: a) Bertaqwa kepada Tuhan Yang Maha Esa; b) Memiliki moral, etika dan kepribadian yang baik di dalam menyelesaikan tugasnya; c) Berperan sebagai warga negara yang bangga dan cinta tanah air serta mendukung perdamaian dunia; d) Mampu bekerja sama dan memiliki kepekaan sosial dan kepedulian yang tinggi terhadap masyarakat dan lingkungannya; e) Menghargai keanekaragaman budaya, pandangan, kepercayaan, dan agama serta pendapat/temuan original orang lain; dan f) Menjunjung tinggi penegakan hukum serta memiliki semangat untuk mendahulukan kepentingan bangsa serta masyarakat luas.

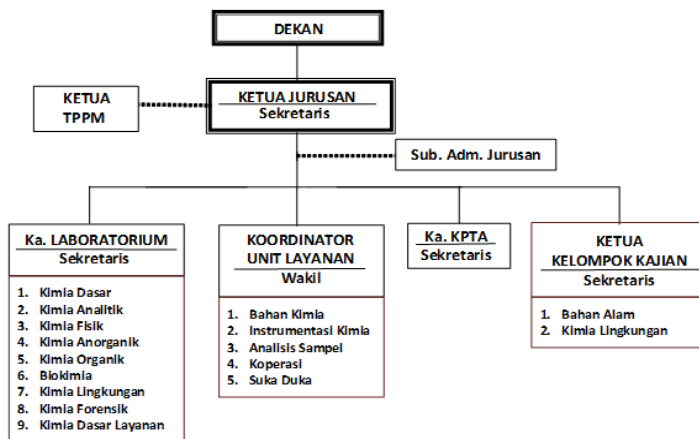
Luaran S1 dalam KKNI menduduki kualifikasi level 6. Capaian kompetensi pada kualifikasi level 6 meliputi: (1) Mampu mengaplikasikan bidang keahliannya dan memanfaatkan ilmu pengetahuan, teknologi, dan/atau seni pada bidangnya dalam penyelesaian masalah serta mampu beradaptasi terhadap situasi yang dihadapi. (2) Menguasai konsep teoritis bidang pengetahuan tertentu secara umum dan konsep teoritis bagian khusus dalam bidang

pengetahuan tersebut secara mendalam, serta mampu memformulasikan penyelesaian masalah prosedural.; (3) Mampu mengambil keputusan yang tepat berdasarkan analisis informasi dan data, dan mampu memberikan petunjuk dalam memilih berbagai alternatif solusi secara mandiri dan kelompok. dan (4) Bertanggung jawab pada pekerjaan sendiri dan dapat diberi tanggungjawab atas pencapaian hasil kerja organisasi.

## 2.2 Kegiatan Instansi Tempat PKL

Jurusan Kimia merupakan salah satu jurusan di Fakultas Matematika dan Ilmu Pengetahuan Alam yang memiliki aktivitas – aktivitas berupa seminar nasional, pengabdian kepada masyarakat, rapat jurusan, PKKMB jurusan dan kegiatan perkuliahan.

## 2.3 Struktur Instansi Tempat PKL



**Gambar 2. 1 Struktur Organisasi Jurusan Kimia FMIPA Universitas Udayana**

Dalam suatu instansi diperlukan adanya kejelasan mengenai struktur organisasi. Hal tersebut sangat diperlukan karena struktur organisasi dapat digunakan sebagai kerangka yang dapat merepresentasikan susunan atau perwujudan pola tetap dari

keterkaitan antar fungsi masing – masing posisi yang menunjukkan kedudukan, wewenang, tanggung jawab serta pembagian kerja, sehingga tujuan suatu instansi dapat tercapai secara efektif dan maksimal. Adapun struktur organisasi Jurusan Kimia dapat dilihat pada Gambar 2.1 sebagai berikut.

## **2.4 Visi, Misi dan Tujuan Instansi Tempat PKL**

Karakteristik Jurusan Kimia Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana saat sangat dipengaruhi oleh kekuatan serta kelemahan internal jurusan serta peluang dan ancaman yang terdapat pada eksternal sistem. Karakter jurusan yang hendak dibangun juga akan ditentukan oleh visi, misi, dan tujuan pendidikan yang hendak dikembangkan. Untuk itu, akan diuraikan visi, misi, serta tujuan pendidikan yang menjadi penciri karakteristik Jurusan Kimia FMIPA Universitas Udayana.

### **2.1.1 Visi Jurusan Kimia**

Visi Jurusan Kimia Fakultas Matematika dan Ilmu pengetahuan Alam Universitas Udayana adalah “Terwujudnya Jurusan Kimia sebagai pusat pengembangan ilmu pengetahuan dan aplikasi bidang kimia melalui pendidikan, penelitian dan pengabdian kepada masyarakat yang menghasilkan lulusan yang unggul, mandiri, dan berbudaya”.

### **2.1.2 Misi Jurusan Kimia**

Beberapa misi dari Jurusan Kimia adalah sebagai berikut.

1. Menyelenggarakan pendidikan kimia melalui proses pembelajaran yang bermutu, memiliki relevansi dan kompetensi yang tinggi.
2. Mengembangkan penelitian unggulan yang dapat diaplikasikan untuk kebutuhan masyarakat melalui kerjasama penelitian dengan berbagai institusi di dalam dan luar negeri.

### **2.1.3 Tujuan Jurusan Kimia**

Beberapa tujuan pendidikan Jurusan Kimia adalah sebagai berikut.

1. Menghasilkan sarjana kimia yang berkepribadian baik, menguasai ilmu kimia dan penerapannya, serta memiliki

kompetensi tinggi sehingga dapat bersaing di pasar kerja lokal, nasional dan internasional.

2. Menghasilkan produk penelitian unggul di bidang kimia yang dapat diaplikasikan untuk pengembangan ilmu kimia dan bermanfaat bagi masyarakat.

## **BAB III**

### **KAJIAN PUSTAKA**

#### **3.1 Sistem Informasi Manajemen**

Pada laporan praktek kerja lapangan ini penulis mengambil judul terkait rancang bangun sistem informasi perpustakaan. Berdasarkan judul ini maka penulis membutuhkan suatu kajian pustaka yang terkait tentang sistem informasi. Pada bagian ini akan dibahas mengenai pengertian sistem informasi, komponen sistem informasi dan elemen sistem informasi.

##### **3.1.1 Pengertian Sistem Informasi Manajemen**

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan. Ada beragam definisi sistem informasi, sebagaimana tercantum di bawah ini.

1. Menurut Alter (1992)  
Sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.
2. Menurut Gelinas, Oram, dan Wiggins (1990)  
Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada pemakai.
3. Menurut Turban, McLean, dan Wetherbe (1999)  
Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik.

Dari berbagai definisi tersebut, dapat disimpulkan bahwa sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan. Informasi dapat didefinisikan sebagai hasil dari pengelolaan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian yang nyata yang digunakan untuk pengambilan keputusan. Informasi merupakan data yang telah diklasifikasikan atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan.

Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima.

### 3.1.2 Komponen Sistem Informasi Manajemen

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), yang terdiri dari komponen input, komponen model, komponen *output*, komponen teknologi, komponen *hardware*, komponen *software*, komponen basis data, dan komponen kontrol. Semua komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran.

#### 1. Komponen *input*

Input mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

#### 2. Komponen model

Komponen ini terdiri dari kombinasi prosedur, logika, dan model matematika yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3.    Komponen *output*  
Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.
4.    Komponen teknologi  
Teknologi merupakan “*Tool Box*” dalam sistem informasi, Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, neghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.
5.    Komponen *hardware*  
*Hardware* berperan penting sebagai suatu media penyimpanan vital bagi sistem informasi berfungsi sebagai tempat untuk menampung database atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.
6.    Komponen *software*  
*Software* berfungsi sebagai tempat untuk mengolah, menghitung dan memanipulasi data yang diambil dari hardware untuk menciptakan suatu informasi.
7.    Komponen basis data  
Basis data (database) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (*Database Management System*).
8.    Komponen kontrol  
Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan-

kecurangan, kegagalan-kegagalan sistem itu sendiri, ketidak efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

### 3.1.3 Elemen Sistem Informasi Manajemen

Sistem informasi merupakan kumpulan elemen-elemen yang terdiri dari orang, prosedur, perangkat keras, perangkat lunak, basis data, jaringan komputer dan komunikasi data. Semua elemen ini merupakan komponen fisik.

1. Orang  
Orang atau personil yang di maksudkan yaitu operator komputer, analis sistem, *programmer*, *personal data entry*, dan manajer sistem informasi/EDP
2. Prosedur  
Prosedur merupakan elemen fisik. Hal ini di sebabkan karena prosedur disediakan dalam bentuk fisik seperti buku panduan dan instruksi. Ada 3 jenis prosedur yang dibutuhkan, yaitu instruksi untuk pemakai, instruksi untuk penyiapan masukan, instruksi pengoperasian untuk karyawan pusat komputer.
3. Perangkat keras  
Perangkat keras bagi suatu sistem informasi terdiri atas komputer (pusat pengolah, unit masukan/keluaran), peralatan penyiapan data, dan terminal masukan/keluaran.
4. Perangkat lunak  
Perangkat lunak dapat dibagi dalam 3 jenis utama:
  - a. Sistem perangkat lunak umum, seperti sistem pengoperasian dan sistem manajemen data yang memungkinkan pengoperasian sistem komputer.
  - b. Aplikasi perangkat lunak umum, seperti model analisis dan keputusan.
  - c. Aplikasi perangkat lunak yang terdiri atas program yang secara spesifik dibuat untuk setiap aplikasi.



5. Basis data

*File* yang berisi program dan data dibuktikan dengan adanya media penyimpanan secara fisik seperti *diskette*, *harddisk*, *magnetictape*, dan sebagainya. *File* juga meliputi keluaran tercetak dan catatan lain diatas kertas, mikro film, dan lain sebagainya.

6. Jaringan komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung dalam satu kesatuan. Informasi dan data bergerak melalui kabelkabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data.

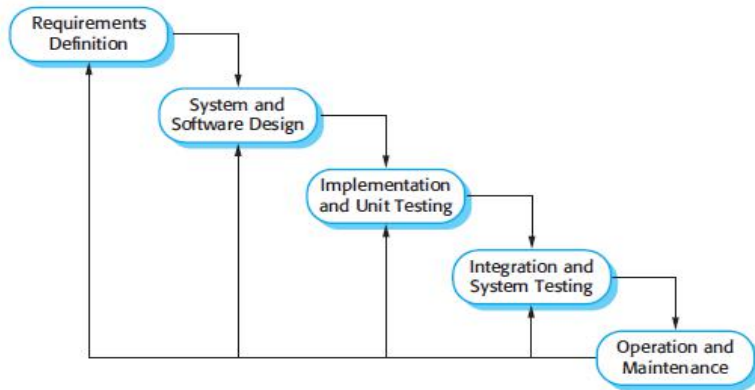
7. Komunikasi data

Komunikasi data adalah merupakan bagian dari telekomunikasi yang secara khusus berkenaan dengan transmisi atau pemindahan data dan informasi diantara komputer-komputer dan piranti-piranti yang lain dalam bentuk digital yang dikirimkan melalui media komunikasi data. Data berarti informasi yang disajikan oleh isyarat digital. Komunikasi data merupakan bagian vital dari suatu sistem informasi karena sistem ini menyediakan infrastruktur yang memungkinkan komputerkomputer dapat berkomunikasi satu sama lain.

### 3.2 Metode Pengembangan Perangkat Lunak

Dalam perancangan dan implementasi sistem informasi manajemen rumah sakit ini digunakan metodologi pengembangan perangkat lunak *Software Development Life Cycle* (SDLC) model proses *Waterfall*. Dimana meodel proses tersebut termasuk dalam model proses klasik yang bersifat sistematis, berurutan dari satu tahap ke tahap lain dalam membangun software (Sommerville, 2011). Model *waterfall* ini mengusulkan suatu pendekatan kepada pengembangan perangkat lunak yang bersifat sistematis dan sekuensial. Dimana hal tersebut dimulai dari tingkat pengembangan sistem pada seluruh analisis, desain sistem, implementasi, pengujian hingga pemeliharaan dari sistem.

Memiliki beberapa tahapan dalam prosesnya, dimana setiap tahapan tersebut harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap yang berikut. Gambar 3.1 berikut merupakan skema dari tahapan – tahapan yang terdapat pada model proses waterfall yaitu sebagai berikut.



**Gambar 3. 1 SDLC Model Proses Waterfall  
(Sommerville, 2011)**

Berikut merupakan pemaparan dari masing – masing tahapan dari skema di atas (Sommerville, 2011), yaitu sebagai berikut.

1. *Requirements Analysis and Definition*, tahap awal yang menetapkan layanan sistem, kendala, dan tujuan yang diperoleh berdasarkan hasil konsultasi dengan pengguna sistem. Kemudian didefinisikan secara rinci yang nantinya dapat difungsikan sebagai spesifikasi dari sistem.
2. *System and Software Design*, tahap ini meliputi mengidentifikasi dan merancang arsitektur atau dasar dari sistem perangkat lunak yang akan dibangun.
3. *Implementation and Unit Testing*, tahap ini ialah mengimplementasikan perancangan fungsi – fungsi perangkat lunak ke dalam bentuk kode program serta dilakukan pengujian agar setiap unit telah memenuhi spesifikasinya.
4. *Integration and System Testing*, tahapan dimana dilakukan pengintegrasian dari unit program individu kemudian

dilakukan pengujian sebagai satu kesatuan sistem yang lengkap guna memastikan telah terpenuhinya persyaratan perangkat lunak. Setelah pengujian dilakukan, maka sistem perangkat lunak telah siap disampaikan kepada pengguna agar dapat dilakukannya *user testing*.

5. *Operation and Maintenance*, merupakan tahapan yang memiliki masa waktu paling lama. Hal tersebut dikarenakan pemeliharaan meliputi beberapa hal seperti memperbaiki kesalahan yang tidak ditemukan pada tahap awal pengembangan, meningkatkan implementasi dari unit sistem dan meningkatkan pelayanan sistem terhadap kebutuhan tambahan yang baru ditemukan.

Keuntungan dari model proses *waterfall* ialah dokumentasi dihasilkan pada setiap tahapan tersebut, dimana hal tersebut dapat berguna untuk model proses perangkat lunak yang lain namun masalah utama dari model proses ini ialah tidak flexibel, pada tahap awal semua kebutuhan harus diketahui secara jelas dan rinci, model proses ini akan sulit untuk merespon perubahan dari kebutuhan pelanggan. Maka dari itu model proses *waterfall* baik digunakan ketika persyaratan telah dipahami dengan baik oleh pengembang sistem (Sommerville, 2011)

### 3.3 Pemrograman Berorientasi Object

Metode pendekatan sistem digunakan untuk mendeskripsikan sistem yang telah maupun akan dirancang dengan metode pengembangan tertentu. Dalam analisis dan perancangan ini digunakan metode pendekatan sistem yang berorientasi objek (Object-Oriented Analysis and Design).

Munawar (2005 : 34), mengemukakan bahwa konsep object oriented dibangun atas beberapa prinsip dasar. Objek adalah contoh atau instance dari sebuah class. Beberapa objek mempunyai attribute dan operation yang sama akan membentuk class. Inheritance, polymorphisme dan encapsulation adalah prinsip-prinsip dasar pada metode Object Oriented. Inheritance lebih berorientasi ke penurunan sifat, polymorphisme lebih menekankan ke penggunaan terminologi operasi yang sejenis. Sedangkan encapsulation lebih menekankan ke

penyembunyian informasi untuk menyederhanakan operasi kepada objek lainnya.

### 3.3.1 Ekapsulasi

Enkapsulasi adalah pembungkus, pembungkus disini dimaksudkan untuk menjaga suatu proses program agar tidak dapat diakses secara sembarangan atau diintervensi oleh program lain. Konsep enkapsulasi sangat penting dilakukan untuk menjaga kebutuhan program agar dapat diakses sewaktu-waktu, sekaligus menjaga program tersebut. Manfaat dari proses enkapsulasi adalah sebagai berikut.

- a. Modularitas  
Kode sumber dari sebuah objek dapat dikelola secara independen dari kode sumber objek yang lain.
- b. Information Hiding  
Karena kita dapat menentukan hak akses sebuah variabel/method dari objek, dengan demikian kita bisa menyembunyikan informasi yang tidak perlu diketahui objek lain.

### 3.3.2 Inheritansi

Merupakan proses menciptakan kelas baru dengan mewarisi karakteristik kelas yang telah ada, atau pewarisan dari satu kelas ke kelas lain. Pewarisan adalah mekanisme ampuh yang memungkinkan kelas mewarisi fungsionalitas kelas yang telah ada. Untuk menciptakan kelas baru, hanya perlu menspesifikasikan cara kelas baru itu berbeda dari kelas yang telah ada. Keuntungan dari inheritance adalah sebagai berikut.

- a. Tidak perlu mendefinisikan Variabel Bila membutuhkan, Cukup menurunkan dari class induk.
- b. Memudahkan dan Mempercepat pemrograman, mengurangi Code.

### 3.3.3 Polimorfisme

Kata *polimorfisme* yang berarti satu objek dengan banyak bentuk yang berbeda, adalah konsep sederhana dalam bahasa pemrograman berorientasi objek yang berarti kemampuan dari suatu variabel referensi objek untuk memiliki aksi berbeda bila method

yang sama dipanggil, dimana aksi method tergantung dari tipe objeknya. Kondisi yang harus dipenuhi supaya *polimorfisme* dapat diimplementasikan adalah sebagai berikut.

1. Method yang dipanggil harus melalui variabel dari basis class atau superclass.
2. Method yang dipanggil harus juga menjadi method dari basis class.
3. Signature method harus sama baik pada superclass maupun subclass.
4. Method access attribute pada subclass tidak boleh lebih terbatas dari basis class.

### 3.3.4 Overriding

Overriding menyatakan keadaan dimana method pada subclass menolak method pada parent classnya. Overriding dipakai saat kita menggunakan method yang sama tapi berbeda implementasinya. Jadi overriding method mempunyai nama method yang sama,

### 3.4 Unified Modelling Language

Menurut Abdul Kadir (2008), *UML* merupakan sintesis dari tiga metode analisis dan perancangan berbasis objek serta ditambah dengan keunggulan metode – metode berorientasi objek lainnya. *UML* juga merupakan bahasa untuk memspesifikasikan, mevisualisasikan, serta mengonstruksi bangunan dasar sistem perangkat lunak, termasuk melibatkan pemodelan aturan – aturan bisnis atau *UML* merupakan perkakas utama untuk analisis dan perancangan sistem berorientasi objek. Beberapa perusahaan besar yang sudah menggunakan metode *UML* adalah IBM dan Microsoft Corp.

*UML* pertama kali diperkenalkan pada tahun 1990 ketika Grady Booch, Ivar Jacobson, dan James Rumbaugh mulai mengadopsi ide – ide serta kemampuan – kemampuan tambahan dari masing – masing metodenya dan berusaha membuat metodologi terpadu yang disebut *UML* yang merupakan metode pengembangan perangkat lunak dengan metode grafis yang relatif mudah dipahami. Usaha pengembangan *UML* dimulai pada Oktober 1994 ketika

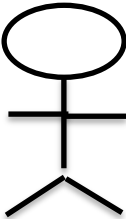

Rumbaugh bergabung dengan Booch di Rational Software Corporation. Tujuan *UML* diantaranya adalah sebagai berikut.






- Memberikan model yang siap pakai, bahasa pemodelan *visual* yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
- Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.

### 3.4.1 Use Case Diagram

*Use case diagram* adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. Berikut pengertian *use case diagram* menurut Bennet et al. (2005, p20) “*Use Case Diagram* merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili user atau sistem lain yang berinteraksi dengan sistem yang dimodelkan.”

**Tabel 3. 1 Simbol Use Case Diagram**

Simbol	Penjelasan
	<i>Actor</i> adalah pengguna sistem. Actor tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan <i>input</i> atau memberikan <i>output</i> , maka aplikasi tersebut juga bisa dianggap sebagai <i>actor</i>
	<i>Association</i> digunakan untuk menghubungkan <i>actor</i> dengan <i>usecase</i> . Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara <i>actor</i> dengan <i>usecase</i>

	<i>Use Case</i> digambarkan sebagai lingkaran elips dengan nama <i>usecase</i> dituliskan didalam elips tersebut
	<i>System</i> Tempat seluruh aktivitas-aktivitas sistem yang sedang berjalan.
	<i>Dependancy</i> Untuk menggambarkan ketergantungan sebuah <i>use case</i> dengan <i>use case</i> lainnya.
<p data-bbox="263 534 408 558">&lt;&lt;Include&gt;&gt;</p> 	<i>Include</i> , menggambarkan bahwa keseluruhan dari sebuah <i>use case</i> merupakan fungsionalitas <i>use case</i> lainnya.
<p data-bbox="263 702 408 726">&lt;&lt;Extend&gt;&gt;</p> 	<i>Extend</i> , menggambarkan hubungan antar <i>use case</i> dimana bahwa sebuah <i>use case</i> merupakan fungsionalitas <i>use case</i> lainnya apabila kondisi tertentu terpenuhi.

*Use Case* diagram menyajikan interaksi antara *use case* dan aktor. Dimana, aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun dan juga menggambarkan fungsionalitas sistem atau persyaratan – persyaratan yang harus dipenuhi sistem dari pandangan pemakai.

### 3.4.2 Activity Diagram

Activity diagram menurut Martin Fowler (2005 : 163) adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa 21 hal, activity diagram memainkan peran mirip diagram alir, tetapi perbedaan prinsip antara notasi diagram alir adalah activity diagram mendukung behavior paralel. Node pada sebuah activity diagram disebut sebagai action, sehingga

diagram tersebut menampilkan sebuah activity yang tersusun dari action.

### 3.4.3 Class Diagram

Class diagram menurut Munawar (2005 : 28) merupakan himpunan dari objek-objek yang sejenis. Sebuah objek memiliki keadaan sesaat (state) dan perilaku (behavior). State sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam attribute/properties. Sedangkan perilaku suatu objek mendefinisikan bagaimana sebuah objek bertindak/beraksi dan memberikan reaksi.

### 3.5 Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam system secara abstrak. Sehingga jelas bahwa ERD berbeda dengan DFD yang merupakan suatu model jaringan fungsi yang akan dilaksanakan oleh system, sedangkan ERD merupakan model jaringan data yang menekankan pada struktur-struktur *relationship* data. *Entity Relationship Diagram* adalah notasi grafik dari sebuah model data atau sebuah model jaringan yang menjelaskan tentang data yang tersimpan (*storage data*) dalam system secara abstrak. Diagram hubungan entitas tidak menyatakan bagaimana memanfaatkan data, membuat data, mengubah data dan menghapus data. Elemen-elemen diagram hubungan entitas terdiri atas beberapa elemen sebagai berikut.

#### 1. *Entity*

Pada E-R diagram, *entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang terdapat di dalam system, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu didalamnya).

#### 2. *Relationship*

Pada E-R diagram, *relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung atau hubungan (*relationship*) diberi nama dengan



kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat pasif dan bisa dengan kalimat aktif). Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang.

### 3. Atribut

Secara umum atribut adalah sifat atau karakteristik dari setiap entitas maupun tiap *relationship*, sehingga atribut adalah sesuatu yang menjelaskan apa yang sebenarnya dimaksud entitas maupun *relationship*, sehingga sering dikatakan atribut adalah elemen dari setiap entitas dan *relationship*. Atribut *value* atau nilai atribut adalah suatu *occurrence* tertentu dari sebuah atribut di dalam suatu *entity* atau *relationship*. Terdapat dua jenis atribut yaitu sebagai berikut.

- a. *Identifier (key)* digunakan untuk menentukan suatu *entity* secara unik (*primary key*).
- b. *Descriptor (non key attribute)* digunakan untuk spesifikasi karakteristik dari suatu *entity* yang tidak unik.

### 4. Kardinalitas

Kardinalitas relasi menunjukkan jumlah maksimum *tupel* yang dapat memiliki relasi dengan entitas yang lain. Dari sejumlah kemungkinan banyaknya hubungan antar entitas tersebut, kardinalitas relasi merujuk pada hubungan maksimum yang terjadi dari entitas satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat tiga macam kardinalitas relasi yaitu sebagai berikut.

- a. *One to One*  
Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.
- b. *One to Many* atau *Many to One*  
Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu, tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang

pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas pertama.

c. *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika setiap kejadian pada sebuah entitas akan mempunyai hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua.

### 3.6 Normalisasi Database

Proses normalisasi pertama kali diperkenalkan oleh E.F.Codd pada tahun 1972. normalisasi sering dilakukan sebagai suatu uji coba pada suatu relasi secara berkelanjutan untuk menentukan apakah relasi tersebut sudah baik atau masih melanggar aturan-aturan standar yang diperlakukan pada suatu relasi yang normal (sudah dapat dilakukan proses insert, update, delete, dan modify pada satu atau beberapa atribut tanpa mempengaruhi integritas data dalam relasi tersebut). Proses normalisasi merupakan metode yang formal/standar dalam mengidentifikasi dasar relasi bagi primary keynya (atau candidate key dalam kasus BCNF), dan dependensi fungsional diantara atribut-atribut dari relasi tersebut. Normalisasi akan membantu perancang basis data dengan menyediakan suatu uji coba yang berurutan yang dapat diimplementasikan pada hubungan individual sehingga skema relasi dapat di normalisasi ke dalam bentuk yang lebih spesifik untuk menghindari terjadinya error atau inkonsistensi data, bila dilakukan update terhadap relasi tersebut dengan Anomaly.

Beberapa langkah – langkah normalisasi database adalah sebagai berikut.

a. First Normal Form (1NF)

Bentuk normal 1NF terpenuhi jika sebuah tabel tidak memiliki atribut bernilai banyak (multivalued attribute), atribut composite atau kombinasinya dalam domain data yang

sama. Setiap atribut dalam tabel tersebut harus bernilai atomic (tidak dapat dibagi-bagi lagi)

Pada intinya bentuk normalisasi 1NF ini mengelompokkan beberapa tipe data atau kelompok data yang sejenis agar dapat dipisahkan sehingga anomali data dapat di atasi. Contoh adalah ketika kita ingin menghapus, mengupdate, atau menambahkan data peminjam, maka kita tidak bersinggungan dengan data buku atau data penerbit. Sehingga inkonsistensi data dapat mulai di jaga.

b. Second Normal Form (2NF)

Syarat untuk menerapkan normalisasi bentuk kedua ini adalah data telah dibentuk dalam 1NF, berikut adalah beberapa fungsi normalisasi 2NF.

- Menghapus beberapa subset data yang ada pada tabel dan menempatkan mereka pada tabel terpisah.
- Menciptakan hubungan antara tabel baru dan tabel lama dengan menciptakan foreign key.
- Tidak ada atribut dalam tabel yang secara fungsional bergantung pada candidate key tabel tersebut.

c. Thrid Normal Form (3NF)

Normalisasi database dalam bentuk 3NF bertujuan untuk menghilangkan seluruh atribut atau field yang tidak berhubungan dengan primary key. Dengan demikian tidak ada ketergantungan transitif pada setiap kandidat key. Syarat dari bentuk normal ketiga atau 3NF adalah sebagai berikut.

- Memenuhi semua persyaratan dari bentuk normal kedua.
- Menghapus kolom yang tidak tergantung pada primary key.

### 3.7 MySQL

MySQL merupakan *database* yang dikembangkan dari bahasa *Structure Query Language* (SQL) yang merupakan bahasa yang terstruktur yang digunakan untuk interaksi antara script program dengan *database server* dalam hal pengelolaan data (Yank,

2012). Penggunaan SQL memungkinkan pembuatan tabel nantinya diisi dengan data, memanipulasi data, serta membuat suatu perhitungan dengan berdasarkan data yang ditemukan.

Sbagai *software* resmi, MySQL dikembangkan oleh perusahaan Swedia bernama MySQL AB, yang waktu itu bernama TcX Data Konsult AB. Awalnya MySQL memakai nama mSQL sebagai antarmuka yang digunakan. Ternyata penggunaan mSQL itu mengalami banyak hambatan seperti sangat lambat dan tidak fleksibel. Guna menanggulangi hal tersebut, Michael Widenius berusaha mengembangkan *interface* yang lebih baik sehingga ditemukan MySQL. Saat itu MySQL didistribusikan secara gratis untuk keperluan nonkomersial, sedangkan diharuskan membayar lisensi untuk kebutuhan komersial. MySQL dikategorikan sebagai *software* berlisensi GPL dimana dapat digunakan secara gratis untuk kebutuhan apapun sejak kemunculan versi 3.23.19.

### 3.8 Structured Query Language (SQL)

Merupakan bahasa *query* yang digunakan untuk mengakses *database* relasional yang menjadi bahasa *database* standar. Terdiri dari berbagai jenis *statement* yang didesain agar memungkinkan berhubungan dengan *database* secara interaktif. Penggunaan SQL pada DBMS (*Database Management System*) sudah cukup luas. SQL dapat dipakai oleh berbagai kalangan, misalnya DBA (*Database Administrator*), *programmer* ataupun pengguna, dikarenakan beberapa alasan berikut yaitu :

1. SQL sebagai bahasa administrasi *database*, dimana dalam hal ini SQL dipakai oleh *database administrator* untuk menciptakan serta mengendalikan pengaksesan *database*.
2. SQL sebagai bahasa *query* interaktif, dimana pengguna dapat memberikan berbagai perintah untuk mengakses *database* yang sesuai kebutuhan.
3. SQL sebagai bahasa pemrograman *database*, dimana programmer dapat menggunakan berbagai perintah SQL dalam program aplikasi yang dibuat.
4. SQL sebagai bahasa *client/server*, dimana SQL dapat digunakan untuk mengimplementasikan sistem *client/server*. Sebuah *client* dapat menjalankan suatu aplikasi yang

mengakses *database* sehingga dalam hal ini sistem operasi antara *server* dan *client* bisa berbeda. Selain itu SQL juga diterapkan pada *internet* atau *intranet* untuk mengakses *database* melalui halaman web untuk mendukung konsep web dinamis.

### 3.8.1 Kelompok Pernyataan SQL

Pernyataan SQL dapat dikelompokkan menjadi 2 kelompok yaitu DDL (*Data Definition Language*) dan DML (*Data Manipulation Language*). Berikut pemaparan mengenai hal tersebut.

1. DDL (*Data Definition Language*)  
DDL merupakan kelompok perintah yang berfungsi untuk mendefinisikan atribut – atribut *database*, tabel, atribut (kolom), batasan – batasan terhadap suatu atribut serta hubungan antar tabel. Yang termasuk kelompok DDL ini ialah *CREATE* untuk menciptakan tabel atau indeks, *ALTER* untuk mengubah struktur table, dan *DROP* untuk menghapus tabel atau indeks
2. DML (*Data Manipulation Language*)  
DML adalah kelompok perintah yang berfungsi untuk memanipulasi data, misalnya untuk pengambilan, penyisipan pengubahan dan penghapusan data. Yang termasuk DML ialah *SELECT* untuk memilih data, *INSERT* untuk menambah data, *DELETE* untuk menghapus data, dan *UPDATE* untuk mengubah data.

### 3.8.2 Kelebihan MySQL

Ada beberapa alasan mengapa MySQL menjadi program *database* yang sangat populer dan digunakan oleh banyak orang. Alasan – alasan tersebut adalah sebagai berikut.

1. MySQL ialah *free software database* yang memiliki kecepatan yang tinggi dalam melakukan pemrosesan data, dapat diandalkan, dan mudah digunakan serta mudah dipelajari.
2. *Fully Multi Threaded* dengan *kernel thread* dimana dapat dengan mudah mempergunakan multiple CPU bila ada.
3. Dapat melakukan koneksi dengan *client* menggunakan *protocol* TCP/IP, Unix socket (Unix), atau Named Pipes (NT).

4. Mendukung banyak bahasa pemrograman seperti C, C++, Perl, Python, Java, dan PHP. Selain itu, dengan bantuan ODBC, MySQL juga mampu berinteraksi dengan berbagai pemrograman visual seperti Delphi, Visual Basic, Java, dan sebagainya.
5. Dapat menangani data dengan skala yang sangat besar dengan jumlah record mencapai lebih dari 50 juta, menampung 60 ribu tabel, dan juga bisa menampung 5 milyar baris data.
6. *Multiuser* yang memungkinkan dalam satu *database server* pada MySQL dapat diakses oleh beberapa user dalam waktu yang sama tanpa mengalami konflik atau kemacetan sistem.
7. Dalam hal relasi antartabel MySQL menerapkan metode *one-sweep multijoin*, sehingga sangat efisien dalam mengelola informasi yang kita minta dari beberapa tabel sekaligus.
8. *Security* yang dimiliki database MySQL dikenal baik, karena memiliki lapisan *security* seperti *levelsubnetmask*, nama *host* dan izin akses *user* dengan sistem perizinan yang khusus serta *password* yang dimiliki setiap *user* dalam bentuk data terenkripsi.

## **BAB IV**

### **PELAKSANAAN PKL**

#### **4.1 Perancangan Sistem Informasi Perpustakaan**

Pada sistem informasi manajemen perpustakaan melibatkan dua entitas yaitu anggota perpustakaan dan petugas perpustakaan. Anggota perpustakaan hanya dapat melihat data buku dan data skripsi yang tersedia. Sedangkan petugas perpustakaan dapat melakukan pengelolaan terhadap data buku, data skripsi, data anggota, data peminjaman dan data pengembalian buku yang terjadi di perpustakaan. Perancangan sistem informasi perpustakaan ini diawali dengan pendefinisian kebutuhan awal hingga tahap implementasi dengan memanfaatkan metodologi pengembangan perangkat lunak.

##### **4.1.1 Metode Pengembangan Sistem**

Dalam perancangan dan implementasi sistem informasi manajemen perpustakaan digunakan metodologi pengembangan perangkat lunak *Software Development Life Cycle (SDLC)* model proses *Waterfall*. Dimana model proses tersebut termasuk dalam model proses klasik yang bersifat sistematis, berurutan dari satu tahap ke tahap lain dalam pembangunan software. Model *waterfall* ini mengusulkan suatu pendekatan kepada pengembangan perangkat lunak yang bersifat sistematis dan sekuensial. Dimana hal tersebut dimulai dari tingkat pengembangan sistem pada seluruh analisis, desain sistem, implementasi pengujian hingga pemeliharaan dari sistem. Tahapan yang dilakukan dalam pengembangan sistem informasi manajemen perpustakaan adalah sebagai berikut.

1. *Requirements Analysis and Definition*, dimana pada tahap ini dilakukan pendefinisian terkait kebutuhan pengguna terhadap sistem, menetapkan layanan atau fitur pada sistem, data dasar atau informasi yang akan dihasilkan, dan tujuan yang diperoleh berdasarkan hasil konsultasi dengan pengguna sistem. Pendefinisian secara rinci tersebut digunakan sebagai acuan dalam penentuan spesifikasi dari sistem. Dengan kata lain dalam

hal ini penulis melakukan studi lapangan dengan metode wawancara terhadap petugas perpustakaan.

**Tabel 4. 1 Analisis Kebutuhan Sistem Perpustakaan**

No	Pengguna	Kebutuhan	Deskripsi Kebutuhan
1	Anggota Perpustakaan	Menampilkan data buku dan data skripsi	Sistem dapat memberikan informasi mengenai buku yang tersedia di perpustakaan
			Sistem dapat memberikan informasi mengenai skripsi yang tersedia di perpustakaan
			Sistem dapat melakukan pencarian data buku berdasarkan inputan pencarian dari <i>user</i>
			Sistem dapat melakukan pencarian data skripsi berdasarkan inputan pencarian dari <i>user</i>
2	Petugas Perpustakaan	<i>Maintance</i> data buku	Sistem dapat melakukan penambahan data buku



			Sistem dapat melakukan <i>update</i> data buku
			Sistem dapat melakukan penghapusan data buku
3	Petugas Perpustakaan	<i>Maintance</i> data skripsi	Sistem dapat melakukan penambahan data skripsi
			Sistem dapat melakukan <i>update</i> data skripsi
			Sistem dapat melakukan penghapusan data skripsi
4	Petugas Perpustakaan	<i>Maintance</i> data anggota	Sistem dapat melakukan penambahan data anggota
			Sistem dapat melakukan <i>update</i> data anggota
			Sistem dapat melakukan penghapusan data anggota

5	Petugas Perpustakaan	<i>Maintance</i> data peminjaman buku	Sistem dapat melakukan penambahan data peminjaman buku
			Sistem dapat melakukan <i>update</i> data peminjaman buku
			Sistem dapat melakukan penghapusan data buku
6	Petugas Perpustakaan	<i>Maintance</i> data pengembalian buku	Sistem dapat melakukan penambahan data pengembalian buku
			Sistem dapat melakukan <i>update</i> data pengembalian buku
			Sistem dapat melakukan penghapusan data pengembalian buku

2. *System and Software Design*, tahap ini dilaksanakan perancangan arsitektur atau desain dari sistem beserta fitur yang diperlukan. Hal ini dilakukan dengan cara berkonsultasi langsung dengan pengguna sehingga masukan dan arahan dari pengguna mengenai desain sistem, fitur sistem, serta

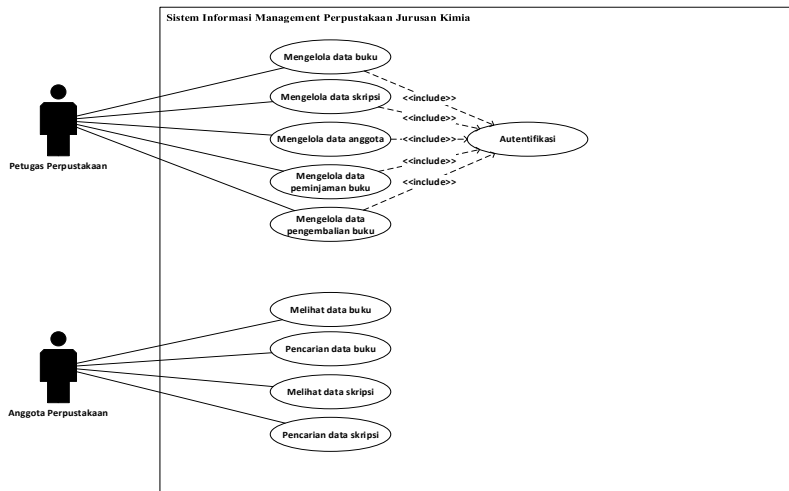
- fungsionalitas dari sistem yang diinginkan dapat dijadikan acuan untuk membangun sistem yang lebih *user-friendly*.
3. *Implementation and Unit Testing*, setelah tahap pendefinisian kebutuhan sistem dan desain sistem selesai, maka selanjutnya mengimplementasikan informasi dari tahap sebelumnya kedalam bentuk kode program. Dalam implementasi ini digunakan bahasa pemrograman java. Pada tahap ini juga dilakukan pengujian agar setiap unit dapat memenuhi spesifikasinya.
  4. *Integration and system testing*, pada tahap ini dilaksanakan pengintegrasian dari segala fitur yang telah dibuat sehingga seluruh fitur dari berbagai unit dapat tersinkronisasi sehingga mampu berjalan dengan baik sesuai dengan kebutuhan fungsional sistem. Setelah pengujian dilakukan, maka sistem perangkat lunak siap disampaikan kepada pengguna agar dapat dilakukannya *user testing* melalui training atau pendampingan penggunaan sistem terhadap masing – masing *user*, pada masing – masing unit. Hal tersebut dilakukan untuk menghindari kesalahan operasional dari *user*, agar *user* dapat beradaptasi dan mampu menggunakan sistem dengan baik.
  5. *Operation and Maintenance*, tahapan ini sistem telah digunakan sepenuhnya oleh *user*, namun tim pengembang tetap melaksanakan pengawasan atau memantau perkembangan sistem guna menanggulangi timbulnya kendala yang tidak ditemukan pada tahap awal, meningkatkan implementasi dari unit sistem dan meningkatkan pelayanan sistem terhadap kebutuhan tambahan yang baru ditemukan yang dapat digunakan sebagai acuan untuk pengembangan berikutnya.

## 4.2 *Unified Modelling Language(UML)*

### 4.2.1 *Use Case Diagram*

Pada *use case diagram* diatas terdapat dua aktor yang terlibat dalam sistem informasi perpustakaan yaitu petugas perpustakaan dan anggota perpustakaan. Petugas melakukan pengelolaan terhadap data skripsi, anggota, peminjaman, pengembalian dan buku, yang dimana dalam pengelolaan tersebut meliputi *insert*, *update*, dan *delete* pada

masing – masing data. Sebelum melakukan proses *insert*, *update* dan *delete* petugas perpustakaan harus melakukan *login* terlebih dahulu. Sedangkan aktor anggota perpustakaan bisa melihat dan melakukan pencarian data yang tersedia di perpustakaan yang meliputi data skripsi dan data buku.



**Gambar 4. 1 Use Case Diagram Sistem Informasi Management Perpustakaan**

Definisi aktor pada *Use Case Diagram* Sistem Informasi Management Pperpustakaan ditunjukkan pada tabel 4.2.

**Tabel 4. 2 Definisi Aktor**

No	Aktor	Deskripsi
1	Petugas Perpustakaan	Petugas perpustakaan bertindak sebagai admin sistem yang bertugas mengelola data buku, data skripsi, data anggota, data peminjaman dan data pengembalian .

2	Anggota Perpustakaan	Anggota perpustakaan dapat melihat dan melakukan pencarian data buku dan data skripsi
---	----------------------	---

Definisi *Use Case* pada *Use Case* Diagram Sistem Informasi *Management* Perpustakaan ditunjukkan pada tabel 4.3.

**Tabel 4. 3 Definisi *Use Case***

No	<i>Use Case</i>	Deskripsi
1	Autentifikasi	Proses pengecekan dan validasi dari <i>username</i> dan <i>password</i> yang diinputkan oleh <i>user</i> atau admin
2	Mengelola data buku	Proses untuk mengeloa data buku yang terdapat sistem yang meliputi tambah, <i>edit</i> , hapus dan lihat data
3	Mengelola data skripsi	Proses untuk mengeloa data skripsi yang terdapat pada sistem yang meliputi tambah, <i>edit</i> , hapus dan lihat data
4	Mengelola data anggota	Proses untuk mengelola data anggota yang terdapat pada sistem yang meliputi tambah, <i>edit</i> , hapus dan lihat data
5	Mengelola data peminjaman buku	Proses untuk mengeloa data peminjaman yang terdapat pada sistem yang meliputi tambah, <i>edit</i> , hapus dan lihat data

6	Mengelola data pengembalian buku	Proses untuk mengelola data anggota yang terdapat pada sistem yang meliputi tambah, <i>edit</i> , hapus dan lihat data
7	Lihat data buku	Proses dimana <i>user</i> dapat melihat data buku yang tersedia di perpustakaan
8	Lihat data skripsi	Proses dimana <i>user</i> dapat melihat data skripsi yang tersedia di perpustakaan
9	Pencarian data buku	Proses dimana <i>user</i> dapat memperoleh informasi data buku tertentu sesuai dengan <i>key word</i> yang diinputkan <i>user</i>
10	Pencarian data skripsi	Proses dimana <i>user</i> dapat memperoleh informasi data skripsi tertentu sesuai dengan <i>key word</i> yang diinputkan <i>user</i>

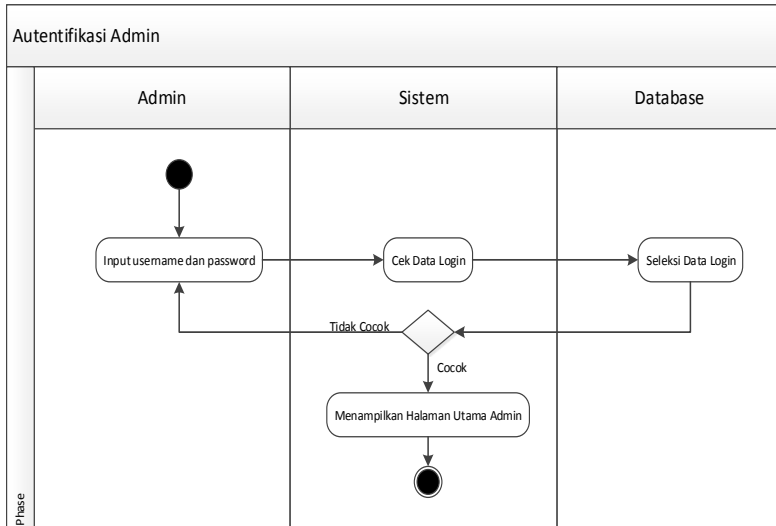
## 4.2.2 Activity Diagram

*Activity Diagram* menjabarkan aktivitas yang terdapat pada setiap *use case* pada *use case diagram*. *Activity diagram* dibagi menjadi dua yaitu *activity diagram* yang menjabarkan aktivitas yang dilakukan oleh admin dan *activity diagram* yang menjabarkan aktivitas yang dilakukan oleh *user*.

### 4.2.2.1 Activity Diagram untuk Admin

*Activity diagram* pada gambar 4.2 menggambarkan proses autentifikasi yang dilakukan oleh admin. Pada proses autentifikasi pada gambar 4.2, pertama admin diminta untuk memasukkan *username* dan *password*. Jika *username* dan *password* yang dimasukkan sudah sesuai dengan data *login* yang terdapat pada

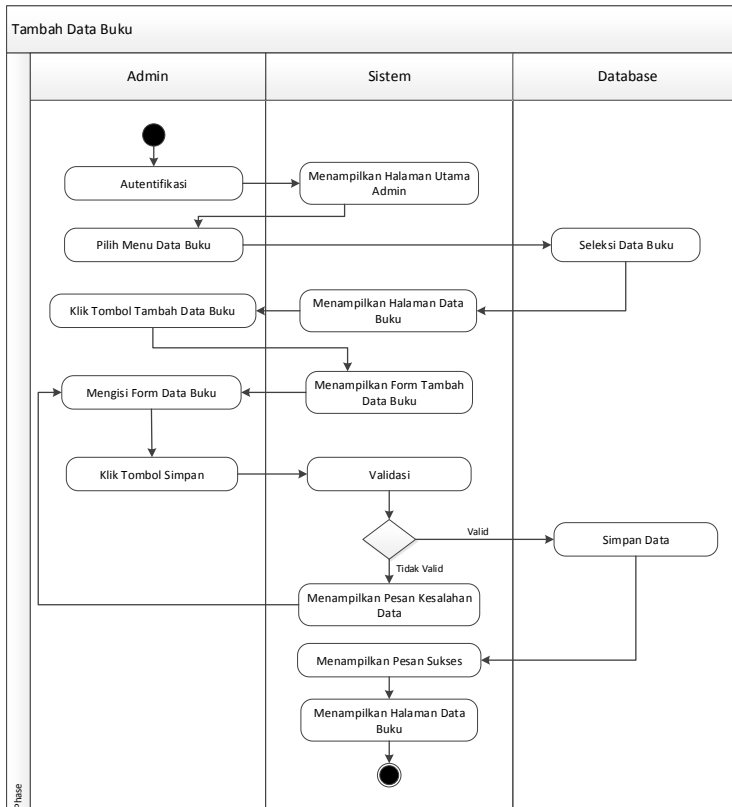
*database* maka admin berhak untuk mengakses halaman utama admin. Namun jika tidak sesuai dengan data *login* yang terdapat pada *database* maka admin harus memasukkan kembali *username* dan *password*.



**Gambar 4. 2 Activity Diagram Autentifikasi untuk Admin**

Gambar 4.3 menggambarkan *activity diagram* tambah data buku. Tahap pertama yang dilakukan admin adalah melakukan *login* ke sistem, jika proses *login* berhasil maka akan ditampilkan form utama admin. Setelah *form* utama admin ditampilkan maka dipilih menu data buku, untuk dapat melakukan penambahan data buku dipilih *button* tambah data maka sistem akan memunculkan *form* penginputan data buku. Setelah selesai mengisi semua data pada form dan dipilih *button* simpan maka dilakukan pengecekan terhadap data yang diinputkan, jika data yang diinputkan tidak valid maka sistem akan mengeluarkan pesan kesalahan data dan pengisian data harus diulang kembali sedangkan jika data yang diinputkan

valid, data akan disimpan pada *database* dan admin dialihkan ke halaman data buku.

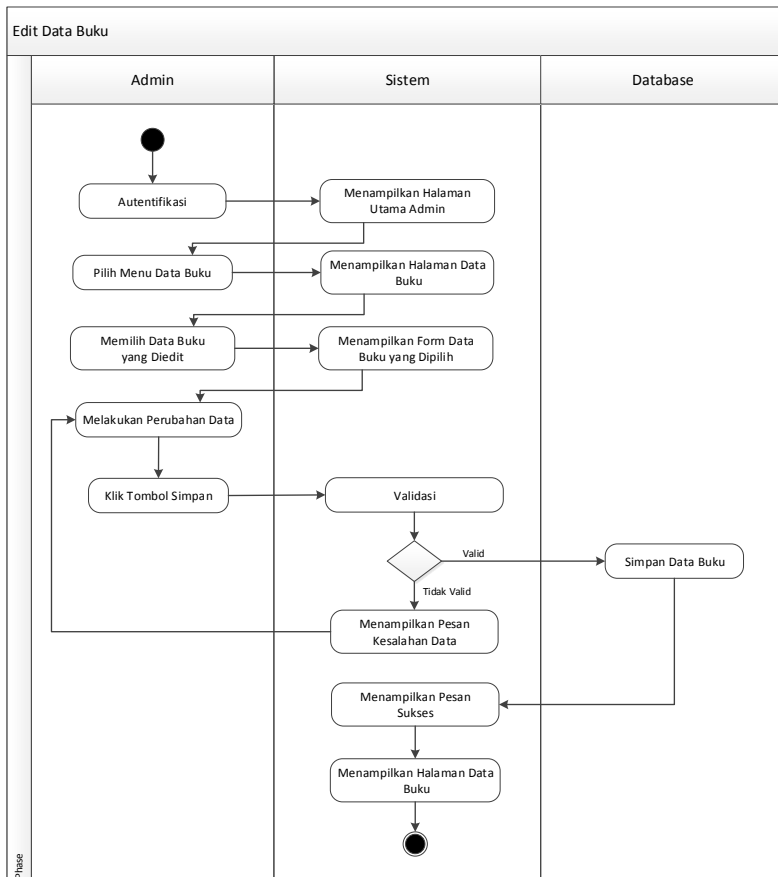


**Gambar 4.3 Activity Diagram Tambah Data Buku**

Gambar 4.4 menggambarkan proses edit data buku. Admin akan memilih terlebih dahulu data yang akan diedit, kemudian sistem akan menampilkan *form* yang berisikan data buku yang dipilih. Admin kemudian melakukan perubahan data dan setelah itu admin memilih *button* simpan. Data yang

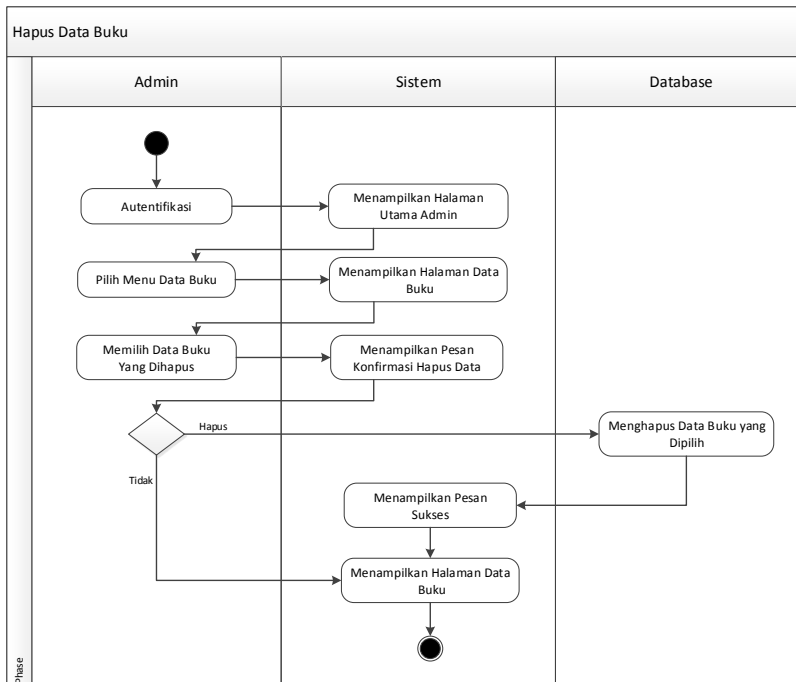


telah diedit kemudian divalidasi oleh sistem. Apabila data tersebut valid maka semua perubahan akan disimpan kedalam *database* dan admin diarahkan kembali ke halaman data buku namun jika tidak valid admin melakukan pengecekan kembali terhadap data yang diedit.



**Gambar 4.4 Activity Diagram Edit Data Buku**

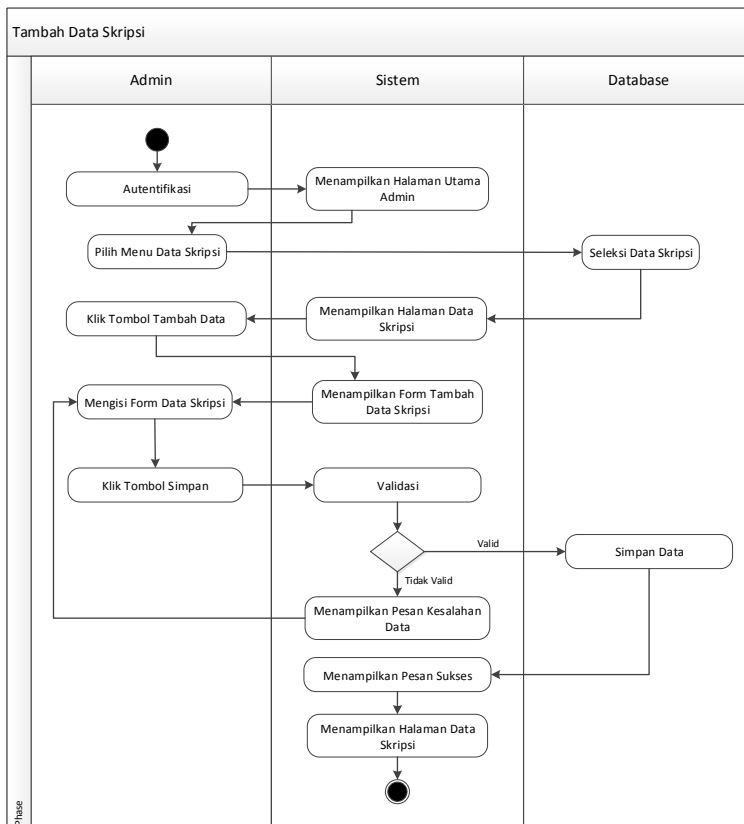
Gambar 4.5 menggambarkan proses hapus data buku. Admin memilih salah satu data buku yang akan dihapus dan kemudian memilih *button* hapus. Kemudian sistem akan menampilkan pesan konfirmasi. Jika admin mengkonfirmasi penghapusan data, maka data yang bersangkutan dihapus dari *database* dan admin dialihkan ke halaman data buku.



**Gambar 4.5 Activity Diagram Hapus Data Buku**

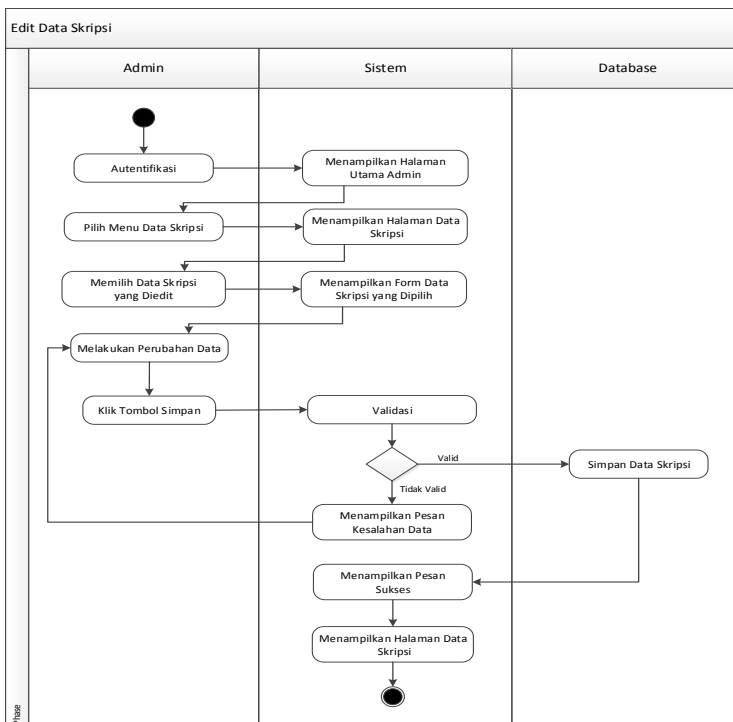
Gambar 4.6 menggambarkan *activity diagram* tambah data skripsi. Tahap pertama yang dilakukan admin adalah melakukan *login* ke sistem, jika proses *login* berhasil maka akan ditampilkan form utama admin. Setelah *form* utama admin ditampilkan maka dipilih menu data skripsi, untuk dapat melakukan penambahan data skripsi dipilih *button* tambah data

maka sistem akan memunculkan *form* penginputan data skripsi. Setelah selesai mengisi semua data pada form dan dipilih *button* simpan maka dilakukan pengecekan terhadap data yang diinputkan, jika data yang diinputkan tidak valid maka sistem akan mengeluarkan pesan kesalahan data dan pengisian data harus diulang kembali sedangkan jika data yang diinputkan valid, data akan disimpan pada *database* dan admin dialihkan ke halaman data skripsi.



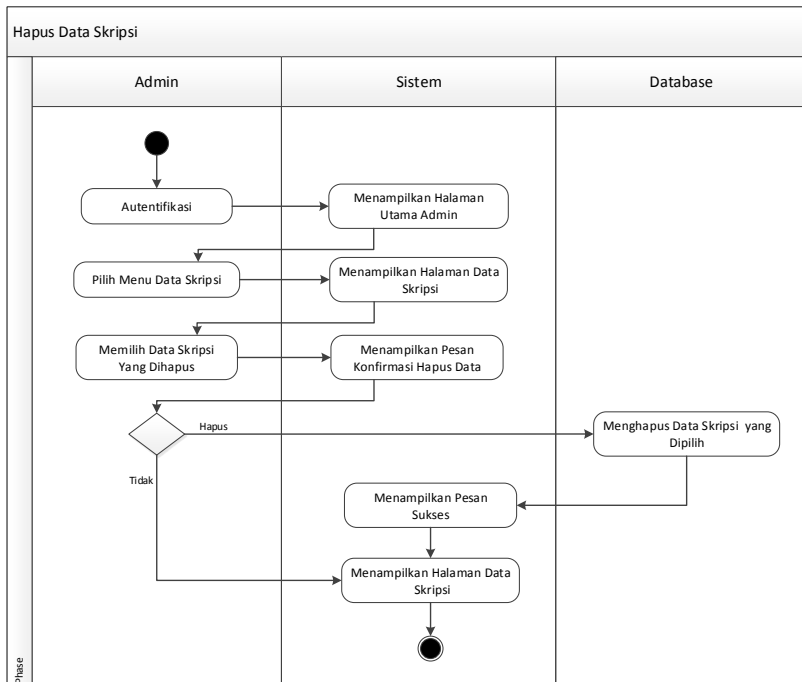
**Gambar 4. 6 Activity Diagram Tambah Data Skripsi**

Gambar 4.7 menggambarkan proses edit data skripsi. Admin akan memilih terlebih dahulu data yang akan diedit, kemudian sistem akan menampilkan *form* yang berisikan data skripsi yang dipilih. Admin kemudian melakukan perubahan data dan setelah itu admin memilih *button* simpan. Data yang telah diedit kemudian divalidasi oleh sistem. Apabila data tersebut valid maka semua perubahan akan disimpan kedalam *database* dan admin diarahkan kembali ke halaman data skripsi namun jika tidak valid admin melakukan pengecekan kembali terhadap data yang diedit.



**Gambar 4.7 Activity Diagram Edit Data Skripsi**

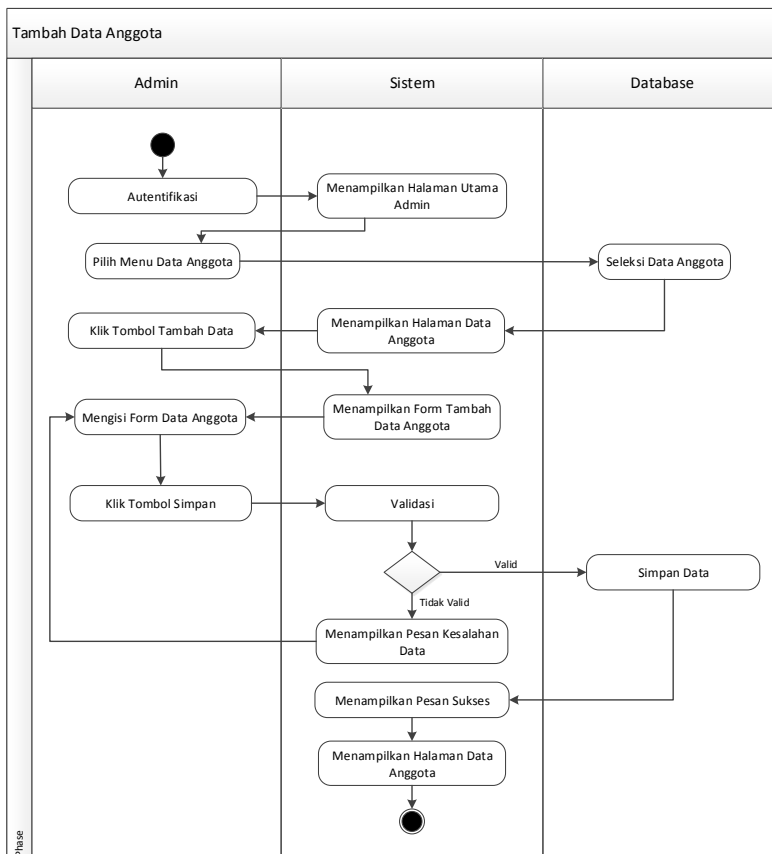
Gambar 4.8 menggambarkan proses hapus data skripsi. Admin memilih salah satu data skripsi yang akan dihapus dan kemudian memilih *button* hapus. Kemudian sistem akan menampilkan pesan konfirmasi. Jika admin mengkonfirmasi penghapusan data, maka data yang bersangkutan dihapus dari *database* dan admin dialihkan ke halaman data skripsi.



**Gambar 4.8 Activity Diagram Hapus Data Skripsi**

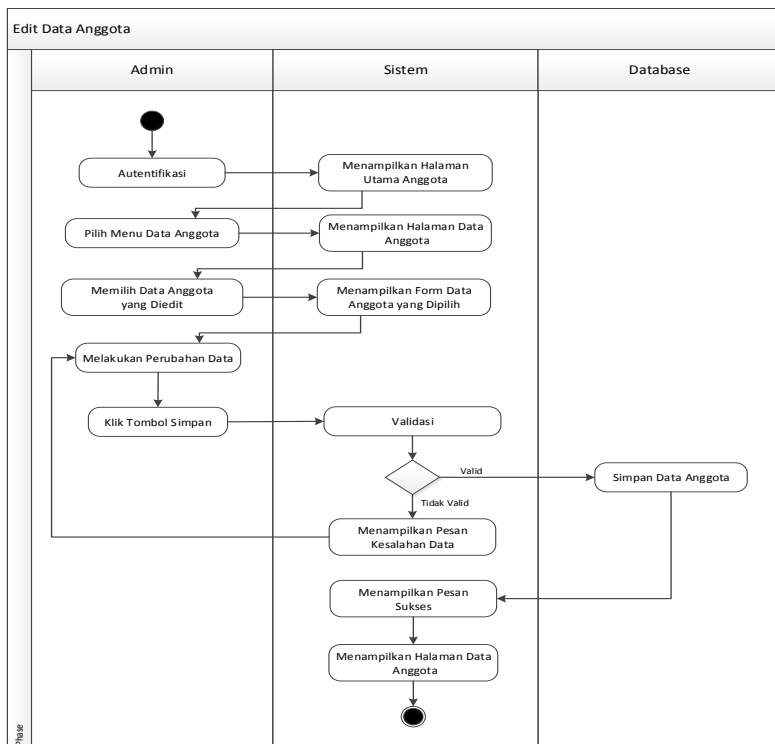
Gambar 4.9 menggambarkan *activity diagram* tambah data anggota. Tahap pertama yang dilakukan admin adalah melakukan *login* ke sistem, jika proses *login* berhasil maka akan ditampilkan form utama admin. Setelah *form* utama admin ditampilkan maka dipilih menu data anggota, untuk dapat melakukan penambahan data skripsi dipilih *button*

tambah data maka sistem akan memunculkan *form* penginputan data anggota. Setelah selesai mengisi semua data pada form dan dipilih *button* simpan maka dilakukan pengecekan terhadap data yang diinputkan, jika data yang diinputkan tidak valid maka sistem akan mengeluarkan pesan kesalahan data dan pengisian data harus diulang kembali sedangkan jika data yang diinputkan valid, data akan disimpan pada *database* dan admin dialihkan ke halaman data anggota.



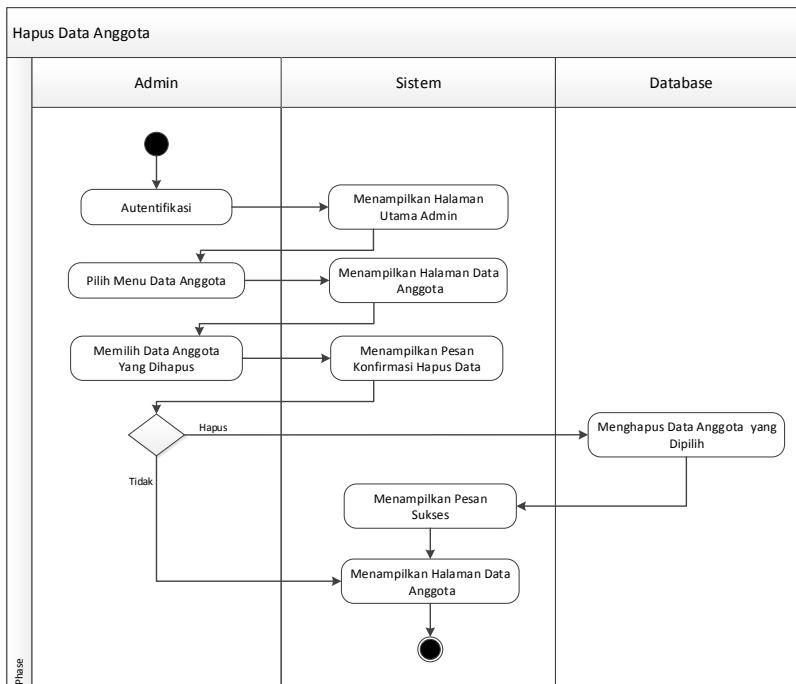
**Gambar 4. 9 Activity Diagram Tambah Data Anggota**

Gambar 4.10 menggambarkan proses edit data anggota. Admin akan memilih terlebih dahulu data yang akan diedit, kemudian sistem akan menampilkan *form* yang berisikan data anggota yang dipilih. Admin kemudian melakukan perubahan data dan setelah itu admin memilih *button* simpan. Data yang telah diedit kemudian divalidasi oleh sistem. Apabila data tersebut valid maka semua perubahan akan disimpan kedalam *database* dan admin diarahkan kembali ke halaman data anggota namun jika tidak valid admin melakukan pengecekan kembali terhadap data yang diedit.



**Gambar 4. 10 Activity Diagram Edit Data Anggota**

Gambar 4.11 menggambarkan proses hapus data anggota. Admin memilih salah satu data anggota yang akan dihapus dan kemudian memilih *button* hapus. Kemudian sistem akan menampilkan pesan konfirmasi. Jika admin mengkonfirmasi penghapusan data, maka data yang bersangkutan dihapus dari *database* dan admin dialihkan ke halaman data anggota.

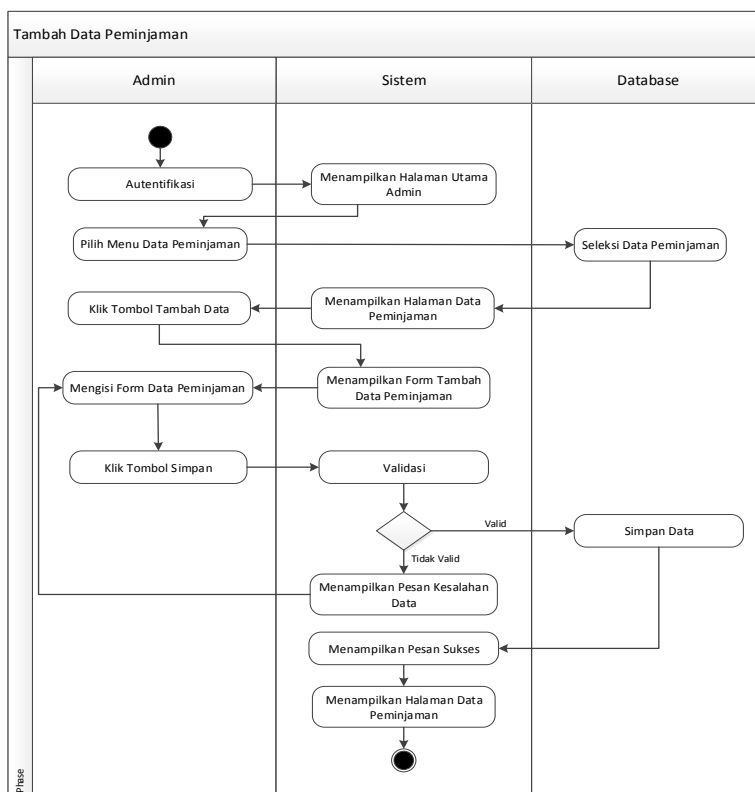


**Gambar 4. 11 Activity Diagram Hapus Data Anggota**

Gambar 4.12 menggambarkan *activity diagram* tambah data peminjaman buku. Tahap pertama yang dilakukan admin adalah melakukan *login* ke sistem, jika proses *login* berhasil maka akan ditampilkan form utama admin. Setelah *form* utama

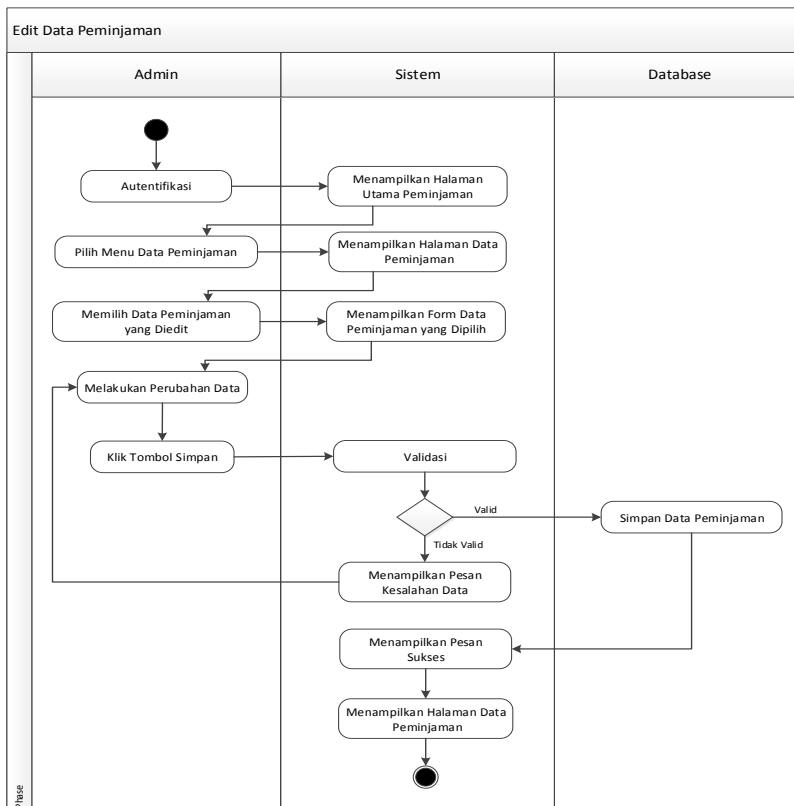


admin ditampilkan maka dipilih menu data peminjaman, untuk dapat melakukan penambahan data peminjaman dipilih *button* tambah data maka sistem akan memunculkan *form* penginputan data peminjaman. Setelah selesai mengisi semua data pada form dan dipilih *button* simpan maka dilakukan pengecekan terhadap data yang diinputkan, jika data yang diinputkan tidak valid maka sistem akan mengeluarkan pesan kesalahan data dan pengisian data harus diulang kembali sedangkan jika data yang diinputkan valid, data akan disimpan pada *database* dan admin dialihkan ke halaman data peminjaman.



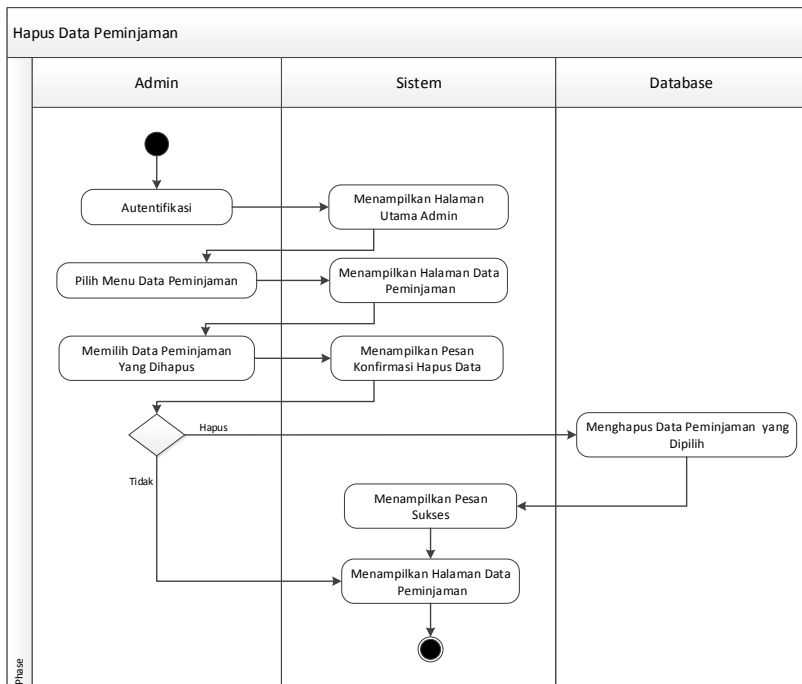
**Gambar 4.12 Activity Diagram Tambah Data Peminjaman**

Gambar 4.13 menggambarkan proses edit data pinjaman. Admin akan memilih terlebih dahulu data yang akan diedit, kemudian sistem akan menampilkan *form* yang berisikan data pinjaman yang dipilih. Admin kemudian melakukan perubahan data dan setelah itu admin memilih *button* simpan. Data yang telah diedit kemudian divalidasi oleh sistem. Apabila data tersebut valid maka semua perubahan akan disimpan kedalam *database* dan admin diarahkan kembali ke halaman data pinjaman namun jika tidak valid admin melakukan pengecekan kembali terhadap data yang diedit.



**Gambar 4. 13 Activity Diagram Edit Data Pinjaman**

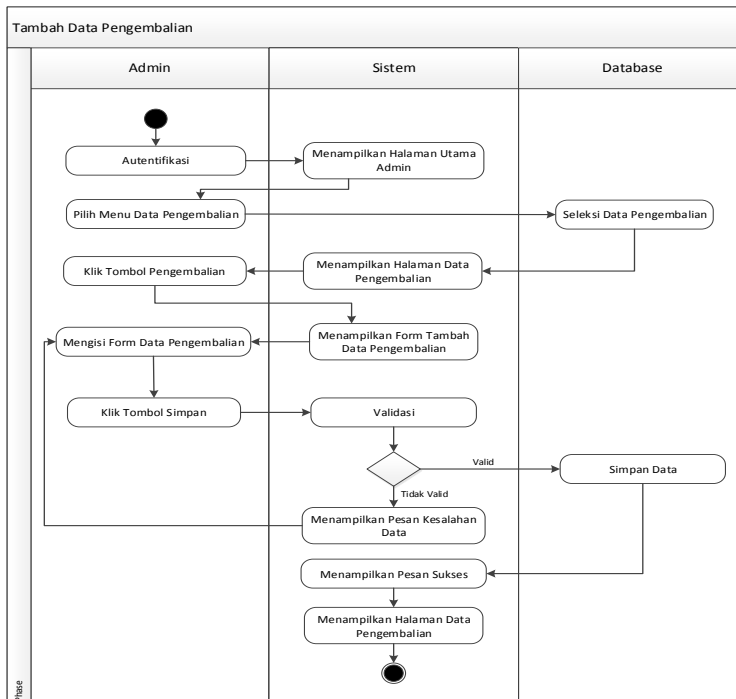
Gambar 4.14 menggambarkan proses hapus data peminjaman. Admin memilih salah satu data peminjaman yang akan dihapus dan kemudian memilih *button* hapus. Kemudian sistem akan menampilkan pesan konfirmasi. Jika admin mengkonfirmasi penghapusan data, maka data yang bersangkutan dihapus dari *database* dan admin dialihkan ke halaman data peminjaman.



**Gambar 4. 14 Activity Diagram Hapus Data Peminjaman**

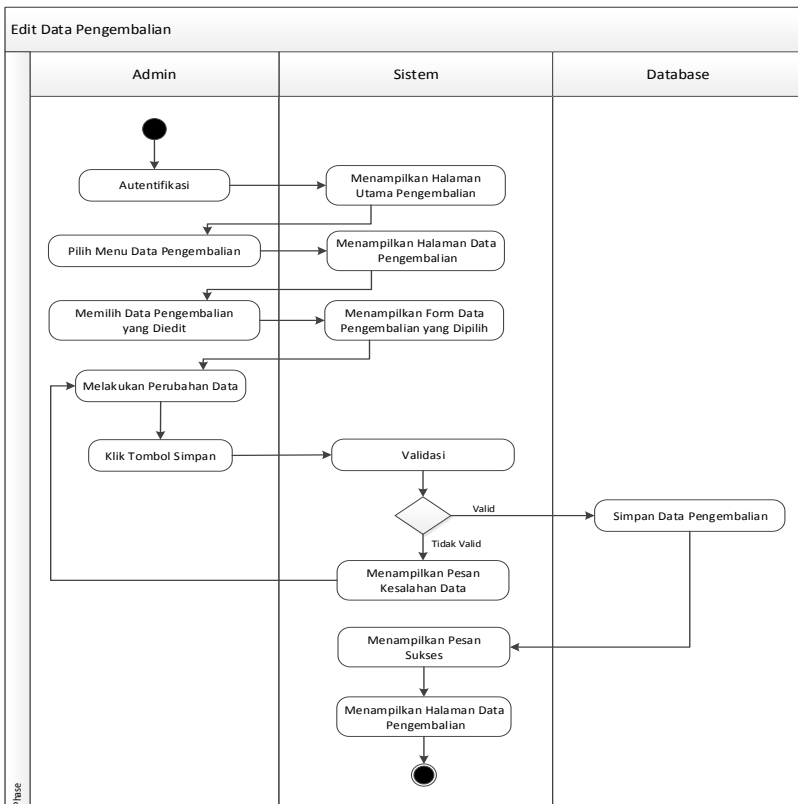
Gambar 4.15 menggambarkan *activity diagram* tambah data pengembalian. Tahap pertama yang dilakukan admin adalah melakukan *login* ke sistem, jika proses *login* berhasil maka akan ditampilkan form utama admin. Setelah *form* utama

admin ditampilkan maka dipilih menu data pengembalian, untuk dapat melakukan penambahan data pengembalian dipilih *button* tambah data maka sistem akan memunculkan *form* penginputan data pengembalian. Setelah selesai mengisi semua data pada form dan dipilih *button* simpan maka dilakukan pengecekan terhadap data yang diinputkan, jika data yang diinputkan tidak valid maka sistem akan mengeluarkan pesan kesalahan data dan pengisian data harus diulang kembali sedangkan jika data yang diinputkan valid, data akan disimpan pada *database* dan admin dialihkan ke halaman data pengembalian.



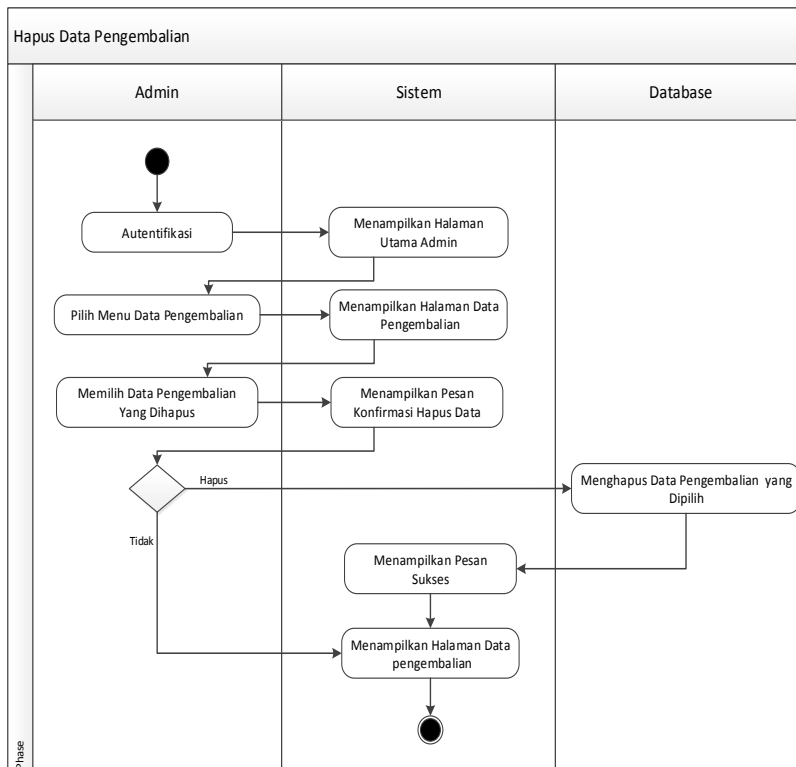
**Gambar 4. 15 Activity Diagram Tambah Data Pengembalian**

Gambar 4.16 menggambarkan proses edit data pengembalian. Admin akan memilih terlebih dahulu data yang akan diedit, kemudian sistem akan menampilkan *form* yang berisikan data pengembalian yang dipilih. Admin kemudian melakukan perubahan data dan setelah itu admin memilih *button* simpan. Data yang telah diedit kemudian divalidasi oleh sistem. Apabila data tersebut valid maka semua perubahan akan disimpan kedalam *database* dan admin diarahkan kembali ke halaman data pengembalian namun jika tidak valid admin melakukan pengecekan kembali terhadap data yang diedit.



**Gambar 4. 16 Activity Diagram Edit Data Pengembalian**

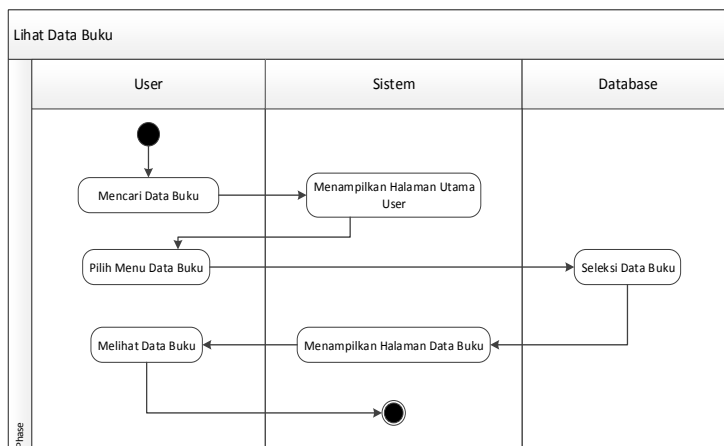
Gambar 4.17 menggambarkan proses hapus data pengembalian. Admin memilih salah satu data pengembalian yang akan dihapus dan kemudian memilih *button* hapus. Kemudian sistem akan menampilkan pesan konfirmasi. Jika admin mengkonfirmasi penghapusan data, maka data yang bersangkutan dihapus dari *database* dan admin dialihkan ke halaman data pengembalian.



**Gambar 4. 17 Activity Diagram Hapus Data Pengembalian**

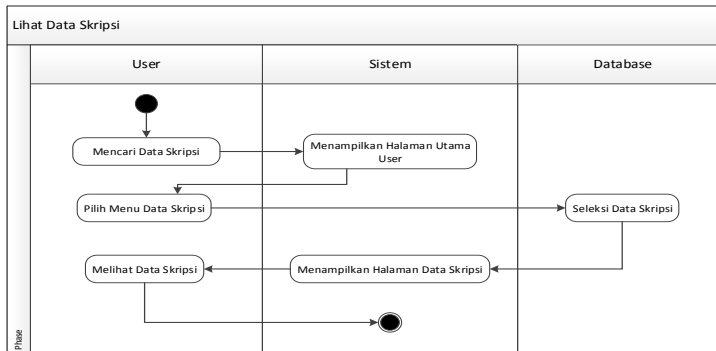
#### 4.2.2.2 Activity Diagram untuk User

Gambar 4.18 menggambarkan *activity diagram* untuk melihat data buku yang tersedia di perpustakaan. Tahap pertama yang akan dilakukan *user* adalah memilih menu data buku. Permintaan dari *user* kemudian diproses oleh sistem dengan mengakses *database* yang menyimpan data buku. Setelah itu sistem akan menampilkan data buku yang dibutuhkan oleh *user*. Informasi yang terdapat pada data buku adalah daftar buku – buku yang tersedia di perpustakaan dan jumlah yang masih tersedia di perpustakaan.



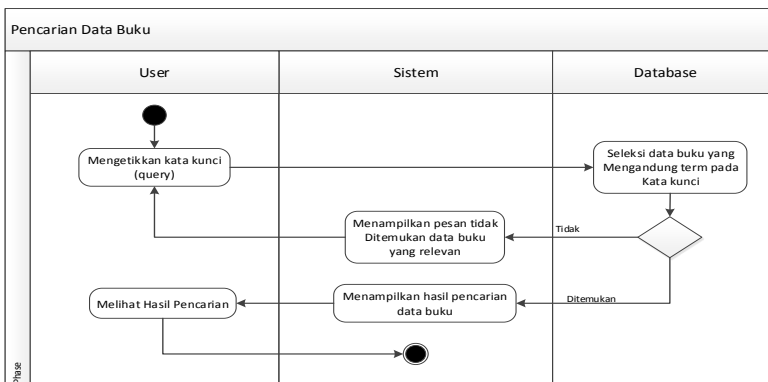
**Gambar 4. 18 Activity Diagram Lihat Data Buku**

Gambar 4.19 menggambarkan *activity diagram* untuk melihat data skripsi yang tersedia di perpustakaan. Tahap pertama yang akan dilakukan *user* adalah memilih menu data skripsi. Permintaan dari *user* kemudian diproses oleh sistem dengan mengakses *database* yang menyimpan data skripsi. Setelah itu sistem akan menampilkan data skripsi yang dibutuhkan oleh *user*. Informasi yang terdapat pada data skripsi adalah daftar skripsi – skripsi yang tersedia di perpustakaan.



**Gambar 4. 19 Activity Diagram Lihat Data Skripsi**

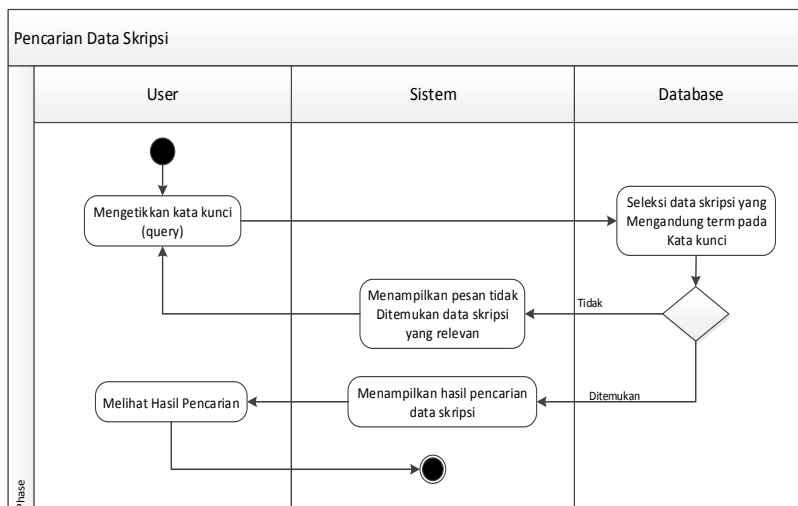
Gambar 4.20 menjelaskan alur aktivitas ketika *User* melakukan pencarian data buku. Pertama *User* mengetik atau menginput kata kunci. Setelah itu sistem akan mengambil data buku pada *database* yang mengandung setidaknya 1 *term* pada kata kunci. Apabila dokumen yang relevan dengan kata kunci tidak ditemukan pada *database* maka sistem akan menampilkan pesan tidak ditemukan data buku yang relevan, sedangkan jika relevan maka akan ditampilkan data buku yang dibutuhkan *user*.



**Gambar 4. 20 Activity Diagram Pencarian Data Buku**



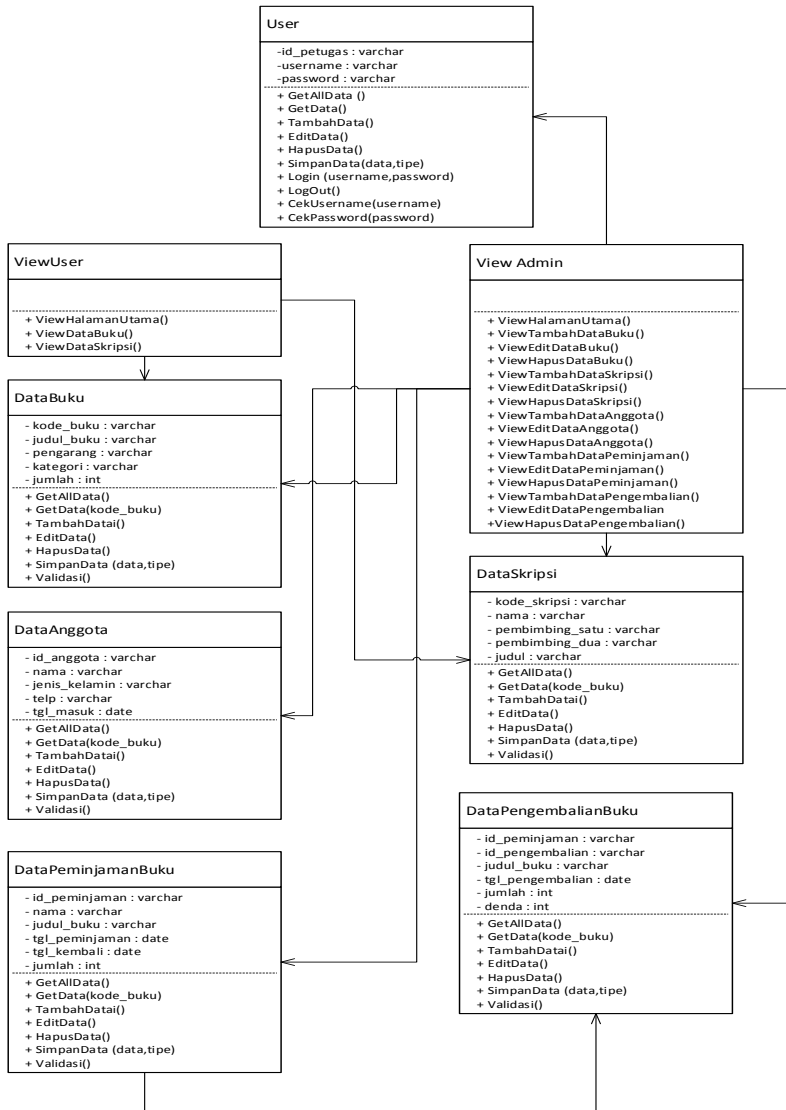
Gambar 4.21 menjelaskan alur aktivitas ketika *User* melakukan pencarian data skripsi. Pertama *User* mengetik atau menginput kata kunci. Setelah itu sistem akan mengambil data skripsi pada *database* yang mengandung setidaknya 1 *term* pada kata kunci. Apabila dokumen yang relevan dengan kata kunci tidak ditemukan pada *database* maka sistem akan menampilkan pesan tidak ditemukan data skripsi yang relevan, sedangkan jika relevan maka akan ditampilkan data skripsi yang dibutuhkan *user*.



**Gambar 4. 21 Activity Diagram Pencarian Data**

### 4.2.3 Class Diagram

Gambar 3.17 merupakan perancangan sistem penentuan pola penjualan obat dan rekomendasi pembelian obat terhadap *supplier* yang dimodelkan dengan *Class Diagram* yang dibangun berdasarkan pendefinisian dari *Use Case Diagram* dimana masing – masing fungsi yang terdapat pada setiap kelas dibuat untuk memenuhi kebutuhan proses yang terdapat pada *Use Case*.



**Gambar 4. 22 Class Diagram Sistem Informasi Management Perpustakaan**

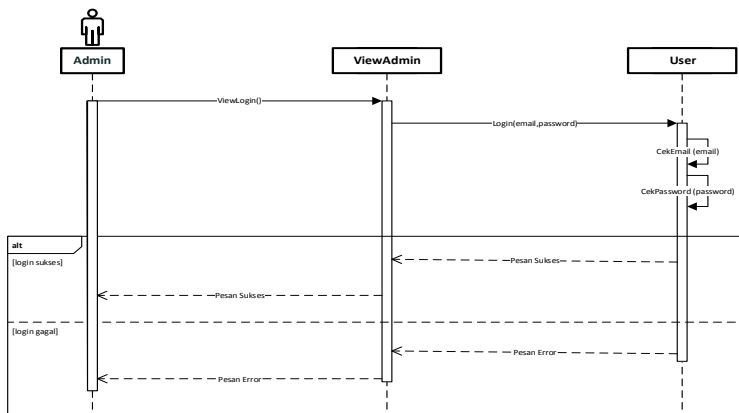
**Tabel 4. 4 Hubungan Class dan Use Case Diagram**

No	Kelas	Use Case	Fungsi
1	<i>User</i>	Autentifikasi	Menangani proses pengecekan hak akses <i>user</i> pada sistem
2	<i>View Admin</i>	Mengelola Data Buku	Menangani proses penampilan data pada saat admin melakukan pengolahan data
		Mengelola Data Skripsi	
		Mengelola Data Anggota	
		Mengelola Data Peminjaman	
		Mengelola Data Pengembalian	
3	<i>View User</i>	Melihat Data Buku	Menangani proses penampilan pada saat <i>user</i> melihat data pola penjualan obat dan rekomendasi pembelian obat terhadap <i>supplier</i>
		Melihat Data Skripsi	
4	Data Buku	Mengelola Data Buku	Menangani operasi pengolahan data buku dan akses data pada sistem
5	Data Anggota	Mengelola Data Anggota	Menangani operasi pengolahan data anggota dan akses data pada sistem
6	Data Skripsi	Mengelola Data Skripsi	Menangani operasi pengolahan data

			skripsi dan akses data pada sistem
7	Data Peminjaman	Mengelola Data Peminjaman	Menangani operasi pengolahan data skripsi dan akses data pada sistem
8	Data Pengembalian	Mengelola Data Pengembalian	Menangani operasi pengolahan data skripsi dan akses data pada sistem

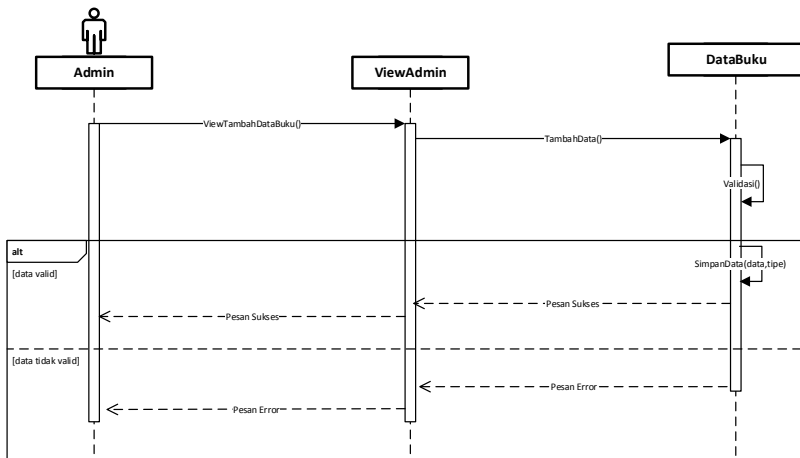
#### 4.2.4 *Sequence Diagram*

Bagian ini membahas mengenai perancangan sistem yang dimodelkan dengan menggunakan *sequence diagram* yang menggambarkan kelakuan atau perilaku *class* pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar *class*. Penjelasan dari masing – masing *sequence diagram* adalah sebagai berikut.



**Gambar 4. 23 *Sequence Diagram* Autentifikasi**

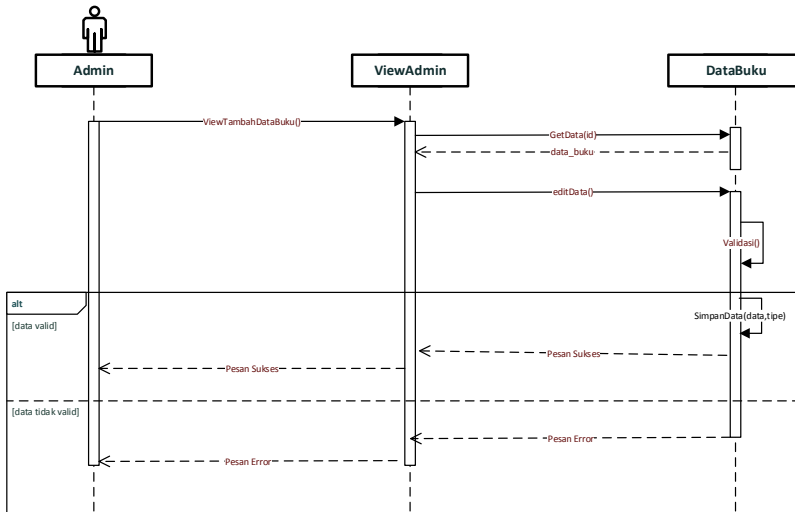
Gambar 4.23 menggambarkan perilaku dan waktu yang digunakan *class* pada proses autentifikasi. Pertama admin mengakses halaman *login* melalui pemanggilan fungsi “*ViewLogin()*” pada kelas “*ViewAdmin()*”. Kemudian setelah admin mengklik *button login* maka akan dijalankan fungsi “*Login*” pada kelas “*User*” dengan parameter *email* dan *password* yang diinputkan admin. Sistem akan mengecek apakah *email* terdaftar pada sistem dengan fungsi “*cekEmail(email)*”. Jika *email* terdaftar selanjutnya akan dicek kembali apakah *email* dan *password* cocok dengan fungsi “*cekPassword(password)*”. Jika semua kondisi terpenuhi maka admin berhasil *login* ke sistem namun jika kondisi *login* tidak terpenuhi maka akan dikirimkan pesan *error*.



**Gambar 4. 24 Sequence Diagram Tambah Data Buku**

Gambar 4.24 menjelaskan perilaku dan waktu yang digunakan *class* pada proses tambah data buku. Pertama admin mengakses halaman tambah data buku melalui pemanggilan fungsi “*viewTambahDataBuku()*” pada kelas “*ViewAdmin()*”. Kemudian setelah data pada *form* telah terisi dan admin mengklik *button* “*Simpan*” maka akan dijalankan fungsi “*tambahData()*” pada kelas “*DataBuku()*” untuk melakukan proses penambahan data. Sistem

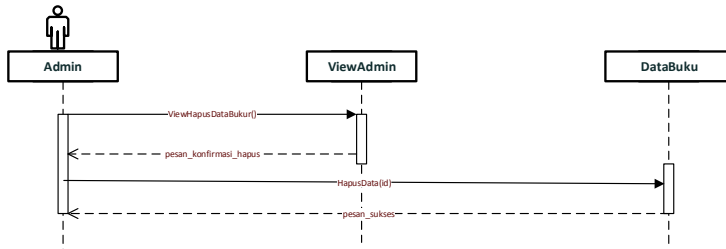
akan mengecek apakah seluruh data yang diinputkan *valid* dengan fungsi “*validasi()*”. Jika data *valid* maka data akan disimpan dengan fungsi “*simpanData()*” dan sistem mengirimkan pesan sukses. Jika data tidak *valid* maka akan dikirimkan pesan *error*.



**Gambar 4. 25 Sequence Diagram Edit Data Buku**

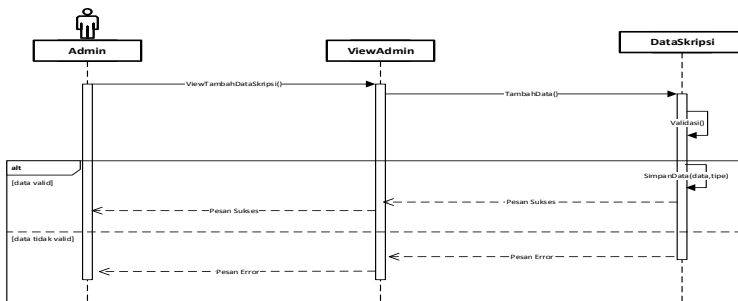
Gambar 4.25 menggambarkan perilaku dan waktu yang digunakan *class* pada proses edit data buku. Ketika Admin mengklik tombol “Edit” pada salah satu data buku maka akan dilakukan pemanggilan fungsi “*viewEditBuku()*” pada kelas “*ViewAdmin()*” dengan parameter “id” data buku yang dipilih. Kemudian akan dijalankan fungsi “*getData(id)*” pada kelas “*DataBuku()*” untuk mengambil data buku berdasarkan parameter id\_buku untuk ditampilkan pada *form* edit data. Setelah data pada *form* telah diedit dan admin mengklik *button* “Simpan” maka akan dijalankan fungsi “*editData()*” pada kelas “*DataBuku()*” untuk melakukan proses perubahan data. Sistem akan mengecek apakah seluruh data yang diinputkan *valid* dengan fungsi “*validasi()*”. Jika data *valid* maka perubahan

data akan disimpan dengan fungsi “*simpanData()*” dan sistem mengirimkan pesan sukses. Jika data tidak *valid* maka akan dikirimkan pesan *error*.



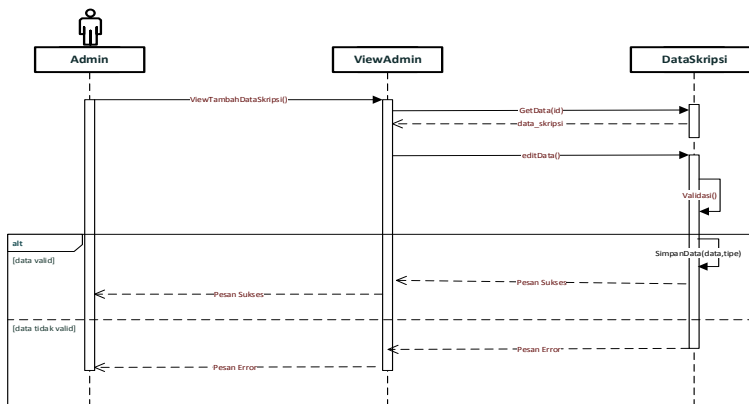
**Gambar 4. 26 Sequence Diagram Hapus Data Buku**

Gambar 4. 26 menggambarkan perilaku dan waktu yang digunakan *class* pada proses hapus data buku. Ketika Admin mengklik *button* “Hapus” pada salah satu data buku maka akan dilakukan pemanggilan fungsi “*ViewHapusDataBuku()*” pada kelas “*ViewAdmin()*” dengan parameter “*id*” data buku yang dipilih. Jika Admin mengkonfirmasi penghapusan data maka akan dijalankan fungsi “*hapusData(id)*” pada kelas “*DataBuku()*” untuk melakukan penghapusan data pada *database*.



**Gambar 4. 27 Sequence Diagram Tambah Data Skripsi**

Gambar 4.27 menjelaskan perilaku dan waktu yang digunakan *class* pada proses tambah data skripsi. Pertama admin mengakses halaman tambah data skripsi melalui pemanggilan fungsi “*viewTambahDataSkripsi()*” pada kelas “*ViewAdmin()*”. Kemudian setelah data pada *form* telah terisi dan admin mengklik *button* “Simpan” maka akan dijalankan fungsi “*tambahData()*” pada kelas “*DataSkripsi()*” untuk melakukan proses penambahan data. Sistem akan mengecek apakah seluruh data yang diinputkan *valid* dengan fungsi “*validasi()*”. Jika data *valid* maka data akan disimpan dengan fungsi “*simpanData()*” dan sistem mengirimkan pesan sukses. Jika data tidak *valid* maka akan dikirimkan pesan *error*.

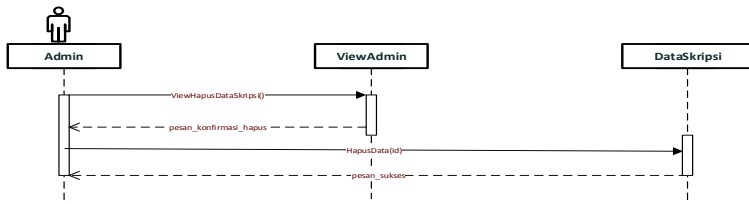


**Gambar 4. 28 Sequence Diagram Edit Data Skripsi**

Gambar 4.28 menggambarkan perilaku dan waktu yang digunakan *class* pada proses edit data skripsi. Ketika Admin mengklik tombol “Edit” pada salah satu data skripsi maka akan dilakukan pemanggilan fungsi “*viewEditDataSkripsi()*” pada kelas “*ViewAdmin()*” dengan parameter “id” data skripsi yang dipilih. Kemudian akan dijalankan fungsi “*getData(id)*” pada kelas “*DataSkripsi()*” untuk mengambil data skripsi berdasarkan parameter *id\_skripsi* untuk ditampilkan pada *form* edit data. Setelah data pada *form* telah diedit dan admin mengklik *button* “Simpan” maka akan dijalankan fungsi “*editData()*” pada kelas

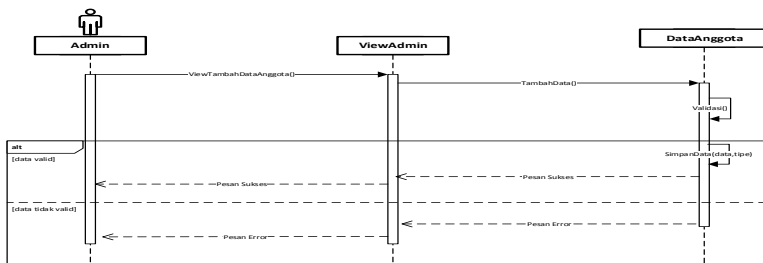


“*DataSkripsi()*” untuk melakukan proses perubahan data. Sistem akan mengecek apakah seluruh data yang diinputkan *valid* dengan fungsi “*validasi()*”. Jika data *valid* maka perubahan data akan disimpan dengan fungsi “*simpanData()*” dan sistem mengirimkan pesan sukses. Jika data tidak *valid* maka akan dikirimkan pesan *error*.



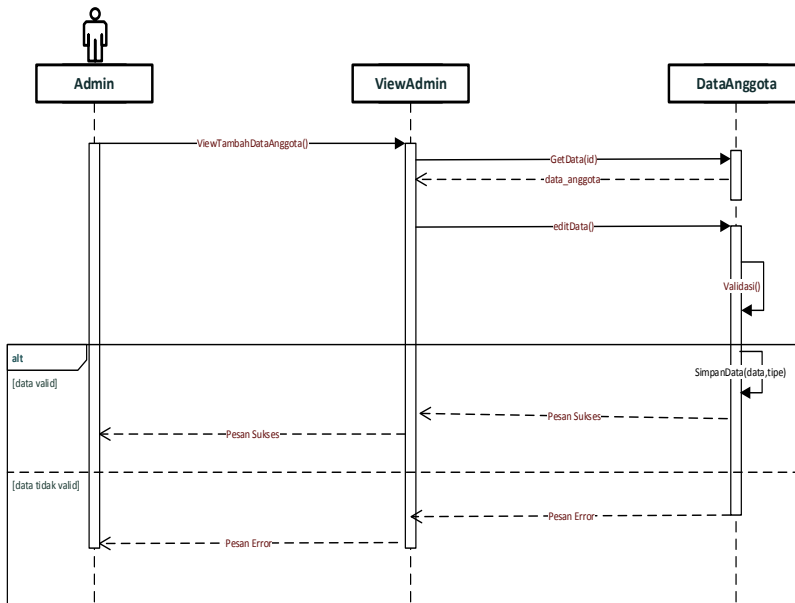
**Gambar 4. 29 Sequence Diagram Hapus Data Skripsi**

Gambar 4. 29 menggambarkan perilaku dan waktu yang digunakan *class* pada proses hapus data skripsi. Ketika Admin mengklik *button* “Hapus” pada salah satu data skripsi maka akan dilakukan pemanggilan fungsi “*ViewHapusSkripsi()*” pada kelas “*ViewAdmin()*” dengan parameter “*id*” data skripsi yang dipilih. Jika Admin mengkonfirmasi penghapusan data maka akan dijalankan fungsi “*hapusData(id)*” pada kelas “*DataSkripsi()*” untuk melakukan penghapusan data pada *database*.



**Gambar 4. 30 Sequence Diagram Tambah Data Anggota**

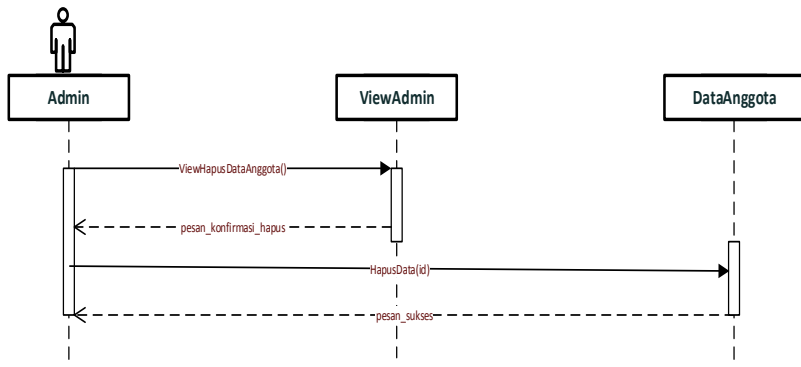
Gambar 4.30 menjelaskan perilaku dan waktu yang digunakan *class* pada proses tambah data anggota. Pertama admin mengakses halaman tambah data anggota melalui pemanggilan fungsi “*viewTambahDataAnggota()*” pada kelas “*ViewAdmin()*”. Kemudian setelah data pada *form* telah terisi dan admin mengklik *button* “Simpan” maka akan dijalankan fungsi “*tambahData()*” pada kelas “*DataAnggota()*” untuk melakukan proses penambahan data. Sistem akan mengecek apakah seluruh data yang diinputkan *valid* dengan fungsi “*validasi()*”. Jika data *valid* maka data akan disimpan dengan fungsi “*simpanData()*” dan sistem mengirimkan pesan sukses. Jika data tidak *valid* maka akan dikirimkan pesan *error*.



**Gambar 4. 31 Sequence Diagram Edit Data Anggota**

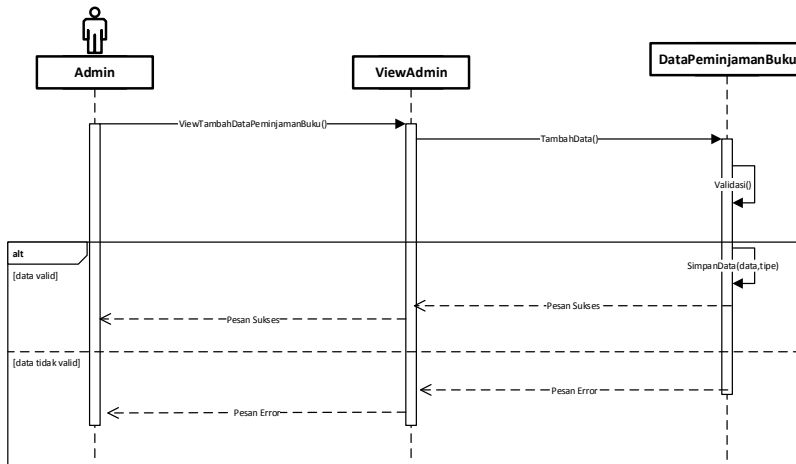
Gambar 4.31 menggambarkan perilaku dan waktu yang digunakan *class* pada proses edit data anggota. Ketika Admin mengklik tombol “Edit” pada salah satu data anggota maka akan dilakukan pemanggilan fungsi “*viewEditDataAnggota()*” pada kelas

“*ViewAdmin()*” dengan parameter “*id*” data anggota yang dipilih. Kemudian akan dijalankan fungsi “*getData(id)*” pada kelas “*DataAnggota()*” untuk mengambil data anggota berdasarkan parameter *id\_anggota* untuk ditampilkan pada *form* edit data. Setelah data pada *form* telah diedit dan admin mengklik *button* “Simpan” maka akan dijalankan fungsi “*editData()*” pada kelas “*DataAnggota()*” untuk melakukan proses perubahan data. Sistem akan mengecek apakah seluruh data yang diinputkan *valid* dengan fungsi “*validasi()*”. Jika data *valid* maka perubahan data akan disimpan dengan fungsi “*simpanData()*” dan sistem mengirimkan pesan sukses. Jika data tidak *valid* maka akan dikirimkan pesan *error*.



**Gambar 4. 32 Sequence Diagram Hapus Data Anggota**

Gambar 4. 32 menggambarkan perilaku dan waktu yang digunakan *class* pada proses hapus data anggota. Ketika Admin mengklik *button* “Hapus” pada salah satu data anggota maka akan dilakukan pemanggilan fungsi “*ViewHapusAnggota()*” pada kelas “*ViewAdmin()*” dengan parameter “*id*” data anggota yang dipilih. Jika Admin mengkonfirmasi penghapusan data maka akan dijalankan fungsi “*hapusData(id)*” pada kelas “*DataAnggota()*” untuk melakukan penghapusan data pada *database*.

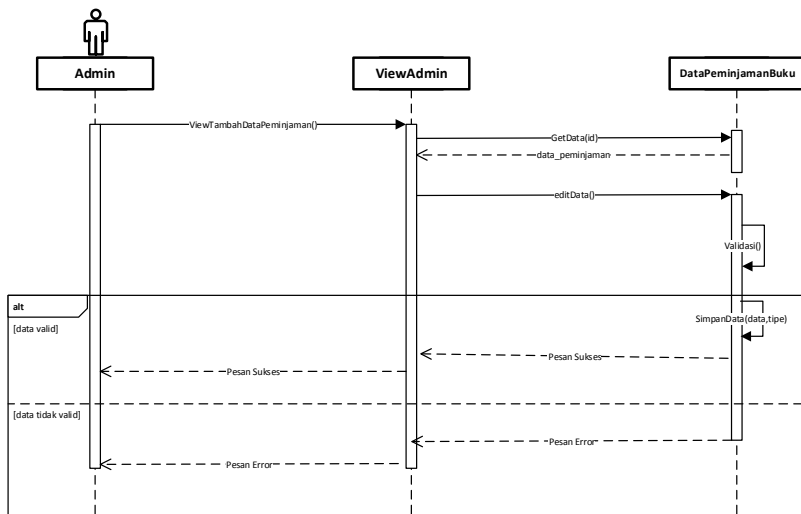


**Gambar 4. 33 Sequence Diagram Tambah Data Peminjaman Buku**

Gambar 4.33 menjelaskan perilaku dan waktu yang digunakan *class* pada proses tambah data peminjaman. Pertama admin mengakses halaman tambah data peminjaman melalui pemanggilan fungsi “*viewTambahDataPeminjaman()*” pada kelas “*ViewAdmin()*”. Kemudian setelah data pada *form* telah terisi dan admin mengklik *button* “Simpan” maka akan dijalankan fungsi “*tambahData()*” pada kelas “*DataPeminjamanBuku()*” untuk melakukan proses penambahan data. Sistem akan mengecek apakah seluruh data yang diinputkan *valid* dengan fungsi “*validasi()*”. Jika data *valid* maka data akan disimpan dengan fungsi “*simpanData()*” dan sistem mengirimkan pesan sukses. Jika data tidak *valid* maka akan dikirimkan pesan *error*.

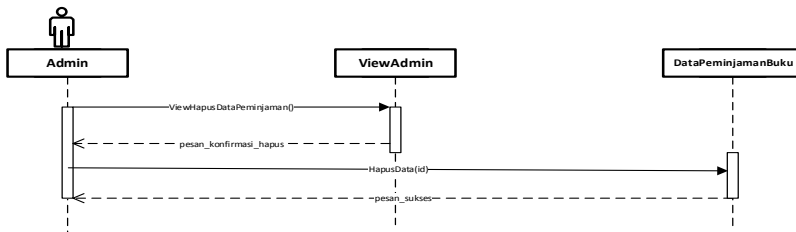
Gambar 4.34 menggambarkan perilaku dan waktu yang digunakan *class* pada proses edit data peminjaman buku. Ketika Admin mengklik tombol “Edit” pada salah satu data peminjaman maka akan dilakukan pemanggilan fungsi “*viewEditPeminjaman()*” pada kelas “*ViewAdmin()*” dengan parameter “*id*” data peminjaman buku yang dipilih. Kemudian akan dijalankan fungsi “*getData(id)*” pada kelas “*DataPeminjamanBuku()*” untuk mengambil data peminjaman berdasarkan parameter *id\_peminjaman* untuk

ditampilkan pada *form* edit data. Setelah data pada *form* telah diedit dan admin mengklik *button* “Simpan” maka akan dijalankan fungsi “*editData()*” pada kelas “*DataPeminjamanBuku()*” untuk melakukan proses perubahan data. Sistem akan mengecek apakah seluruh data yang diinputkan *valid* dengan fungsi “*validasi()*”. Jika data *valid* maka perubahan data akan disimpan dengan fungsi “*simpanData()*” dan sistem mengirimkan pesan sukses. Jika data tidak *valid* maka akan dikirimkan pesan *error*.



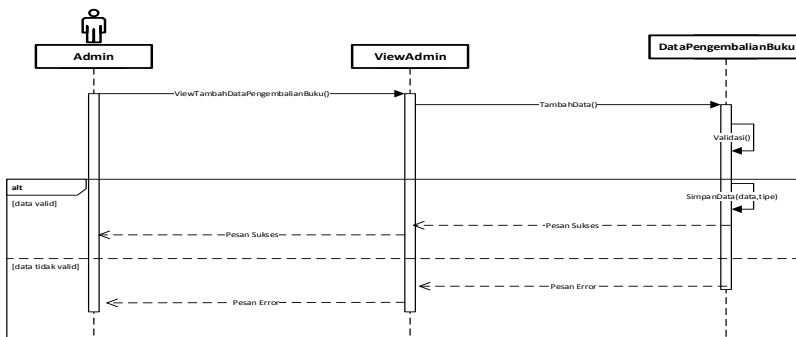
**Gambar 4. 34 Sequence Diagram Edit Data Peminjaman Buku**

Gambar 4. 35 menggambarkan perilaku dan waktu yang digunakan *class* pada proses hapus data peminjaman buku. Ketika Admin mengklik *button* “Hapus” pada salah satu data peminjaman buku maka akan dilakukan pemanggilan fungsi “*ViewHapusPeminjaman()*” pada kelas “*ViewAdmin()*” dengan parameter “*id*” data peminjaman yang dipilih. Jika Admin mengkonfirmasi penghapusan data maka akan dijalankan fungsi “*hapusData(id)*” pada kelas “*DataPeminjamanBuku()*” untuk melakukan penghapusan data pada *database*.



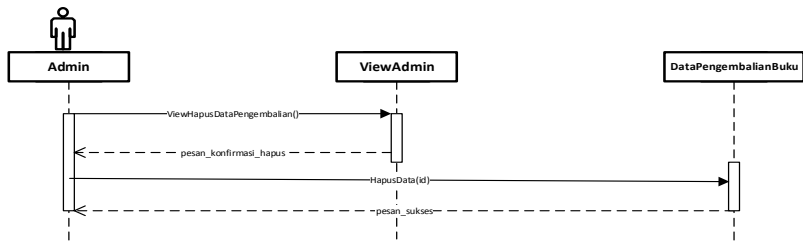
**Gambar 4. 35 Sequence Diagram Hapus Data Peminjaman Buku**

Gambar 4.36 menjelaskan perilaku dan waktu yang digunakan *class* pada proses tambah data pengembalian. Pertama admin mengakses halaman tambah data pengembalian melalui pemanggilan fungsi “*viewTambahDataPengembalian()*” pada kelas “*ViewAdmin()*”. Kemudian setelah data pada *form* telah terisi dan admin mengklik *button* “Simpan” maka akan dijalankan fungsi “*tambahData()*” pada kelas “*DataPengembalianBuku()*” untuk melakukan proses penambahan data. Sistem akan mengecek apakah seluruh data yang diinputkan *valid* dengan fungsi “*validasi()*”. Jika data *valid* maka data akan disimpan dengan fungsi “*simpanData()*” dan sistem mengirimkan pesan sukses. Jika data tidak *valid* maka akan dikirimkan pesan *error*.



**Gambar 4. 36 Sequence Diagram Tambah Data Pengembalian**

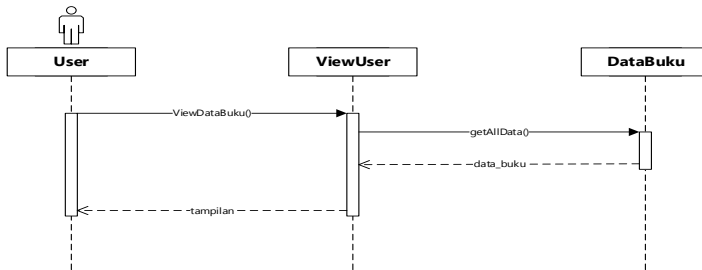
Gambar 4.37 menggambarkan perilaku dan waktu yang digunakan *class* pada proses edit data pengembalian buku. Ketika Admin mengklik tombol “Edit” pada salah satu data pengembalian buku maka akan dilakukan pemanggilan fungsi “*viewEditDataPengembalianBuku()*” pada kelas “*ViewAdmin()*” dengan parameter “id” data pengembalian buku yang dipilih. Kemudian akan dijalankan fungsi “*getData(id)*” pada kelas “*DataPengembalianBuku()*” untuk mengambil data pengembalian berdasarkan parameter *id\_pengembalian* untuk ditampilkan pada *form* edit data. Setelah data pada *form* telah diedit dan admin mengklik *button* “Simpan” maka akan dijalankan fungsi “*editData()*” pada kelas “*DataPengembalianBuku()*” untuk melakukan proses perubahan data. Sistem akan mengecek apakah seluruh data yang diinputkan *valid* dengan fungsi “*validasi()*”. Jika data *valid* maka perubahan data akan disimpan dengan fungsi “*simpanData()*” dan sistem mengirimkan pesan sukses. Jika data tidak *valid* maka akan dikirimkan pesan *error*.



**Gambar 4. 37 Sequence Diagram Hapus Data Pengembalian**

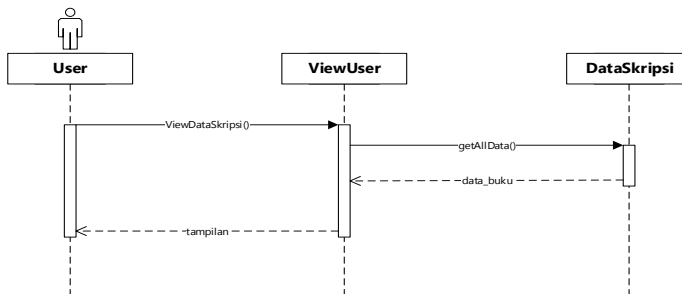
Gambar 4. 35 menggambarkan perilaku dan waktu yang digunakan *class* pada proses hapus data pengembalian buku. Ketika Admin mengklik *button* “Hapus” pada salah satu data pengembalian buku maka akan dilakukan pemanggilan fungsi “*ViewHapusPengembalian()*” pada kelas “*ViewAdmin()*” dengan parameter “id” data penegmbalian yang dipilih. Jika Admin mengkonfirmasi penghapusan data maka akan dijalankan fungsi “*hapusData(id)*” pada kelas

“*DataPengembalianBuku()*” untuk melakukan penghapusan data pada *database*.



**Gambar 4. 38 Sequence Diagram Lihat Data Buku**

Gambar 4.38 menggambarkan perilaku user dalam melihat data buku. Ketika user memilih menu data buku maka sistem akan memanggil kelas data buku melalui kelas “*ViewUser()*”. Setelah itu akan dijalankan fungsi “*ViewDataBuku()*” dan data buku akan ditampilkan pada *user*.



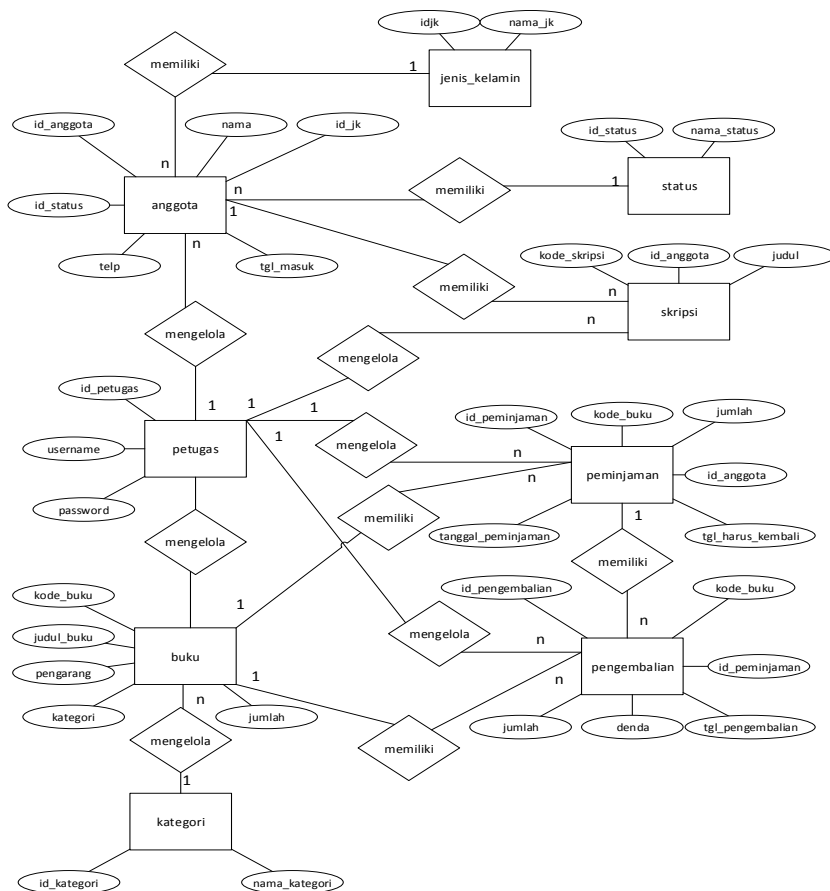
**Gambar 4. 39 Sequence Diagram Lihat Data Skripsi**

Gambar 4.38 menggambarkan perilaku user dalam melihat data skripsi. Ketika user memilih menu data skripsi maka sistem akan memanggil kelas data skripsi melalui kelas “*ViewUser()*”. Setelah itu akan dijalankan fungsi “*ViewDataSkripsi()*” dan data skripsi akan ditampilkan pada *user*.



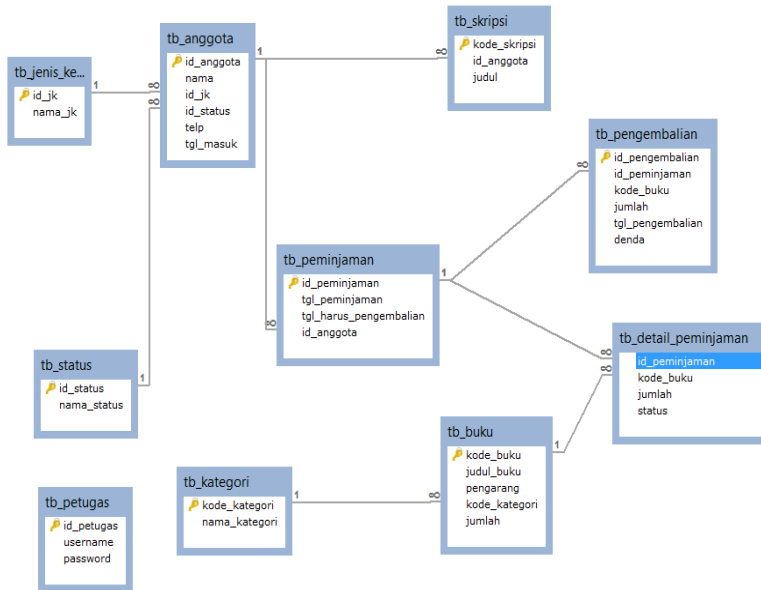
### 4.3 Entity Relationship Diagram (ERD)

Dalam perancangan database dari sistem informasi perpustakaan, perencanaannya akan direpresentasikan atau ditampilkan menggunakan Entity Relationship Diagram (ERD). Pada rancangan database sistem informasi perpustakaan terdapat 9 entitas yaitu entitas buku, entitas anggota, entitas petugas, entitas skripsi, entitas peminjaman, entitas pengembalian, entitas kategori, entitas status dan entitis jenis kelamin.



**Gambar 4. 40 ERD Sistem Perpustakaan**

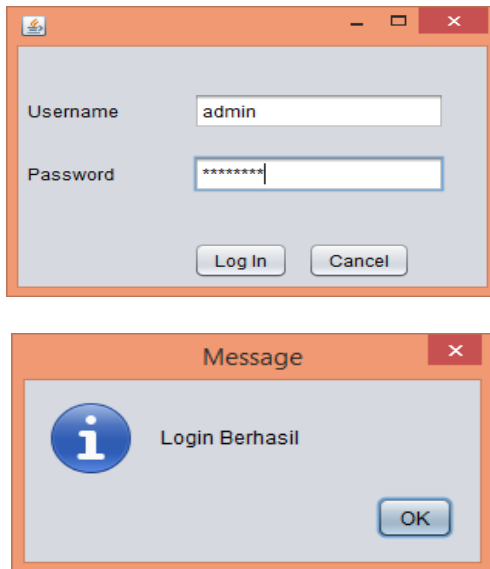
Pengimplementasian ERD Sistem perpustakaan kedalam skema database adalah sebagai berikut.



**Gambar 4. 41 Skema Database Sistem Perpustakaan**

#### 4.4 Implementasi Sistem

Pada sistem informasi manajemen perpustakaan ini hak akses *user* dibedakan menjadi dua yaitu sebagai anggota perpustakaan dan sebagai petugas. Anggota perpustakaan hanya dapat melihat data buku dan data skripsi, sedangkan petugas perpustakaan bertugas untuk mengelola data buku, data skripsi, data anggota, data peminjaman dan data pengembalian. Sebelum melakukan pengelolaan data, terlebih dahulu petugas perpustakaan harus melakukan *login* ke sistem. Jika *username* dan *password* yang diinputkan oleh petugas perpustakaan sesuai dengan data yang tersimpan pada database maka petugas perpustakaan akan diarahkan ke halaman utama admin.



The image shows two windows from a library management system. The top window is a login form with fields for 'Username' (containing 'admin') and 'Password' (containing '\*\*\*\*\*'). It has 'Log In' and 'Cancel' buttons. The bottom window is a 'Message' dialog with an information icon and the text 'Login Berhasil' (Login Successful), with an 'OK' button.

**Gambar 4. 42 Form Login Sistem Perpustakaan**

Pada form *login* terlebih dahulu petugas perpustakaan memasukkan *username* dan *password* terlebih dahulu. Jika *username* dan *password* yang diinputkan benar maka sistem akan memberikan notif login berhasil.

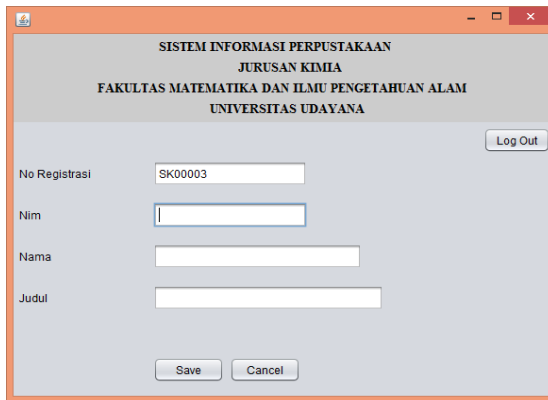


The image shows the main menu of the 'SISTEM INFORMASI PERPUSTAKAAN JURUSAN KIMIA, FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM, UNIVERSITAS UDAYANA'. It features a tabbed interface with 'Data Skripsi' selected. Below the tabs is a search bar with a 'Cari' button. To the right are 'Insert Data' and 'Log Out' buttons. A table displays a list of theses with columns for 'Kode Skripsi', 'Nim', 'Nama', and 'Judul'. The first two rows of data are visible.

Kode Skripsi	Nim	Nama	Judul	Edit	Hapus
SK. 15.629	1108195019	A.A.I. Rahma Prabawati	Uji Efektivitas dan Identifikasi Senyawa Anti Ekstrak Daun Sirsak sebagai Pelembab Nafas L.	Edit	Hapus
SK. 15.630	1108195022	Nilu Made Dwi Desianti	Karakteristik Mutu Getas dari Kulit Ayam Broiler melalui Proses Perendaman Asam dan K.	Edit	Hapus

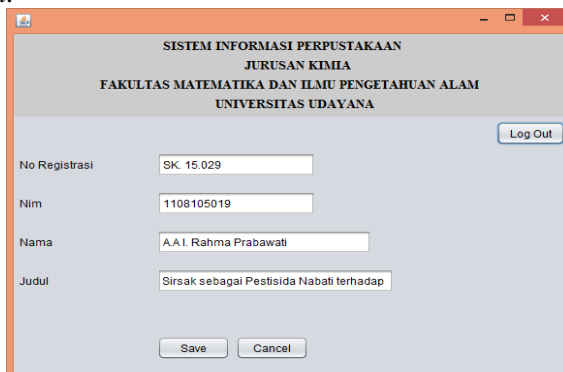
**Gambar 4. 43 Form Menu Skripsi Sistem Perpustakaan**

Gambar 4.43 merupakan form untuk menampilkan data skripsi yang tersedia di perpustakaan. Jika *user* ingin menambahkan data skripsi maka dapat memilih *button insert* data pada bagian atas. Sedangkan jika ingin melakukan edit atau hapus dapat memilih *button edit* dan hapus yang terdapat pada tabel.



**Gambar 4. 44 Form Tambah Data Skripsi**

Gambar 4.44 merupakan form untuk melakukan penambahan data skripsi. Jika semua data telah terisi maka *user* dapat memilih *button save* untuk menyimpan data yang sudah diisi pada *form*.



**Gambar 4. 45 Form Edit Data Skripsi**

Gambar 4.5 merupakan *form* untuk mengedit data skripsi. Ketika *user* memilih *button edit* pada tabel maka *form* ini akan langsung muncul dan data pada tabel akan langsung tampil pada *form*. Pada data skripsi yang terdapat pada gambar 4.43, jika *user* memilih hapus maka akan muncul notif terlebih dahulu apakah yakin ingin menghapus data tersebut, jika *user* memilih *yes* maka data skripsi akan terhapus.

ID	Nama	Jenis Kelamin	Status	Telp	Tanggal Masuk	Gaji	Hakus
1100100115	A.A. Rahma Prah	Wanita	Mahasiswa	08572004545	2016-07-15	0.00	Hapus
1100100222	Ni Made Dwi Dini	Wanita	Mahasiswa	085782307671	2016-07-15	0.00	Hapus

**Gambar 4. 46 Form Tampil Data Anggota**

Gambar 4.46 merupakan *form* untuk menampilkan data yang anggota perpustakaan.

Kode Buku	Judul Buku	Pengarang	Kategori	Jumlah	Rata	Hakus
NZ 1.001.1	Buku: cara jendral B.	Harold MacGill	Karya nyata	4	0.00	Hapus

**Gambar 4. 47 Form Tampil Data Buku**

Id Pe	Nip/Nid	Nama	Status	Kode	Judul	Tanggal	Jumlah
TR00...	11081...	A.A.I...		K1.1.9...	Basic...	2016...	1

**Gambar 4. 48 Form Peminjaman Buku**

Gambar 4.48 merupakan form untuk menginputkan data peminjaman buku. Sekali peminjaman anggota perpustakaan hanya dapat meminjam maksimal dua buku. Namun jika peminjaman lebih dari bata yang sudah ditentukan maka sistem akan memberikan notif mengenai buku yang dapat dipinjam

**Gambar 4. 49 Form Pengembalian Buku**

Gambar 4.49 merupakan form pengembalian buku. Ketika user menginputkan tanggal sekarang maka akan secara otomatis sistem menghitung denda jika terjadi keterlamabatan pengembalian dari tanggal seharusnya.

Kode Skripsi	Nim	Nama	Judul Skripsi
SK. 15.029	1108105019	A.A.I Ratnisa Pratawati	Uji Efektifitas dan Identifikasi S...
SK. 15.030	1108105022	Ni Made Dwi Desiani	Karakteristik Mutu Getin dari ...

**Gambar 4. 50 Form Menampilkan Data Skripsi**

Gambar 4.50 merupakan form untuk menampilkan data skripsi. Ketika anggota perpustakaan ingin mengetahui data skripsi yang tersedia di perpustakaan dapat memilih menu data skripsi pada menu utama *user*.

Kode Buku	Judul	Pengarang	Kategori	Jumlah
KI.1.001.1	Basic one and two dim...	Horst Fiebolin	Kimia Analitik	3

**Gambar 4. 51 Form Menampilkan Data Buku**

Gambar 4.51 merupakan form untuk menampilkan data buku yang tersedia di perpustakaan. Sehingga dengan adanya sistem perpustakaan yang dapat menampilkan data buku, dapat memudahkan anggota perpustakaan dalam mengecek ada atau tidaknya buku tersebut di perpustakaan.

#### 4.5 Pengujian *Black Box*

Pengujian *black box* merupakan pengujian yang dilakukan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Apabila sudah sesuai harapan maka kesimpulan yang diperoleh adalah skenario pengujian yang bernilai valid.

**Tabel 4. 5 Pengujian dengan *Black Box Testing***

No	Skenario Pengujian	Keluaran Pengujian	Hasil Pengujian	Kesimpulan
1	Menginputkan username dan password yang benar dan klik tombol login pada halaman login petugas	Sistem mendirect ke halaman utama petugas	Sesuai harapan	Valid
2	Menginputkan username dan password yang salah atau yang tidak terdapat pada database lalu klik tombol login	Sistem menolak akses ke halaman user dan menampilkan pesan kesalahan username atau password	Sesuai harapan	Valid
3	Menginputkan data buku	Data baru yang	Sesuai harapan	Valid



		diinputkan ditampilkan pada tabel data buku		
4	Mengupdate data buku	Sistem mengubah data yang dipilih user	Sesuai harapan	Valid
5	Mendelete data buku	Sistem menghapus data yang dipilih user	Sesuai harapan	Valid
6	Mengklik tombol cari	Sistem menampilkan data pencarian buku	Sesuai harapan	Valid
7	Menginput data skripsi	Data baru yang diinputkan ditampilkan pada tabel data skripsi	Sesuai harapan	Valid
8	Mengupdate data skripsi	Sistem mengubah data skripsi yang dipilih user	Sesuai harapan	Valid
9	Mendelete data skripsi	Sistem menghapus data skripsi yang dipilih user	Sesuai harapan	Valid
10	Mengklik tombol cari	Sistem menampilkan data	Sesuai harapan	Valid

		pencarian skripsi		
11	Menginput data transaksi	Data baru yang diinputkan ditampilkan pada tabel data transaksi	Sesuai harapan	Valid
12	Mengupdate data transaksi	Sistem mengubah data transaksi yang dipilih user	Sesuai harapan	Valid
13	Mendelete data transaksi	Sistem menghapus data transaksi yang dipilih user	Sesuai harapan	Valid
14	Mengklik tombol cari	Sistem menampilkan data pencarian transaksi	Sesuai harapan	Valid

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Adapun kesimpulan yang dapat diperoleh dari pelaksanaan pratek kerja lapangan ini adalah sebagai berikut :

1. Penulis memperoleh gambaran nyata mengenai bagaimana situasi dalam dunia kerja sehingga dapat mempersiapkan diri dalam persaingan di dunia kerja nantinya. Selain itu adanya pengarahan terhadap penerapan ilmu pengetahuan tentang pemanfaatan teknologi informasi pada khususnya dan ilmu yang didapatkan dari perkuliah pada umumnya dalam dunia kerja, sehingga menjadi pengalaman yang tentunya sangat bermanfaat.
2. Dari segi sistem yang dibangun dalam pelaksanaan kegiatan ini, uji coba serta evaluasi kualitas informasi dari perancangan dan implementasi sistem yang dilakukan secara keseluruhan telah mampu menjawab kebutuhan dasar dari pengguna sistem baik keseluruhan unit pada umumnya serta khususnya pada perpustakaan.
3. Berjalannya seluruh kebutuhan dari pengguna secara normal sesuai dengan fungsinya pada sistem merupakan tolak ukur dari keberhasilan pengembangan sistem informasi manajemen perpustakaan ini.

#### **5.2 Saran**

Adapun saran yang dapat diberikan oleh penulis adalah sebagai berikut .

1. Pada tahap pengoperasian sistem diperlukan adanya pendampingan dari seseorang yang memiliki keahlian khusus dalam hal pendampingan terhadap *user* terhadap segala kemungkinan kesalahan yang terjadi, mampu mengerti dan menghadapi user dengan baik, dan mampu menerjemahkan permasalahan serta kebutuhan tambahan dari *user* sebagai pedoman dalam pengembangan sistem selanjutnya.

## DAFTAR PUSTAKA

- Fatansyah. 1999. Basis Data. Bandung: Informatika.
- Jogiyanto, HM. 2001. Analisis dan Desain Sistem Informasi. Yogyakarta: Andi.
- Ladjamudin, A. 2005. Analisis dan Desain Sistem Informasi. Yogyakarta: Graha Ilmu.
- Rosenblatt. 2013. *System Analysis and Design – Tenth Edition Shelly Cashman Series*. USA: Course Technology
- Sommerville, I. 2011. *Software Engineering. Ninth Edition*. Harlow: Pearson Education Limited. Pearson Education, Inc.
- Sutanta. 2003. Sistem Basis Data. Yogyakarta: Graha Ilmu.

## Lampiran 1. Aktivitas Harian

### AKTIVITAS HARIAN PKL

Nama : NIKKI DINIA PRADITA PERMITA  
 NIM : 1806020011  
 Lokasi PKL : JURUSAN KIMIA FAKULTAS ILMU  
 Waktu Pelaksanaan : 1 MARET 2019 - 20 MARI 2019



No.	Nama Penanggung jawab/Jabatan	Pelaksanaan PKL			Ket	Paraf
		Tanggal	Lokasi	Aktivitas		
1	I Pengah Wirajana (Ketua Jurusan Kimia)	Senin, 12 Maret 2019	Jurusan Kimia	Menganalisis menu navigasi yang diperlukan untuk web chemistry.uns.ac.id		
2	I Pengah Wirajana (Ketua Jurusan Kimia)	Rabu, 13 Maret 2019	Jurusan Kimia	Mengumpulkan-tentukan hasil analisis menu navigasi web chemistry.uns.ac.id		
3	I Pengah Wirajana (Ketua Jurusan Kimia)	Kamis 14 Maret 2019	Jurusan Kimia	Membuat form pengantar PKL Form Ipti lembar berpetak, form pengajuan ujian PKL		
4	I Pengah Wirajana (Ketua Jurusan Kimia)	Jumat 15 Maret 2019	Jurusan Kimia	Menginput form-form pada point 3 ke web, menginput data sejarah jurusan		
5	-	Sabtu, 16 Maret 2019	-	Libur	Libur Hari Sabtu	-
6	-	Minggu 17 Maret 2019	-	Libur	Libur Hari Minggu	-
7	I Pengah Wirajana (Ketua Jurusan Kimia)	Senin, 18 Maret 2019	Jurusan Kimia	Menginput data visi dan misi, fasilitas dan spesifikasi jurusan kimia ke web		
8	-	Selasa, 19 Maret 2019	-	Libur	Pengurusahan	-
-	-	Rabu, 20 Maret 2019	-	Libur	Kegagalan	-

### AKTIVITAS HARIAN PKL

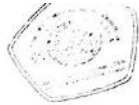
Nama : NIKKI DINIA PRADITA PERMITA  
 NIM : 1806020011  
 Lokasi PKL : JURUSAN KIMIA FAKULTAS ILMU  
 Waktu Pelaksanaan : 1 MARET 2019 - 20 MARI 2019



No.	Nama Penanggung jawab/Jabatan	Pelaksanaan PKL			Ket	Paraf
		Tanggal	Lokasi	Aktivitas		
10	-	Kamis, 21 Maret 2019	-	Libur	Libur Hengkab hari	-
11	I Pengah Wirajana (Ketua Jurusan Kimia)	Jumat 22 Maret 2019	Jurusan Kimia	Menginput data kurikulum dalam bahasa mda nama dan data PKL ke websitechemistry		
12	-	Sabtu, 23 Maret 2019	-	Libur	Libur Hari Sabtu	-
13	-	Minggu 24 Maret 2019	-	Libur	Libur Hari Minggu	-
14	I Pengah Wirajana (Ketua Jurusan Kimia)	Senin, 25 Maret 2019	Jurusan Kimia	Menginput data dosen ke website Chemistry.uns.ac.id		
15	I Pengah Wirajana (Ketua Jurusan Kimia)	Selasa 26 Maret 2019	Jurusan Kimia	Menginput data CV Dosen		
16	I Pengah Wirajana (Ketua Jurusan Kimia)	Rabu 27 Maret 2019	Jurusan Kimia	Menginput data CV Dosen		
17	I Pengah Wirajana (Ketua Jurusan Kimia)	Kamis 28 Maret 2019	Jurusan Kimia	Menginput data tugas akhir dan data tenaga kependidikan		
18	I Pengah Wirajana (Ketua Jurusan Kimia)	Jumat 29 Maret 2019	Jurusan Kimia	Menginput data CV Dosen		

# AKTIVITAS HARIAN PKL

Nama : NADA DINAR PRATIWI PRATIWI  
 NIM : 1305403041  
 Lokasi PKL : JURUSAN KIMIA FAKULTAS UINUJ  
 Waktu Pelaksanaan : 1 Maret 2016 - 30 Mei 2016



No.	Nama Penanggung Jawab/Jabatan	Pelaksanaan PKL			Ket	Paraf
		Tanggal	Lokasi	Aktivitas		
19		Gablu, 19 Maret 2016	-	Libur	Libur Hari Sabtu	-
20		Minggu, 20 Maret 2016	-	Libur	Libur Hari Minggu	-
21	I Nengah Wirajana (Ketua Jurusan Kimia)	Senin, 21 Maret 2016	Jurusan Kimia	Menginput data alumni dan menambahkan link untuk data penelitian dosen di website chemistry		23
22	I Nengah Wirajana (Ketua Jurusan Kimia)	Selasa, 22 Maret 2016	Jurusan Kimia	Menginput data CV Dosen dan data Pelengkap penelitian TA di website Chemistry		23
23		Rabu, 23 Maret 2016	-	Libur	Kegiatan produksi di ekstrakurikuler	-
24	I Nengah Wirajana (Ketua Jurusan Kimia)	Kamis, 24 Maret 2016	Jurusan Kimia	Menginput foto dosen pada data dosen dan CV dosen di website chemistry		23
25		Jumat, 25 Maret 2016	-	Libur	Wakat Tesas Kertas	-
26		Sabtu, 26 Maret 2016	-	Libur	Libur Hari Sabtu	-
27		Minggu, 27 Maret 2016	-	Libur	Libur Hari Minggu	-

Pelaksanaan PKL

# AKTIVITAS HARIAN PKL

Nama : NADA DINAR PRATIWI PRATIWI  
 NIM : 1305403041  
 Lokasi PKL : JURUSAN KIMIA FAKULTAS UINUJ  
 Waktu Pelaksanaan : 1 Maret 2016 - 30 Mei 2016



No.	Nama Penanggung Jawab/Jabatan	Pelaksanaan PKL			Ket	Paraf
		Tanggal	Lokasi	Aktivitas		
28	I Nengah Wirajana (Ketua Jurusan Kimia)	Senin, 28 Maret 2016	Jurusan Kimia	Menginput foto kegiatan (Rapat Jurusan Kimia, Talenta, HUT HIMA KIM)		23
29	I Nengah Wirajana (Ketua Jurusan Kimia)	Selasa, 29 Maret 2016	Jurusan Kimia	Menginput data CV Dosen dan foto kegiatan (Gala rer, Oxxx, Oxxx)		23
30	I Nengah Wirajana (Ketua Jurusan Kimia)	Rabu, 30 Maret 2016	Jurusan Kimia	Menginput foto kegiatan (Bakti Dharma, Green Kampus, HMS Senior 2016)		23
31	I Nengah Wirajana (Ketua Jurusan Kimia)	Kamis, 31 Maret 2016	Jurusan Kimia	Menginput data buku perustakaan		23
32	I Nengah Wirajana (Ketua Jurusan Kimia)	Jumat, 1 April 2016	Jurusan Kimia	Menambahkan data pada news, agenda dan gallery		23
33		Sabtu, 2 April 2016	-	Libur	Libur Hari Sabtu	-
34		Minggu, 3 April 2016	-	Libur	Libur Hari Minggu	-
35	I Nengah Wirajana (Ketua Jurusan Kimia)	Senin, 4 April 2016	Jurusan Kimia	Menginput data buku perpustakaan ke website chemistry		23
36	I Nengah Wirajana (Ketua Jurusan Kimia)	Selasa, 5 April 2016	Jurusan Kimia	Menambahkan data agenda di website chemistry dan data gallery		23

No.	Nama Penanggung Jawab/Jabatan	Tanggal	Lokasi	Aktivitas	Ket.	Paraf
89	I Nengah Wirajana (Ketua Jurusan Kimia)	8 Mei 2018	Jurusan Kimia	Menginput data mahasiswa kelas PBM dan menginput data jadwal kuliah		
90	I Nengah Wirajana (Ketua Jurusan Kimia)	8 Mei 2018	Jurusan Kimia	Menginput dokumen form penyesuaian nilai (rata-rata) 80, 90, 100, untuk peninjauan		
91	I Nengah Wirajana (Ketua Jurusan Kimia)	8 Mei 2018	Jurusan Kimia	Menginput data struktur tata rancang dan data silabus ke website chemistry		
92	Gusti 8 Mei 2018	-	Libur		Jukur Hari Gusti	-
93	Margo 10 April 2018	-	Libur		Jukur Hari Margo	-
94	I Nengah Wirajana (Ketua Jurusan Kimia)	8 Mei 2018	Jurusan Kimia	Menginput data kerjasama dalam negeri dan luar negeri		
95	I Nengah Wirajana (Ketua Jurusan Kimia)	8 Mei 2018	Jurusan Kimia	Menginput foto tenaga kependidikan dan data HIKMAT		
96	I Nengah Wirajana (Ketua Jurusan Kimia)	8 Mei 2018	Jurusan Kimia	Menginput data bidang kefarmasian dosen dan menambahkan foto dosen yang belum lengkap		
97	I Nengah Wirajana (Ketua Jurusan Kimia)	8 Mei 2018	Jurusan Kimia	Menginputkan kebulatan pengumpulan keabsahan sistem informasi: pengumpulan		

No.	Nama Penanggung Jawab/Jabatan	Tanggal	Lokasi	Pelaksanaan PKL	Ket	Paraf
46	I Nengah Wirajana (Ketua Jurusan Kimia)	Jumat, 13 April 2016	Jurusan Kimia	Membuat tutorial untuk maintenance data web chemistry.unsw.ac.id		
47	-	Sabtu, 14 April 2016	-	Libur	Libur Hari Sabtu	-
48	-	Minggu, 17 April 2016	-	Libur	Libur Hari Minggu	-
49	I Nengah Wirajana (Ketua Jurusan Kimia)	Senin, 18 April 2016	Jurusan Kimia	Membuat tutorial untuk maintenance data web chemistry.unsw.ac.id		
50	I Made Windartha	Selasa, 19 April 2016	Jurusan Ilmu Komputer	Gimbingan Draf Laporan PKL		
51	I Nengah Wirajana (Ketua Jurusan Kimia)	Sabtu, 20 April 2016	Jurusan Kimia	Analisa kebutuhan sistem informasi		
52	I Nengah Wirajana (Ketua Jurusan Kimia)	Senin, 21 April 2016	Jurusan Kimia	Membuat perancangan sistem informasi: persyaratan (use case, activity, class diagram)		
53	I Nengah Wirajana (Ketua Jurusan Kimia)	Jumat, 23 April 2016	Jurusan Kimia	Membuat perancangan database dan pemasangan tampilan sistem informasi: persyaratan		
54	-	Sabtu, 23 April 2016	-	Libur	Libur Hari Sabtu	-

# AKTIVITAS HARIAN PKL



Nama : MADE DINIA PRADITA PRANITA  
 NIM : 1208000201  
 Lokasi PKL : JURUSAN KIMIA FAKULTAS UMMU  
 Waktu Pelaksanaan : 1 MARET 2016 - 20 MEI 2016

No.	Nama Penanggung Jawab/Jabatan	Pelaksanaan PKL			Ket	Paraf
		Tanggal	Lokasi	Aktivitas		
55	-	Minggu, 26 April 2016	-	Libur	Libur Hari Minggu	-
56	I Made Widhiarta (Ketua Jurusan Kimia)	Senin, 26 April 2016	Jurusan Ilmu Komputer	Bimbingan Draft Laporan PKL		CP
57	I Nengah Wirajana (Ketua Jurusan Kimia)	Selasa, 26 April 2016	Jurusan Kimia	Mengimplementasikan rancangan database dalam bentuk query		CP
58	I Nengah Wirajana (Ketua Jurusan Kimia)	Rabu, 27 April 2016	Jurusan Kimia	Mengimplementasikan hasil rancangan sistem ke dalam sourcecode program		CP
59	I Nengah Wirajana (Ketua Jurusan Kimia)	Kamis, 28 April 2016	Jurusan Kimia	Mengimplementasikan hasil rancangan sistem ke dalam sourcecode program		CP
60	I Nengah Wirajana (Ketua Jurusan Kimia)	Jumat, 29 April 2016	Jurusan Kimia	Mengimplementasikan hasil rancangan sistem ke dalam sourcecode program		CP
61	-	Sabtu, 30 April 2016	-	Libur	Libur Hari Sabtu	-
61	-	Minggu, 1 Mei 2016	-	Libur	Libur Hari Minggu	-
62	I Nengah Wirajana (Ketua Jurusan Kimia)	Senin, 3 Mei 2016	Jurusan Kimia	Pengujian aplikasi sistem informasi perustakaan		CP

# AKTIVITAS HARIAN PKL



Nama : MADE DINIA PRADITA PRANITA  
 NIM : 1208000201  
 Lokasi PKL : JURUSAN KIMIA FAKULTAS UMMU  
 Waktu Pelaksanaan : 1 MARET 2016 - 20 MEI 2016

No.	Nama Penanggung Jawab/Jabatan	Pelaksanaan PKL			Ket	Paraf
		Tanggal	Lokasi	Aktivitas		
64	I Nengah Wirajana (Ketua Jurusan Kimia)	Senin, 3 Mei 2016	Jurusan Kimia	Pengujian aplikasi sistem informasi perustakaan		CP
65	I Nengah Wirajana (Ketua Jurusan Kimia)	Rabu, 4 Mei 2016	Jurusan Kimia	Pengujian aplikasi sistem informasi perustakaan		CP
66	-	Kamis, 5 Mei 2016	-	Libur	Kendaraan Yang Berstatus	-
67	-	Jumat, 6 Mei 2016	-	Libur	Ira Masyah Rabbul Muhammadiyah	-
68	-	Sabtu, 7 Mei 2016	-	Libur	Libur Hari Sabtu	-
69	-	Minggu, 8 Mei 2016	-	Libur	Libur Hari Minggu	-
70	I Nengah Wirajana (Ketua Jurusan Kimia)	Senin, 9 Mei 2016	Jurusan Kimia	Pengujian aplikasi sistem informasi perustakaan		CP
71	I Nengah Wirajana (Ketua Jurusan Kimia)	Selasa, 10 Mei 2016	Jurusan Kimia	Operasi dan maintenance sistem informasi perustakaan		CP
72	I Nengah Wirajana (Ketua Jurusan Kimia)	Rabu, 11 Mei 2016	Jurusan Kimia	Operasi dan maintenance sistem informasi perustakaan		CP