BAB 3
ANALISIS DAN PERANCANGAN

3.1 Deskripsi Umum

Pengerjaan tugas akhir ini adalah menerapkan *database* nosql terhadap e-library Universitas Pendidikan Indonesia, khususnya pada otomasi perpustakaan upi. Kemudian dilakukan pengujian performa *database* nosql dengan *database* mysql.

3.1.1 Analisis Masalah

Universitas Pendidikan Indonesia mempunyai mahasiswa yang sangat banyak, sehingga untuk menunjang kegiatan belajar mahasiswa perpustakaan digital Universitas Pendidikan Indonesia harus mampu menyajikan data dalam jumlah yang banyak pula.Salah satu faktor yang dapat memenuhi hal tersebut adalah penggunaan *database* yang tepat.

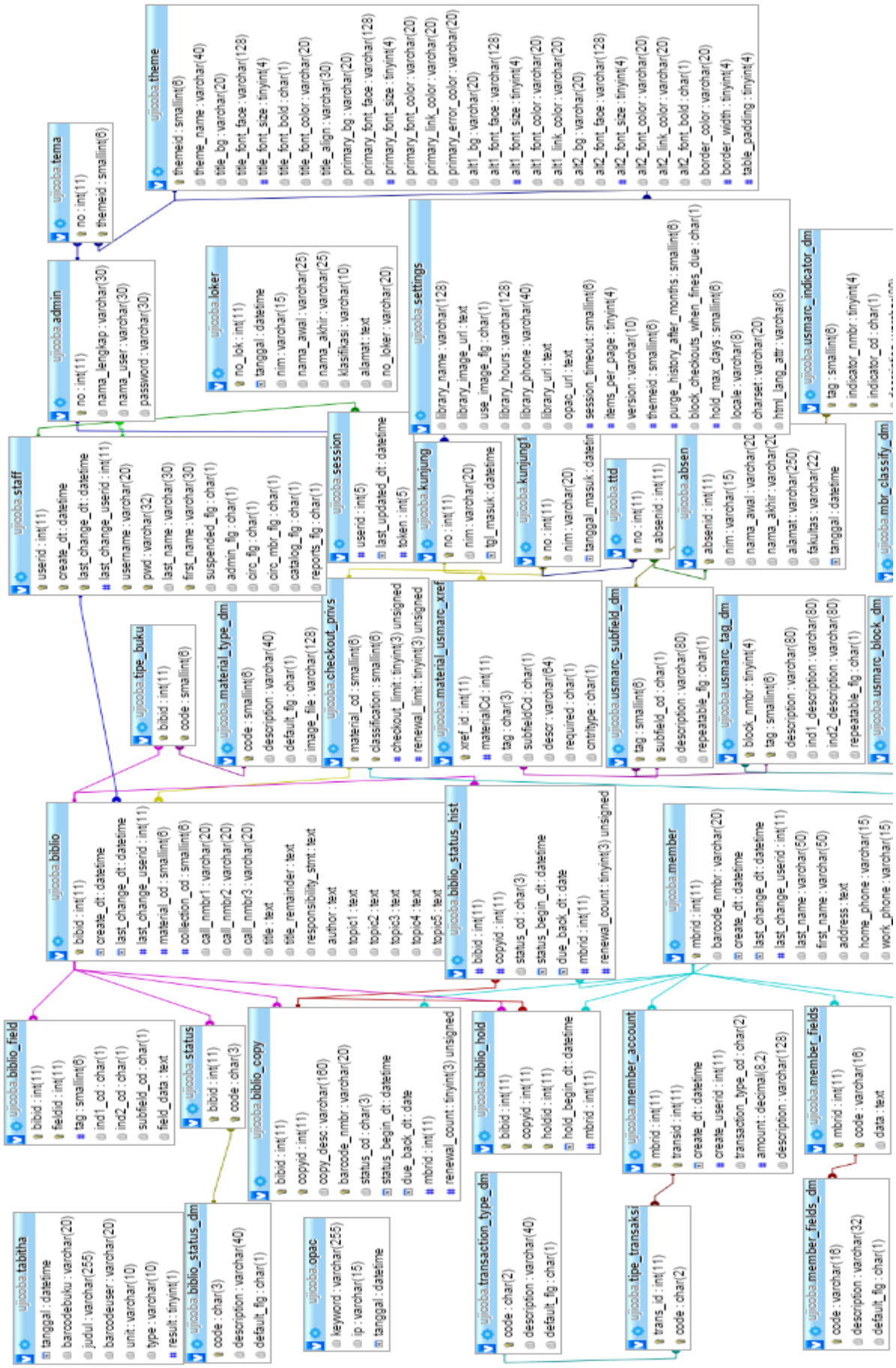
Otomasi Perpustakaan Universitas Pendidikan Indonesia masih menggunakan *database* mysql dimana database mysql mempunyai kendala pada penanganan data dalam jumlah yang besar.

3.1.2 Analisis *Database* yang Sedang Berjalan

Analisis *database* dilakukan untuk mengetahui seberapa besar permasalahan yang terdapat pada sistem *database* perpustakaan, sehingga proses analisis dapat menghasilkan sebuah informasi.

3.1.2.1 Diagram Relasi

Diagram relasi merupakan hubungan antar tabel yang terdapat dalam suatu RDBMS.Untuk keperluan penelitian, *database* UPI telah dilakukan perubahan dengan ditambahkannya tabel sebagai penunjang skema relasi.Berikut skema relasi *database* otomasi perpustakaan upi.



Gambar 3. 1 Diagram Relasi Database Otomasi Perpustakaan UPI

3.1.2.2 Daftar Tabel

Database yang digunakan otomasi perpustakaan Universitas Pendidikan Indonesia ialah RDBMS mysql.Struktur tabel dari otomasi perpustakaan online UPI mempunyai 33 tabel.*Database* otomasi perpustakaan upi terlihat pada gambar 3.1.

Tabel 3. 1 Struktur Table Database Otomasi Perpustakaan UPI

Tabel	**Jumlah Baris**	**Keterangan**
Absen	2.413	Absen Kehadiran
Admin	2	Admin Perpustakaan
Biblio	36.877	Buku
biblio_copy	68.245	Status buku
biblio_field	445.454	Kopian buku
biblio_hold	5	Buku yang ditahan
biblio_statu_dm	9	Status Pinjaman Buku
biblio_status_hist	268.783	Peminjaman
checkout_privs	5	Status Checkout Matrial
collection_dm	9	Sanksi Keterlambatan
Klasifikasi	0	Tabel baru dari member dan member klasifikasi
Kunjung	38.141	Kunjungan Mahasiswa1
kunjung1	20	Kunjungan Mahasiswa2
material_type_dm	4	Jenis Material
material_usmarc_xref	0	Xref material
mbr_classify_dm	4	Klasifikasi Member
Member	32.982	Member Perpustakaan
member_account	1.591	Akun Member
member_fields	61.737	Asal Member
member_fields_dm	3	Klasifikasi Asal member
Pengunjung	727	Pengunjung Perpustakaan
Session	71	Batas Waktu Online Admin
Setting	1	Pengaturan identitas perpustakaan
Staff	31	Staff Perpustakaan
Status	0	Tabel Baru dari buku dan status buku
Tema	0	Tabel baru dari tema dan admin
Theme	7	Tema halaman admin

Tipe_buku	0	Tabel baru dari buku dan jenis material
Ttd	0	Tabel baru dari kinjungan dan absen
usmarc_block_dm	10	Buku yang diblok
usmarc_indicator_dm	642	Indikator Blok buku
*usmarc_subfield_*	1.678	Elemen Kerusakan Buku
usmarc_tag_dm	175	Elemen Kerusakan Buku

3.1.3 Rutinitas *Table*

Rutinitas table merupakan aktifitas data pada *table table* dari database otomasi perpustakaan UPI, aktifitas table tersebut dirangkum sebagai rutinitas yang sering hingga jarang dilakukan. Rutinitas *table* yang diambil pada bulan juni.

Tabel 3.2 Rutinitas Penggunaan *Table Database* UPI pada bulan juni

Rutinitas tersebut merupakan rangkuman rutinitas harian dari penggunaan tabel yang dirangkum kedalam empat periode selama satu bulan.

Tabel	**Pekan ke 1**	**Pekan ke 2**	**Pekan ke 3**	**Pekan Ke 4**	**Total**
Absen	4	4	4	4	16
Admin	9	7	7	7	34
Biblio	23	34	27	33	117
biblio_copy	8	9	9	7	33
biblio_field	2	1	0	1	4
biblio_statu_dm	11	8	9	13	41
biblio_status_hist	2	1	2	1	6
collection_dm	0	0	0	0	0
klasifikasi	0	0	0	0	0
kunjung	23	34	27	33	117
kunjung1	23	34	27	33	117
material_type_dm	12	11	15	9	47
material_usmarc_xref	0	0	0	0	0
mbr_classify_dm	3	1	2	2	8
member	23	34	27	33	117
member_account	15	23	17	22	77

member_fields	15	23	17	22	77
member_fields_dm	3	1	2	1	7
pengunjung	23	34	27	33	117
session	4	4	4	4	16
Setting	4	4	4	4	16
Staff	9	8	7	7	31
Status	0	0	0	0	0
Tema	0	0	0	0	0
Theme	1	1	1	1	4
Tipe_buku	0	0	0	0	0
Ttd	0	0	0	0	0
usmarc_block_dm	2	3	2	3	10
usmarc_indicator_dm	2	3	2	3	10
*usmarc_subfield_*	2	3	2	3	10
usmarc_tag_dm	2	3	2	3	10

Keterangan: total pengaksesan 0 tidak ada pengaksesan (~), total pengaksesan 1-30 = jarang, dan total pengaksesan lebih dari 30 = sering.

Berdasarkan rutinitas dari penggunaan table pada tabel 3.2, maka rutinitas *table* database otomasi perpustakaan UPI disajikan pada tabel 3.3.

Tabel 3. 3 Runititas table database otomasi perpustakaan UPI.

No	Nama Tabel	*Read*	*Upadate*	*Delete*
1	*Member*	Sering	Jarang	Jarang
2	*Member_fields*	Sering	Jarang	Jarang
3	*Member_fields_dm*	Jarang	Jarang	Jarang
4	*Mbr_clasiffy_dm*	Jarang	Jarang	Jarang
5	*Member_account*	Sering	Tidak ada	Tidak ada
6	*Klasifikasi*	~	~	~
7	*Tipe_transaksi*	~	~	~
8	*Biblio*	Sering	jarang	Tidak ada
9	*Collection_dm*	~	Jarang	Jarang
10	*biblio_status_dm*	Sering	Sering	Tidak ada
11	*Biblio_copy*	Sering	Jarang	Tidak ada

12	*Biblio_field*	Jarang	Sering	Jarang
13	*Material_type_dm*	Sering	Tidak ada	Tidak ada
14	*Biblio_status_hits*	Jarang	Jarang	Jarang
15	*Status*	~	~	~
16	*Tipe_buku*	~	~	~
17	*Staff*	Sering	Jarang	Jarang
18	*Admin*	Sering	Sering	Jarang
29	*Tema*	~	~	~
20	*Theme*	Jarang	Jarang	Jarang
21	*Setting*	Jarang	Jarang	Jarang
22	*Absen*	~	~	~
23	*Kunjung*	Sering	Jarang	Jarang
24	*kunjung1*	Sering	Jarang	Jarang
25	*Material_usmark*	~	~	~
26	*Pengunjung*	Sering	Jarang	Jarang
27	*Session*	Jarang	Jarang	Jarang
28	*Ttd*	~	~	~
29	*usmarc_block_dm*	Jarang	Jarang	Jarang
30	*usmarc_indicator_dm*	Jarang	Jarang	Jarang
31	*usmarc_subfield_*	Jarang	Jarang	Jarang
32	*usmarc_tag_dm*	Jarang	Jarang	Jarang

Berdasarkan informasi yg didapat pada tabel 3.3, maka analisis yang didapat

a. Data yang dihasilkan terpusat pada tabel member, biblio, staff, maka tabel member, biblio, staff, perpus_seting, pinjam, denda dan konten merupakan sebuah collection.
b. Berdasarkan data pada tabel 3.3, dari 32 tabel *database* tersebut rutinitas yang sering dilakukan untuk *read* 12 tabel, *update* 3 tabel dan *delete* 0 tabel. Berdasarkan rutinitas tersebut maka perlu dilakukan

analisis query *read* data untuk mengoptimalkan proses read data pada aplikasi yang akan dibangun.

3.1.4 Analisis Query

Analisis ini bertujuan untuk meganalisa bagaimana Query MySQL serta Query MongoDB dalam menampilkan data perpustakaan. Sebagai contoh penggunaan query diambil pada pencarian data anggota.

3.1.4.1 Pencarian data

Query ini bertujuan untuk memperoleh informasi data anggota perpustakaan. *Table* yang digunakan dalam *query* ini adalah *member* dan *staff.*

a. *Query* pencarian data MySQL

Query MySQL untuk memanggil data dimana kebutuhan data berasal dari satu tabel.

```
mysql_query(mysql_fetch_array("select * from katalog where kode_buku="101""))
```

Berdasarkan query tersebut dilakukan perhitungan jumlah karakter, maka pada query tersebut terdapat 70 karakter

Query MySQL untuk memanggil data dimana kebutuhan data berasal dari dua tabel menggunakan join.

```
“$sql = $this->mkSQL("select member.*, staff, userfname from member"."left join staf on member.last_change_userid = staff.userid"."where mbrid=%N ", $mbrid);
```

Berdasarkan query tersebut dilakukan perhitungan jumlah karakter, maka pada query tersebut terdapat 170 karakter.

b. *Query* pencarian data MongoDB

Query yang digunakan oleh MongoDB dalam pemanggilan data

```
db.katalog.find( { "kode_buku" : "101" } )
```

Berdasarkan perhitungan karakter *query* tersebut, diperoleh sebanyak 42 karakter.

Query MongoDB untuk kebutuhan data yang berasal dari dua *collection* dalam MongoDB sama. Hal ini disebabkan skema mongodb bersifat dinamis, maka pada collection katalog bisa disisipkan dokumen dari collection anggota.

3.1.4.2 Insert Data

a. Query Insert data Mysql

```
$query="insert into katalog values('kode_buku' :
   '101','judul_buku' : 'Dev C','pengarang' :
   'andre','jenis_buku' : 'sirkulasi','penerbit':
   'elang','tahun_terbit' : '2010','stok':
   '10','deskripsi' : 'basic programming',
   'tanggal_update': '01/February/2016','tanggal':
   '01/February/2016')";
    $exe=mysql_query($query);
```

Berdasarkan query tersebut dilakukan perhitungan jumlah karakter, maka pada query tersebut terdapat 288 karakter.

b. Query insert data Mongodb

```
db.katalog,insert {
        "kode_buku" : "101",
        "judul_buku" : "Dev C",
        "pengarang" : "andre",
        "jenis_buku" : "sirkulasi",
        "penerbit" : "elang",
        "tahun_terbit" : "2010",
        "stok" : "10",
        "deskripsi" : "basic programming",
        "tanggal_update" : "01/February/2016",
        "tanggal" : "01/February/2016"
   }
```

Berdasarkan query tersebut dilakukan perhitungan jumlah karakter, maka pada query tersebut terdapat 250 karakter.

3.1.4.3 Update Data

a. Query update mysql

```
$query= update katalog set 'kode_buku' :
   '101','judul_buku' : 'Dev C','pengarang' :
   'andre','jenis_buku' : 'sirkulasi','penerbit':
   'elang','tahun_terbit' : '2010','stok':
   '10','deskripsi' : 'basic
   programming','tanggal_update':
   '01/February/2016','tanggal': '01/February/2016';
$exe=mysql_query($query,$conn);
```

Berdasarkan query tersebut dilakukan perhitungan jumlah karakter, maka pada query tersebut terdapat 288 karakter.

b. Query update mongodb

```
db.katalog,update(
        {    "kode_buku" : "101" },    {
        $set{
             "judul_buku" : "Dev C",
             "pengarang" : "andre",
             "jenis_buku" : "sirkulasi",
             "penerbit" : "elang",
             "tahun_terbit" : "2010",
             "stok" : "10",
             "deskripsi" : "basic programming",
             "tanggal_update" : "01/February/2016",
             "tanggal" : "01/February/2016"
        }
```

Berdasarkan query tersebut dilakukan perhitungan jumlah karakter, maka pada query tersebut terdapat 258 karakter.

3.1.4.4 Delete Data

a. Query delete data mysql

```
Delete from katalog where kode_buku=101;
```

Berdasarkan query tersebut dilakukan perhitungan jumlah karakter, maka pada query tersebut terdapat 36 karakter

b. Query delete data mongodb

```
db.katalog.remove{ "kode_buku" : "101" }
```

Berdasarkan query tersebut dilakukan perhitungan jumlah karakter, maka pada query tersebut terdapat 36 karakter

Berdasarkan hasil perhitungan karakter dari setiap query, maka dapat diperoleh kesimpulan bahwa dengan memakai mongodb dapat mempersingkat proses pembacaan karena jumlah karakter dalam mongodb lebih sedikit dibandingkan dengan jumlah karakter pada mysql.

3.1.5 Analisis Kebutuhan Data

Berdasarkan sifat NoSQL yang bersifat *schemaless,* berarti mendefinisikan kebutuhan data secara sederhana, kemudian dapat langsung membuat database NoSQL MonggoDB, Analisis kebutuhan data diperlukan untuk proses penerapan yang akan dilakukan, penggunaan tabel pada RDBMS MySQL mencerminkan kebutuhan data untuk pemodelan *database* NoSQL MongoDB. Adapun analisis kebutuhan data yang dimaksud dapat dilihat pada tabel 3.3

Tabel 3. 2 Tabel Kebutuhan Data

No	Data	Tabel yang digunakan
1	Anggota	*member, member_fields, member_fields_dm, mbr_classify_dm, member_account, klasifikasi, tipe_transaksi*
2	Buku (Katalog)	*biblio, material_type_dm, collection_dm, biblio_status_dm, biblio_copy, status, tipe_buku, biblio_field, biblio_hold, biblio_status_hits.*
3	Staff	*Staff, admin, tema, theme.*

Berdasarkan diagram relasi *database* otomasi perpustakaan UPI maka didapatkan kebutuhan data apa saja yang akan diperlukan pada *database* NoSQL. Maka didapatkanlah tiga kebutuhan data utama, diantaranya :

1. Data Kebutuhan Penglolaan Anggota

   Data kebutuhan pengelolaan Anggota yaitu tabel *member, member_fields, member_fields_dm, mbr_classify_dm, member_account, klasifikasi, tipe_transaksi.*Contoh kebutuhan data pengolahan anggota yaitu

   - Pencarian anggota, *table* yang digunakan adalah *member, mbr_clasiffy_dm dan klasifikasi*
   - Penambahan anggota, *table* yang digunakan adalah *member, member_fields_dm, mbr_clasiffy_dm.*
   - Pengelolaan tipe anggota, *table* yang digunakan adalah *mbr_classify_dm, member* dan *klasifikasi.*

2. Data Kebutuhan Pengelolaan Buku

   Data kebutuhan pengelolaan buku yaitu *table biblio, material_type_dm, collection_dm, biblio_status_dm, biblio_copy, status, tipe_buku, biblio_field, biblio_hold, biblio_status_hits.*Contoh kebutuhan Pengelolaan buku yaitu :

   - Pencarian Katalog, *table* yang digunakan adalah *biblio, material_type_dm, biblio_status_dm, biblio_copy, status, tipe_buku.*
   - Pengembalian buku, *table* yang digunakan adalah *biblio, biblio_copy*
   - Penambahan Bibliografi, *table* yang digunakan adalah *biblio, material_type_dm* dan *type_buku.*
   - Pengelolaan tipe material, *table* yang digunakan adalah *material_type_dm, biblio* dan *type_buku*

2. Data Kebutuhan Pengelolaan Staff

   Data kebutuhan pengelolaan Staff yaitu *tableStaff, admin, tema, theme, Setting*. Contoh kebutuhan pengelolaan staff yaitu :

   - Pengelolaan staff admin, *table* yang digunakan adalah *staff* dan *admin*
   - Pengelolaan informasi perpustakaan, *table* yang digunakan adalah *admin, setting*
   - Penggantian Tema, *table* yang digunakan adalah *admin, tema* dan *theme.*

3.1.6 Konversi Database MySQL ke MongoDB

Berdasarkan proses analisis sebelumnya, pertimbangan terhadap analisis kebutuhan data menghasilkan rangkuman *collection* yang terbentuk, *collection* merupakan koleksi dari beberapa dokumen. Collection diibaratkan sama dengan *table* pada RDBMS. beberapa *collection* yang terbentuk diantaranya

a. *Collection katalog* yang didapat berdasarkan hasil analisis terhadap kebutuhan data pada halaman pencarian (user), pencarian katalog (admin), sirkulasi pengembalian (admin), katalog bibbliografi baru (admin), tipe material, koleksi, pengaturan pinjaman (admin).
b. *Collection anggota* yang didapat berdasarkan hasil analisis terhadap kebutuhan data pada halaman sirkulasi pencarian anggota, tambah anggota, admin staf admin, tipe anggota.
c. *Collection staff* yang didapat berdasarkan hasil analisis terhadap kebutuhan data halaman setting.

Analisis ini menghasilkan tiga *collection* dengan model *document oriented* adapun daftar *collection* NoSQL MonggoDB untuk sistem otomasi perpustakaan Universitas Pendidikan Indonesia disajikan pada tabel 3.4.

Tabel 3. 3 Daftar Collection NoSQL MonggoDB

Collection	***Dokumen***	***References***
Katalog	Pencarian user, pencarian catalog, sirkulasi pengembalian, bibliografi baru, tipe material, koleksi.	*biblio, material_type_dm, collection_dm, biblio_status_dm, biblio_copy, status, tipe_buku, biblio_field, biblio_hold, biblio_status_hits*
Anggota	Pencarian anggota, tambah anggota, staff admin, tipe anggota.	*member, member_fields, member_fields_dm, mbr_classify_dm, member_account, klasifikasi, tipe_transaksi*
Staff	Setting	*setting Staff, admin, tema, theme, Setting*

Berdasarkan tabel 3.3 rincian model yang akan dilakukan konversi secara pragmantik adalah sebagai berikut :

1) ***Collection katalog***

Db.createcollection(“Katalog”,{ ,

create_date=>' ',

last_change_dt=>' ',

last_change_userid=>' ',

material_cd=>' ',

collection_cd=>’ ‘,

call_nmbr1=>’ ‘,

call_nmbr2=>’ ‘,

call_nmber3=>’ ‘,

title=>’ ‘,

title_responsibility_smst=>’ ‘,

author=>’ ‘,

topic1=>’ ‘,

topic2=>’ ‘,

topic3=>’ ‘,

topic4=>’ ‘,

topic5=>’ ‘,

oprn_flg=>’ ‘,

copyid=>’ ‘,

copy_desk=>’ ‘,

barcode_nmbr=>’ ‘,

status_cd=>’ ‘,

status_begin_dt=>’ ‘,

due_back_dt=>’ ‘,

renewal_count=>’ ‘,

fieldid=>’ ‘,

tag=>’ ‘,

ind1_cd=>’ ‘,

ind2_cd=>’ ‘,

subfield_cd=>’ ‘,

field_data);

2) *Colletion anggota*

Db.createcollection (“anggota”,{

mbrid=>’ ’,

code=>’ ‘,

data=>’ ‘,

code=>’ ’

description=>’ ‘,

default_flg=>’ ‘,

description=>’ ’,

default_flg=>’ ‘,

max_fines=>’ ‘,

staff=>’’,);

3) *Collection staff*

Db.createcollection (“staff”,{

User_id=>’ ‘,

Create_date=>’ ‘,

Last_change_dt=>’ ‘

Username=>’ ‘,

Password=>’ ‘,

Last_name=>’ ‘,

First_name=>’ ‘,

Last_change_userid=>’ ‘,

Library_name=>’ ‘,

Library_img_url=>’ ‘,

Use_img_flg=>’ ‘,

Library_hourse=>’ ‘,

Library_phone=>’ ‘,

Library_url=>’ ‘,

Session_time_out=>’ ‘})

Tiga *collection* diatas dibuat dengan menggunakan konsep *document oriented database* yang memang managemen datanya hanya menggunakan array, tidak menggunakan *table* ataupun *reqord*.

3.2 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai pada pengimplementasian dari sistem tersebut. Pada analisis kebutuhan sistem non-fungsional ini dijelaskan analisis mengenai *hardware, software,* dan *brainware.*

3.2.1 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam menerapkan *database* NoSQL MongoDB dapat dilihat pada tabel 3.5

Tabel 3. 4 Tabel Kebutuhan Perangkat Keras

Nama Perangkat	**Spesifikasi**
Processor	Dual Core 1.0 GHz
RAM	4 Gb
HDD	500 Gb

3.2.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam menerapkan *database* NOSQL MonggoDB dapat dilihat pada tabel 3.6

Tabel 3. 5 Tabel Kebutuhan Perangkat Lunak

Nama Perangkat Lunak	**Spesifikasi**
Sistem Operasi	*Windows, Mac OS, Linux*
Web Server	*NodeJS v0.10.32*
Database Server	*Localhost XAMPP, MonggoDB*
Web Browser	*Chrome, Firefox*

3.2.3 Analisis Pengguna

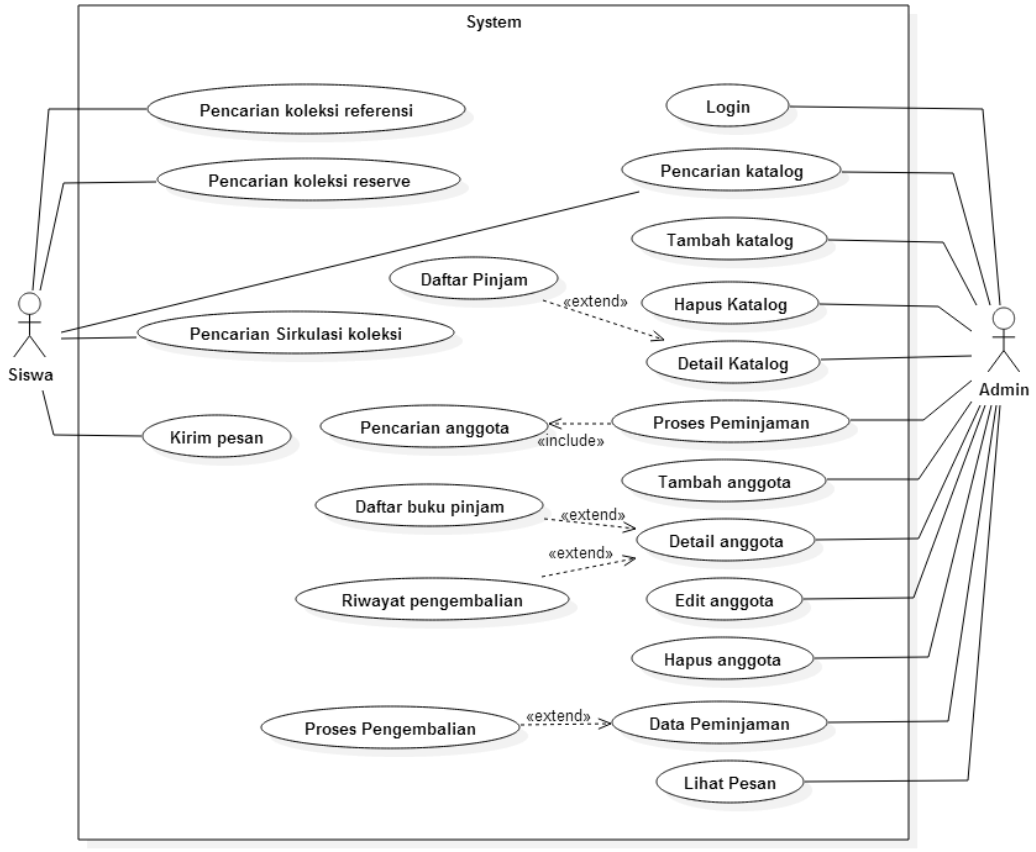
Sistem Otomasi Perpustakaan yang akan dibangun memiliki target pengguna, untuk user selaku pengunjung ditentukan minimal seorang yang bisa menjalankan computer dan untuk admin ditentukan minimal seorang database administrator yang biasa mengelola sebuah *database.*

3.3 Analisis Kebutuhan Fungsional

Analisis kebutuhan membahas perancangan sistem yang dibangun dalam bentuk analisis diagram UML(*Unified Modeling Language*). UML digunakan sebagai alat untuk menganalisis kebutuhan fungsional terhadap sistem yang akan dibangun.

3.3.1 Use Case Diagram

Use case Diagram menunjukan interaksi antara *Use case* dan dua actor yaitu user dan admin.



Gambar 3. 2 Use Case Diagram

3.3.2 Use Case Skenario

Bagian *use case* memiliki scenario yang menunjukan proses apa saja yang terjadi.

a. Skenario Login

Tabel 3. 6 Skenario Login

Use Case Name	Login	
Goal In Context	Menampilkan Tampilan Aplikasi	
Preconditions	Menampilkan Input text	
Successful Condition	Berhasil menampilkan halaman aplikasi	
Failed End Condition	Menampilkan pesan gagal login	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna mengisi form username dan password	
Main Flow	***Step***	***Action***
	1	Pengguna mengisi form username dan password
	2	Sistem melakukan Validasi dan Verifikasi
	3	Sistem menampilkan tampilan aplikasi
Extensions	***Step***	***Branching Action***
	3.1	Menampilkan pesan gagal login

b. Skenario Pencarian Katalog

Tabel 3. 7 Skenario Pencarian Katalog

Use Case Name	Pencarian Katalog	
Goal In Context	Menampilkan Data Katalog	
Preconditions	Menampilkan tampilan awal	
Successful Condition	Berhasil menampilkan data katalog	
Failed End Condition	Menampilkan pesan data tidak ditemukan	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna melkukan *request* katalog	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* katalog
	2	Sistem menampilkan daftar katalog
	3	Pengguna mengisi data katalog
	4	Sistem mencari data katalog
	5	Sistem menampilkan data yang ditemukan
Extensions	***Step***	***Branching Action***

	5.1	Menampilkan pesan data tidak ditemukan

c. Skenario Tambah Katalog

Tabel 3. 8 Skenario Tambah Katalog

Use Case Name	Tambah katalog	
Goal In Context	Menambah Data katalog	
Preconditions	Menampilkan menu katalog untuk pengguna	
Successful Condition	Berhasil mmenambah menu katalog	
Failed End Condition	Menampilkan pesan gagal	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* tambah katalog	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* tambah katalog
	2	Sistem menampilkan input text
	3	Pengguna mengisi data katalog
	4	Pengguna memilih *action* simpan
	5	Menampilkan pesan sukses
	6	Sistem menampilkan menu katalog
Extensions	***Step***	***Branching Action***
	5.1	Menampilkan pesan gagal

d. Skenario Hapus Katalog

Tabel 3. 9 Skenario Hapus Katalog

Use Case Name	Hapus katalog	
Goal In Context	Menghapus data katalog	
Preconditions	Menampilkan daftar katalog untuk pengguna	
Successful Condition	Berhasil menghapus data katalog	
Failed End Condition	data katalog gagal dihapus	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* katalog	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* katalog
	2	Sistem menampilkan pesan konfirmasi penghapusan
	3	Menghapus data katalog
Extensions	***Step***	***Branching Action***
	3.1	Gagal menghapus data katalog

e. Skenario Detail Katalog

Tabel 3. 10 Skenario Detail Katalog

Use Case Name	Detail Katalog	
Goal In Context	Menampilkan detail katalog	
Preconditions	Menampilkan daftar katalog untuk pengguna	
Successful Condition	Berhasil menampilkan detail katalog	
Failed End Condition	-	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* detail katalog	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* detail katalog
	2	Sistem menampilkan detail katalog

f. Skenario Daftar Peminjam

Tabel 3. 11 Skenario Daftar Peminjaman

Use Case Name	Daftar Peminjam	
Goal In Context	Menampilkan halaman daftar peminjam	
Preconditions	Menampilkan detail katalog	
Successful Condition	Berhasil menampilkan daftar peminjam	
Failed End Condition	-	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* data peminjam	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* data peminjam
	2	Sistem menampilkan daftar pinjam

g. Skenario Edit Katalog

Tabel 3. 12 Skenario Edit Katalog

Use Case Name	Edit Katalog
Goal In Context	Menampilkan detail katalog
Preconditions	Menampilkan daftar katalog untuk pengguna
Successful Condition	Berhasil menampilkan detail katalog
Failed End Condition	Menampilkan pesan katalog gagal diubah
Primary Actor	Admin
Secondary Actor	-
Trigger	Pengguna melakukan *request* edit katalog

Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* edit katalog
	2	Sistem menampilkan input text
	3	Pennggguna mengedit data katalog
	4	Pengguna memilih *action* simpan
	5	Sistem menampilkan pesan katalog berhasil diubah
	6	Menampilkan halaman detail katalog
Extensions	***Step***	***Branching Action***
	5.1	Menampilkan pesan gagal mengubah katalog

h. Skenario Proses Peminjaman

Tabel 3. 13 Skenario Proses Peminjaman

Use Case Name	Proses Peminjaman	
Goal In Context	Menampilkan detail katalog	
Preconditions	Menampilkan daftar katalog untuk pengguna	
Successful Condition	Berhasil menampilkan detail katalog	
Failed End Condition	Menampilkan pesan gagal meminjam	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* pinjam	
Main Flow	***Step***	***Action***
	1	Pengguna memilih *action* pinjam
	2	Sistem menampilkan peminjaman
	3	Pengguna mengisi nomor anggota
	4	Pengguna memilih *action* peminjaman
	5	Sistem menampilkan pesan berhasil meminjam
	6	Sistem menampilkan detail katalog
Extensions	***Step***	***Branching Action***
	5.1	Sistem menampilkan pesan gagal meminjam
	6	Sistem menampilkan detail katalog

i. Skenario Pencarian Anggota

Tabel 3. 14 Skenario Pencarian Anggota

Use Case Name	Pencarian anggota
Goal In Context	Menampilkan data anggota
Preconditions	Menampilkan tampilan awal untuk pengguna
Successful Condition	Berhasil menampilkan data anggota
Failed End Condition	Menampilkan pesan data tidak ditemukan

Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* anggota	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* anggota
	2	Sistem menampilkan daftar anggota
	3	Pengguna mengisi data pencarian
	4	Sistem menampilkan data anggota
Extensions	***Step***	***Branching Action***
	4.1	Sistem menampilkan pesan data tidak ditemukan

j. Skenario Tambah Anggota

Tabel 3. 15 Skenario Tambah Anggota

Use Case Name	Tambah anggota	
Goal In Context	Menambah data anggota	
Preconditions	Menampilkan menu anggota untuk pengguna	
Successful Condition	Berhasil menambah anggota	
Failed End Condition	Menampilkan pesan gagal	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* tambah anggota	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* tambah anggota
	2	Sistem menampilkan input text
	3	Pengguna menambah anggota
	4	Pengguna memilih *action* simpan
	5	Sistem menampilkan pesan suskses menambah anggota
	6	Kembali ke menu anggota
Extensions	***Step***	***Branching Action***
	3.1	Sistem menampilkan pesan gagal menambah anggota

k. Skenario Detail Anggota

Tabel 3. 16 Skenario Detail Anggota

Use Case Name	Detail anggota
Goal In Context	Menampilkan detail anggota
Preconditions	Menampilkan daftar anggota untuk pengguna
Successful Condition	Berhasil menambahkan detail anggota
Failed End Condition	-

Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* detail anggota	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* detail anggota
	2	Sistem menampilkan detail anggota

l. Skenario Daftar Buku Pinjam

Tabel 3. 17 Skenario Daftar Buku Pinjam

Use Case Name	Daftar buku pinjam	
Goal In Context	Menampilkan daftar buku yang dipinjam	
Preconditions	Menampilkan detail anggota	
Successful Condition	Berhasil menampilkan daftar buku yang dipinjam	
Failed End Condition	-	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* sedang dipinjam	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* sedang dipinjam
	2	Sistem menampilkan data daftar buku yang dipinjam

m. Skenario Riwayat Pengembalian

Tabel 3. 18 Skenario Riwayat Pengambilan

Use Case Name	Riwayat pengembalian	
Goal In Context	Menampilkan riwayat pengembalian	
Preconditions	Menampilkan detail anggota untuk pengguna	
Successful Condition	Berhasil menampilkan riwayat pengembalian	
Failed End Condition	-	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* riwayat pengembalian	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* riwayat pengembalian
	2	Sistem menampilkan daftar riwayat pengembalian

n. Skenario Edit Anggota

Tabel 3. 19 Skenario Edit Anggota

Use Case Name	Edit anggota	
Goal In Context	Mengubah data anggota	
Preconditions	Menampilkan detail anggota	
Successful Condition	Berhasil mengubah data anggota	
Failed End Condition	Gagal mengubah data anggota	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* edit anggota	
Main Flow	***Step***	***Action***
	1	Pengguna memilih *action* edit anggota
	2	Sistem menampilkan input text
	3	Pengguna mengedit data anggota
	4	Pengguna memilih *action* simpan
	5	Sistem Menampilkan pesan anggota berhasil diubah
	6	Sistem menampilkan edit anggota
Extensions	***Step***	***Branching Action***
	5.1	Sistem menampilkan pesan anggota gagal diubah

o. Skenario Hapus Anggota

Tabel 3. 20 Skenario Hapus Anggota

Use Case Name	Hapus Anggota	
Goal In Context	Menghapus anggota	
Preconditions	Sistem Menampilkan daftar anggota	
Successful Condition	Berhasil menghapus data anggota	
Failed End Condition	data anggota gagal dihapus	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna memilih *action* delete	
Main Flow	***Step***	***Action***
	1	Pengguna memilih *action* delete
	2	Sistem menampilkan pesan konfirmasi penghapusan
	3	Sisgtem menghapus data anggota
Extensions	***Step***	***Branching Action***
	3.1	Gagal menghapus data anggota

p. Skenario Data Peminjaman

Tabel 3. 21 Skenario Data Peminjaman

Use Case Name	Data peminjaman	
Goal In Context	Menampilkan daftar peminjam	
Preconditions	Menampilkan tampilan awal	
Successful Condition	Berhasil menampilkan daftar peminjam	
Failed End Condition	-	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* status	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* status
	2	Sistem menampilkan daftar peminjam

q. Skenario Proses Pengembalian

Tabel 3. 22 Skenario Proses Pengembalian

Use Case Name	Proses pengembalian	
Goal In Context	Mengembalikan katalog ke perpustakaan	
Preconditions	Menampilkan daftar status penjaman	
Successful Condition	Berhasil mengembalikan katalog ke perpustakaan	
Failed End Condition	gagal mengembalikan katalog ke perpustakaan	
Primary Actor	Admin	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* kembalikan	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* kembalikan
	2	Sistem menampilkan pesan kambalikan katalog ke perpustakaan
	3	Sistem Menampilkan detail buku
Extensions	***Step***	***Branching Action***
	3.1	Sistem Menampilkan daftar status

r. Skenario Lihat Pesan

Tabel 3. 23 Skenario Lihat Pesan

Use Case Name	Lihat pesan
Goal In Context	Menampilkan daftar pesan
Preconditions	Menampilkan tampilan awal
Successful Condition	Berhasil menampilkan daftar pesan
Failed End Condition	-
Primary Actor	Admin

Secondary Actor	-	
Trigger	Pengguna melakukan *request* pesan	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* pesan
	2	Sistem menampilkan daftar pesan

s. Skenario Pencarian Koleksi Referensi

Tabel 3. 24 Skenario Pencarian Koleksi Referensi

Use Case Name	Pencarian Koleksi referensi	
Goal In Context	Menampilkan daftar koleksi referensi	
Preconditions	Menampilkan tampilan awal	
Successful Condition	Berhasil menampilkan daftar koleksi referensi	
Failed End Condition	-	
Primary Actor	Siswa	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* koleksi referesi	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* koleksi referensi
	2	Sistem menampilkan daftar koleksi referensi

t. Skenario Pencarian Koleksi Reserve

Tabel 3. 25 Skenario Pencarian Koleksi Reverse

Use Case Name	Pencarian Koleksi reserve	
Goal In Context	Menampilkan daftar koleksi reserve	
Preconditions	Menampilkan tampilan awal	
Successful Condition	Berhasil menampilkan daftar koleksi reserve	
Failed End Condition	-	
Primary Actor	Siswa	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* koleksi reserve	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* koleksi reserve
	2	Sistem menampilkan daftar koleksi reserve

u. Skenario Pencarian Koleksi Sirkulasi

Tabel 3. 26 Skenario Pencarian Koleksi Sirkulasi

Use Case Name	Pencarian Koleksi sirkulasi	
Goal In Context	Menampilkan daftar koleksi sirkulasi	
Preconditions	Menampilkan tampilan utama	
Successful Condition	Berhasil menampilkan daftar koleksi sirkulasi	
Failed End Condition	-	
Primary Actor	Siswa	
Secondary Actor	-	
Trigger	Pengguna melakukan *request* koleksi sirkulasi	
Main Flow	***Step***	***Action***
	1	Pengguna melakukan *request* koleksi sirkulasi
	2	Sistem menampilkan daftar koleksi sirkulasi

v. Skenario Kirim Pesan

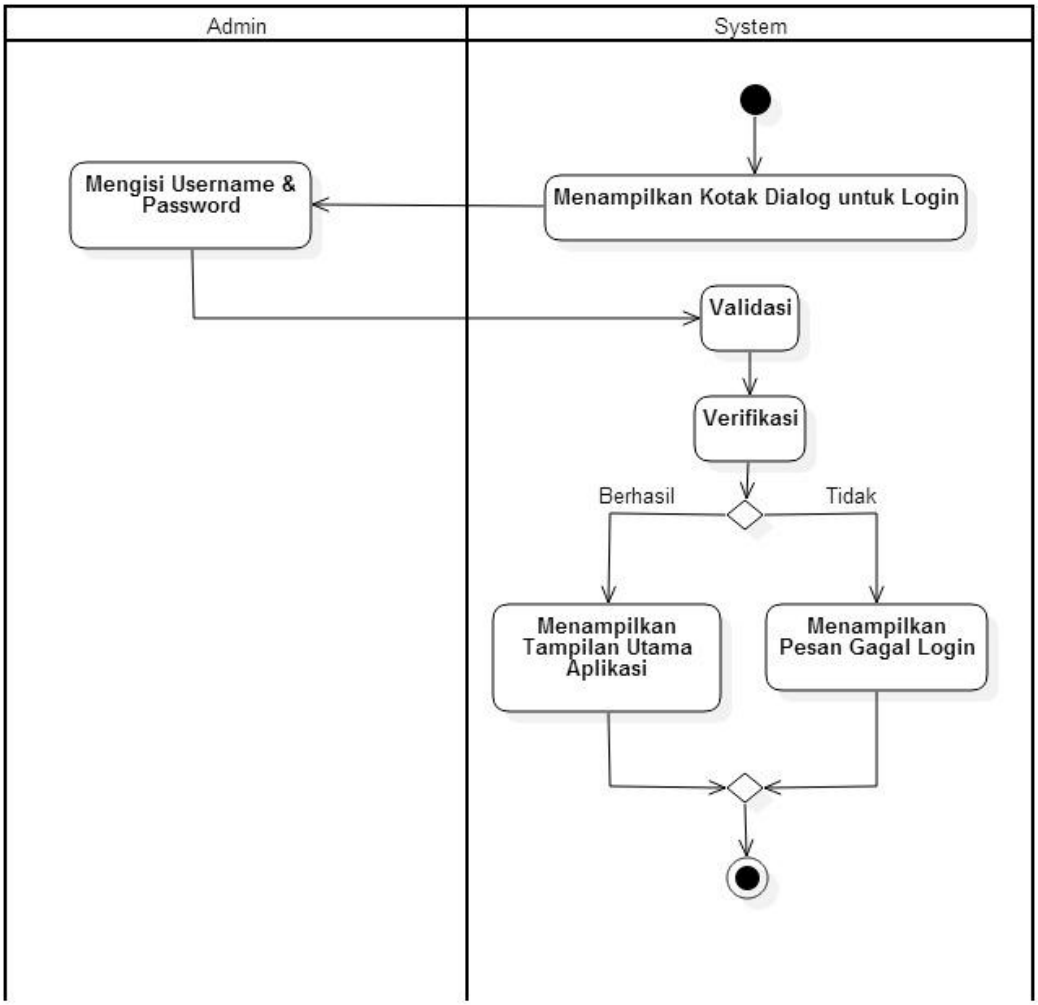
Tabel 3. 27 Skenario Kirim Pesan

Use Case Name	Kirim pesan	
Goal In Context	Mengirimkan pesan ke database	
Preconditions	Menampilkan tampilan awal	
Successful Condition	Berhasil mengirim pesan ke database	
Failed End Condition	-	
Primary Actor	Siswa	
Secondary Actor	-	
Trigger	Pengguna mengisi kontak	
Main Flow	***Step***	***Action***
	1	Pengguna mengisi kontak
	2	Pengguna memilih action kirim pesan
	3	Sistem mengirim pesan ke database

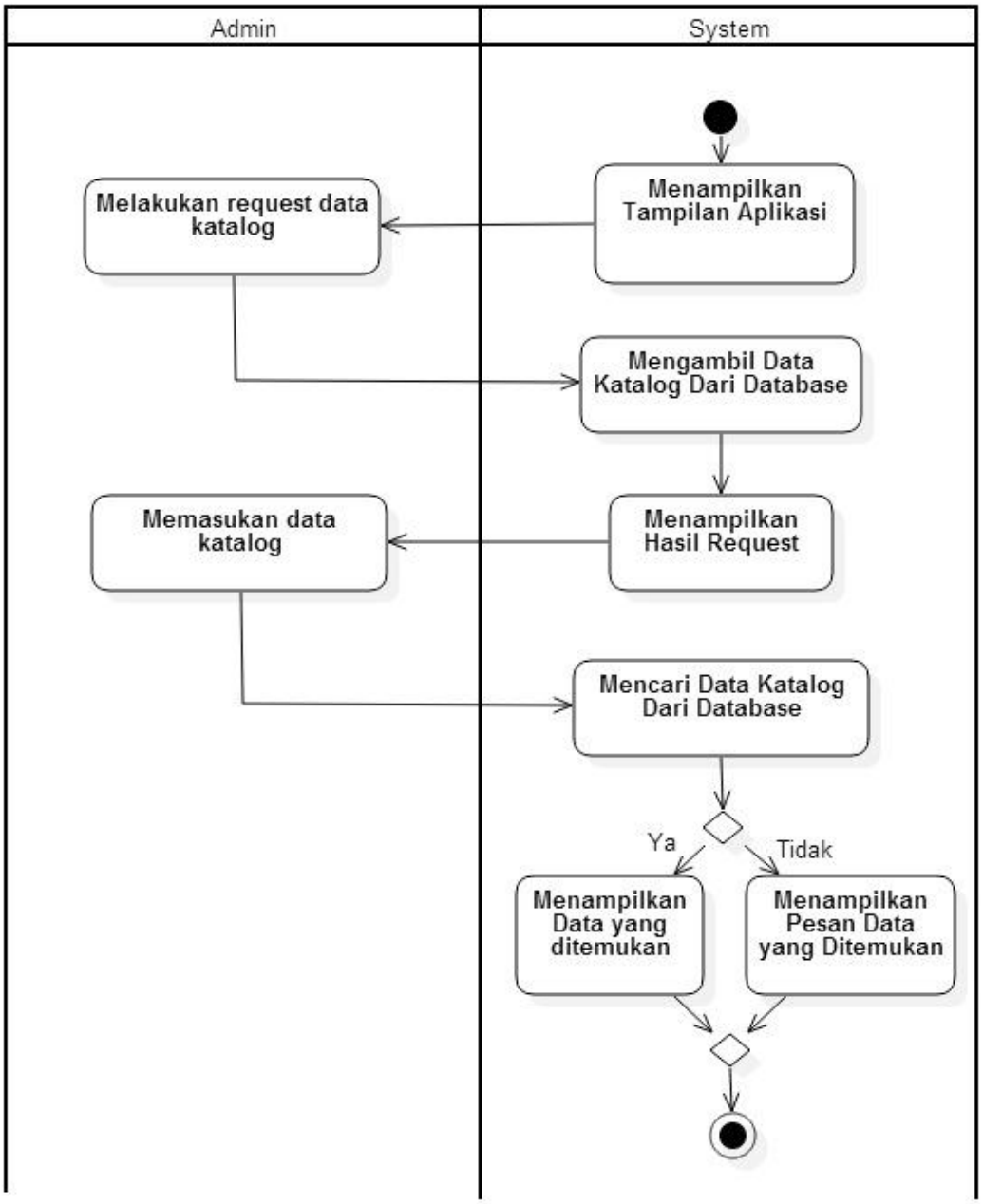
3.3.3 Activity Diagram

Activity diagram merupakan bagian dari pengembangan sistem secara fungsional menjelaskan fungsi-fungsi logika atau fungsi yang terimplementasi oleh kode program. *Activity diagram* memodelkan *event-*

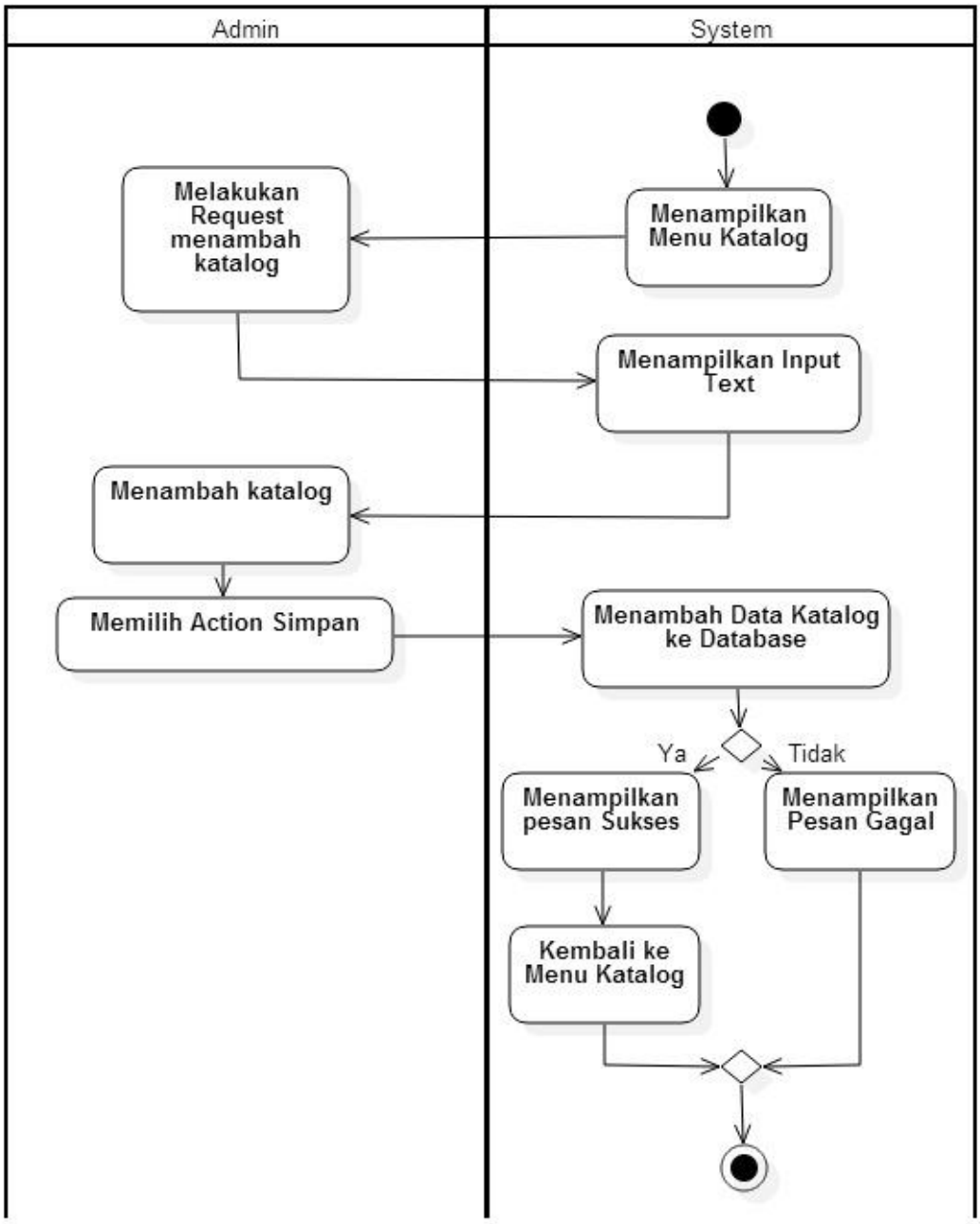
event yang tejadi didalam suatu *use case* dan digunakan untuk pemodelan aspek dinamisdari sistem.



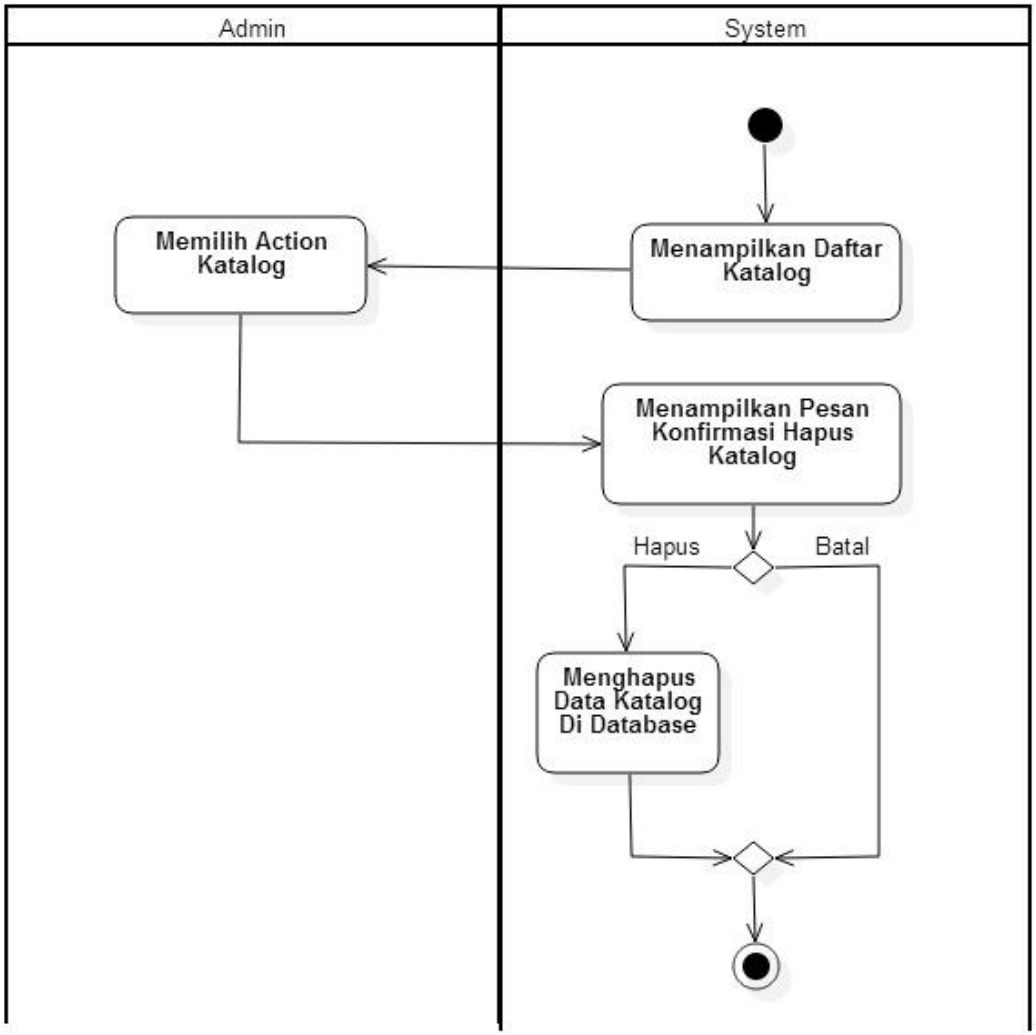
Gambar 3. 3 Activity Diagram Login



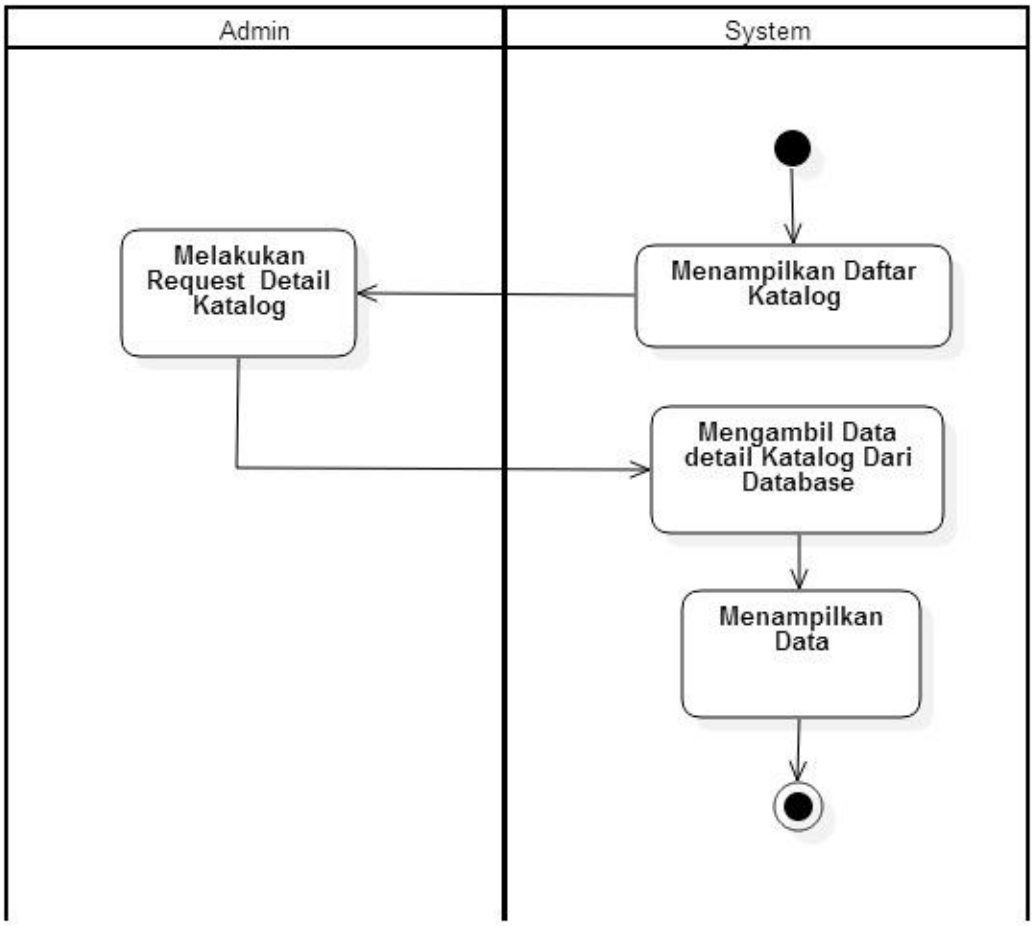
Gambar 3. 4 Activity Diagram Pencarian Katalog



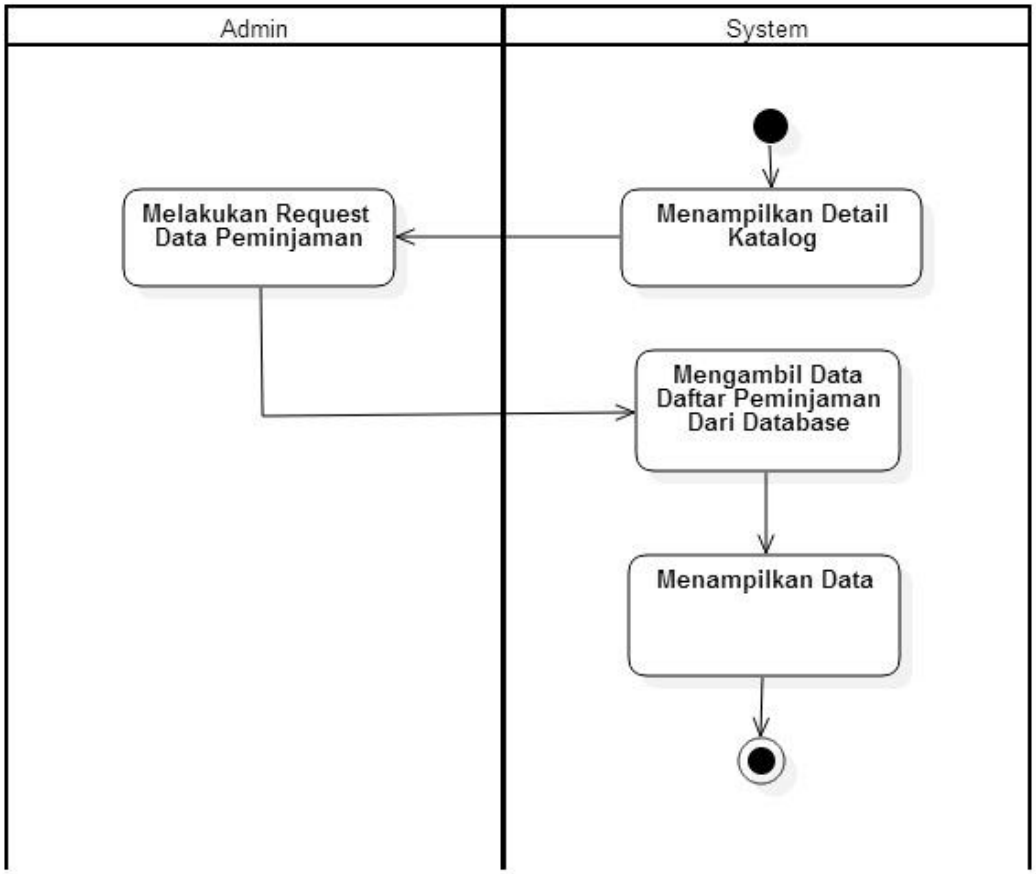
Gambar 3. 5 Activity Diagram Tambah Katalog



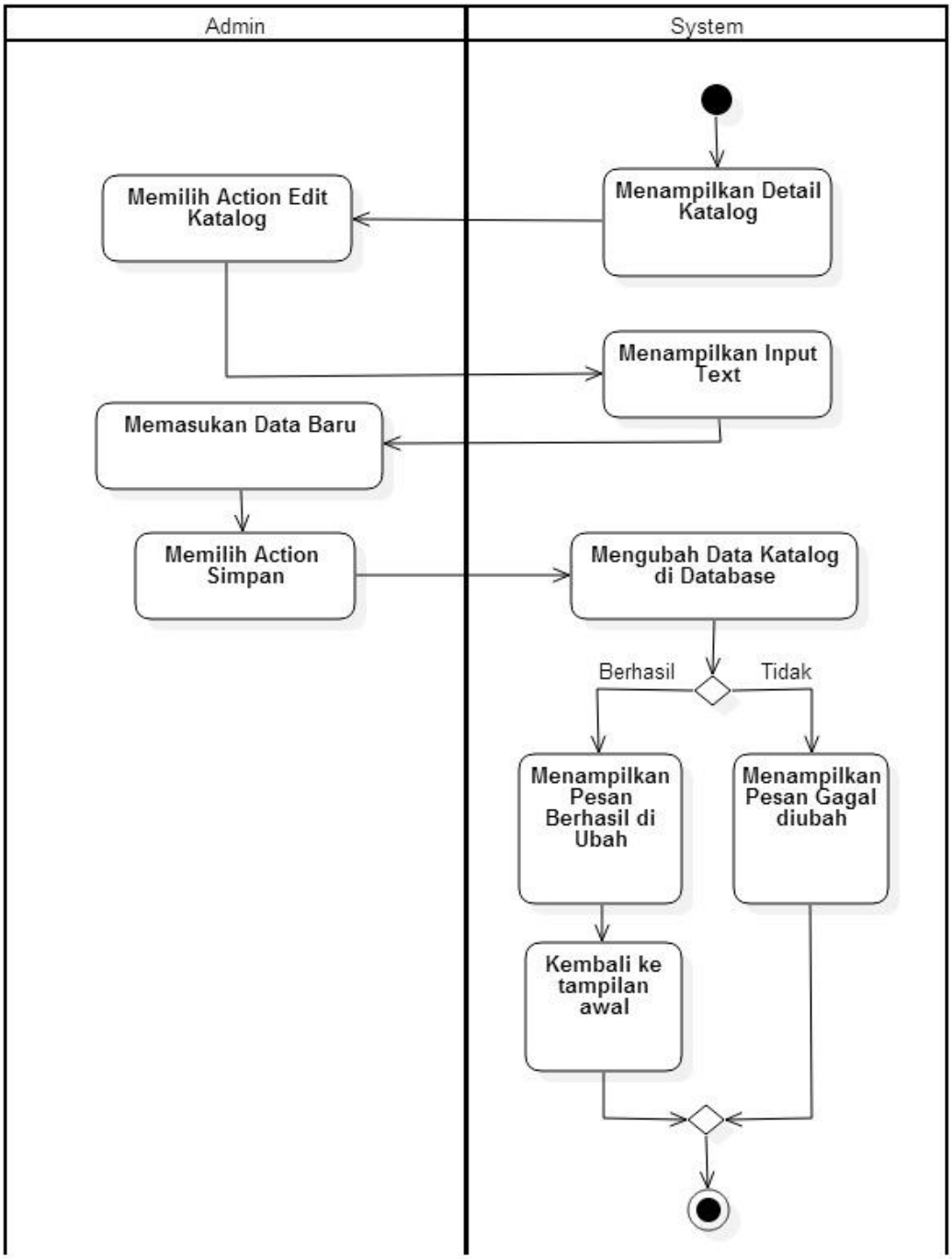
Gambar 3. 6 Activity Diagram Hapus Katalog



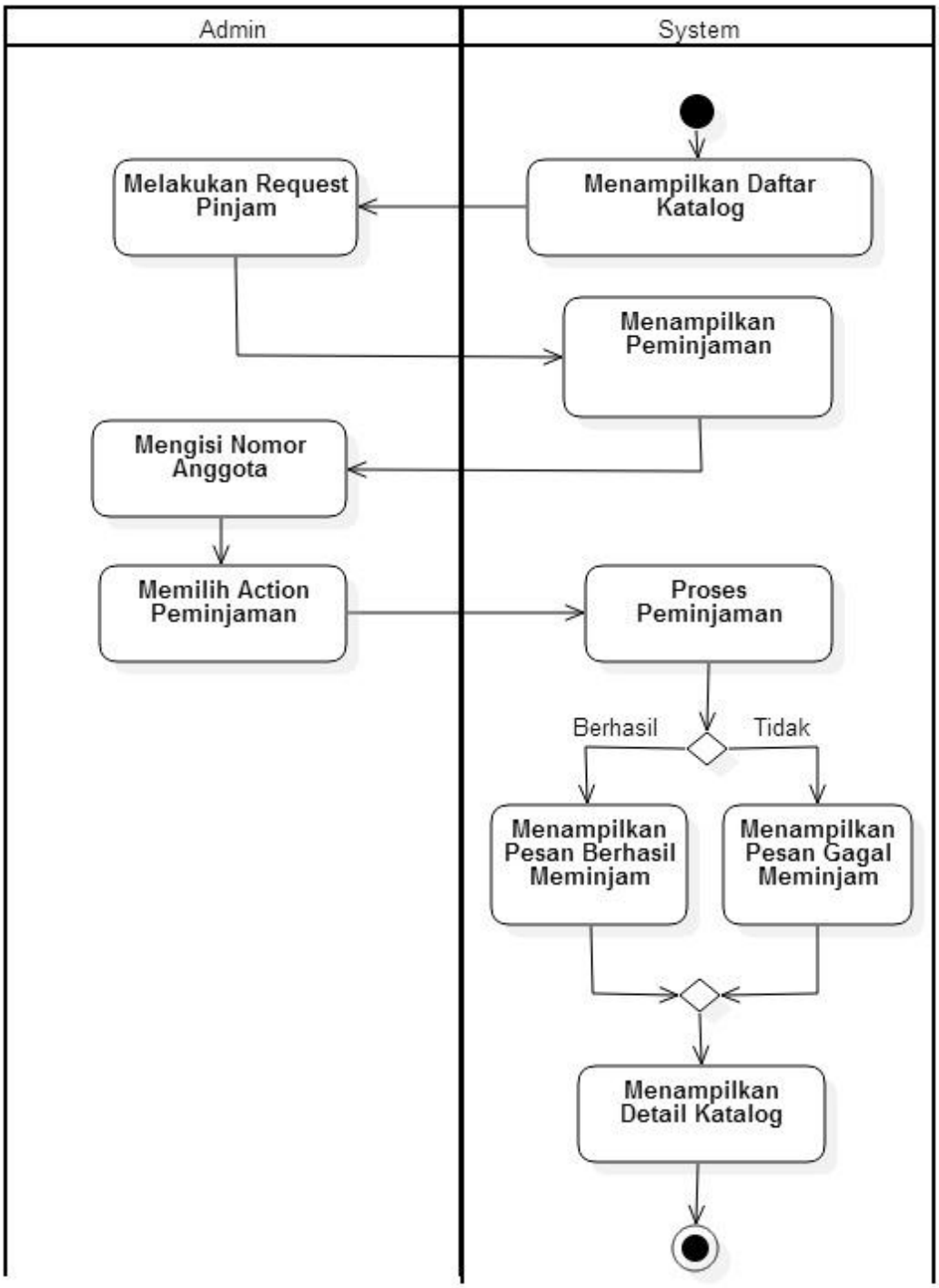
Gambar 3. 7 Activity Diagram Detail Katalog



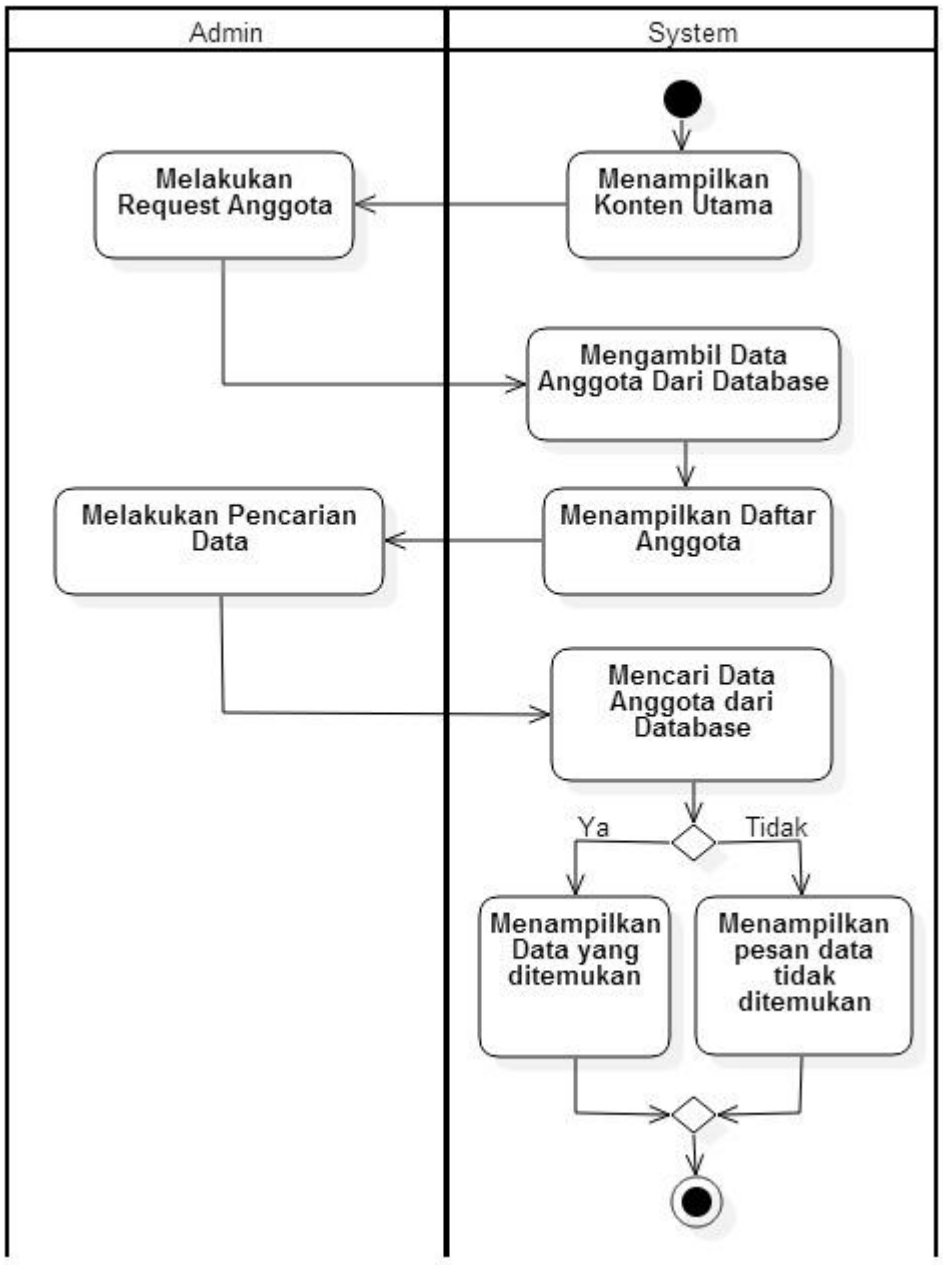
Gambar 3. 8 Activity Diagram Daftar Peminjaman



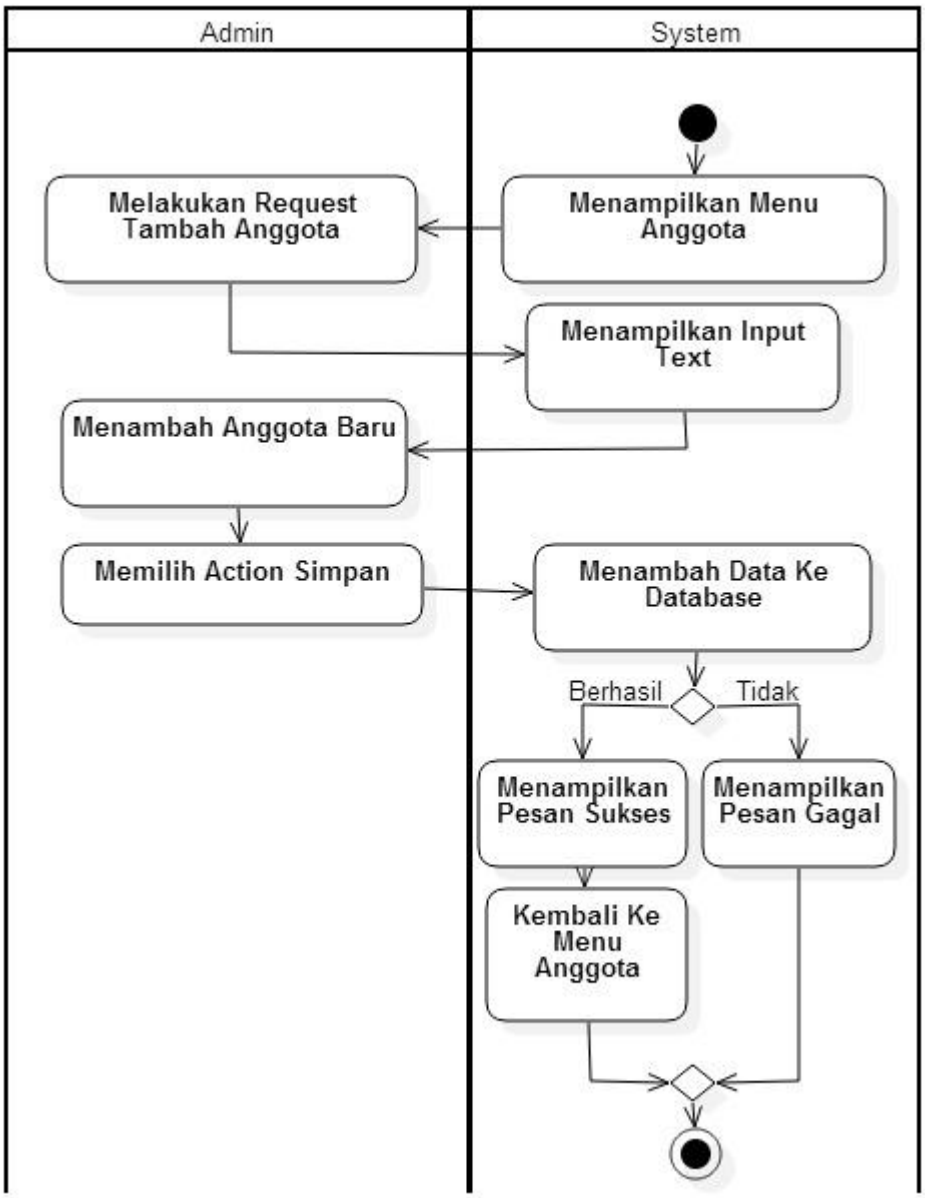
Gambar 3. 9 Acitivity Diagram Edit Katalog



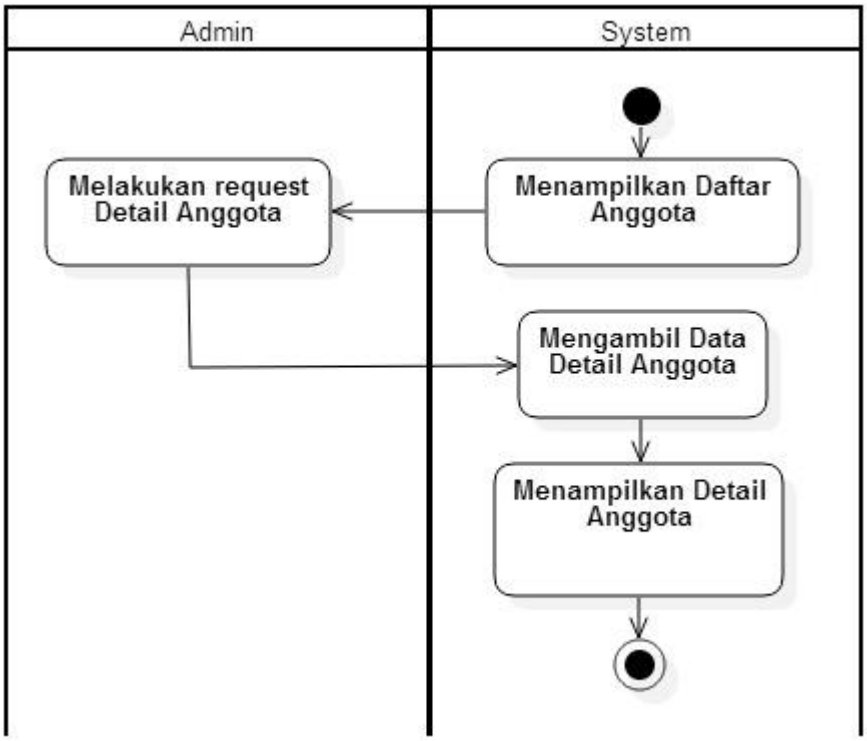
Gambar 3. 10 Acitivity Diagram Proses Peminjaman



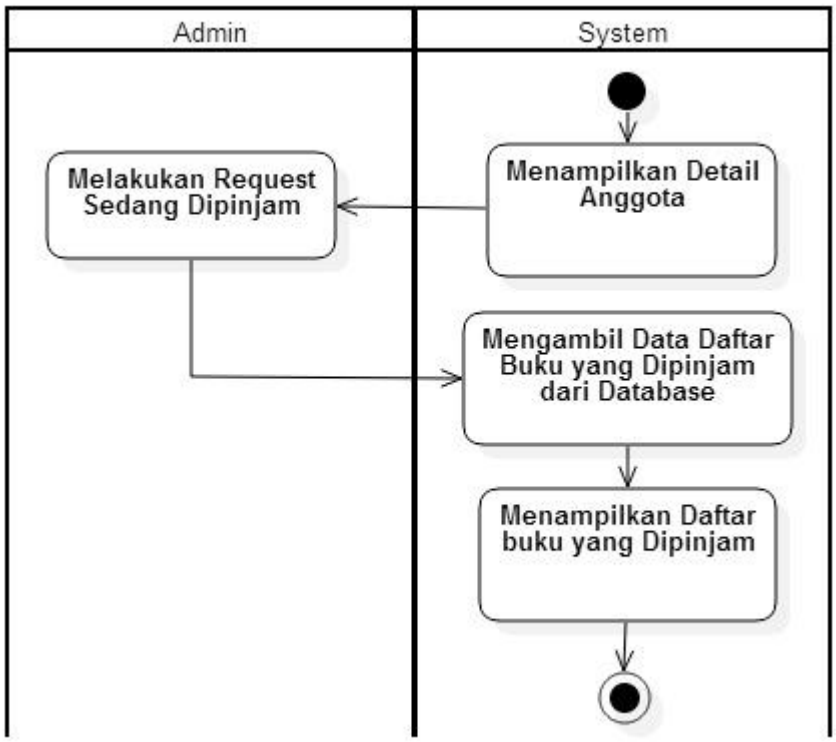
Gambar 3. 11 Acitivity Diagram Pencarian Anggota



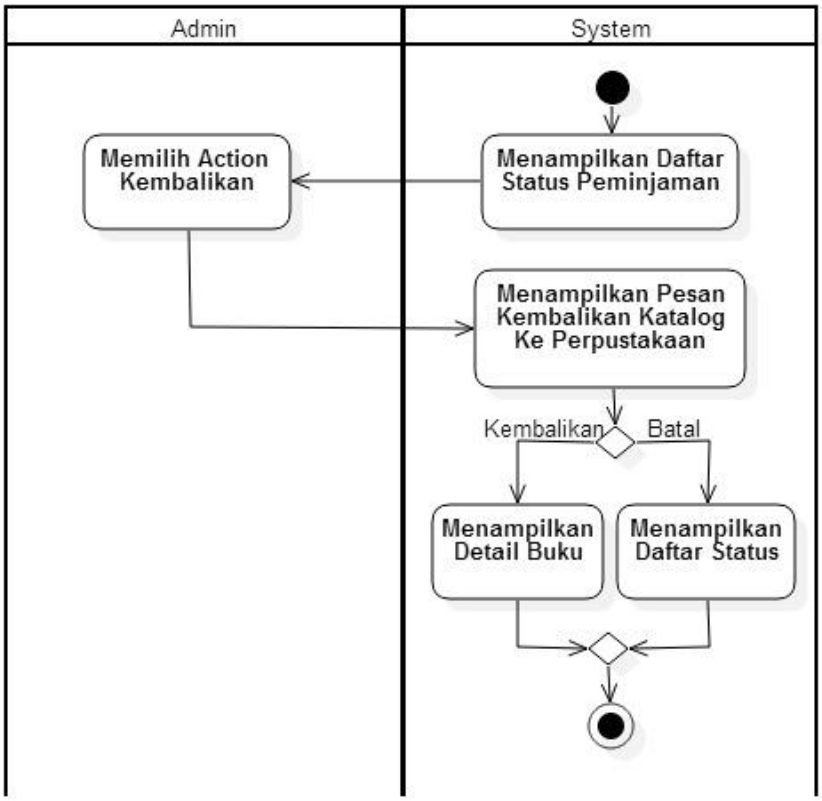
Gambar 3. 12 Acitivity Diagram Tambah Anggota



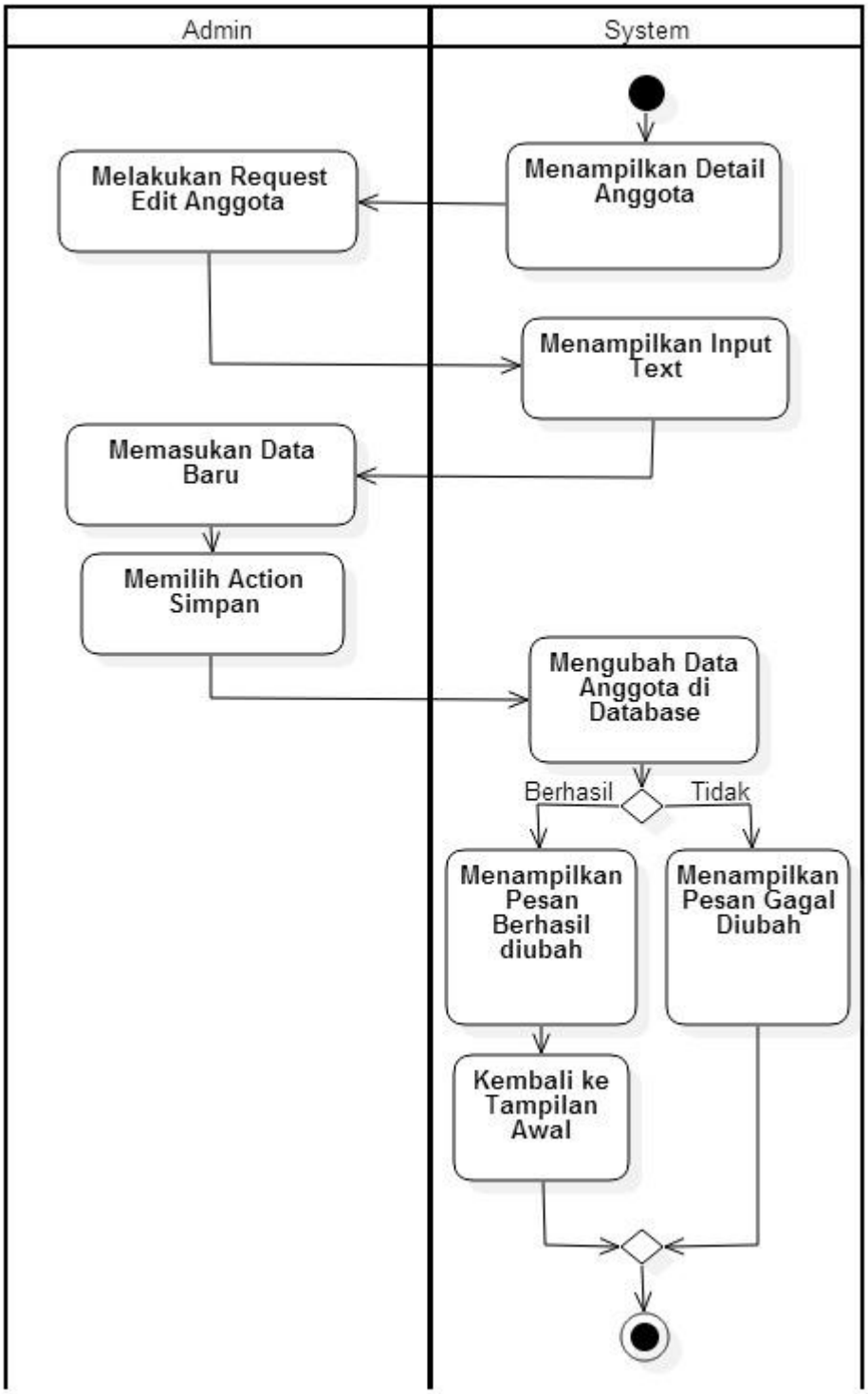
Gambar 3. 13 Acitivity Diagram Detail Anggota



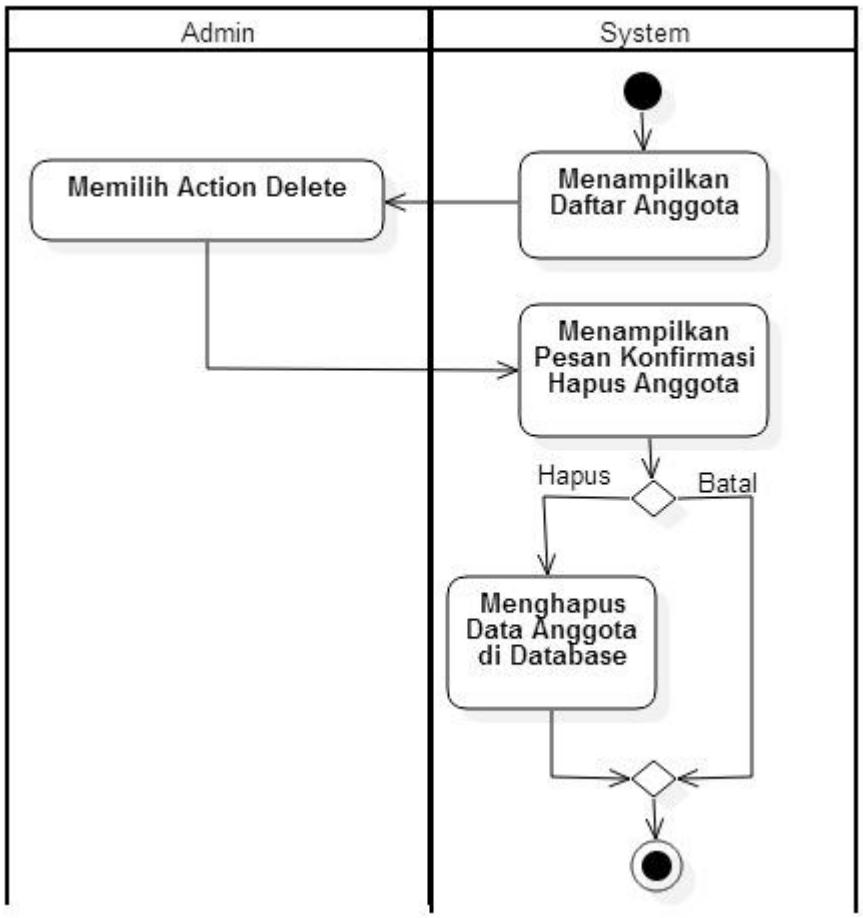
Gambar 3. 14 Acitivity Diagram Daftar Buku Pinjam



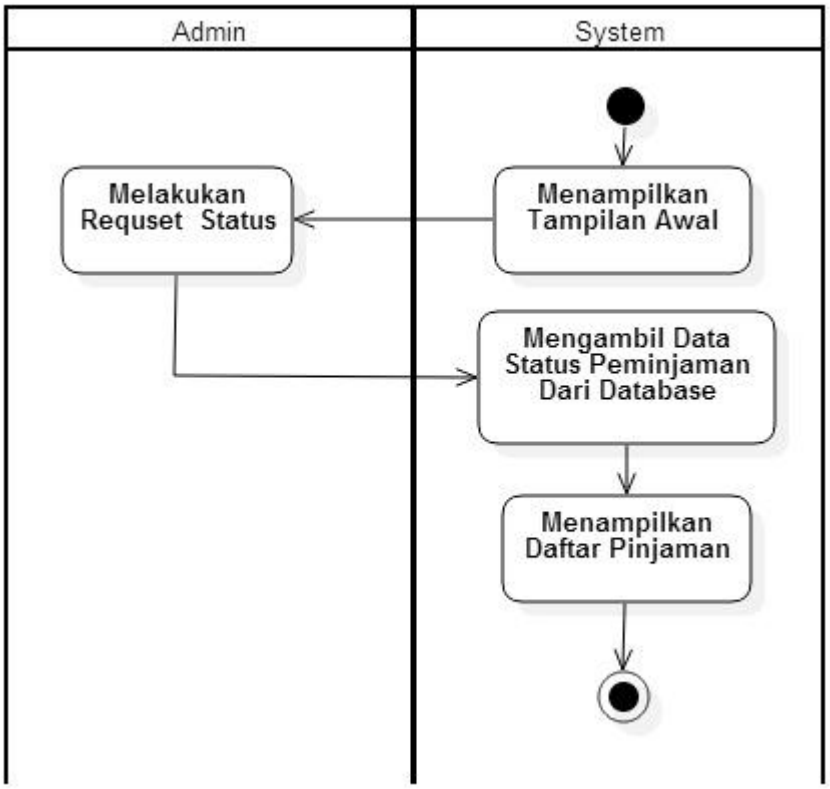
Gambar 3. 15 Acitivity Diagram Riwayat Pengembalian



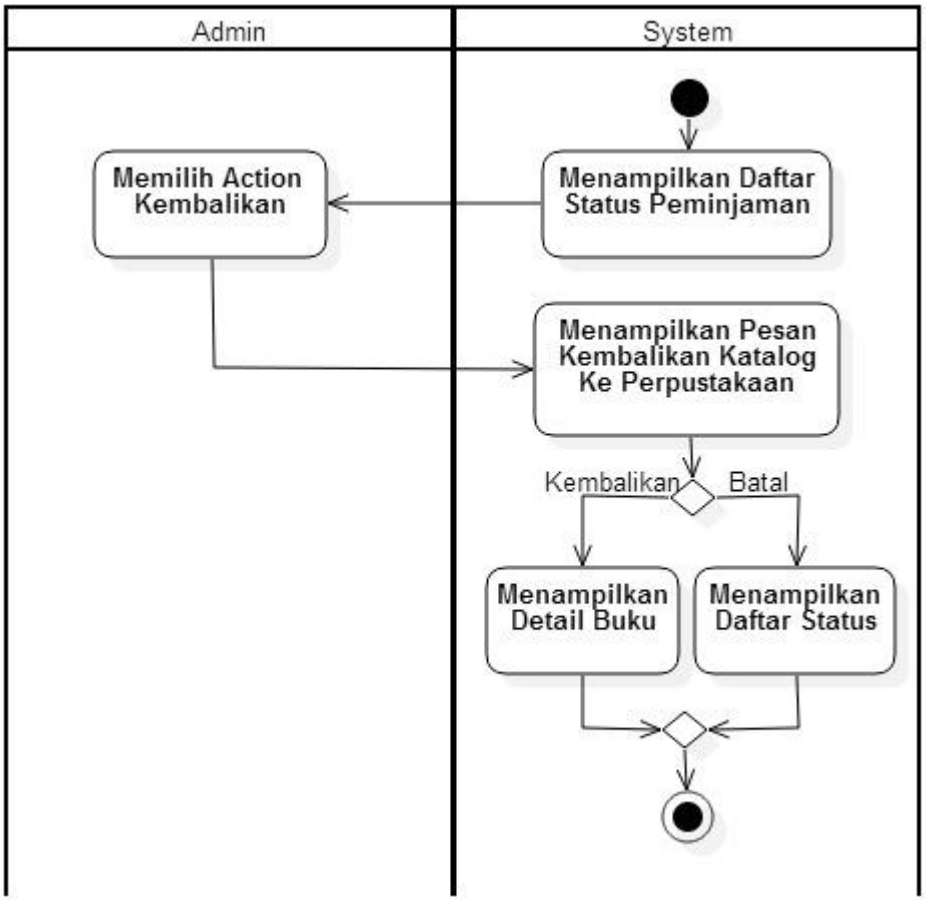
Gambar 3. 16 Acitivity Diagram Edit Anggota



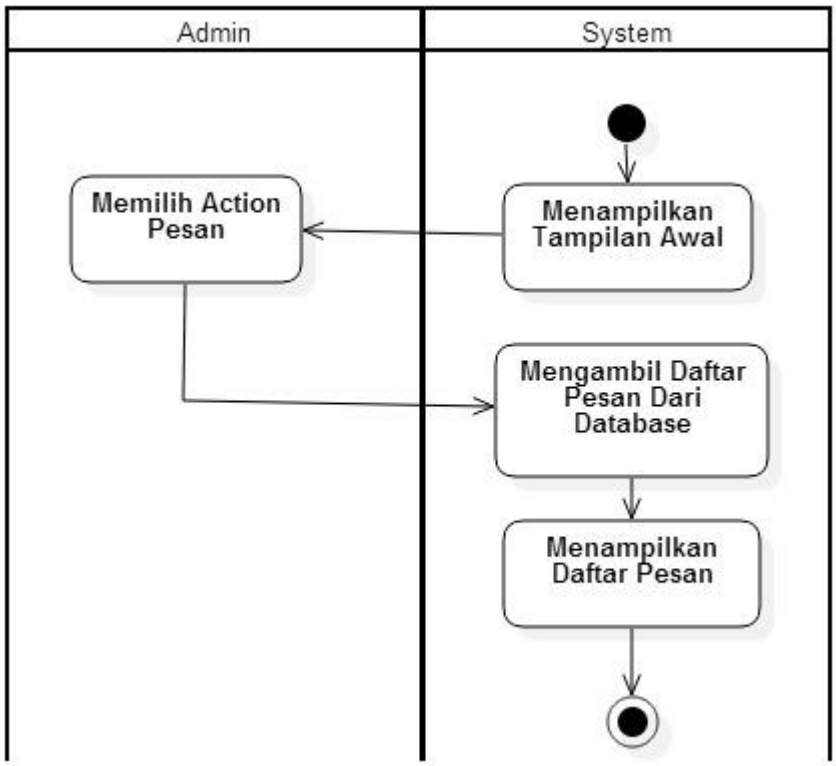
Gambar 3. 17 Acitivity Diagram Hapus Anggota



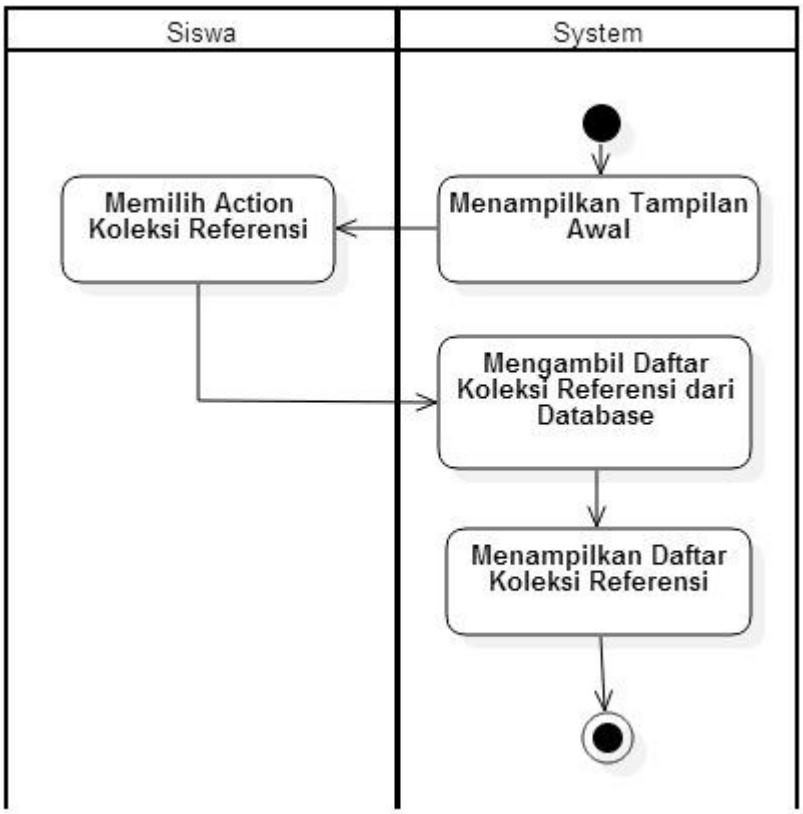
Gambar 3. 18 Acitivity Diagram Data Peminjaman



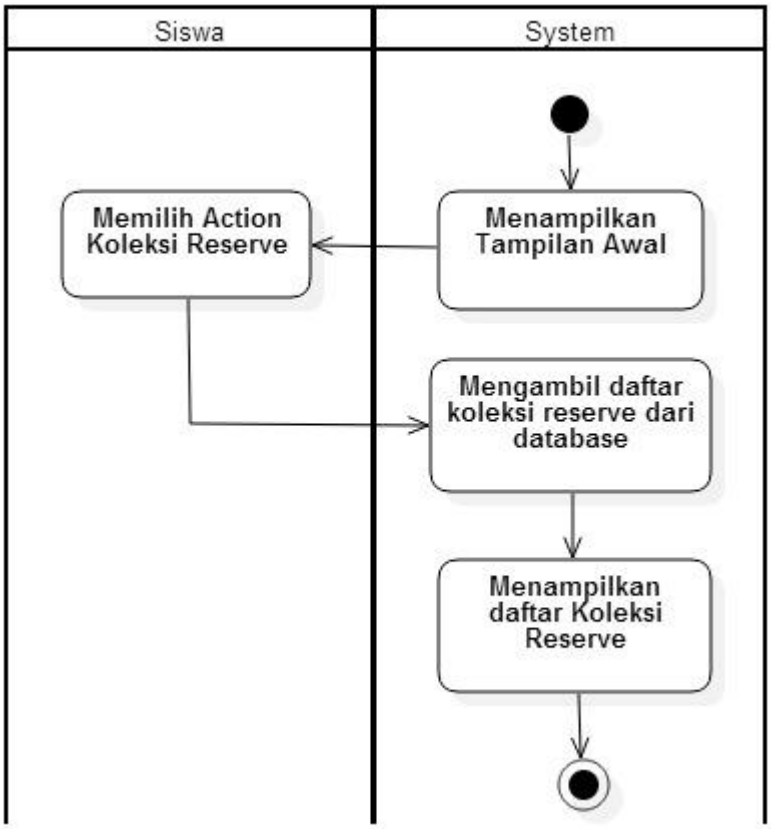
Gambar 3. 19 Acitivity Diagram Proses Pengembalian



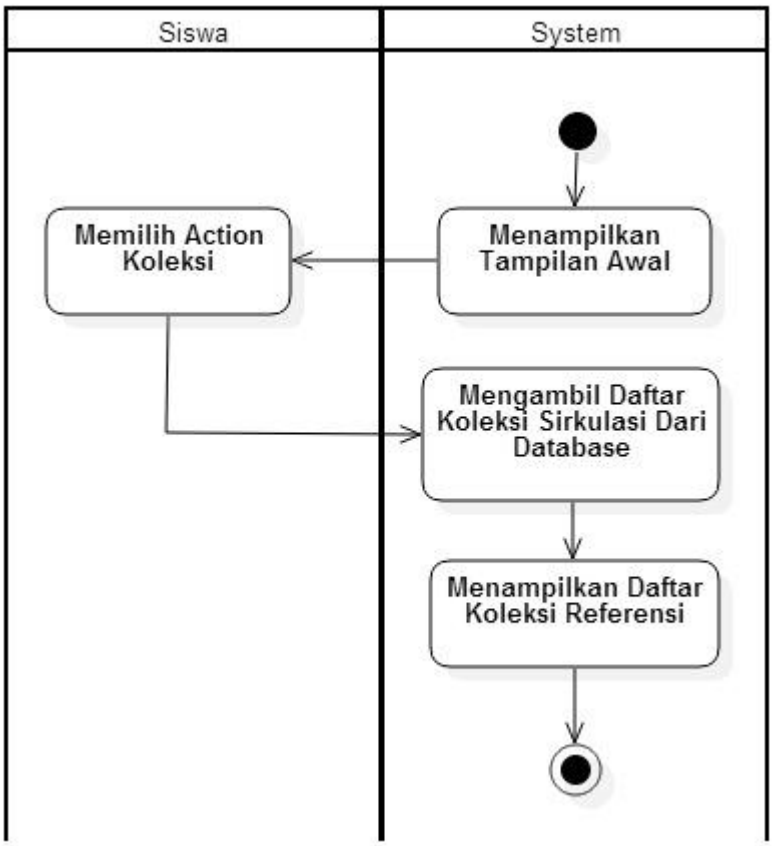
Gambar 3. 20 Acitivity Diagram Lihat Pesan



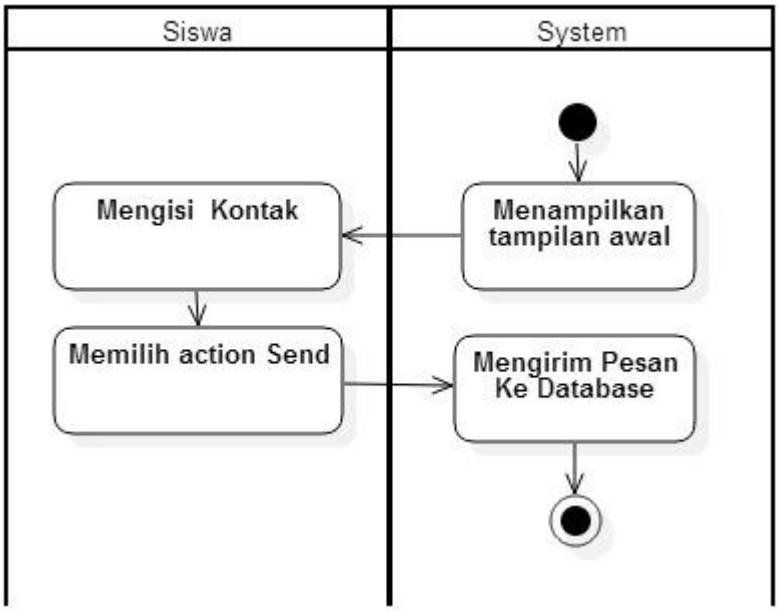
Gambar 3. 21 Acitivity Diagram Pencarian Koleksi Referensi



Gambar 3. 22 Acitivity Diagram Koleksi Reverse



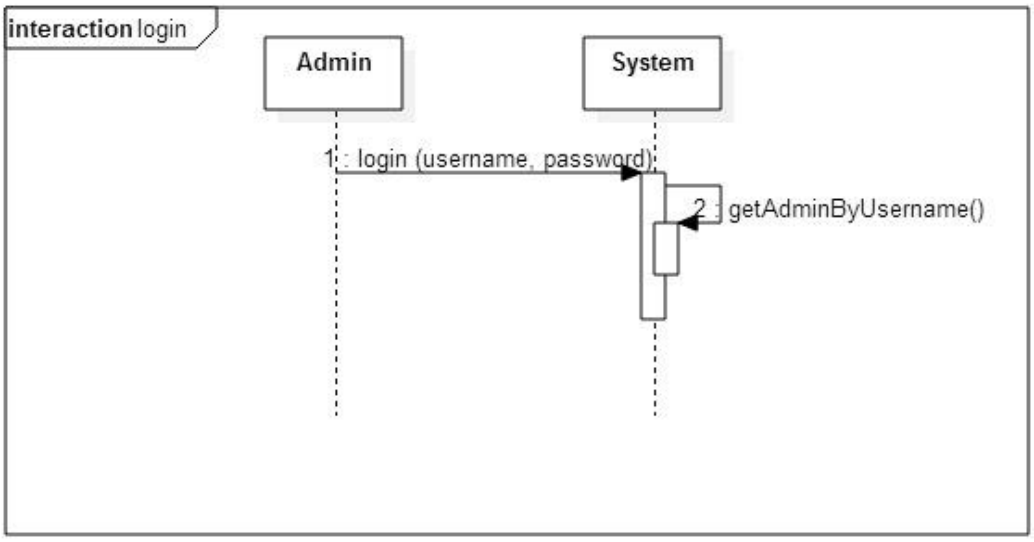
Gambar 3. 23 Acitivity Diagram Koleksi Sirkulasi



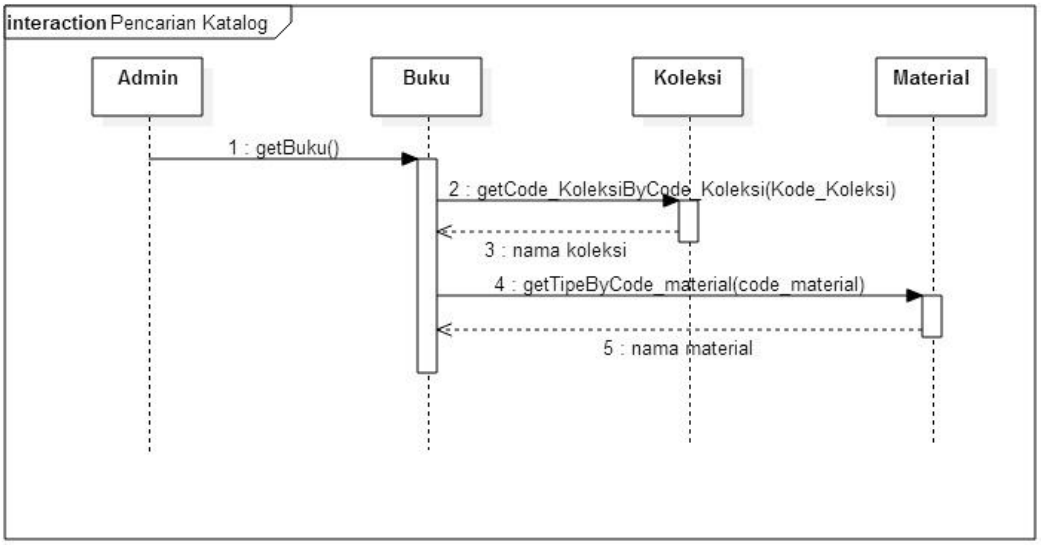
Gambar 3. 24 Acitivity Diagram Kirim Pesan

3.3.4 Sequence Diagram

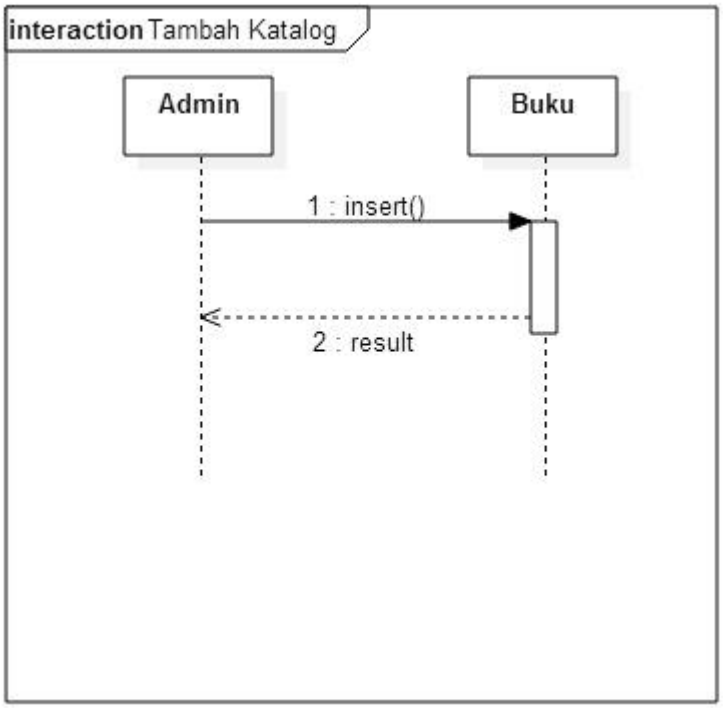
Sequence Diagram menggambarkan interaksi antar masing masing objek pada setiap *Use Case* dalam runtutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek yang saling berinteraksi. *Sequence Diagram* disajikan mulai gambar 3.21 sampai gambar 3.30.



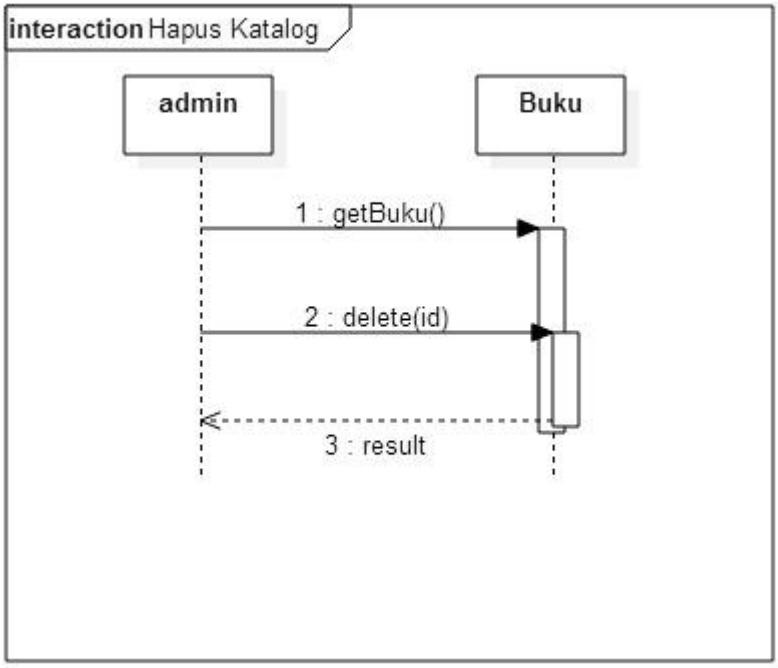
Gambar 3. 25 Sequance Diagram Login



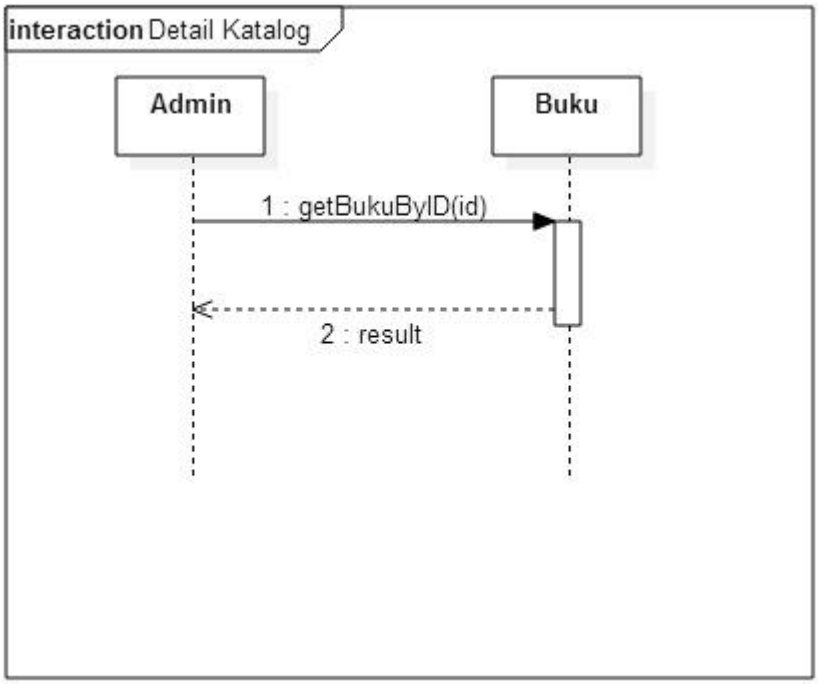
Gambar 3. 26 Sequance Diagram Pencarian Katalog



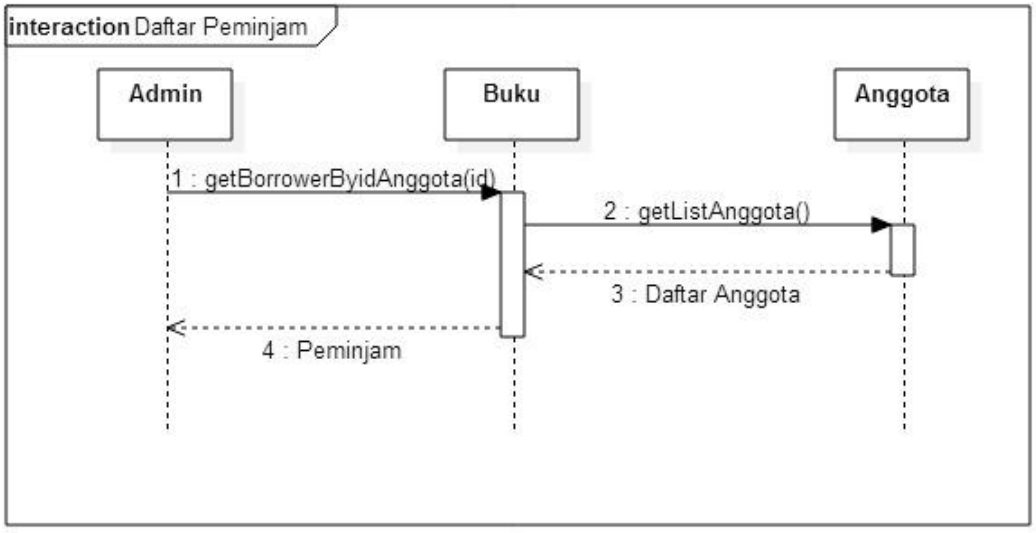
Gambar 3. 27 Sequance Diagram Tambah Katalog



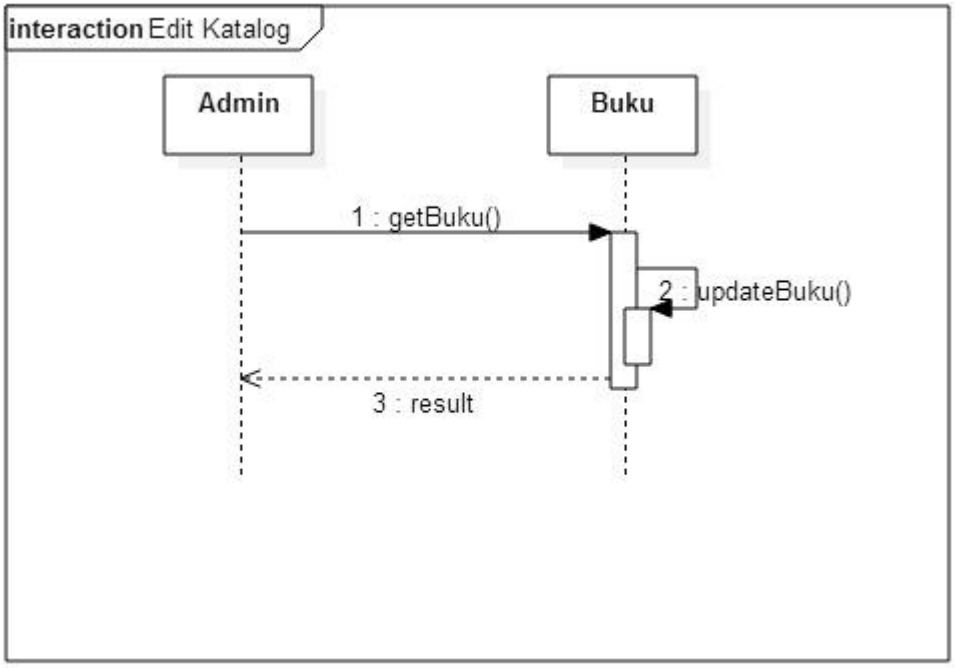
Gambar 3. 28 Sequance Diagram Hapus Katalog



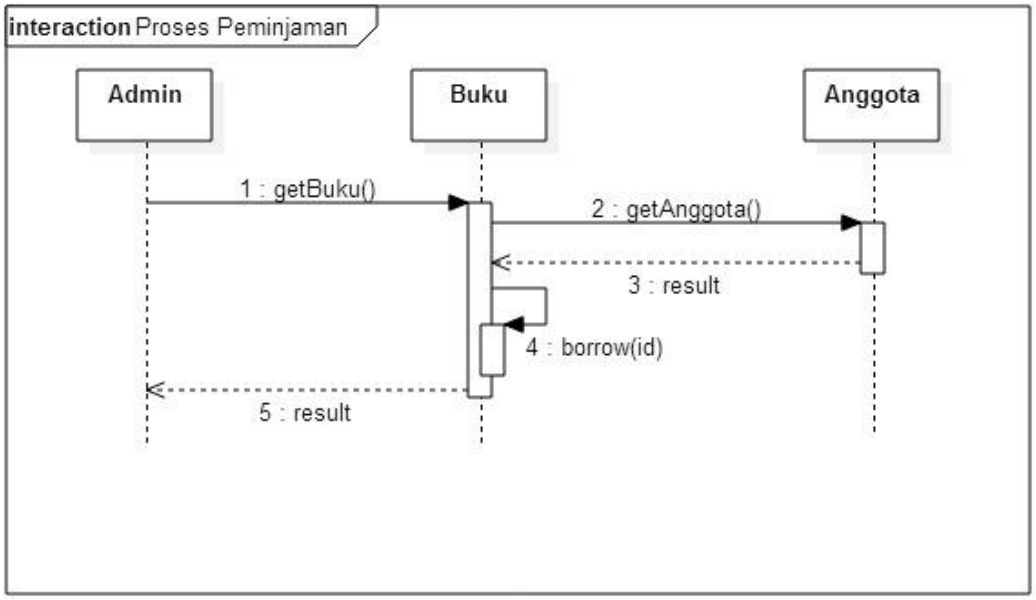
Gambar 3. 29 Sequance Diagram Detail Katalog



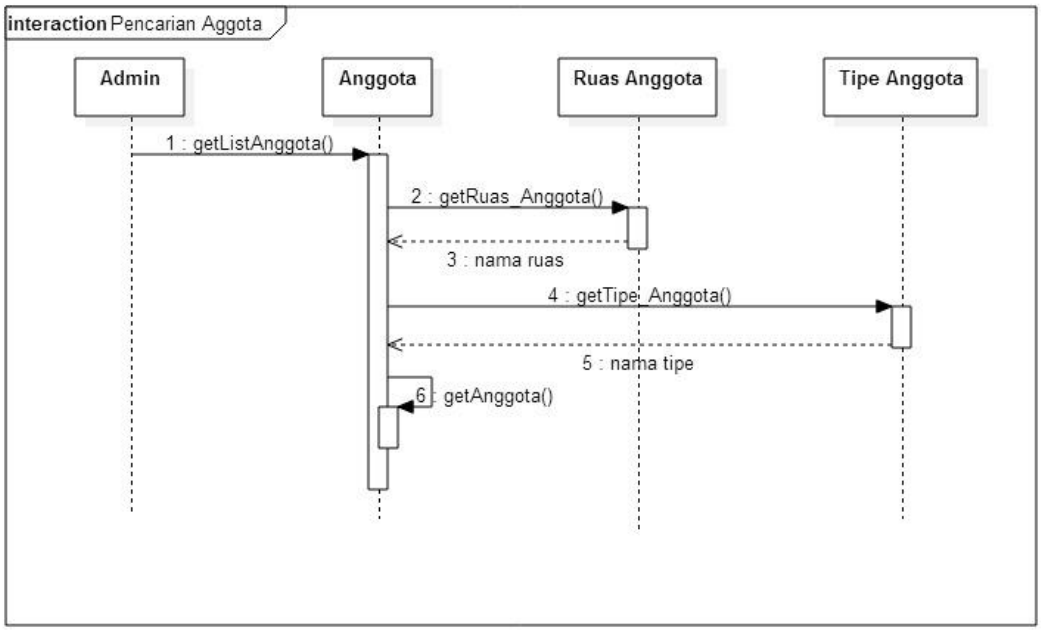
Gambar 3. 30 Sequance Diagram Daftar Peminjaman



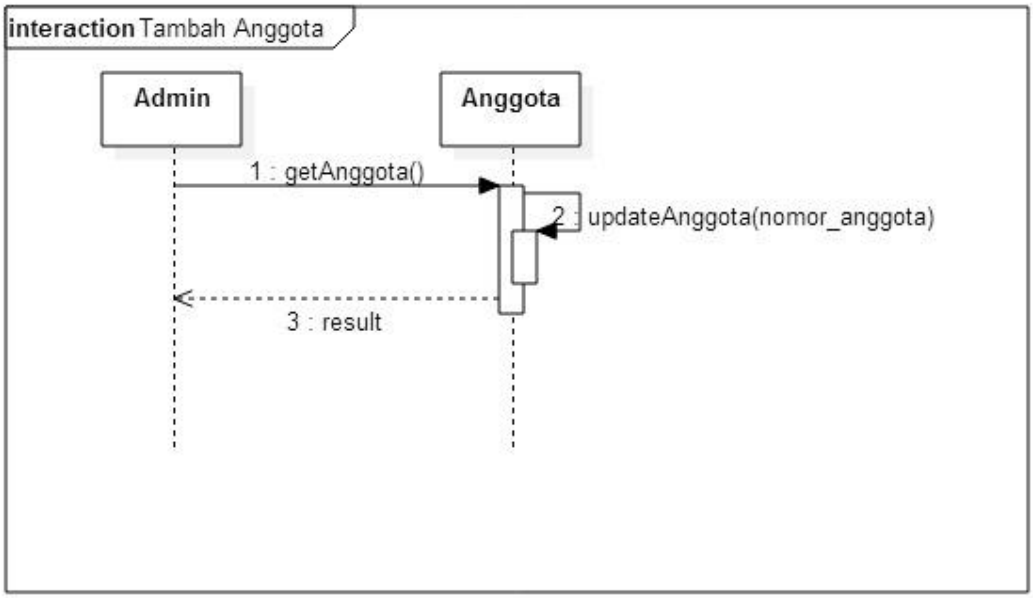
Gambar 3. 31 Sequance Diagram Edit Katalog



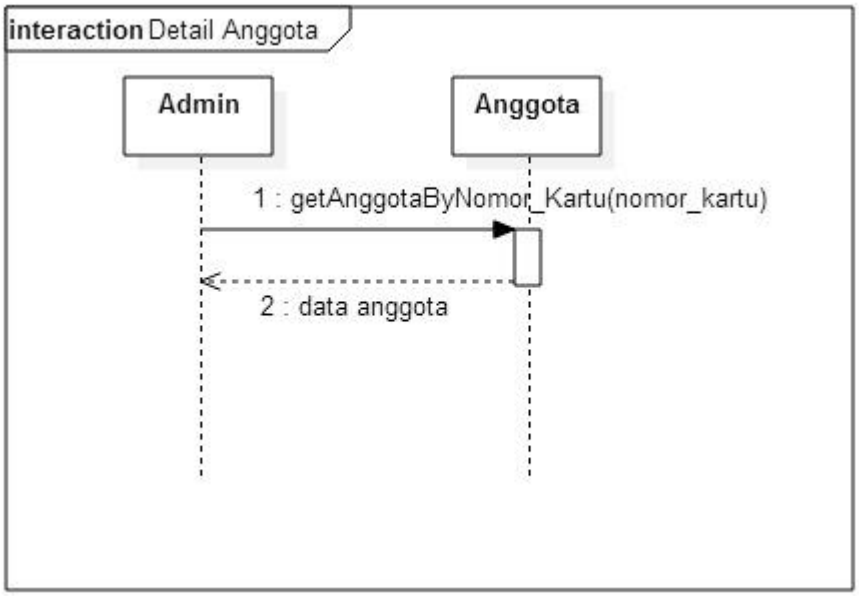
Gambar 3. 32 Sequance Diagram Proses Peminjaman



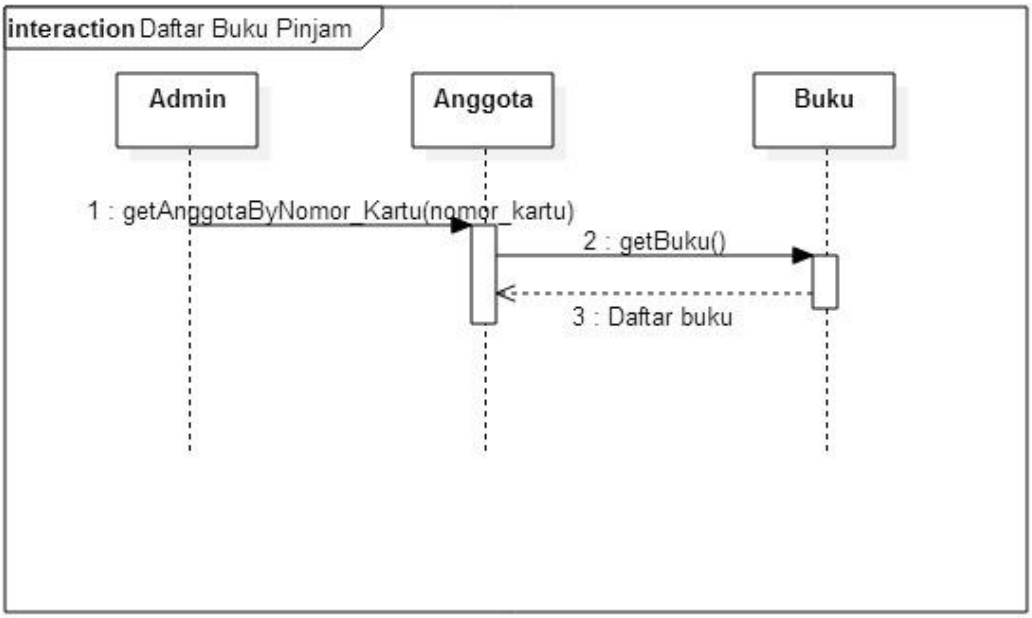
Gambar 3. 33 Sequance Diagram Pencarian Anggota



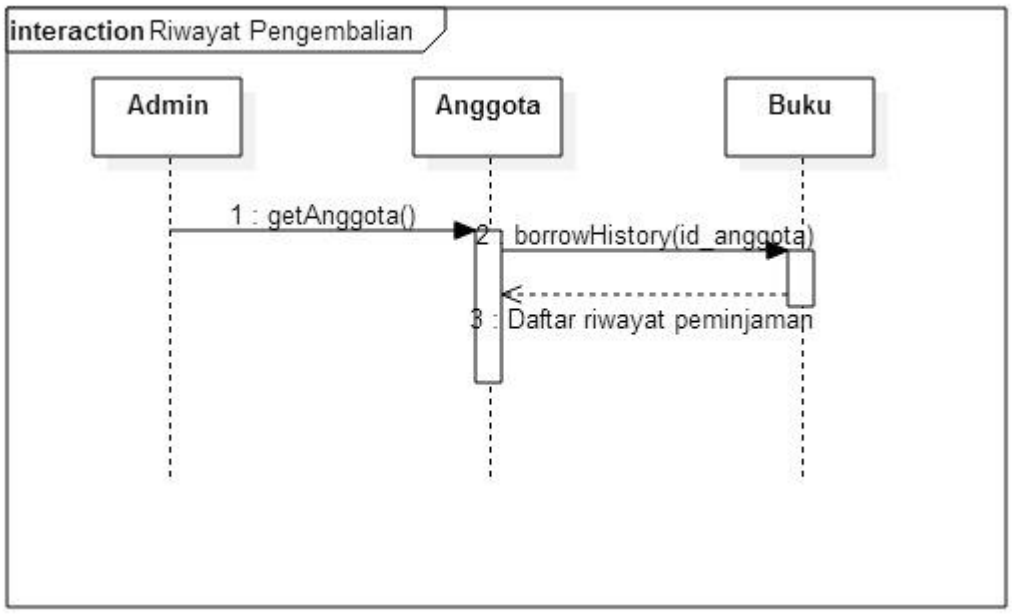
Gambar 3. 34 Sequance Diagram Tambah Anggota



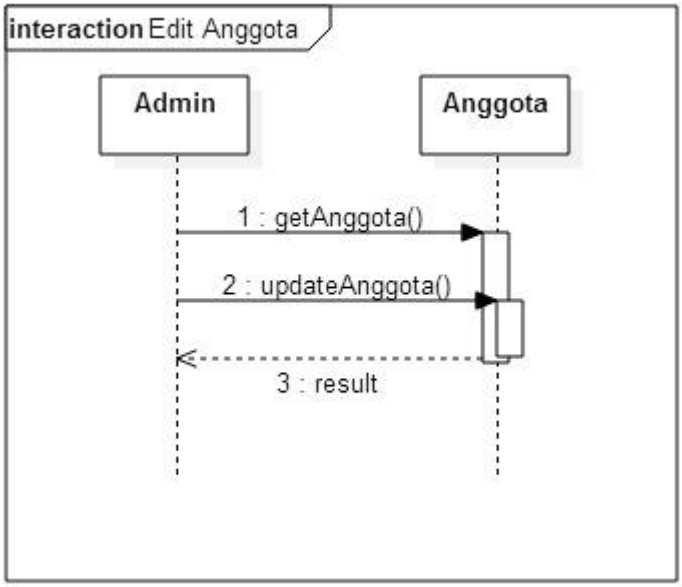
Gambar 3. 35 Sequance Diagram Detail Anggota



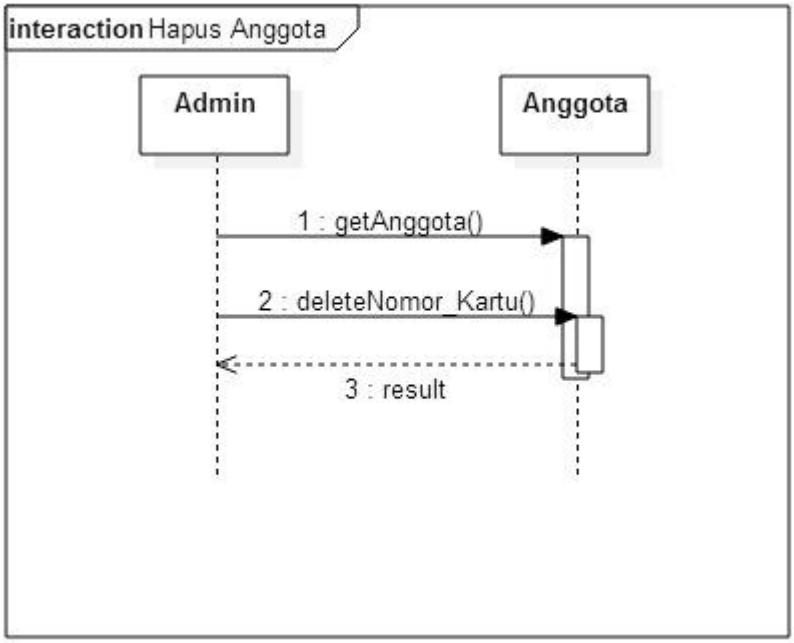
Gambar 3. 36 Sequance Diagram Daftar Buku Pinjam



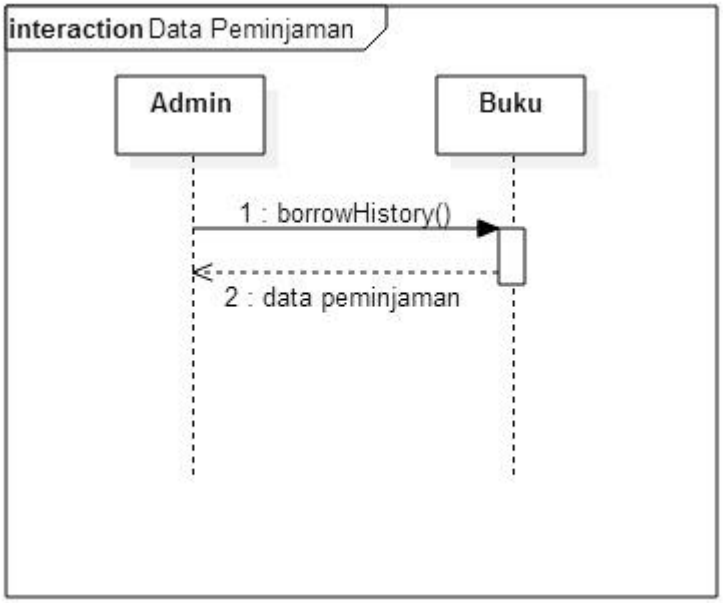
Gambar 3. 37 Sequance Diagram Riwayat Pengembalian



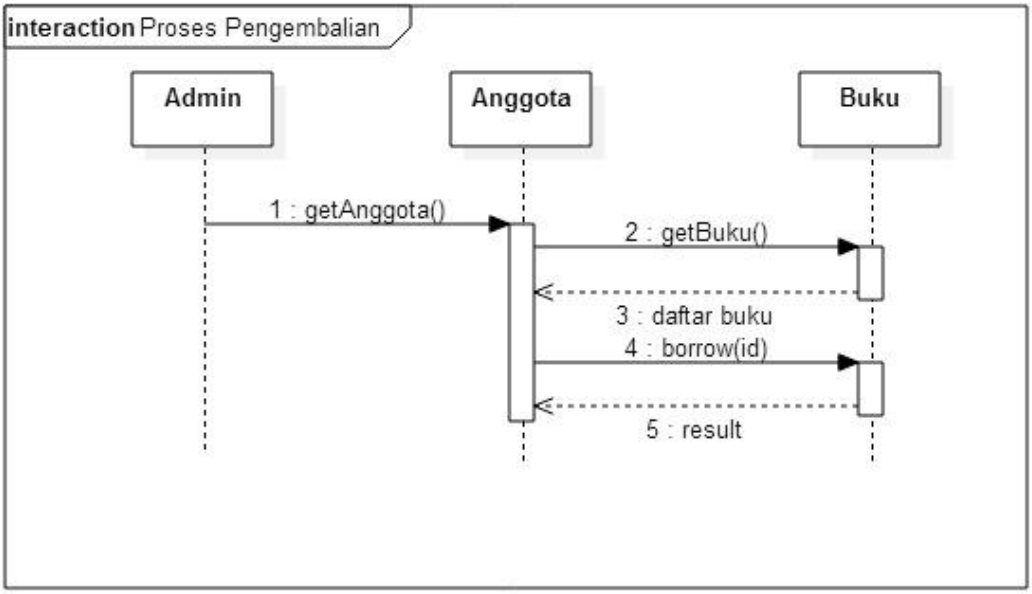
Gambar 3. 38 Sequance Diagram Edit Anggota



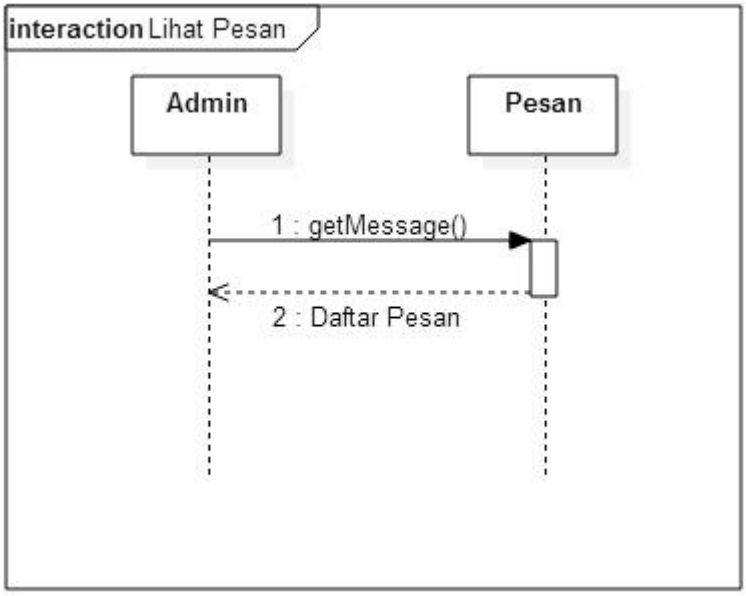
Gambar 3. 39 Sequance Diagram Hapus Anggota



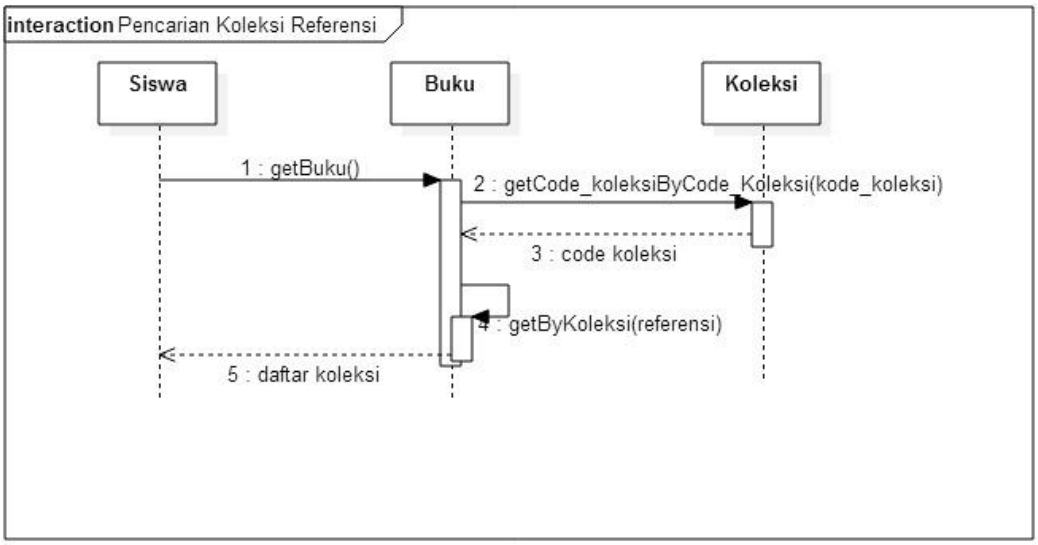
Gambar 3. 40 Sequance Diagram Data Peminjaman



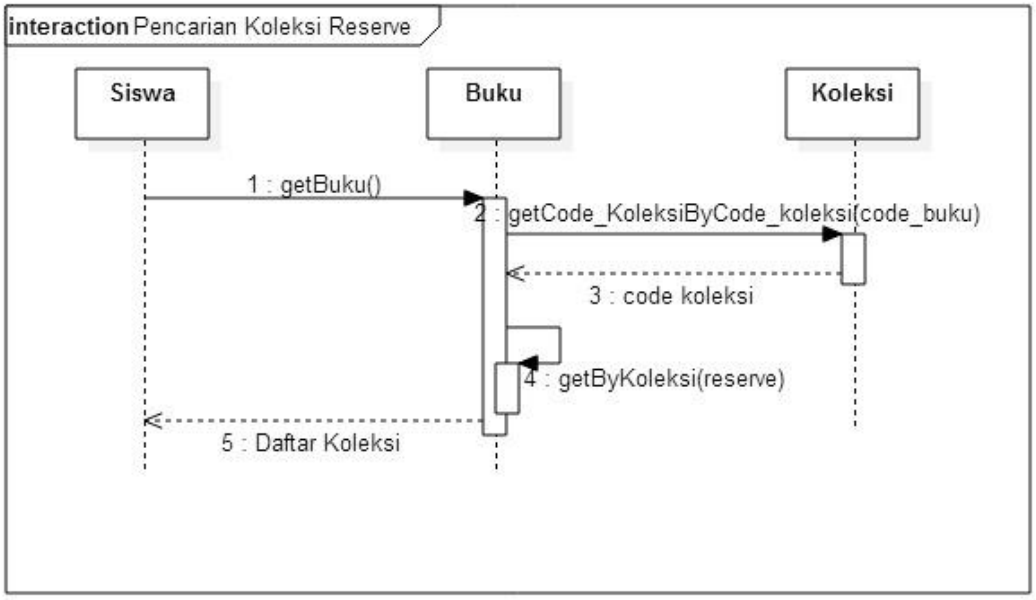
Gambar 3. 41 Sequance Diagram Proses Pengambilan



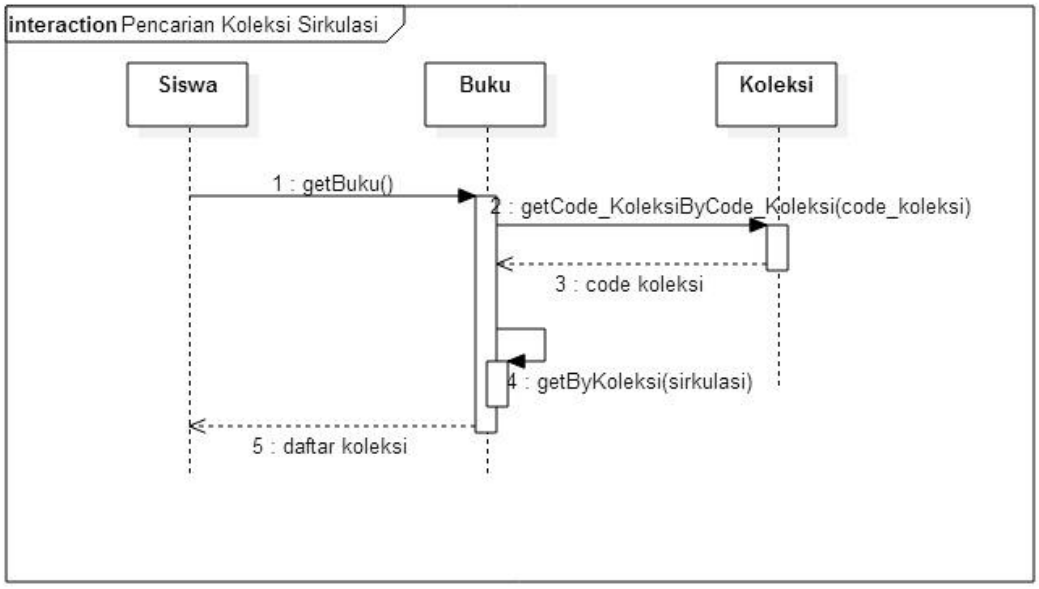
Gambar 3. 42 Sequance Diagram Lihat Pesan



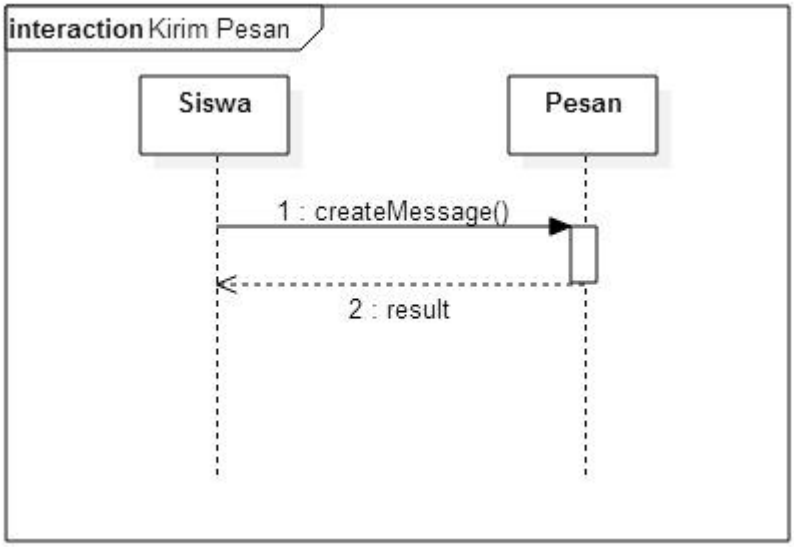
Gambar 3. 43 Sequance Diagram Pencarian Koleksi Referensi



Gambar 3. 44 Sequance Diagram Pencarian Koleksi Reverse



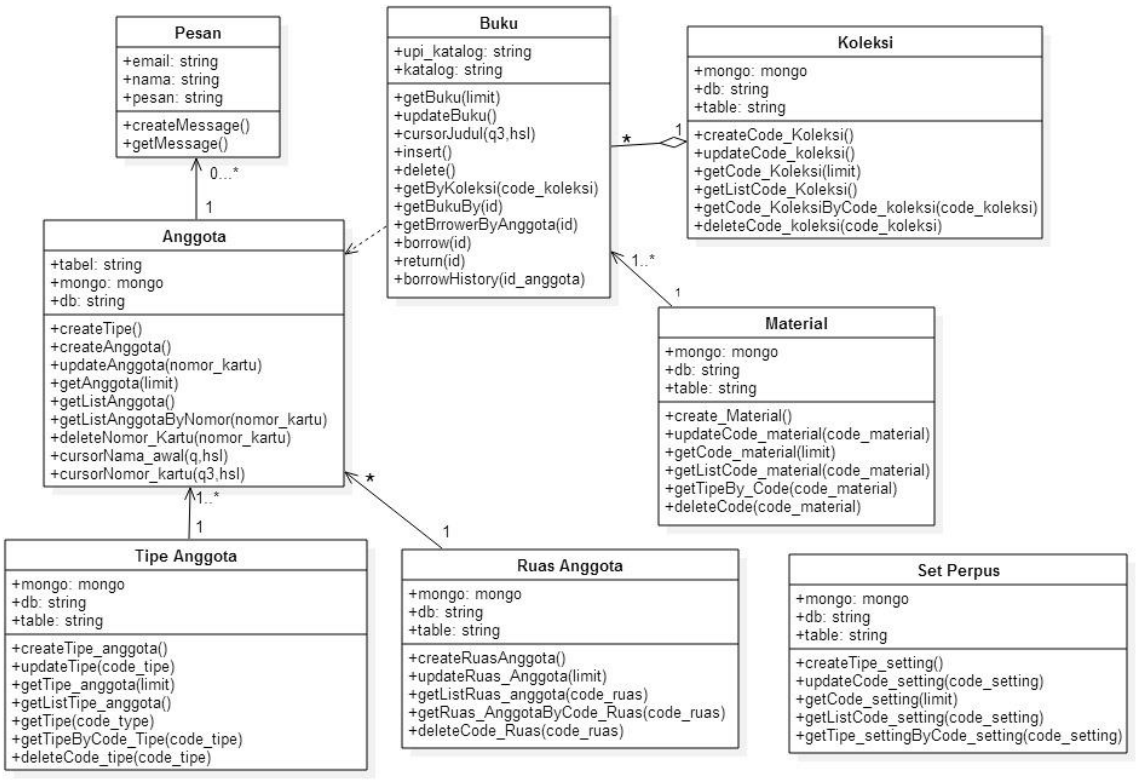
Gambar 3. 45 Sequance Diagram Pencarian Koleksi Sirkulasi



Gambar 3. 46 Sequance Diagram Kirim Pesan

3.3.5 Class Diagram

Class Diagram menggambarkan struktur dan hubungan antar objek yang ada pada sistem. Struktur tersebut meliputi struktur dan method yang ada pada masing masing *class*. *Class diagram* disajikan pada gambar 3.31.



Gambar 3. 47 Class Diagram

3.4 Perancangan Sistem

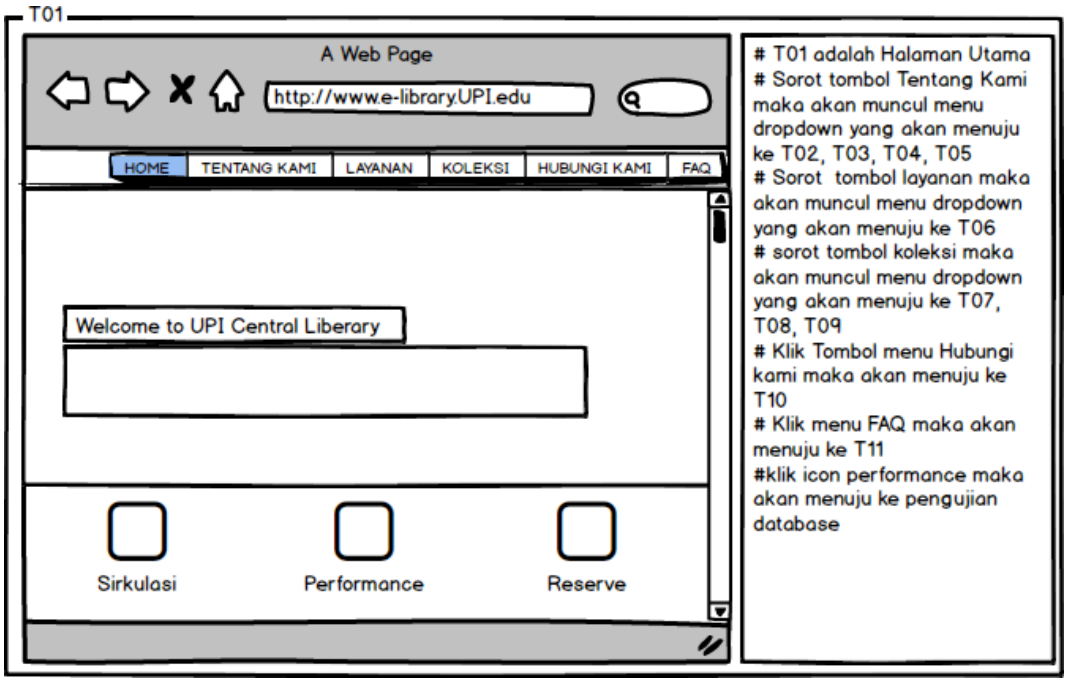
Perancangan sistem merupakan kegiatan untuk merancang aplikasi yang akan dibangun, tahapannya mulai dari perancangan antar muka dan jaringan semantik

3.4.1 Perancangan Antarmuka

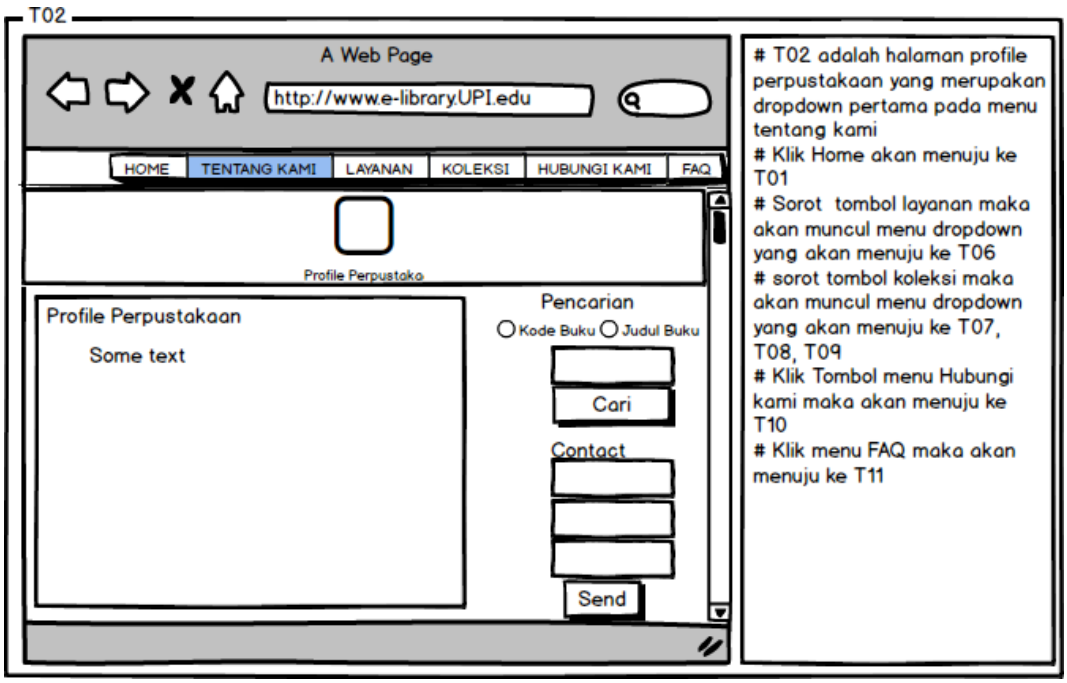
Perancangan antarmuka bertujuan untuk memberikan gambaran visual tentang aplikasi yang akan dibangun. Sehingga akan mempermudah dalam proses pengkodean.

Perancangan antarmuka terbagi menjadi dua, yaitu halaman siswa dan admin. Berikut perancangan antar muka dari aplikasi e-library yang akan dibangun.

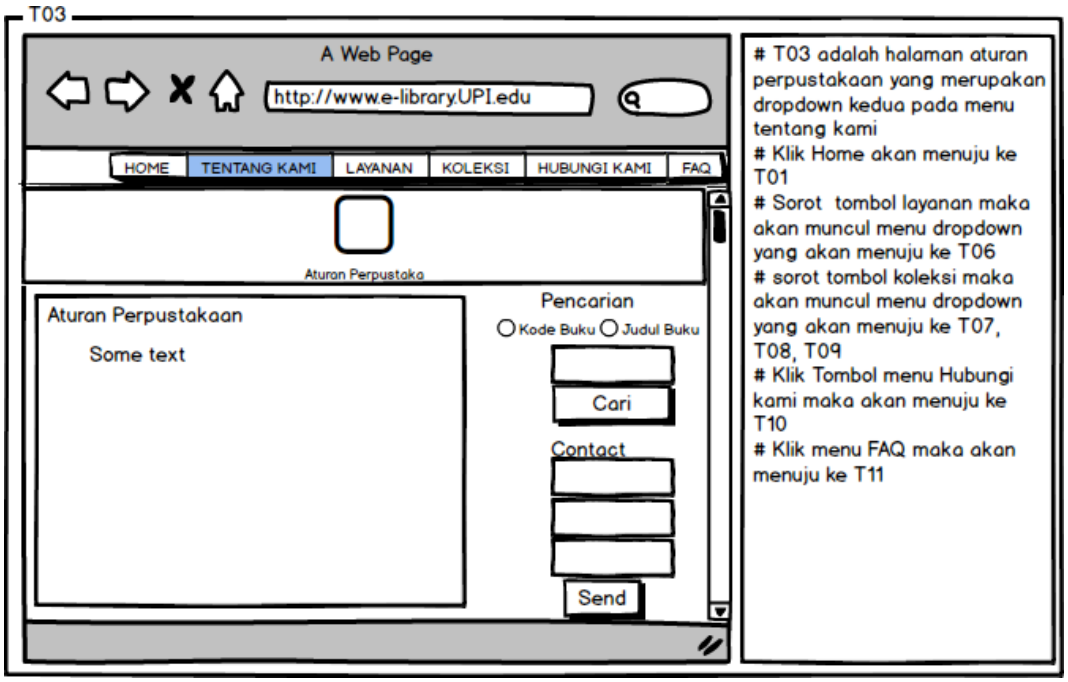
3. Perancangan Antarmuka Siswa



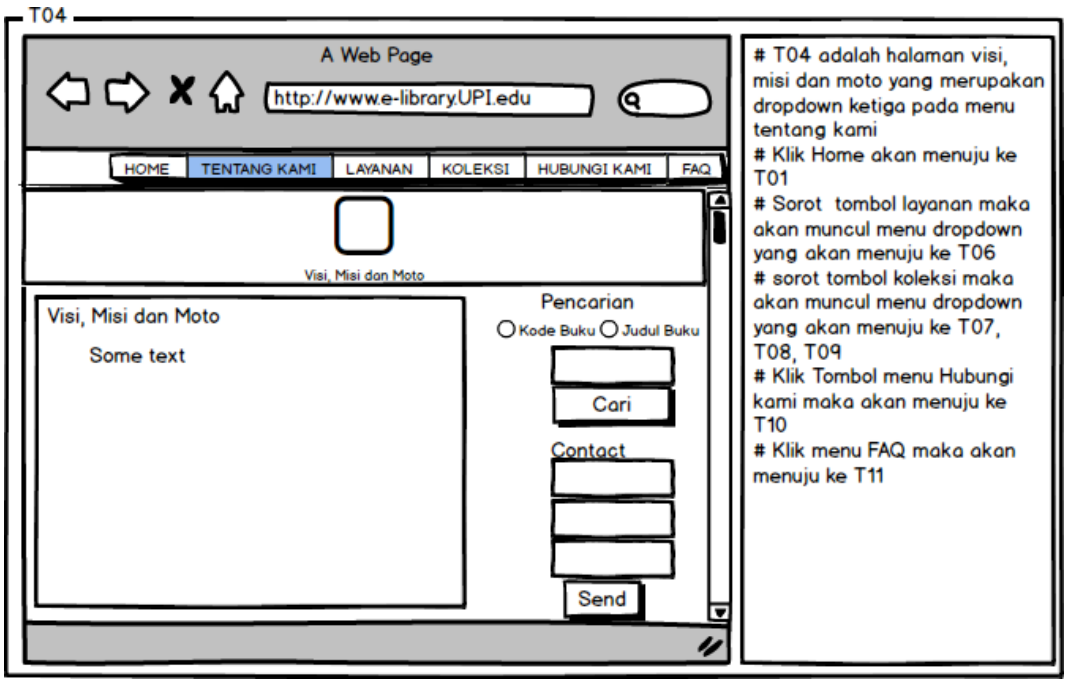
Gambar 3. 48 Perancangan Antarmuka Halaman utama



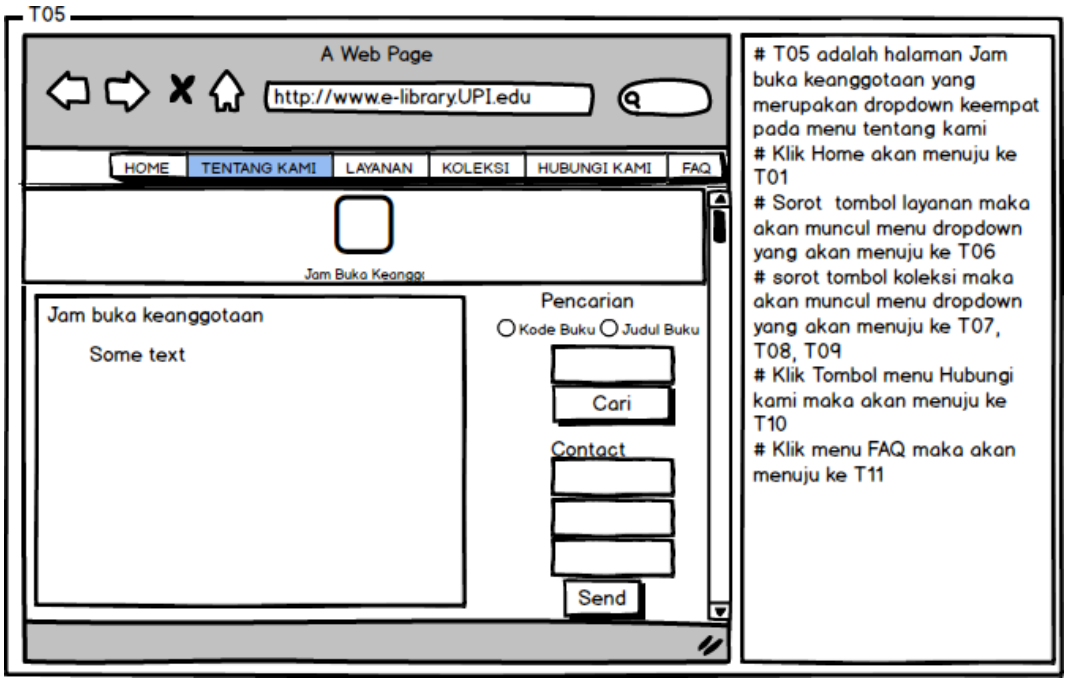
Gambar 3. 49 Perancangan Antarmuka Halaman Profile Perpustakaan



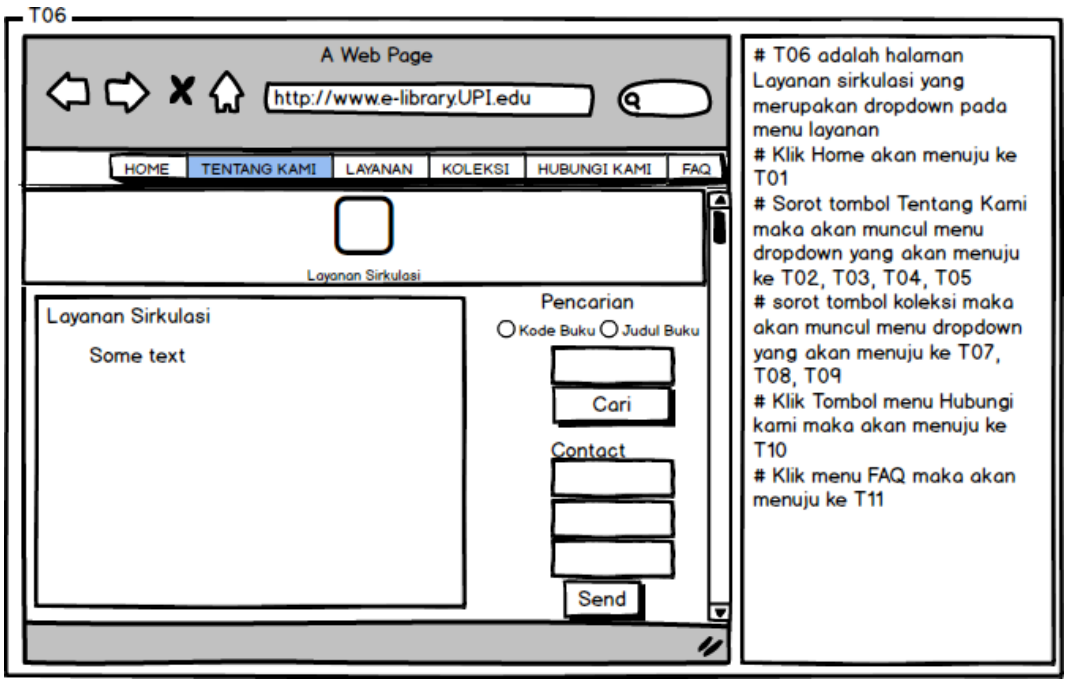
Gambar 3. 50 Perancangan Antarmuka Halaman Aturan Perpustakaan



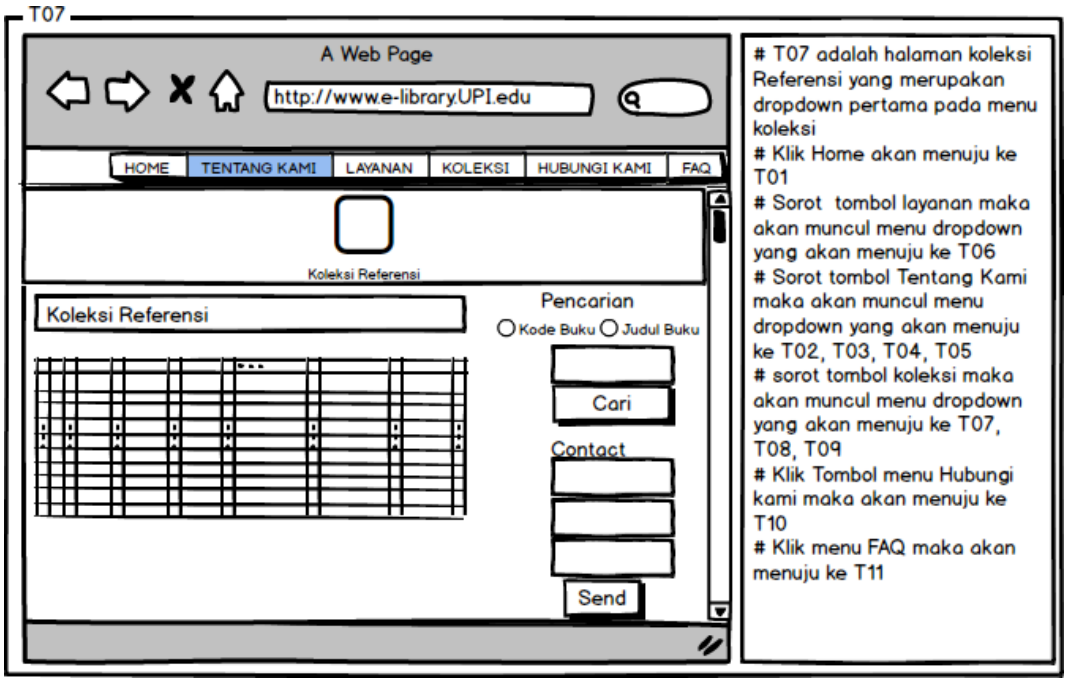
Gambar 3. 51 Perancangan Antarmuka Halaman Visi Misi



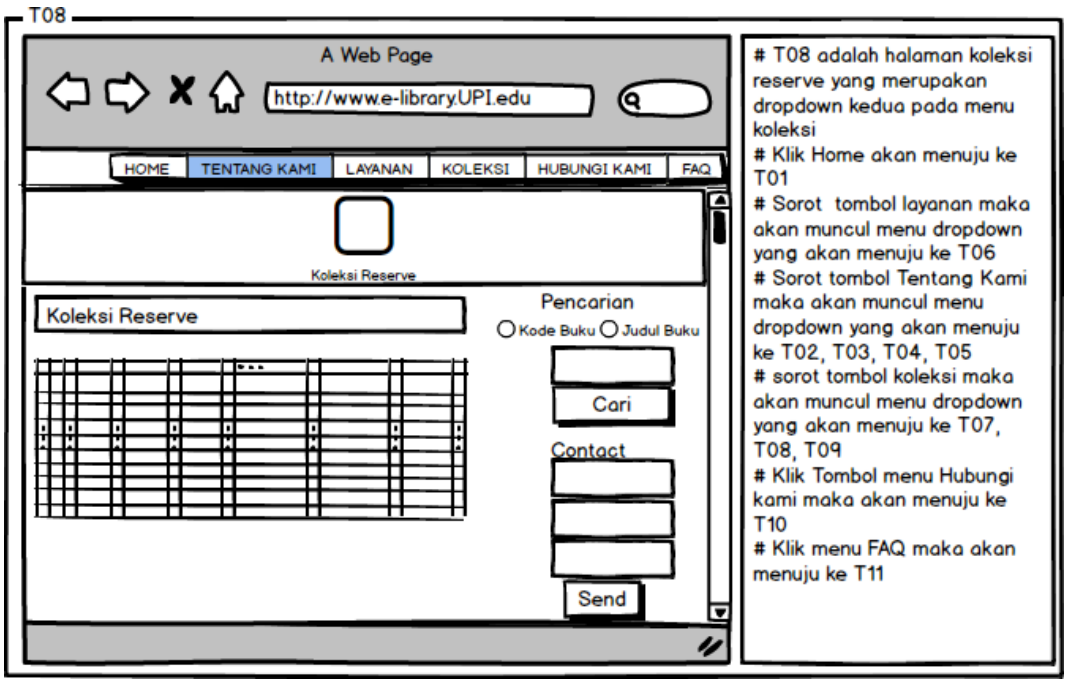
Gambar 3. 52 Perancangan Antarmuka HalamanTentang Kami



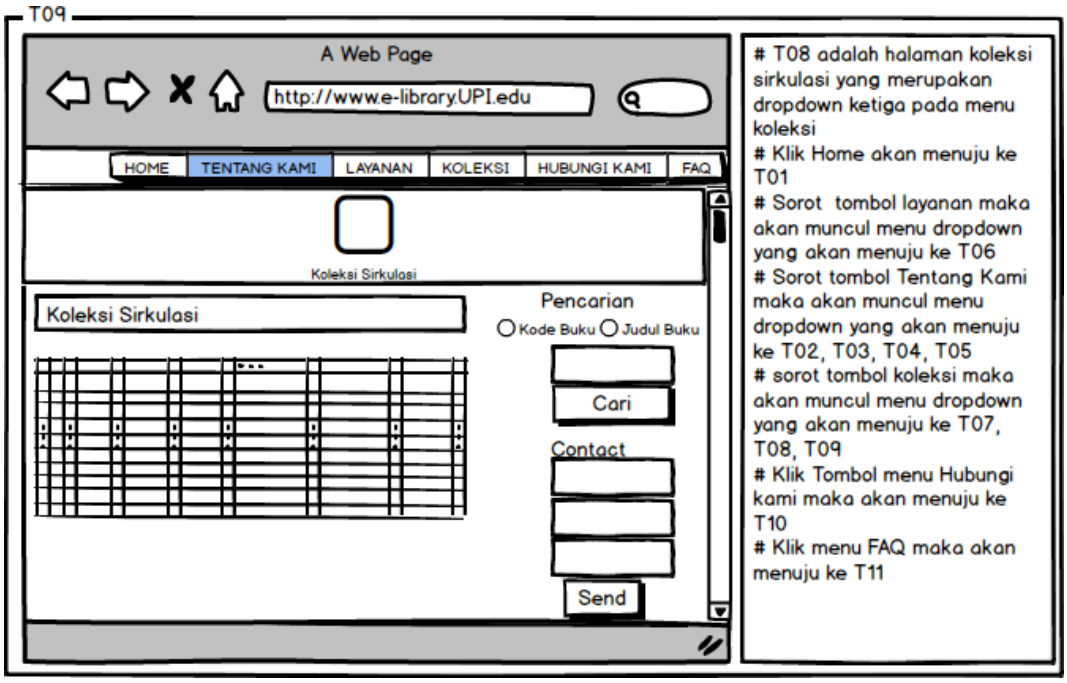
Gambar 3. 53 Perancangan Antarmuka Halaman Layanan



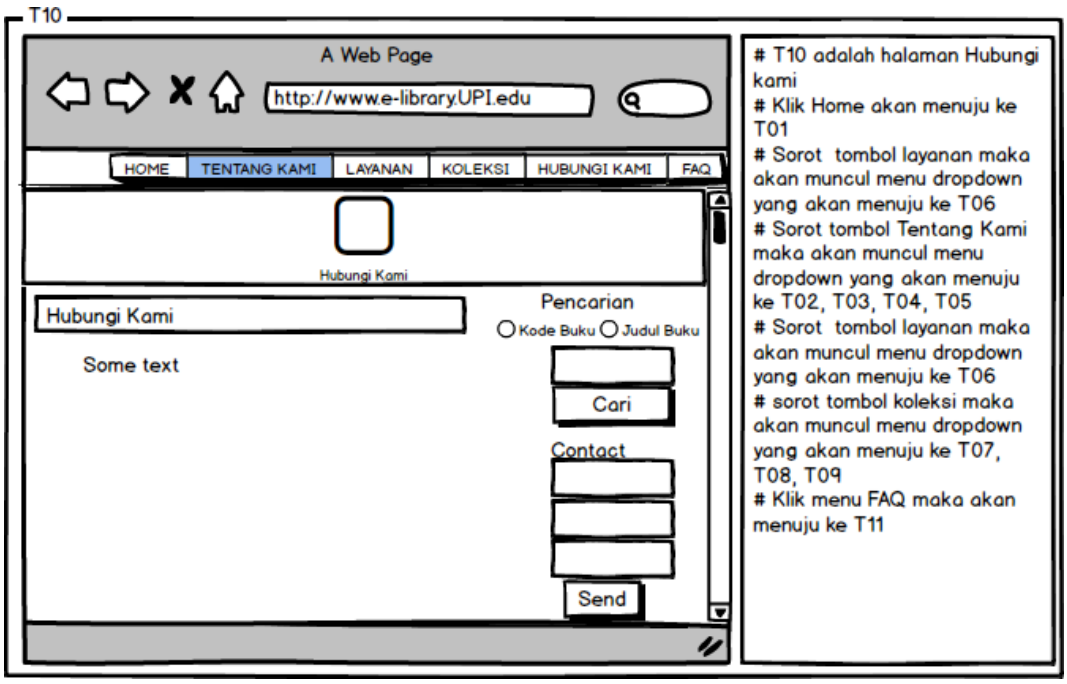
Gambar 3. 54 Perancangan Antarmuka Halaman Koleksi Referensi



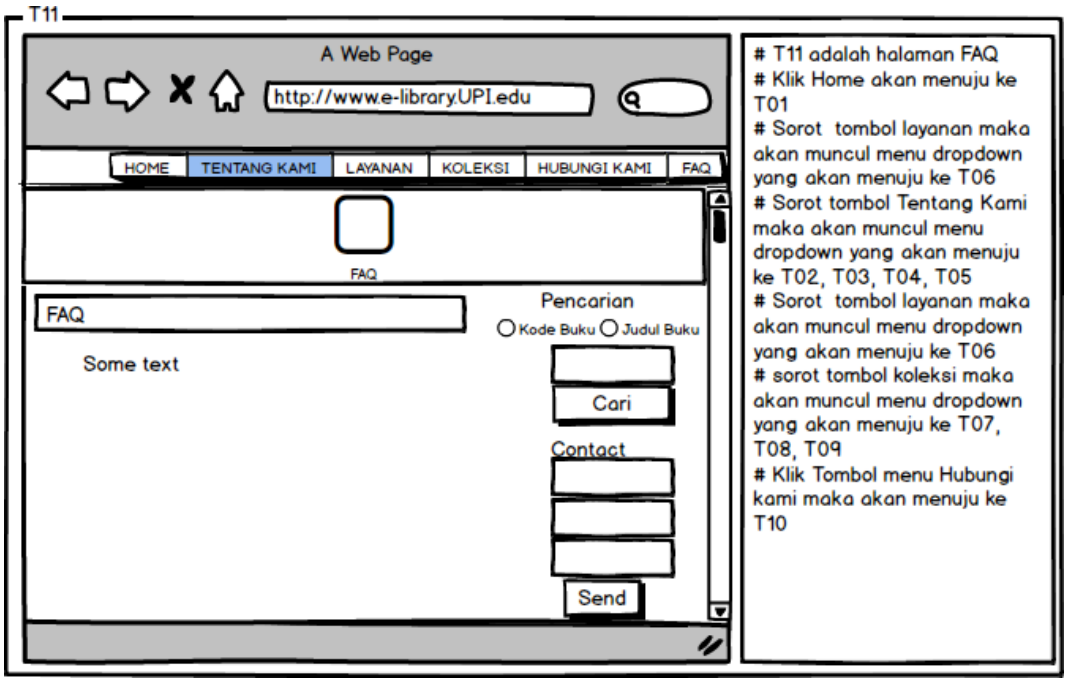
Gambar 3. 55 Perancangan Antarmuka Halaman Koleksi Reserve



Gambar 3. 56 Perancangan Antarmuka Halaman Koleksi Sirkulasi

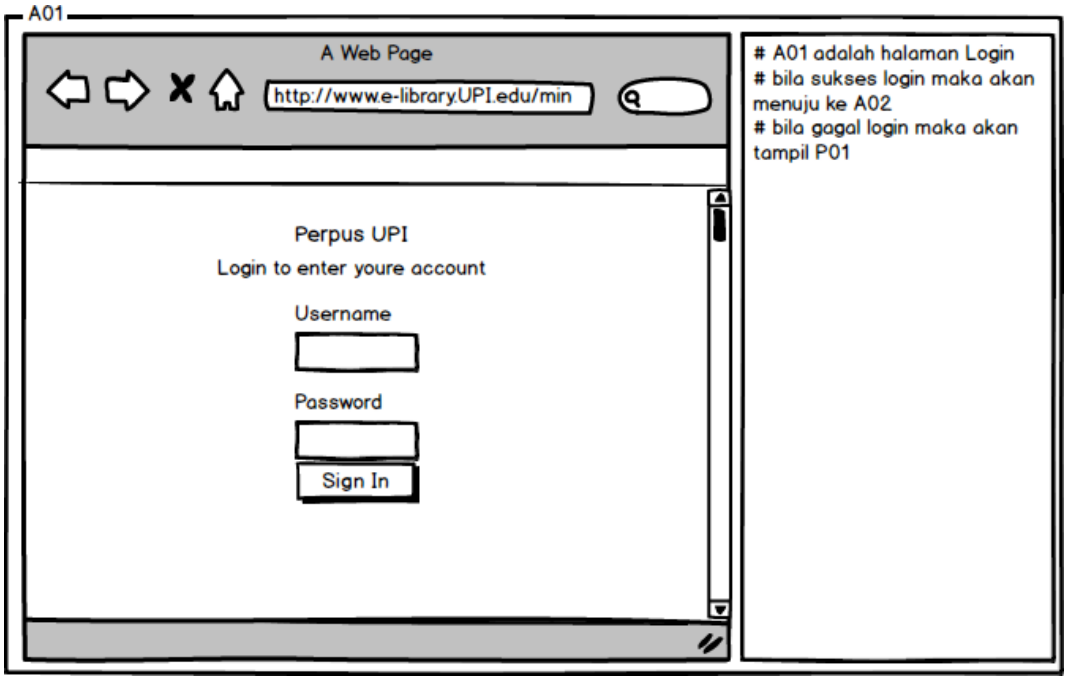


Gambar 3. 57 Perancangan Antarmuka Halaman Hubungi Kami

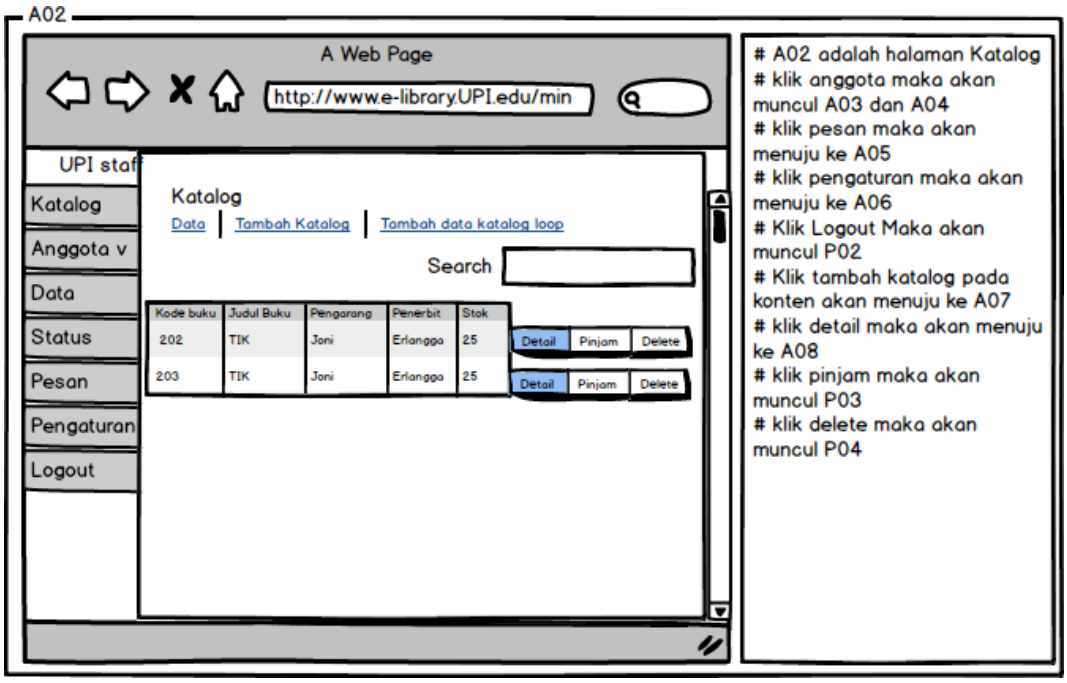


Gambar 3. 58 Perancangan Antarmuka Halaman FAQ

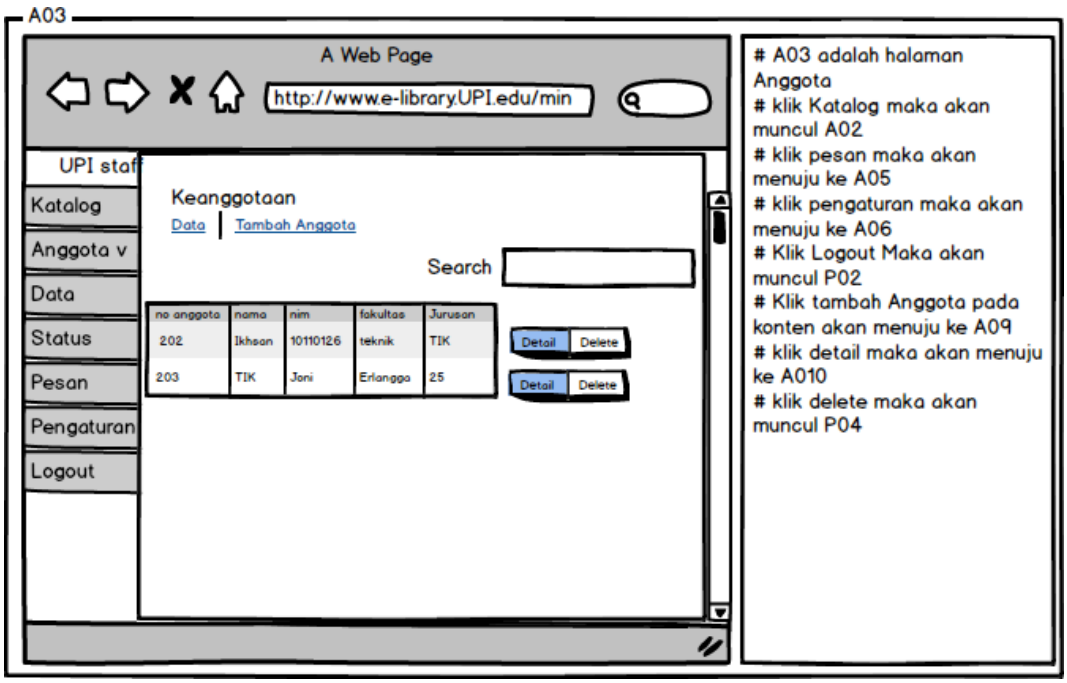
4. Perancangan antarmuka Admin



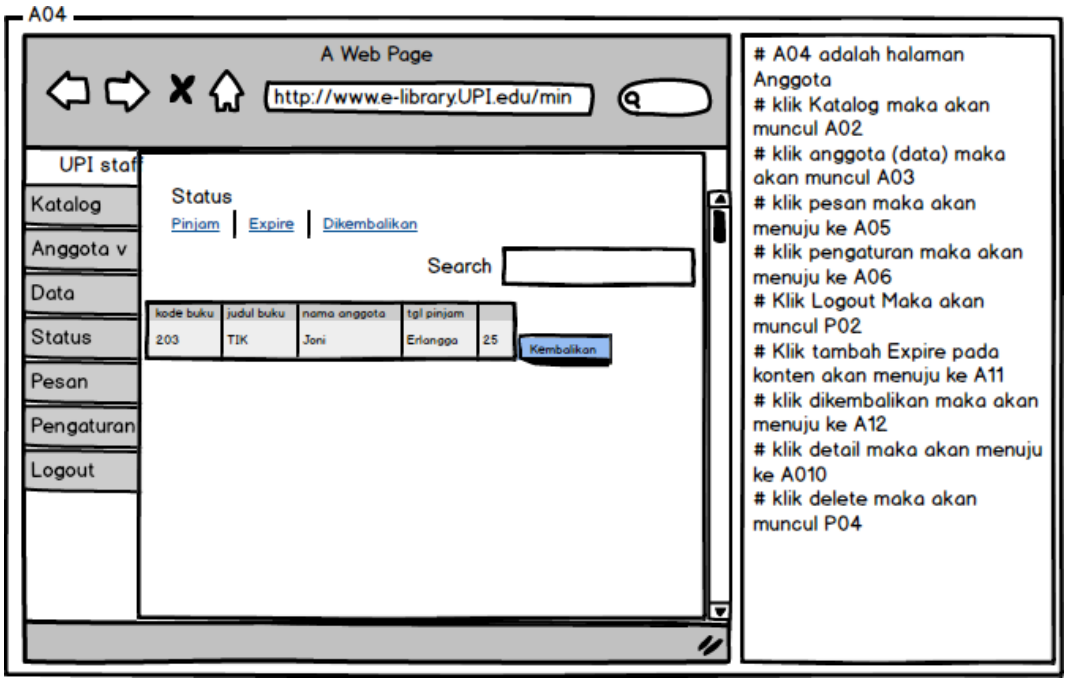
Gambar 3. 59 Perancangan Antarmuka Login



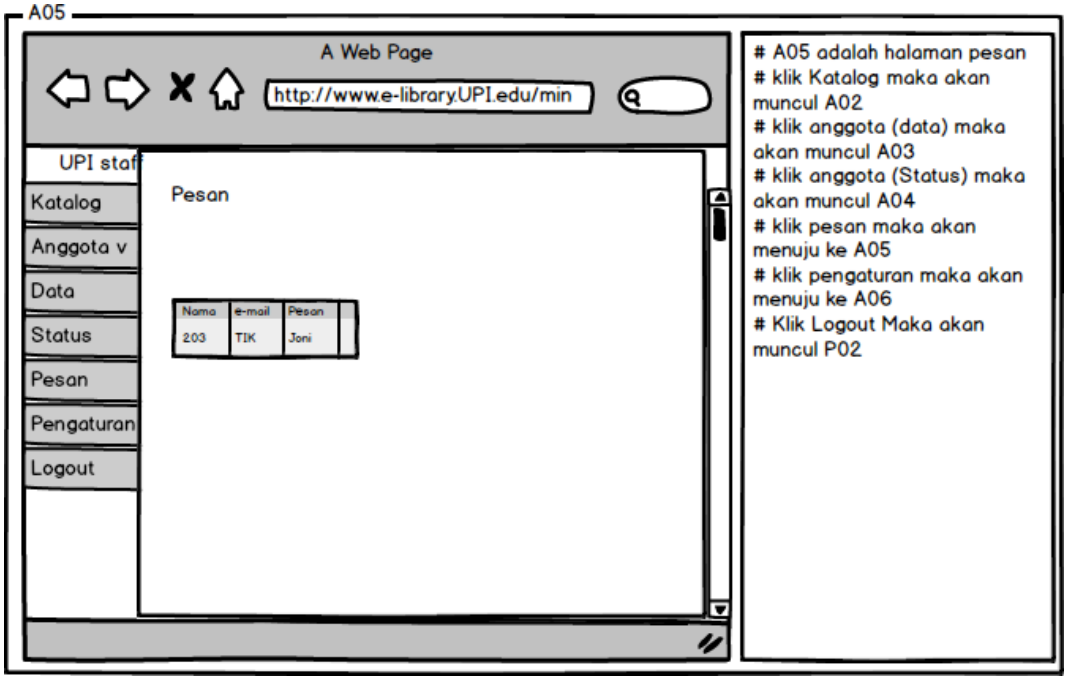
Gambar 3. 60 Perancangan Antarmuka Halaman Katalog



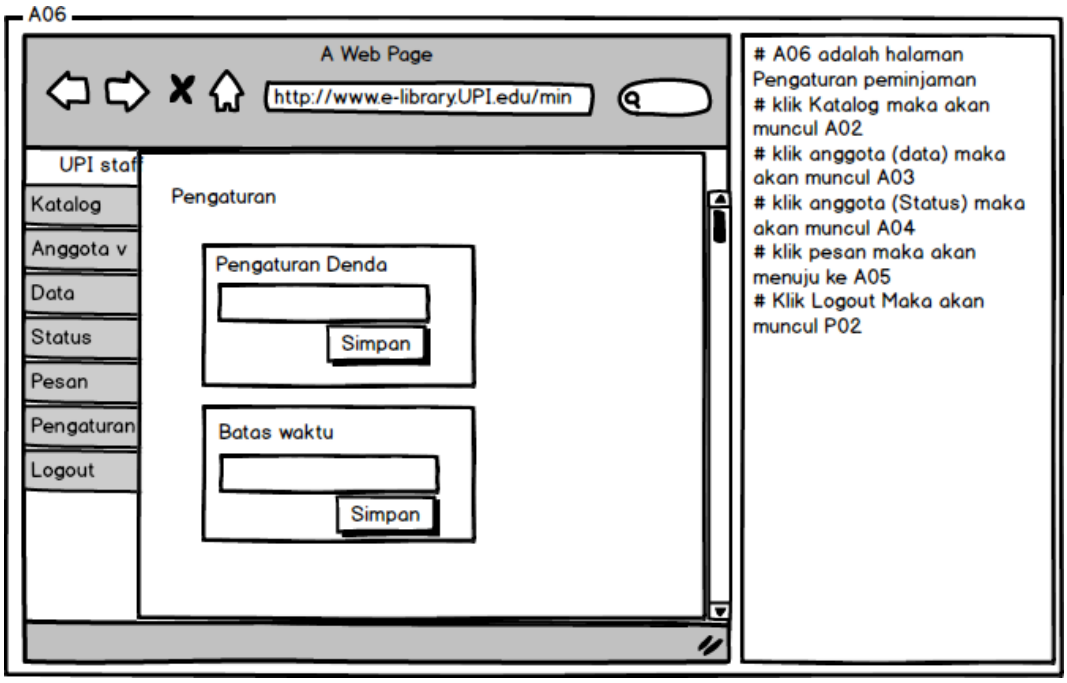
Gambar 3. 61 Perancangan Antarmuka Halaman Keanggotaan



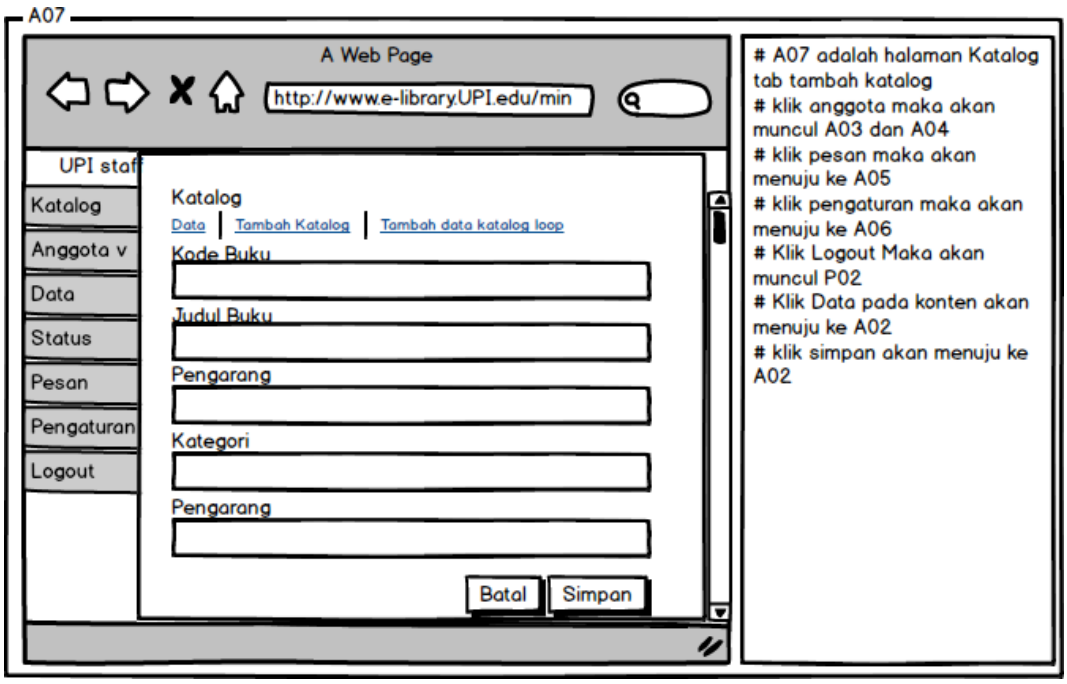
Gambar 3. 62 Perancangan Antarmuka Halaman Status



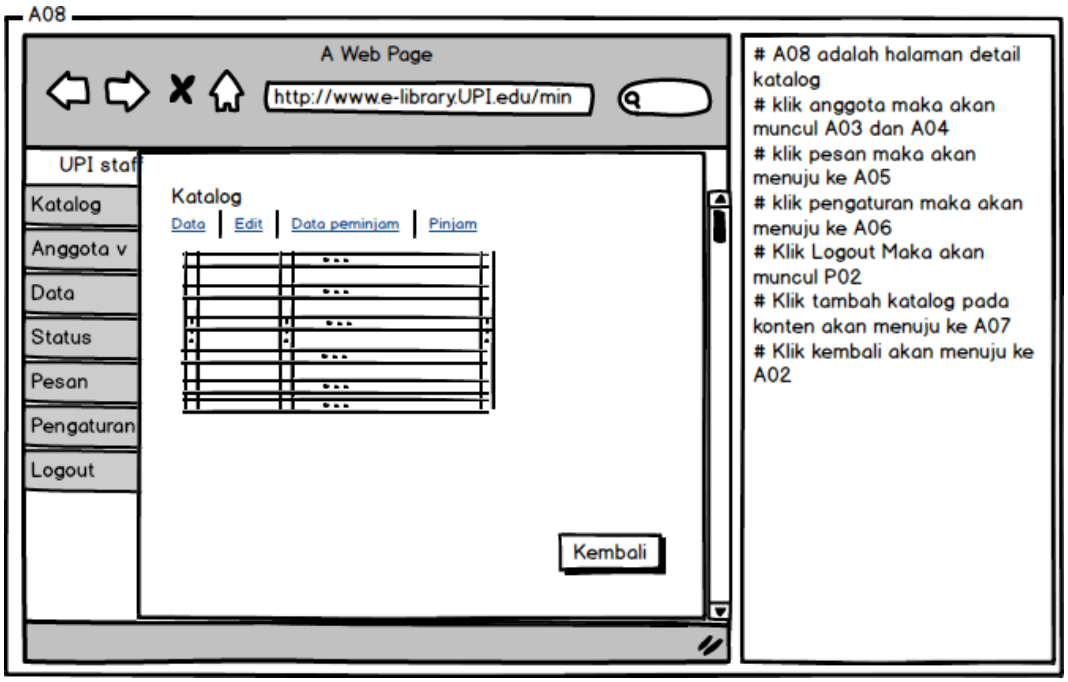
Gambar 3. 63 Perancangan Antarmuka Halaman Pesan



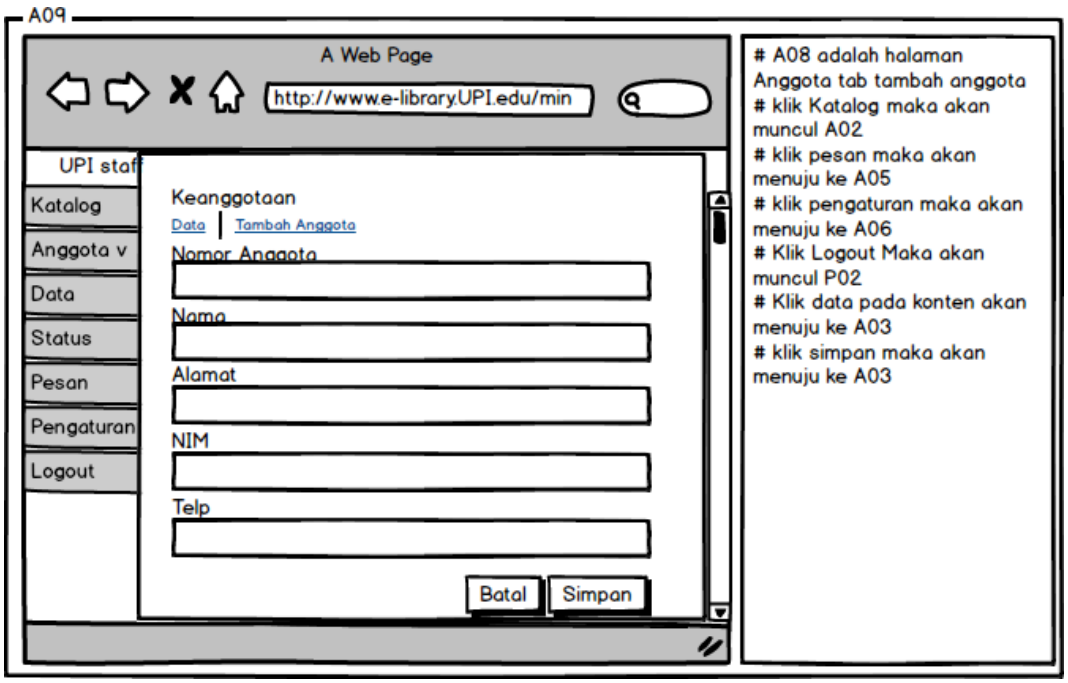
Gambar 3. 64 Perancangan Antarmuka Halaman Pengaturan Peminjaman



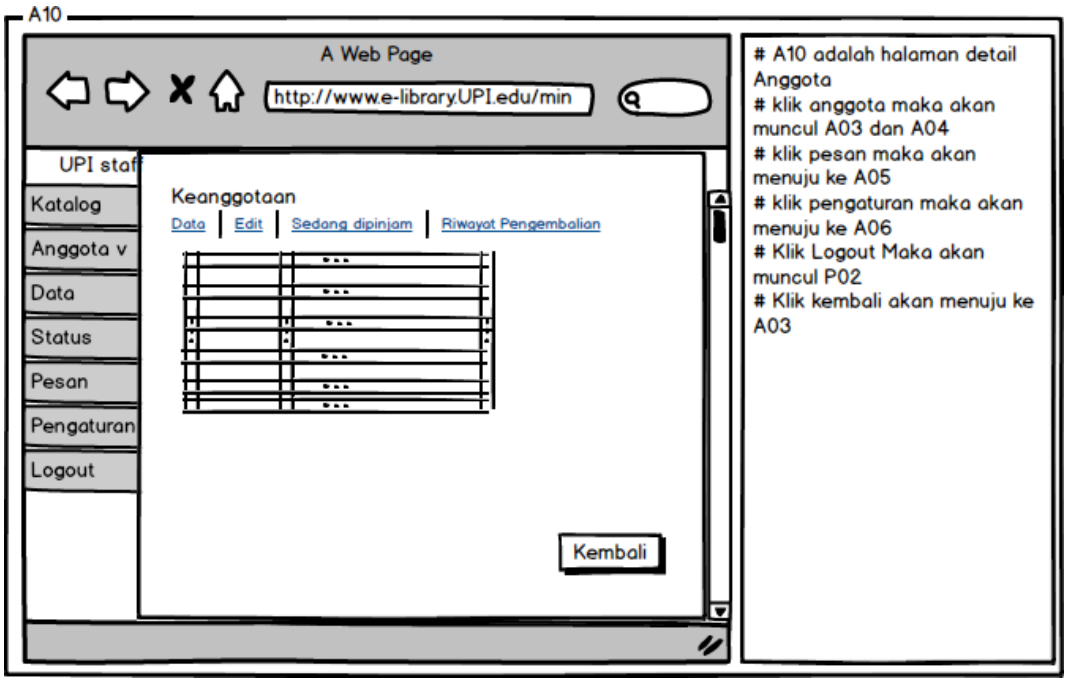
Gambar 3. 65 Perancangan Antarmuka Halaman Tambah Katalog



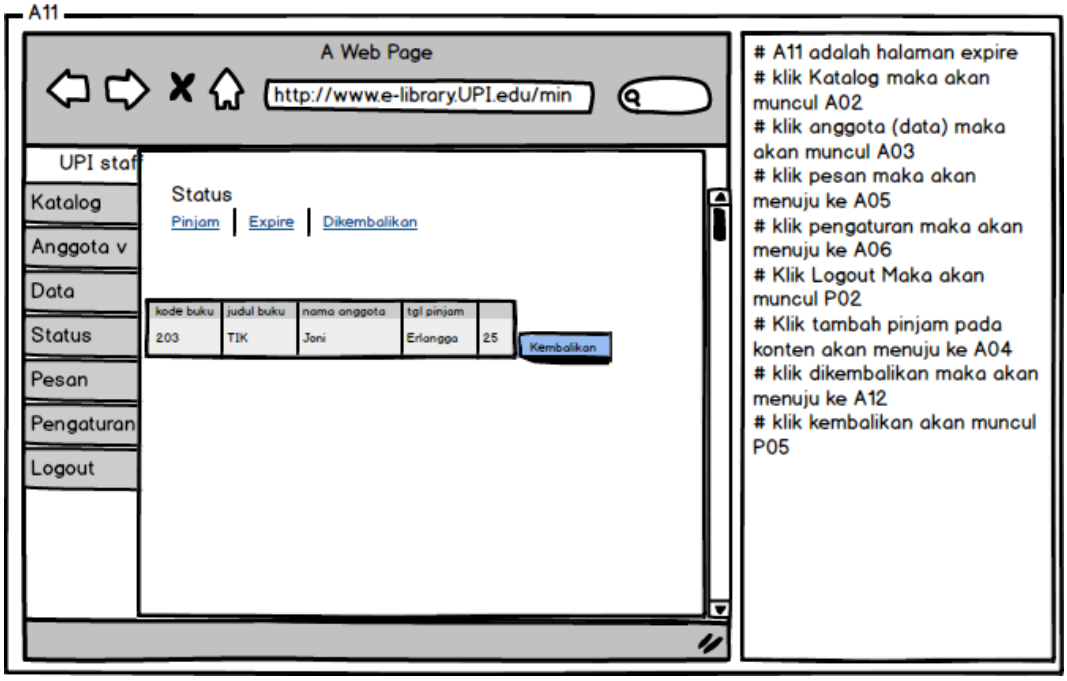
Gambar 3. 66 Perancangan Antarmuka Halaman Detail Katalog



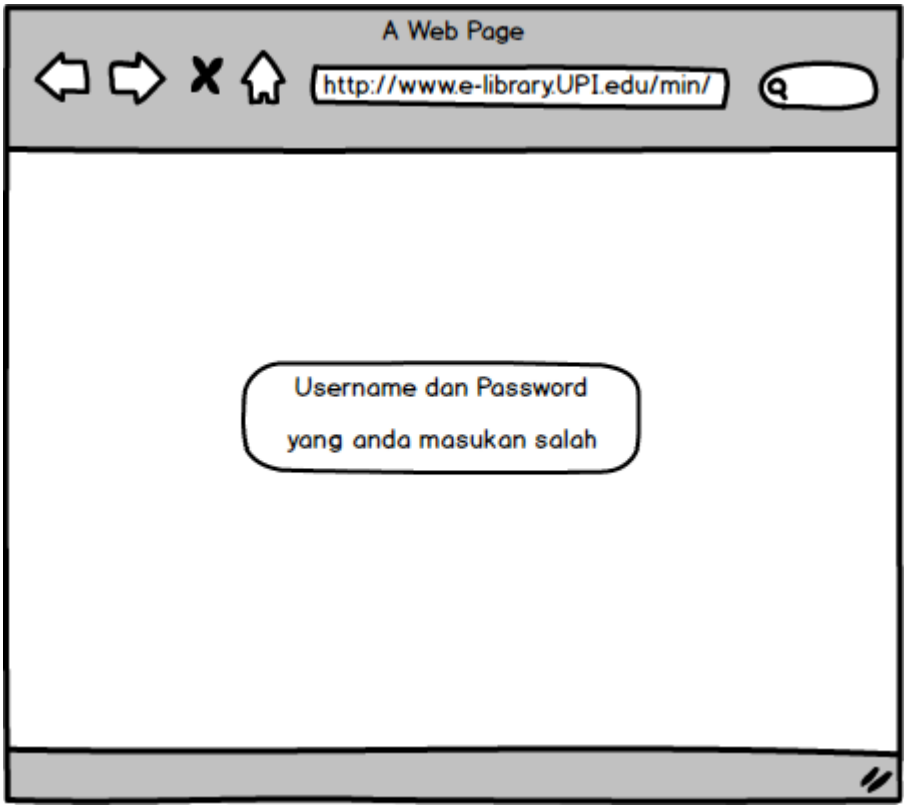
Gambar 3. 67 Perancangan Antarmuka Halaman Tambah Anggota



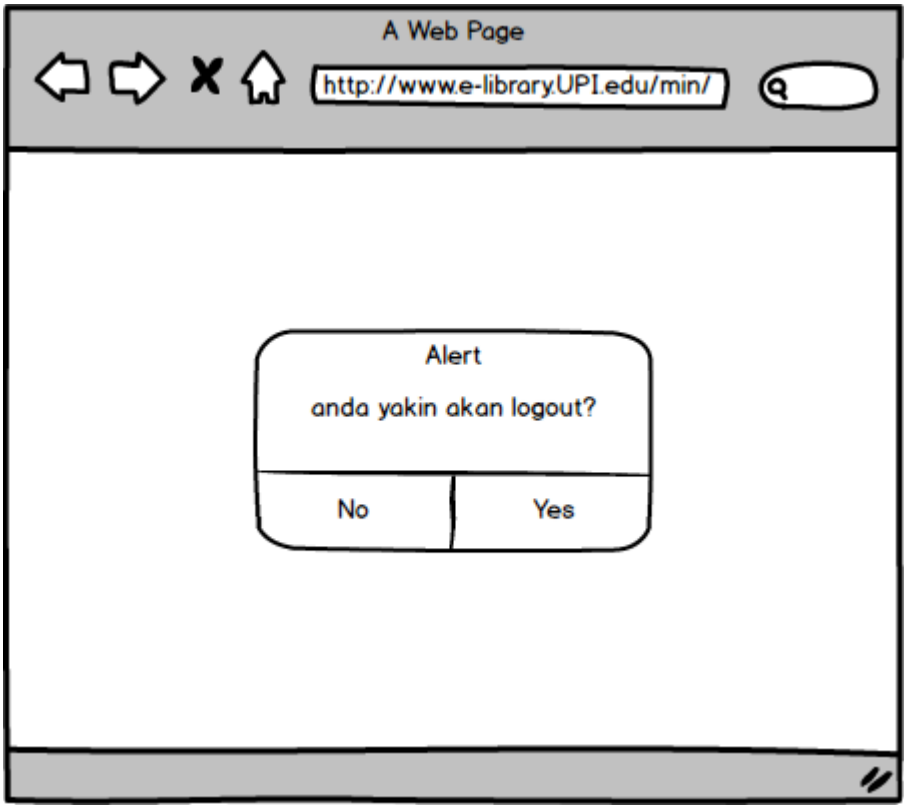
Gambar 3. 68 Perancangan Antarmuka Halaman Detail Anggota



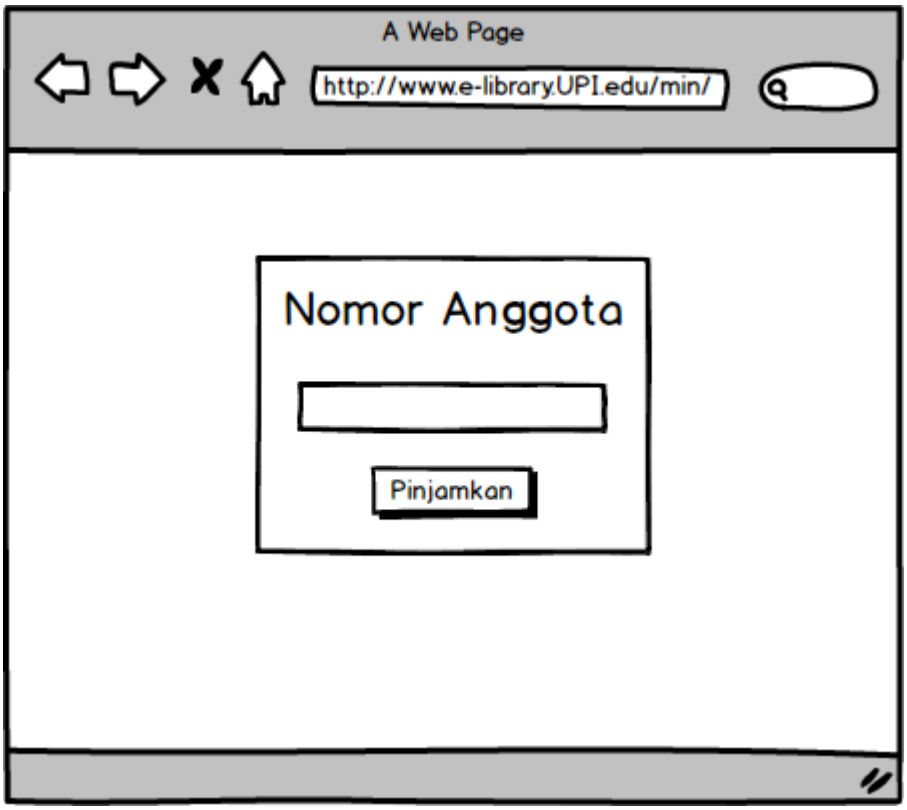
Gambar 3. 69 Perancangan Antarmuka Halaman Expire



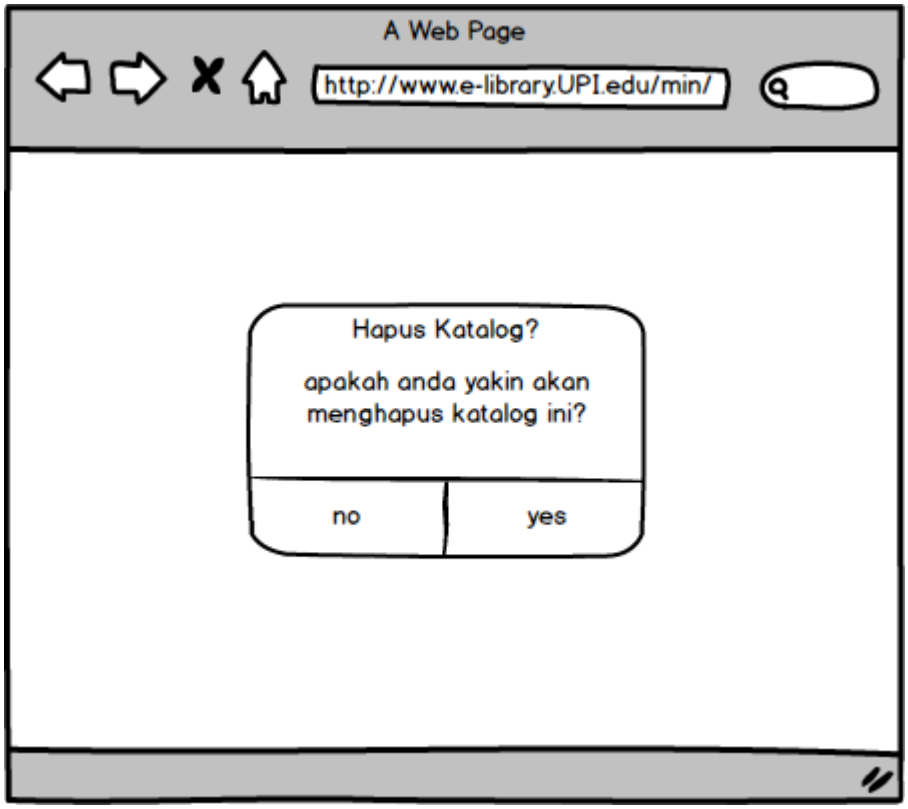
Gambar 3. 70 Perancangan Antarmuka Pesan Gagal Login



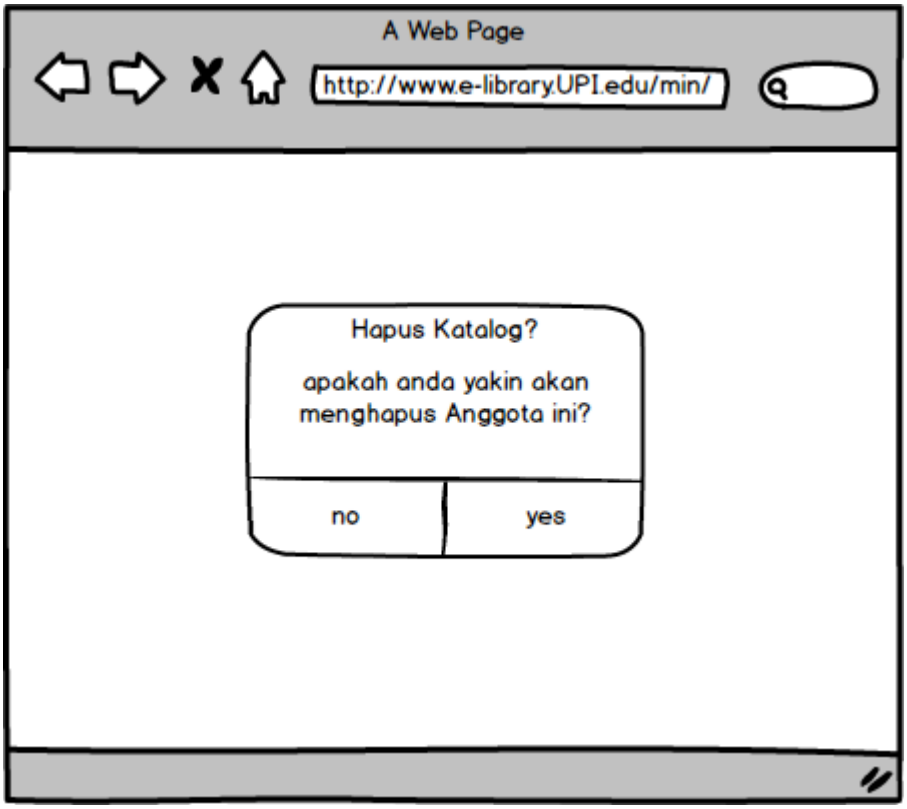
Gambar 3. 71 Perancangan Antarmuka Pesan Logout



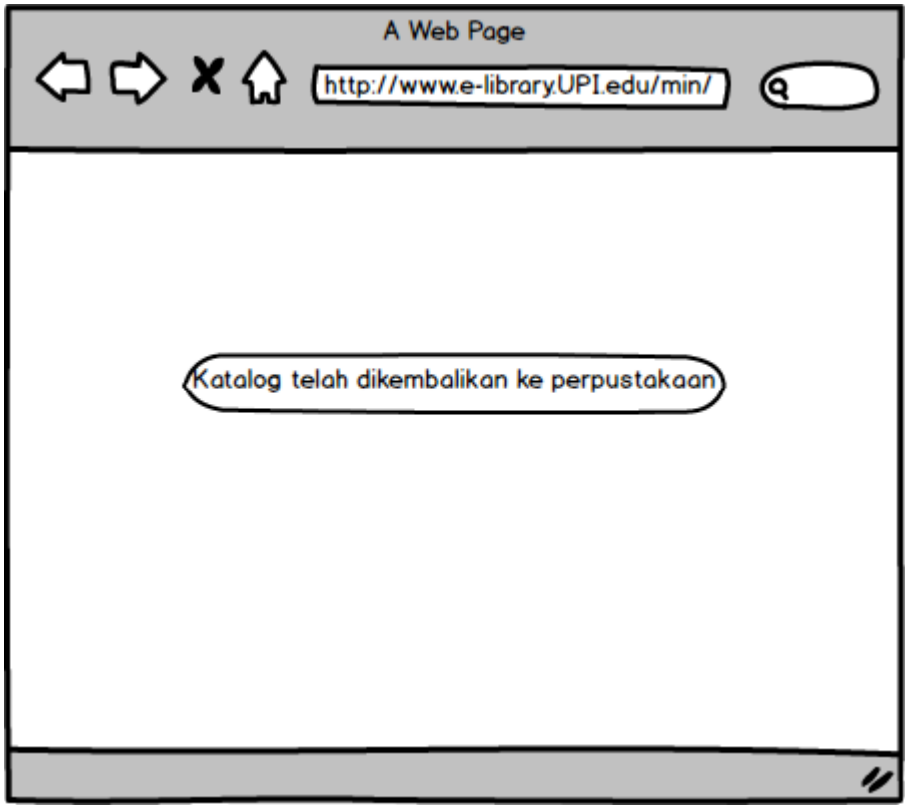
Gambar 3. 72 Perancangan Antarmuka Pesan Masukan Nomor Anggota



Gambar 3. 73 Perancangan Antarmuka Pesan Hapus Katalog



Gambar 3. 74 Perancangan Antarmuka Pesan Hapus Anggota

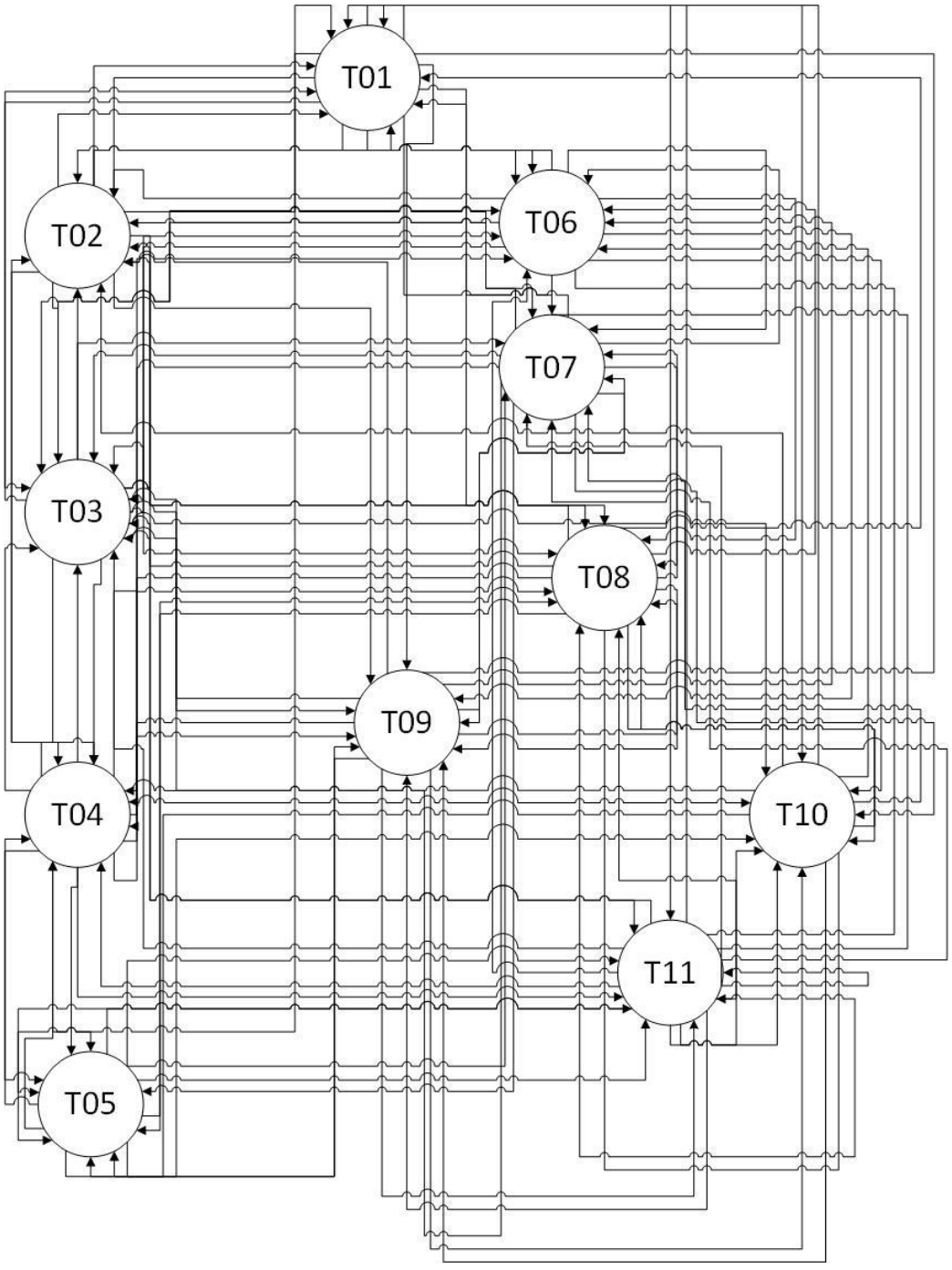


Gambar 3. 75 Perancangan Antarmuka Pesan Pengembalian Katalog

3.4.2 Jaringan Semantik

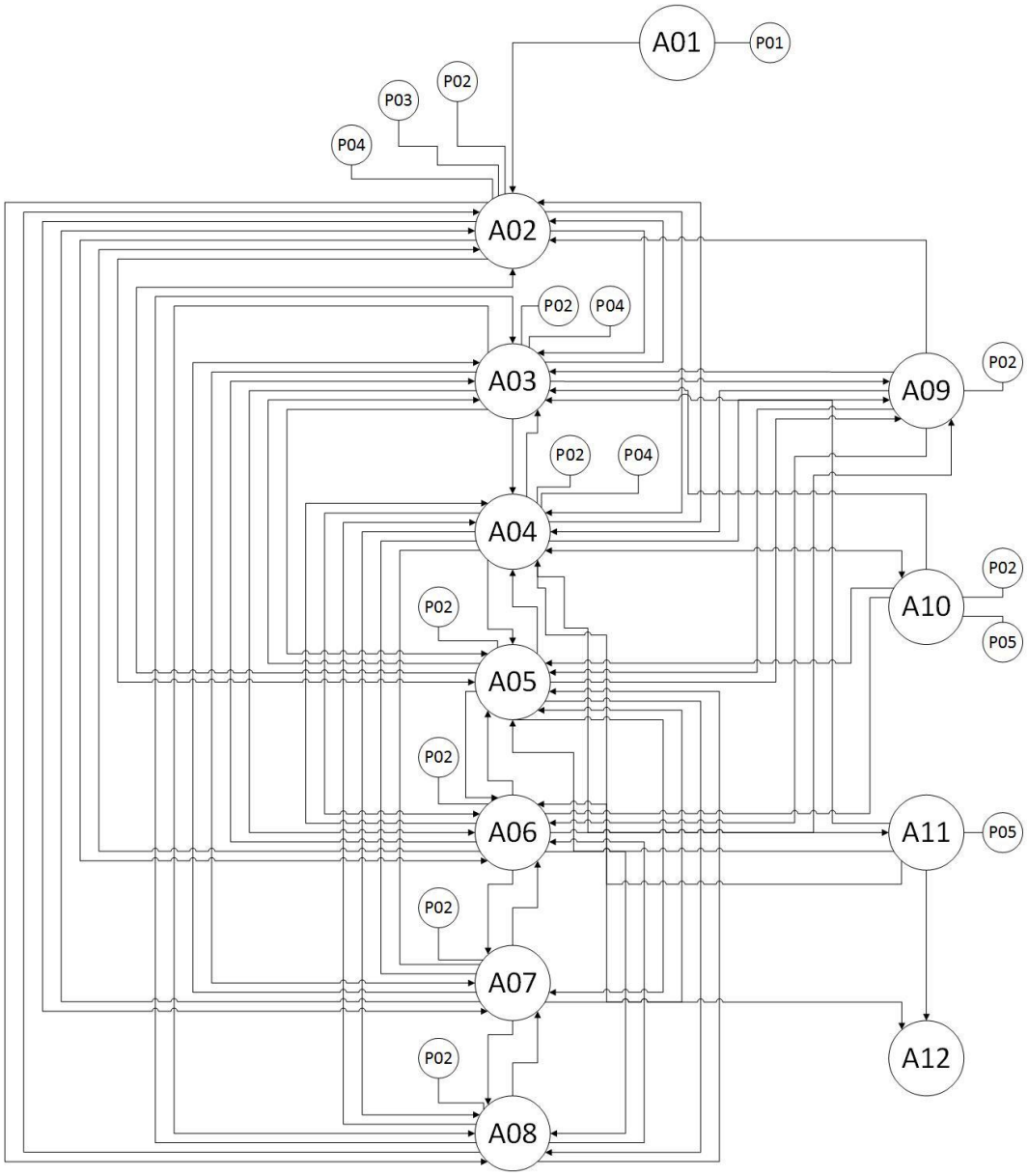
Jaringan semantik merupakan gambaran hubungan tentang navigasi halaman-halaman perancangan terhadap halaman perancangan lainnya. Berikut jaringan semantiknya.

1. Jaringan semantik halaman siswa



Gambar 3. 76 Jaringan Semantik Halaman Siswa

2. Jaringan semantik halaman admin



Gambar 3. 77 Jaringan Semantik Perancangan Admin