BAB 3

ANALISIS DAN PERANCANGAN

3.1. Analisis Sistem

Sedangkan menurut Kristanto (2003) analisis sistem adalah suatu proses mengumpulkan dan menginterpretasikan kenyataan-kenyataan yang ada, mendiagnosa persoalan dan menggunakan keduanya untuk memperbaiki sistem.

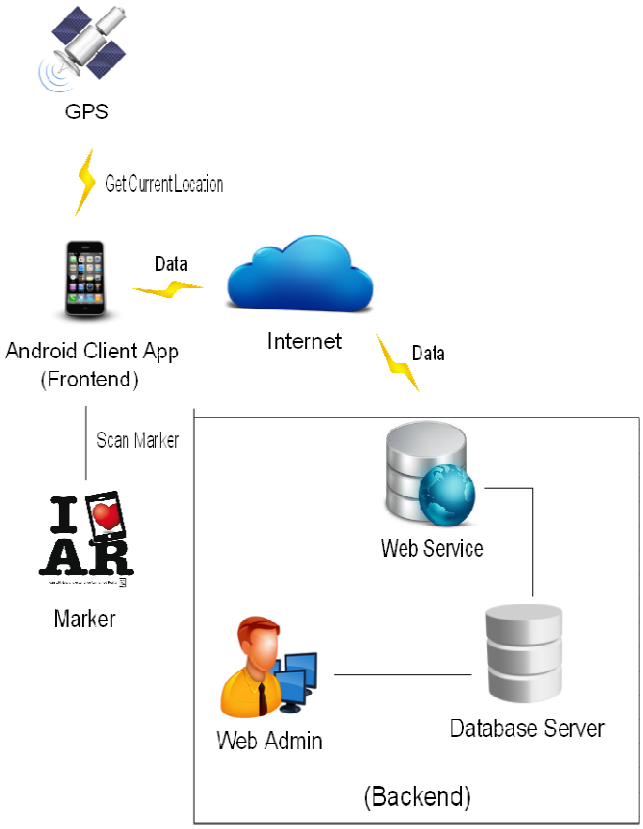
Menurut Yogiyanto (1995) analisis sistem adalah penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan.

3.1.1. Analisis Masalah

Berdasarkan permasalahan yang ada, yaitu terbatasnya informasi yang didapat pengunjung, kurang efektifnya media informasi yang tersedia dan kurangnya penunjuk lokasi di kawasan kebun binatang ini maka dilakukan rancangan untuk membangun aplikasi dan memberikan tampilan antarmuka aplikasi yang mudah dioperasikan sehingga informasi yang diinformasikan mudah untuk dicerna oleh pengunjung.

3.1.2. Analisis Arsitektur Sistem

Sistem yang dibangun menggunakan metode *client server*. Arsitektur fisik sistem terdiri dari komponen *frontend, web service, web admin* serta *database server*. Untuk lebih jelasnya dapat dilihat pada gambar 3.1.



Gambar 3. 1 Arsitektur Sistem.

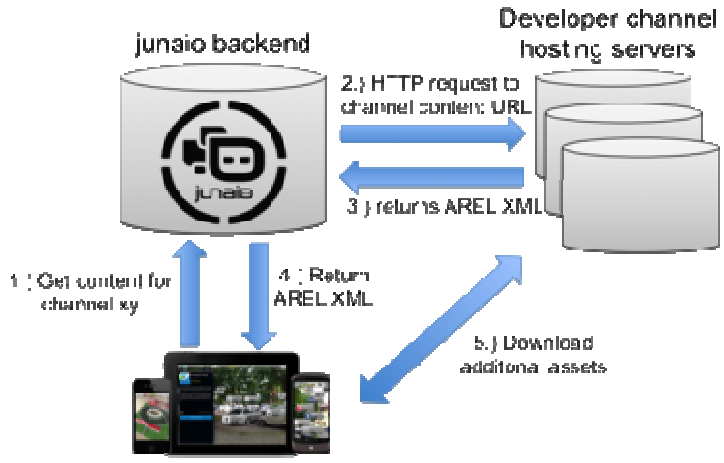
Android Client App (Frontend) merupakan aplikasi yang diinstall di perangkat *mobile* yang berfungsi sebagai antarmuka pengguna dengan sistem dalam mengakses konten-konten yang tersedia di *database.* Pada *frontend* terdapat akses ke *GPS* untuk menentukan posisi keberadaan pengguna. Posisi pengguna ini berfungsi sebagai titik acuan dalam menampilkan keberadaan *POI* di sekitar pengguna. Sedangkan akses dengan *marker* digunakan sebagai *trigger* untuk mengakses konten informasi yang dimiliki *POI.*

Backend merupakan aplikasi pendukung dalam pengolahan data *POI* yang merupakan sumber data bagi sisi *frontend.* Pada sisi ini terbagi menjadi beberapa subbagian yaitu *web service, web admin* dan *database server. Web service* berfungsi sebagai media pertukaran data antar *database server* dengan *frontend.*

Pada pembangunan aplikasi ini *web service* dibangun dengan menggunakan pemrograman *PHP* dan *AREL*. *Web admin* berfungsi sebagai media pengolah sumber basis data yang terdapat pada *database server*. *Web admin* dikelola oleh

admin dan pada penelitian ini dibangun dengan menggunakan bahasa pemrograman *PHP*. Dan bagian *backend* yang terakhir yaitu *database server*, fungsi dari bagian ini adalah sebagai media penyimpanan sumber data aplikasi. Pembangunan *database server* pada penelitian ini menggunakan *MySQL*.

3.1.3. Analisis Implementasi *AR Browser*



Gambar 3. 2 Konsep Kerja AR Browser [Junaio : Doc].

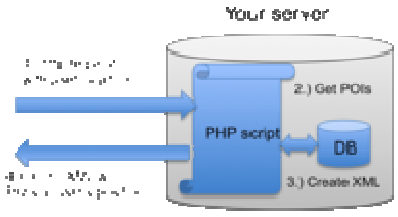
Browser ini menggunakan konsep kerja *channel* dalam pengimplementasian *augmented reality*. Pada dasarnya kerja dari *browser* sama persis dengan konsep saluran televisi. Saluran tersebut pada *browser* disebut dengan *channel* yang terdapat pada bagian *backend browser*. Jadi cara kerjanya *user* akan diarahkan pada *channel* yang telah disesuaikan dengan *query* aplikasi.

Berikut penjelasan rincinya :

1. Setelah mendapatkan *trigger* (untuk *LBS* berupa *current location* pengguna/untuk *marker* berupa *scan marker*) sisi *client* akan mengirimkan permintaan *HTTP* ke *server browser* untuk mendapatkan konten *channel* yang telah dibuat terlebih dahulu pada sisi *backend browser*. Contoh *channel* yang telah dibuat (*id*: 216132, *name*: BonBinTour | Hello World).

2. *Browser backend* melanjutkan *URL channel* dan mengirim permintaan *HTTP* ke *server hosting*. Permintaan ini dapat berisi lokasi pengguna, *POI*, serta jenis perangkat. Contoh *URL channel* yang telah dibuat di *channel*: 216132 (*http://bonbintour.kebunbinatangbandung.com/arel/arel.xml*).
3. *Hosting server* kemudian merespon permintaan dengan *ARELXMLHelper*. Yang berupa file dinamis *XML* melalui *PHP*. Untuk *AR LBS* penyeleksian data yang akan ditampilkan dilakukan di sisi *hosting server.* Untuk *AR scan marker* penyeleksian data *marker* dilakukan juga di sisi *hosting server*, sebelumnya *database marker* telah di-*upload* dan tersimpan di *hosting server.* Contoh file *XML* yang akan di respon adalah *arel.xml*.
4. *Hosting server* melanjutkan *XML (arel.xml)* kepada sisi *client* dan mem-*parsing* respon tersebut agar dapat terbaca di *device client.* Disinilah peran *AREL* sebagai media untuk mem-*parsing* data dari *server* ke *client*, begitu juga sebaliknya.
5. Sisi c*lient* mulai mendownload dan memproses semua *asset* yang diperlukan untuk merepresentasikan *augmented reality* yang diinginkan.

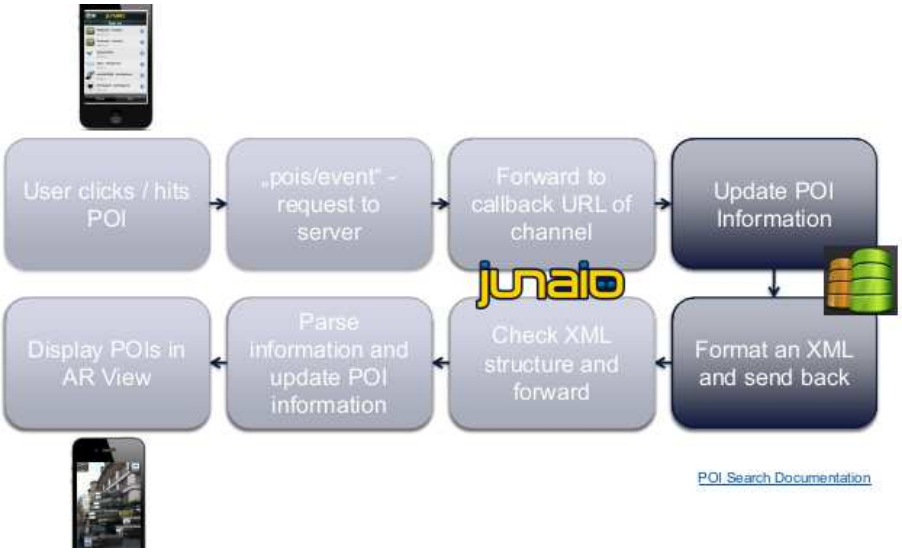
Berikut mengenai penjelasan detail mengenai *dinamic channel* :



Gambar 3. 3 Dinamic Channel Browser [Junaio : Doc].

Pada *dinamic channel* ketika proses *request*, *request* tersebut tidak langsung mengacu pada file *XML* melainkan diarahkan terlebih dahulu ke *script logic PHP* yang akan membuatkan *query database* lalu membuatkan sebuah *XML* dan mengembalikan *XML* tersebut ke sisi *server browser*.

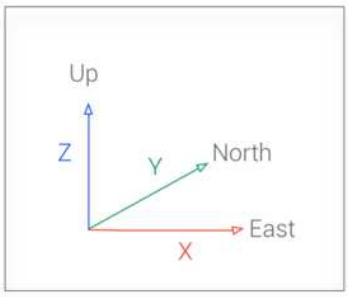
3.1.4. Analisis Pembentukan *AR*



Gambar 3. 4 Proses Pembentukan AR [Junaio : Doc].

Dalam pembentukan *AR* diawali dengan *user* yang melakukan *trigger event* baik berupa identifikasi koordinat pengguna maupun proses *scan marker* yang diinginkan lalu aplikasi mengirimkan permintaan *Event* ke sisi *server browser*, lalu diteruskan dengan mengembalikan *callback url* oleh *server browser* yang mengarah ke sisi *database server*. Pada *database server* akan mengembalikan sebuah nilai dalam format *XML* yang akan dikirimkan kembali ke sisi *server browser* untuk di cek ke-*valid*-annya. Setelah di cek, maka dilanjutkan ke proses *parsing* informasi yang nantinya akan tag kan sehingga *AR* dapat terlihat pada tampilan *browser* di sisi *client*.

3.1.5. Analisis Koordinat *GPS*



Gambar 3. 5 Konsep Dasar Penggambaran Koordinat GPS [Metaio : Doc].

Dalam sistem koordinat *GPS* global, diterjemahkan dan digambarkan dalam suatu ruang bangun 3D yang memiliki 3 sumbu koordinat yaitu X, Y dan Z. Seperti yang digambarkan. X akan digambarkan menuju ke Timur, Y ke Utara dan Z ke atas. Ketiga koordinat itulah yang nanntinya akan disebut dengan koordinat *latitude*, *longitude* dan *altitude*.

3.1.6. Analisis *Marker*

Tahapan dalam menganalisis bentuk dan isi pada *marker* yang digunakan serta marker seperti apa saja yang digunakan.

3.1.6.1. Format *Marker*

Marker yang digunakan berupa gambar berwarna yang merepresentasikan objek dari *POI* yang tersedia. Format gambar yang digunakan pada aplikasi ini ber-*format .png*. Contoh *marker* bisa dilihat pada gambar 3.6.



Gambar 3. 6 Marker.

3.1.6.2. *Tools Marker*

Dalam penelitian ini *marker* didaftarkan ke *database hosting server* dengan menggunakan *tools Metaio Creator*. Yang digunakan pada penelitian ini adalah *Metaio Creator Versi Demo. Database* gambar *marker* tersimpan di *hosting server*, sehingga untuk mengaksesnya (menyeleksi untuk dikenali) hanya perlu memanggil *channel* yang mengacu ke *url database marker* di *hosting server*. Dengan *tools* ini semua konten pada *marker* akan di sesuaikan dengan objek *POI*, sehingga pengguna dapat dengan tepat dan cepat mengakses konten tersebut.

3.1.7. Analisis Suara

Konten suara yang didukung dalam pembangunan aplikasi ini ber-*format .mp3*. Format *.mp3* merupakan *format file* suara yang hampir didukung oleh semua jenis pemutar suara yang ada, oleh karena itu dipilih *format* tersebut. Dalam proses *upload file* yang dilakukan oleh *admin* pada *web admin (backend)* dilakukan validasi jenis *file* yang diizinkan yaitu *.mp3.*

3.1.8. Analisis Video

Konten video yang didukung pada aplikasi ini ber-*format .mp4* yang merupakan *format file video* umum digunakan. Seperti halnya dengan *format file* suara, untuk proses *upload file* di sisi *backend* juga dilakukan *validasi* sehingga jenis *file* yang diizinkan untuk di-*upload* adalah *.mp4.*

3.1.9. Analisis Model

Konten model yang didukung pada aplikasi ini ber-*format .obj* dan *.md2*. *Format .obj* digunakan untuk jenis model *static (non animation).* Sedangkan untuk *format .md2* digunakan untuk jenis model *dynamic (animation).* Nampun dalam proses *upload file* dari *backend* diharuskan ber-*format .zip,* jadi untuk objek yang akan di-*upload* harus di-*compress* terlebih dahulu ke dalam format *.zip.* Hal ini dilakukan untuk mempermudah sistem dalam melakukan pencarian model yang di-*request.* Contoh model yang digunakan pada penelitian ini adalah model *sample* dari *metaio.*

3.1.10. Analisis dan Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional menggambarkan kebutuhan sistem yang menitik-beratkan pada properti perilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat keras, perangkat lunak, serta user sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.1.10.1. Kebutuhan Perangkat Lunak

Berikut ini adalah spesifikasi minimum perangkat lunak yang digunakan dalam membangun aplikasi pemandu wisata kebun binatang bandung ini.

Tabel 3. 1 Spesifikasi Perangkat Lunak.

Komputer	***Handset***
Sistem Operasi *Windows XP*	*Platform Android versi. 3.2* ke atas.
IDE (*Integrated Development Environment*) *Eclipse Helios*	
JDK (*Java Development Kit*) *v.1.6*	
Android SDK (*Standard Developmenet Kit*) *Windows*	
ADT (*Android Development Tools*) *v.8.0.0*	
Junaio 5.0 AR Browser (*Augmented Reality Browser*)	

3.1.10.2. Kebutuhan Perangkat Keras

Komputer dan *handset* terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi-instruksi kepada perangkat keras untuk melakukan tugas tertentu.

Pada pembangunan aplikasi ini, perangkat keras yang digunakan untuk membangun aplikasi adalah sebagai berikut :

Tabel 3. 2 Spesifikasi Perangkat Keras.

Komputer	***Handset***
Processor Intel Atom CPU N270 @1.60 GHz (2CPUs)	*New Andromax i* (*OS Android 4.1.2 Jelly Beans*)
RAM 1014 MB	
Hardisk 150 GB	
Mouse dan *Keyboard*	

3.1.10.3. Spesifikasi *User*

User dibutuhkan sebagai pengguna aplikasi, spesifikasi pengguna dalam aplikasi ini dibagi menjadi dua bagian, yaitu :

1. Pengguna aplikasi pemandu wisata kebun binatang bandung (*user client*) yaitu para pengguna *handset* dengan *platform android* versi 3.2 atau lebih tinggi. Dalam menggunakan sistem ini, pengguna diharuskan terhubung dengan koneksi jaringan *internet*, dan sensor dalam *handset* aktif seperti *GPS receiver*, *digital compass*, *accelerometer*, dan kamera. Untuk bisa menggunakan aplikasi ini.
2. *Admin* sebagai pengelola data memiliki kapabilitas dalam mengelola *database* seperti menambah, mengubah, dan menghapus data melalui *web admin*.

3.1.11. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan. Analisis kebutuhan fungsional ini meliputi analisis kebutuhan sistem, analisis kebutuhan data, spesifikasi sistem dan pemodelan sistem.

3.1.11.1. Analisis Kebutuhan Sistem

Sebelum melakukan perancangan sistem, terlebih dahulu dilakukan analisis kebutuhan sistem yaitu dengan cara melakukan wawancara terhadap responden. Hal ini dimaksudkan agar dapat mengatasi ketidaksesuaian antara aplikasi yang dirancang dengan kebutuhan pengguna. Adapun kebutuhan sistem yang diperlukan antara lain :

1. Memberikan visualiasi informasi keberadaan *POI* (*Point Of Interest*) yang terdapat di sekitar pengguna..
2. Memberikan informasi data hewan, tumbuhan dan fasilitas umum beserta dengan detail dari objek-objek tersebut dengan lengkap dan rinci melalui *scan marker object* dan menampilkan informasi multimedia *augmented reality* objek yang dikunjungi.

3.1.11.2. Analisis Kebutuhan Data

Setelah melakukan proses analisis kebutuhan sistem, maka analisis selanjutnya adalah analisis kebutuhan data. Analisis ini bertujuan untuk memudahkan dalam perancangan informasi. Adapun analisis kebutuhan data yang diperlukan antara lain :

1. Data *POI*

   Data *POI* sebenarnya terbagi ke dalam 2 kategori yang meliputi data hewan, tumbuhan dan fasilitas umum. Dalam 2 kategori data *POI* tersebut memiliki informasi sebagai berikut :

   a. Hewan/Tumbuhan

      Berisi informasi mengenai hewan/tumbuhan yang meliputi *kingdom*, *filum*, *kelas*, *ordo*, *famili*, *genus*, *spesies*, *habitat*, makanan (hewan), *behavior*, penyebaran, *populasi*, status dan lain sebagainya.

   b. Fasilitas Umum

      Berisi informasi mengenai fasilitas umum yang ada di sekitar kawasan kebun binatang Bandung yang meliputi *food court*, *toilet*, ruang santai, dan lain sebagainya.

3.1.11.3. Spesifikasi Sistem

Berikut spesifikasi sistem yang dibuat :

1. Sistem dikembangkan dengan menggunakan metode *client server*. Aplikasi ini dikembangkan disisi *client/frontend*. Sedangkan disisi *server/backend* akan dikembangkan aplikasi *web* yang dibuat dengan *Framework Code Igniter* yang merupakan sumber data dari aplikasi *frontend*.
2. Konektivitas antara aplikasi pemandu wisata kebun binatang Bandung pada *platform android* ini dengan aplikasi *backend* di *server* menggunakan koneksi *protocol HTTP* dengan memanfaatkan jaringan *internet* yang ada dalam perangkat (*handset*).
3. Spesifikasi sistem di sisi *frontend* adalah sebagai berikut :
   a. Perangkat yang digunakan harus bisa digunakan dimana saja dan kapan saja selama perangkat (*handset*) memiliki jaringan *internet*.
   b. Agar aplikasi dapat berjalan dengan maksimal, sebaiknya berada dalam cakupan *line of sight* agar data yang diterima relevan dengan lokasi pengguna berada. Cakupan *line of sight* yang baik yaitu ketika berada di lingkungan terbuka atau tidak berada di ruangan tertutup.
4. Spesifikasi sistem di sisi *backend* adalah sebagai berikut :
   a. Sistem dapat memfasilitasi pengelolahan konten informasi.
   b. Sistem dapat memfasilitasi admin untuk melakukan tambah, ubah, dan hapus konten informasi hewan/tumbuhan dan fasilitas umum.

3.1.11.4. Pemodelan Sistem

Pemodelan sistem aplikasi yang dibangun dimodelkan dengan menggunakan *UML (Unified Modeling Language).* Pemodelan disini merupakan penggambaran diagram model yang terdiri dari diagram *Use Case, Activity, Class, Sequence, Statechart, Component* dan *Deployment*.

A. Use Case Diagram

Analisis yang akan dilakukan pada diagram ini meliputi identifikasi *actor*, identifikasi *use case* dan identifikasi *use case scenario.*

1. Identifikasi *Actor*

Actor menggambarkan mengenai pihak-pihak yang terlibat dan berperan dalam sistem. Berikut beberapa *actor* yang berperan terhadap sistem yang dibangun.

Tabel 3. 3 Identifikasi Actor.

No	*Actor*	Deskripsi
A-01	Pengunjung	Merupakan *actor* yang menggunakan aplikasi untuk membantu memandu dalam kunjungan wisata di kebun binatang Bandung. Akses yang diberikan melalui sisi *frontend* yang merupakan *client* berbasis *mobile.*
A-02	*Admin*	Merupakan *actor* yang menggunakan aplikasi untuk mengelola konten serta informasi yang akan disediakan oleh sistem. Akses yang diberikan melalui sisi *backend* yang merupakan *server* baik berbasis *mobile* maupun *website*.

2. Identifikasi *Use Case*

Untuk lebih jelas mengenai pembahasan ini lihat diagram *use case* yang terlampir pada lampiran A.

Berikut pemaparan identifikasi *use case* yang dibangun :

Tabel 3. 4 Identifikasi Use Case Diagram.

Use Case Backend		
No	***Use Case***	**Deskripsi**
UC-01	*Login*	Fungsionalitas untuk mendapatkan hak akses *admin* terhadap sistem.
UC-02	*Logout*	Fungsionalitas untuk menutup dan memutuskan hak akses *admin* terhadap sistem.
UC-03	Tambah *Kingdom*	Fungsionalitas untuk menambah data *kingdom*.
UC-04	Detail *Kingdom*	Fungsionalitas untuk melihat detail data *kingdom*.
UC-05	Ubah *Kingdom*	Fungsionalitas untuk mengubah data *kingdom*.
UC-06	Hapus *Kingdom*	Fungsionalitas untuk menghapus data *kingdom*.
UC-07	Tambah *Filum*	Fungsionalitas untuk menambah data *filum*.
UC-08	Detail *Filum*	Fungsionalitas untuk melihat detail data *filum*.
UC-09	Ubah *Filum*	Fungsionalitas untuk mengubah data *filum*.
UC-10	Hapus *Filum*	Fungsionalitas untuk menghapus data *filum*.
UC-11	Tambah *Kelas*	Fungsionalitas untuk menambah data *kelas*.
UC-12	Detail *Kelas*	Fungsionalitas untuk melihat detail data *kelas*.
UC-13	Ubah *Kelas*	Fungsionalitas untuk mengubah data *kelas*.
UC-14	Hapus *Kelas*	Fungsionalitas untuk menghapus data *kelas*.
UC-15	Tambah *Ordo*	Fungsionalitas untuk menambah data *ordo*.
UC-16	Detail *Ordo*	Fungsionalitas untuk melihat detail data *ordo*.
UC-17	Ubah *Ordo*	Fungsionalitas untuk mengubah data *ordo*.
UC-18	Hapus *Ordo*	Fungsionalitas untuk menghapus data *ordo*.

UC-19	Tambah *Famili*	Fungsionalitas untuk menambah data *famili*.
UC-20	Detail *Famili*	Fungsionalitas untuk melihat detail data *famili*.
UC-21	Ubah *Famili*	Fungsionalitas untuk mengubah data *famili*.
UC-22	Hapus *Famili*	Fungsionalitas untuk menghapus data *famili*.
UC-23	Tambah *Genus*	Fungsionalitas untuk menambah data *genus*.
UC-24	Detail *Genus*	Fungsionalitas untuk melihat detail data *genus*.
UC-25	Ubah *Genus*	Fungsionalitas untuk mengubah data *genus*.
UC-26	Hapus *Genus*	Fungsionalitas untuk menghapus data *genus*.
UC-27	Tambah *Spesies*	Fungsionalitas untuk menambah data *spesies*.
UC-28	Detail *Spesies*	Fungsionalitas untuk melihat detail data *spesies*.
UC-29	Ubah *Spesies*	Fungsionalitas untuk mengubah data *spesies*.
UC-30	Hapus *Spesies*	Fungsionalitas untuk menghapus data *spesies*.
UC-31	Tambah Hewan	Fungsionalitas untuk menambah data hewan.
UC-32	Detail Hewan	Fungsionalitas untuk melihat detail data hewan.
UC-33	Ubah Hewan	Fungsionalitas untuk mengubah data hewan.
UC-34	Hapus Hewan	Fungsionalitas untuk menghapus data hewan.
UC-35	Tambah Tumbuhan	Fungsionalitas untuk menambah data tumbuhan.
UC-36	Detail Tumbuhan	Fungsionalitas untuk melihat detail data tumbuhan.
UC-37	Ubah Tumbuhan	Fungsionalitas untuk mengubah data tumbuhan.
UC-38	Hapus Tumbuhan	Fungsionalitas untuk menghapus data tumbuhan.
UC-39	Tambah Fasilitas Umum	Fungsionalitas untuk menambah data fasilitas umum.
UC-40	Detail Fasilitas Umum	Fungsionalitas untuk melihat detail data fasilitas umum.
UC-41	Ubah Fasilitas Umum	Fungsionalitas untuk mengubah data fasilitas umum.
UC-42	Hapus Fasilitas Umum	Fungsionalitas untuk menghapus data fasilitas umum.

UC-43	Tambah *POI*	Fungsionalitas untuk menambah data *POI*.
UC-44	Detail *POI*	Fungsionalitas untuk melihat detail data *POI*.
UC-45	Ubah *POI*	Fungsionalitas untuk mengubah data *POI*.
UC-46	Hapus *POI*	Fungsionalitas untuk menghapus data *POI*.
UC-47	Identifikasi Koordinat	Fungsionalitas untuk mengidentifikasi koordinat lokasi posisi pengguna berbasis *mobile* yang berupa informasi lintang *(latitude)* dan bujur *(longitude).*
UC-48	Tambah Gambar	Fungsionalitas untuk menambahkan gambar pada data hewan/tumbuhan/fasilitas umum.
UC-49	Tambah Suara	Fungsionalitas untuk menambahkan suara pada data hewan/tumbuhan/fasilitas umum.
UC-50	Tambah Video	Fungsionalitas untuk menambahkan video pada data hewan/tumbuhan/fasilitas umum.
UC-51	Tambah Model	Fungsionalitas untuk menambahkan model 3D pada data hewan/tumbuhan/fasilitas umum.
UC-52	Tampil Hewan Sekitar	Fungsionalitas untuk menampilkan keberadaan hewan di sekitar pengguna.
UC-52	Tampil Hewan Sekitar	Fungsionalitas untuk menampilkan keberadaan hewan di sekitar pengguna.
UC-53	Tampil Tumbuhan Sekitar	Fungsionalitas untuk menampilkan keberadaan tumbuhan di sekitar pengguna.
UC-54	Tampil Fasilitas Umum Sekitar	Fungsionalitas untuk menampilkan keberadaan fasilitas umum di sekitar pengguna.
UC-55	*Scan Marker*	Fungsionalitas untuk melakukan pengidentifikasian marker di lokasi *POI*.
UC-56	Lihat Detail	Fungsionalitas untuk menampilkan informasi detail dari *POI*.
UC-57	Putar Video	Fungsionalitas untuk memutar video yang berkaitan dengan *POI*.
UC-58	Putar Suara	Fungsionalitas untuk memutar suara yang berkaitan dengan *POI*.
UC-59	Tampil Model	Fungsionalitas untuk menampilkan model yang berkaitan dengan *POI*.

3. Identifikasi *Use Case Scenario*

Use case scenario merupakan penjelasan rinci bagian-bagian *use case* yang menggambarkan interaksi antara perintah *user* dengan reaksi sistem sebagai berikut :

Tabel 3. 5 Skenario Use Case Login.

Use Case Name **:** *Login*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk mengizinkan *admin* memasuki sistem.	
Triggering Event **:** *Admin* menginputkan alamat *website backend*, memasukkan *username* & *password* dan menekan tombol *submit.*	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* mengisi *username* dan *password* pada *form login* dan menekan tombol *submit.*	*Form Login.*
2. Sistem mengecek kevalidan data *login.*	Data *admin.*
3. *Admin* memasuki sistem.	Halaman Utama.
Preconditions **:** *Admin* telah memasuki alamat *website backend.*	
Postconditions **:** *Admin* berhasil memasuki sistem *backend*.	
Assumptions **:** *Admin* menginputkan isian *form login* dengan data yang *valid.*	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 6 Skenario Use Case Logout.

Use Case Name **:** *Logout*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk memutuskan hak akses *admin* pada sistem.	
Triggering Event **:** *Admin* menekan tombol keluar.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol keluar.	Halaman Utama.
2. Sistem memutuskan hak akses dan mengembalikan tampilan ke halaman *login*.	*Login Form.*
Preconditions **:** *Admin* telah memasuki sistem *backend*.	
Postconditions **:** *Admin* berhasil keluar sistem *backend*.	
Assumptions **:** *Session login* masih aktif.	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 7 Skenario Use Case Tambah Kingdom.

Use Case Name **:** Tambah *Kingdom*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menambahkan data *kingdom* baru.	
Triggering Event **:** *Admin* mengisi isian data *kingdom* dan menekan tombol *submit.*	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol tambah *kingdom.*	*List Kingdom.*

2. Sistem menampilkan *form* tambah *kingdom*.	*Form Kingdom.*
3. *Admin* menginputkan isian *form kingdom* dan menekan tombol *submit.*	*Form Kingdom.*
4. Sistem memproses penambahan data *kingdom.*	*Form Kingdom, Data Kingdom.*
5. Sistem menampilkan pesan *"Kingdom baru berhasil ditambahkan"*.	*Form Kingdom.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *kingdom*.	
Postconditions **:** *Admin* berhasil menambahkan data *kingdom* baru.	
Assumptions **:** *Admin* menginputkan isian *form kingdom* dengan data yang *valid.*	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 8 Skenario Use Case Detail Kingdom.

Use Case Name **:** Detail *Kingdom*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menampilkan detail data *kingdom.*	
Triggering Event **:** *Admin* menekan tombol detail..	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol detail *kingdom.*	*List Kingdom.*
2. Sistem menampilkan informasi detail *kingdom*.	*Detail Kingdom.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *kingdom*.	
Postconditions **:** Detail *kingdom* berhasil ditampilkan.	
Assumptions **:** Detail *kingdom* yang ditampilkan merupakan data yang sudah tersimpan di *database.*	

Priority	**:** *Medium*.
Risk	**:** *Medium.*

Tabel 3. 9 Skenario Use Case Ubah Kingdom.

Use Case Name	**:** Ubah *Kingdom*
Area	**:** *Backend*
Actor(s)	**:** *Admin*
Description	**:** Fungsional untuk mengubah data *kingdom* di *database.*
Triggering Event	**:** *Admin* mengubah data *kingdom* dan menekan tombol *submit.*
Trigger Type	**:** *External*
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol ubah *kingdom.*	*List Kingdom.*
2. Sistem menampilkan *form* ubah *kingdom*.	*Form Kingdom.*
3. *Admin* mengubah isian *form kingdom* dan menekan tombol *submit.*	*Form Kingdom.*
4. Sistem memproses perubahan data *kingdom.*	*Form Kingdom, Data Kingdom.*
5. Sistem menampilkan pesan "*Ubah data kingdom berhasil"*.	*Form Kingdom.*
Preconditions	**:** *Admin* telah memasuki halaman pengolahan *kingdom*.
Postconditions	**:** *Admin* berhasil mengubah data *kingdom*.
Assumptions	**:** *Admin* mengubah isian *form kingdom* dengan data yang *valid.*
Priority	**:** *Medium*.
Risk	**:** *Medium.*

Tabel 3. 10 Skenario Use Case Hapus Kingdom.

Use Case Name **:** Hapus *Kingdom*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menghapus data *kingdom* .	
Triggering Event **:** *Admin* menekan tombol hapus.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol hapus *kingdom.*	*List Kingdom.*
2. Sistem memproses penghapusan data *kingdom.*	*List Kingdom, Data Kingdom.*
3. Sistem menampilkan pesan *"Data kingdom telah berhasil dihapus"*.	*List Kingdom.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *kingdom*.	
Postconditions **:** *Admin* berhasil menghapus data *kingdom*.	
Assumptions **:** Data *kingdom* yang dihapus merupakan data yang tersimpan di *database*..	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 11 Skenario Use Case Tambah Filum.

Use Case Name **:** Tambah *Filum*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menambahkan data *filum* baru.	
Triggering Event **:** *Admin* mengisi isian data *filum* dan menekan tombol *submit.*	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***

1. *Admin* menekan tombol tambah *filum.*	*List Filum.*
2. Sistem menampilkan *form* tambah *filum*.	*Form Filum.*
3. *Admin* menginputkan isian *form filum* dan menekan tombol *submit.*	*Form Filum.*
4. Sistem memproses penambahan data *filum.*	*Form Filum, Data Filum.*
5. Sistem menampilkan pesan *"Filum baru berhasil ditambahkan"*.	*Form Filum.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *filum*.	
Postconditions **:** *Admin* berhasil menambahkan data *filum* baru.	
Assumptions **:** *Admin* menginputkan isian *form filum* dengan data yang *valid*.	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 12 Skenario Use Case Detail Filum.

Use Case Name **:** Detail *Filum*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menampilkan detail data *filum.*	
Triggering Event **:** *Admin* menekan tombol detail..	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol detail *filum.*	*List Filum.*
2. Sistem menampilkan informasi detail *filum*.	*Detail Filum.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *filum*.	
Postconditions **:** Detail *filum* berhasil ditampilkan.	
Assumptions **:** Detail *filum* yang ditampilkan merupakan data yang sudah tersimpan di *database.*	

Priority	**:** *Medium*.
Risk	**:** *Medium.*

Tabel 3. 13 Skenario Use Case Ubah Filum.

Use Case Name	**:** Ubah *Filum*
Area	**:** *Backend*
Actor(s)	**:** *Admin*
Description	**:** Fungsional untuk mengubah data *filum* di *database.*
Triggering Event	**:** *Admin* mengubah data *filum* dan menekan tombol *submit.*
Trigger Type	**:** *External*
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol ubah *filum.*	*List Filum.*
2. Sistem menampilkan *form* ubah *filum*.	*Form Filum.*
3. *Admin* mengubah isian *form filum* dan menekan tombol *submit.*	*Form Filum.*
4. Sistem memproses perubahan data *filum.*	*Form Filum, Data Filum.*
5. Sistem menampilkan pesan "*Ubah data filum berhasil*".	*Form Filum.*
Preconditions	**:** *Admin* telah memasuki halaman pengolahan *filum*.
Postconditions	**:** *Admin* berhasil mengubah data *filum*.
Assumptions	**:** *Admin* mengubah isian *form filum* dengan data yang *valid.*
Priority	**:** *Medium*.
Risk	**:** *Medium.*

Tabel 3. 14 Skenario Use Case Hapus Filum.

Use Case Name **:** Hapus *Filum*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menghapus data *filum.*	
Triggering Event **:** *Admin* menekan tombol hapus.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol hapus *filum.*	*List Filum.*
2. Sistem memproses penghapusan data *filum.*	*List Filum, Data Filum.*
3. Sistem menampilkan pesan *"Data filum telah berhasil dihapus"*.	*List Filum.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *filum.*	
Postconditions **:** *Admin* berhasil menghapus data *filum*.	
Assumptions **:** Data *filum* yang dihapus merupakan data yang tersimpan di *database*..	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 15 Skenario Use Case Tambah Kelas.

Use Case Name **:** Tambah *Kelas*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menambahkan data *kelas* baru.	
Triggering Event **:** *Admin* mengisi isian data *kelas* dan menekan tombol *submit.*	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***

1. *Admin* menekan tombol tambah *kelas.*	*List Kelas.*
2. Sistem menampilkan *form* tambah *kelas*.	*Form Kelas.*
3. *Admin* menginputkan isian *form kelas* dan menekan tombol *submit.*	*Form Kelas.*
4. Sistem memproses penambahan data *kelas.*	*Form Kelas, Data Kelas.*
5. Sistem menampilkan pesan *"Kelas baru berhasil ditambahkan"*.	*Form Kelas.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *kelas*.	
Postconditions **:** *Admin* berhasil menambahkan data *kelas* baru.	
Assumptions **:** *Admin* menginputkan isian *form kelas* dengan data yang *valid*.	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 16 Skenario Use Case Detail Kelas.

Use Case Name **:** Detail *Kelas*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menampilkan detail data *kelas.*	
Triggering Event **:** *Admin* menekan tombol detail..	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol detail *kelas.*	*List Kelas.*
2. Sistem menampilkan informasi detail *kelas*.	*Detail Kelas.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *kelas*.	
Postconditions **:** Detail *kelas* berhasil ditampilkan.	
Assumptions **:** Detail *kelas* yang ditampilkan merupakan data yang sudah tersimpan di *database.*	

Priority	**:** *Medium.*
Risk	**:** *Medium.*

Tabel 3. 17 Skenario Use Case Ubah Kelas.

Use Case Name	**:** Ubah *Kelas*
Area	**:** *Backend*
Actor(s)	**:** *Admin*
Description	**:** Fungsional untuk mengubah data *kelas* di *database.*
Triggering Event	**:** *Admin* mengubah data *kelas* dan menekan tombol *submit.*
Trigger Type	**:** *External*
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol ubah *kelas* .	*List Kelas.*
2. Sistem menampilkan *form* ubah *kelas* .	*Form Kelas.*
3. *Admin* mengubah isian *form kelas* dan menekan tombol *submit.*	*Form Kelas.*
4. Sistem memproses perubahan data *kelas* .	*Form Kelas, Data Kelas.*
5. Sistem menampilkan pesan *"Ubah data kelas berhasil".*	*Form Kelas.*
Preconditions	**:** *Admin* telah memasuki halaman pengolahan *kelas* .
Postconditions	**:** *Admin* berhasil mengubah data *kelas* .
Assumptions	**:** *Admin* mengubah isian *form kelas* dengan data yang *valid.*
Priority	**:** *Medium.*
Risk	**:** *Medium.*

Tabel 3. 18 Skenario Use Case Hapus Kelas.

Use Case Name **:** Hapus *Kelas*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menghapus data *kelas.*	
Triggering Event **:** *Admin* menekan tombol hapus.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol hapus *kelas.*	*List Kelas.*
2. Sistem memproses penghapusan data *kelas.*	*List Kelas, Data Kelas.*
3. Sistem menampilkan pesan *"Data kelas telah berhasil dihapus"*.	*List Kelas.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *kelas*.	
Postconditions **:** *Admin* berhasil menghapus data *kelas*.	
Assumptions **:** Data *kelas* yang dihapus merupakan data yang tersimpan di *database*..	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 19 Skenario Use Case Tambah Ordo.

Use Case Name **:** Tambah *Ordo*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menambahkan data *ordo* baru.	
Triggering Event **:** *Admin* mengisi isian data *ordo* dan menekan tombol *submit.*	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***

1. *Admin* menekan tombol tambah *ordo.*	*List Ordo.*
2. Sistem menampilkan *form* tambah *ordo*.	*Form Ordo.*
3. *Admin* menginputkan isian *form ordo* dan menekan tombol *submit.*	*Form Ordo.*
4. Sistem memproses penambahan data *ordo.*	*Form Ordo, Data Ordo.*
5. Sistem menampilkan pesan *"Ordo baru berhasil ditambahkan"*.	*Form Ordo.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *ordo*.	
Postconditions **:** *Admin* berhasil menambahkan data *ordo* baru.	
Assumptions **:** *Admin* menginputkan isian *form ordo* dengan data yang *valid.*	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 20 Skenario Use Case Detail Ordo.

Use Case Name **:** Detail *Ordo*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menampilkan detail data *ordo.*	
Triggering Event **:** *Admin* menekan tombol detail..	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol detail *ordo.*	*List Ordo.*
2. Sistem menampilkan informasi detail *ordo*.	*Detail Ordo.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *ordo*.	
Postconditions **:** Detail *ordo* berhasil ditampilkan.	
Assumptions **:** Detail *ordo* yang ditampilkan merupakan data yang sudah tersimpan di *database.*	

Priority	**:** *Medium.*
Risk	**:** *Medium.*

Tabel 3. 21 Skenario Use Case Ubah Ordo.

Use Case Name	**:** Ubah *Ordo*
Area	**:** *Backend*
Actor(s)	**:** *Admin*
Description	**:** Fungsional untuk mengubah data *ordo* di *database.*
Triggering Event	**:** *Admin* mengubah data *ordo* dan menekan tombol *submit.*
Trigger Type	**:** *External*
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol ubah *ordo* .	*List Ordo.*
2. Sistem menampilkan *form* ubah *ordo* .	*Form Ordo.*
3. *Admin* mengubah isian *form ordo* dan menekan tombol *submit.*	*Form Ordo.*
4. Sistem memproses perubahan data *ordo* .	*Form Ordo, Data Ordo.*
5. Sistem menampilkan pesan "*Ubah data ordo berhasil".*	*Form Ordo.*
Preconditions	**:** *Admin* telah memasuki halaman pengolahan *ordo* .
Postconditions	**:** *Admin* berhasil mengubah data *ordo* .
Assumptions	**:** *Admin* mengubah isian *form ordo* dengan data yang *valid.*
Priority	**:** *Medium.*
Risk	**:** *Medium.*

Tabel 3. 22 Skenario Use Case Hapus Ordo.

Use Case Name **:** Hapus *Ordo*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menghapus data *ordo* .	
Triggering Event **:** *Admin* menekan tombol hapus.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol hapus *ordo* .	*List Ordo.*
2. Sistem memproses penghapusan data *ordo.*	*List Ordo, Data Ordo.*
3. Sistem menampilkan pesan *"Data ordo telah berhasil dihapus".*	*List Ordo.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *ordo* .	
Postconditions **:** *Admin* berhasil menghapus data *ordo* .	
Assumptions **:** Data *ordo* yang dihapus merupakan data yang tersimpan di *database*..	
Priority **:** *Medium.*	
Risk **:** *Medium.*	

Tabel 3. 23 Skenario Use Case Tambah Famili.

Use Case Name **:** Tambah *Famili*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menambahkan data *famili* baru.	
Triggering Event **:** *Admin* mengisi isian data *famili* dan menekan tombol *submit.*	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***

1. *Admin* menekan tombol tambah *famili.*	*List Famili.*
2. Sistem menampilkan *form* tambah *famili*.	*Form Famili.*
3. *Admin* menginputkan isian *form famili* dan menekan tombol *submit.*	*Form Famili.*
4. Sistem memproses penambahan data *famili.*	*Form Famili, Data Famili.*
5. Sistem menampilkan pesan *"Famili baru berhasil ditambahkan"*.	*Form Famili.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *famili*.	
Postconditions **:** *Admin* berhasil menambahkan data *famili* baru.	
Assumptions **:** *Admin* menginputkan isian *form famili* dengan data yang *valid*.	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 24 Skenario Use Case Detail Famili.

Use Case Name **:** Detail *Famili*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menampilkan detail data *famili.*	
Triggering Event **:** *Admin* menekan tombol detail..	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol detail *famili.*	*List Famili.*
2. Sistem menampilkan informasi detail *famili*.	*Detail Famili.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *famili*.	
Postconditions **:** Detail *famili* berhasil ditampilkan.	
Assumptions **:** Detail *famili* yang ditampilkan merupakan data yang sudah tersimpan di *database.*	

Priority	**:** *Medium.*
Risk	**:** *Medium.*

Tabel 3. 25 Skenario Use Case Ubah Famili.

Use Case Name	**:** Ubah *Famili*
Area	**:** *Backend*
Actor(s)	**:** *Admin*
Description	**:** Fungsional untuk mengubah data *famili* di *database.*
Triggering Event	**:** *Admin* mengubah data *famili* dan menekan tombol *submit.*
Trigger Type	**:** *External*
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol ubah *famili* .	*List Famili.*
2. Sistem menampilkan *form* ubah *famili* .	*Form Famili.*
3. *Admin* mengubah isian *form famili* dan menekan tombol *submit.*	*Form Famili.*
4. Sistem memproses perubahan data *famili* .	*Form Famili, Data Famili.*
5. Sistem menampilkan pesan *"Ubah data famili berhasil".*	*Form Famili.*
Preconditions	**:** *Admin* telah memasuki halaman pengolahan *famili* .
Postconditions	**:** *Admin* berhasil mengubah data *famili* .
Assumptions	**:** *Admin* mengubah isian *form famili* dengan data yang *valid.*
Priority	**:** *Medium.*
Risk	**:** *Medium.*

Tabel 3. 26 Skenario Use Case Hapus Famili.

Use Case Name **:** Hapus *Famili*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menghapus data *famili* .	
Triggering Event **:** *Admin* menekan tombol hapus.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol hapus *famili.*	*List Famili.*
2. Sistem memproses penghapusan data *famili.*	*List Famili, Data Famili.*
3. Sistem menampilkan pesan *"Data famili telah berhasil dihapus"*.	*List Famili.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *famili*.	
Postconditions **:** *Admin* berhasil menghapus data *famili*.	
Assumptions **:** Data *famili* yang dihapus merupakan data yang tersimpan di *database*..	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 27 Skenario Use Case Tambah Genus.

Use Case Name **:** Tambah *Genus*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menambahkan data *genus* baru.	
Triggering Event **:** *Admin* mengisi isian data *genus* dan menekan tombol *submit.*	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***

1. *Admin* menekan tombol tambah *genus.*	*List Genus.*
2. Sistem menampilkan *form* tambah *genus*.	*Form Genus.*
3. *Admin* menginputkan isian *form genus* dan menekan tombol *submit.*	*Form Genus.*
4. Sistem memproses penambahan data *genus.*	*Form Genus, Data Genus.*
5. Sistem menampilkan pesan *"Genus baru berhasil ditambahkan"*.	*Form Genus.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *genus*.	
Postconditions **:** *Admin* berhasil menambahkan data *genus* baru.	
Assumptions **:** *Admin* menginputkan isian *form genus* dengan data yang *valid*.	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 28 Skenario Use Case Detail Genus.

Use Case Name **:** Detail *Genus*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menampilkan detail data *genus.*	
Triggering Event **:** *Admin* menekan tombol detail..	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol detail *genus.*	*List Genus.*
2. Sistem menampilkan informasi detail *genus*.	*Detail Genus.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *genus*.	
Postconditions **:** Detail *genus* berhasil ditampilkan.	
Assumptions **:** Detail *genus* yang ditampilkan merupakan data yang sudah tersimpan di *database.*	

Priority	**:** *Medium.*
Risk	**:** *Medium.*

Tabel 3. 29 Skenario Use Case Ubah Genus.

Use Case Name	**:** Ubah *Genus*
Area	**:** *Backend*
Actor(s)	**:** *Admin*
Description	**:** Fungsional untuk mengubah data *genus* di *database.*
Triggering Event	**:** *Admin* mengubah data *genus* dan menekan tombol *submit.*
Trigger Type	**:** *External*
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol ubah *genus* .	*List Genus.*
2. Sistem menampilkan *form* ubah *genus* .	*Form Genus.*
3. *Admin* mengubah isian *form genus* dan menekan tombol *submit.*	*Form Genus.*
4. Sistem memproses perubahan data *genus* .	*Form Genus, Data Genus.*
5. Sistem menampilkan pesan *"Ubah data genus berhasil".*	*Form Genus.*
Preconditions	**:** *Admin* telah memasuki halaman pengolahan *genus* .
Postconditions	**:** *Admin* berhasil mengubah data *genus* .
Assumptions	**:** *Admin* mengubah isian *form genus* dengan data yang *valid.*
Priority	**:** *Medium.*
Risk	**:** *Medium.*

Tabel 3. 30 Skenario Use Case Hapus Genus.

Use Case Name **:** Hapus *Genus*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menghapus data *genus* .	
Triggering Event **:** *Admin* menekan tombol hapus.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol hapus *genus.*	*List Genus.*
2. Sistem memproses penghapusan data *genus.*	*List Genus, Data Genus.*
3. Sistem menampilkan pesan *"Data genus telah berhasil dihapus"*.	*List Genus.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *genus*.	
Postconditions **:** *Admin* berhasil menghapus data *genus*.	
Assumptions **:** Data *genus* yang dihapus merupakan data yang tersimpan di *database*..	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 31 Skenario Use Case Tambah Spesies.

Use Case Name **:** Tambah *Spesies*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menambahkan data *spesies* baru.	
Triggering Event **:** *Admin* mengisi isian data *spesies* dan menekan tombol *submit.*	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***

1. *Admin* menekan tombol tambah *spesies.*	*List Spesies.*
2. Sistem menampilkan *form* tambah *spesies*.	*Form Spesies.*
3. *Admin* menginputkan isian *form spesies* dan menekan tombol *submit.*	*Form Spesies.*
4. Sistem memproses penambahan data *spesies.*	*Form Spesies, Data Spesies.*
5. Sistem menampilkan pesan *"Spesies baru berhasil ditambahkan".*	*Form Spesies.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *spesies*.	
Postconditions **:** *Admin* berhasil menambahkan data *spesies* baru.	
Assumptions **:** *Admin* menginputkan isian *form spesies* dengan data yang *valid.*	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 32 Skenario Use Case Detail Spesies.

Use Case Name **:** Detail *Spesies*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menampilkan detail data *spesies.*	
Triggering Event **:** *Admin* menekan tombol detail..	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol detail *spesies.*	*List Spesies.*
2. Sistem menampilkan informasi detail *spesies.*	*Detail Spesies.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *spesies*.	
Postconditions **:** Detail *spesies* berhasil ditampilkan.	
Assumptions **:** Detail *spesies* yang ditampilkan merupakan data yang sudah tersimpan di	

database.	
Priority	**:** *Medium.*
Risk	**:** *Medium.*

Tabel 3. 33 Skenario Use Case Ubah Spesies.

Use Case Name	**:** Ubah *Spesies*
Area	**:** *Backend*
Actor(s)	**:** *Admin*
Description	**:** Fungsional untuk mengubah data *spesies* di *database.*
Triggering Event	**:** *Admin* mengubah data *genus* dan menekan tombol *submit.*
Trigger Type	**:** *External*
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol ubah *spesies* .	*List Spesies.*
2. Sistem menampilkan *form* ubah *spesies* .	*Form Spesies.*
3. *Admin* mengubah isian *form spesies* dan menekan tombol *submit.*	*Form Spesies.*
4. Sistem memproses perubahan data *spesies* .	*Form Spesies, Data Spesies.*
5. Sistem menampilkan pesan *"Ubah data spesies berhasil"*.	*Form Spesies.*
Preconditions	**:** *Admin* telah memasuki halaman pengolahan *spesies* .
Postconditions	**:** *Admin* berhasil mengubah data *spesies* .
Assumptions	**:** *Admin* mengubah isian *form spesies* dengan data yang *valid.*
Priority	**:** *Medium.*
Risk	**:** *Medium.*

Tabel 3. 34 Skenario Use Case Hapus Spesies.

Use Case Name **:** Hapus *Spesies*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menghapus data *spesies* .	
Triggering Event **:** *Admin* menekan tombol hapus.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol hapus *spesies.*	*List Spesies.*
2. Sistem memproses penghapusan data *spesies.*	*List Spesies, Data Spesies.*
3. Sistem menampilkan pesan *"Data spesies telah berhasil dihapus"*.	*List Spesies.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *spesies*.	
Postconditions **:** *Admin* berhasil menghapus data *spesies*.	
Assumptions **:** Data *spesies* yang dihapus merupakan data yang tersimpan di *database*..	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 35 Skenario Use Case Tambah Hewan.

Use Case Name **:** Tambah Hewan	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menambahkan data hewan baru.	
Triggering Event **:** *Admin* mengisi isian data hewan dan menekan tombol *submit.*	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***

1. *Admin* menekan tombol tambah hewan.	*List* Hewan.
2. Sistem menampilkan *form* tambah hewan.	*Form* Hewan.
3. *Admin* menginputkan isian *form* hewan dan menekan tombol *submit.*	*Form* Hewan.
4. Sistem memproses penambahan data hewan.	*Form* Hewan, *Data* Hewan.
5. Sistem menampilkan pesan "*Hewan baru berhasil ditambahkan".*	*Form* Hewan.
Preconditions **:** *Admin* telah memasuki halaman pengolahan hewan.	
Postconditions **:** *Admin* berhasil menambahkan data hewan baru.	
Assumptions **:** *Admin* menginputkan isian *form* hewan dengan data yang *valid.*	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 36 Skenario Use Case Detail Hewan.

Use Case Name **:** Detail Hewan	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menampilkan detail data hewan.	
Triggering Event **:** *Admin* menekan tombol detail..	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol detail hewan.	*List* Hewan.
2. Sistem menampilkan informasi detail hewan.	*Detail* Hewan.
Preconditions **:** *Admin* telah memasuki halaman pengolahan hewan.	
Postconditions **:** Detail hewan berhasil ditampilkan.	
Assumptions **:** Detail hewan yang ditampilkan merupakan data yang sudah tersimpan di	

database.	
Priority	**:** *Medium*.
Risk	**:** *Medium.*

Tabel 3. 37 Skenario Use Case Ubah Hewan.

Use Case Name	**:** Ubah Hewan
Area	**:** *Backend*
Actor(s)	**:** *Admin*
Description	**:** Fungsional untuk mengubah data hewan di *database*.
Triggering Event	**:** *Admin* mengubah data *genus* dan menekan tombol *submit.*
Trigger Type	**:** *External*
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol ubah hewan.	*List* Hewan.
2. Sistem menampilkan *form* ubah hewan.	*Form* Hewan.
3. *Admin* mengubah isian *form* hewan dan menekan tombol *submit.*	*Form* Hewan.
4. Sistem memproses perubahan data hewan.	*Form* Hewan, *Data* Hewan.
5. Sistem menampilkan pesan "*Ubah data* hewan *berhasil"*.	*Form* Hewan.
Preconditions	**:** *Admin* telah memasuki halaman pengolahan hewan.
Postconditions	**:** *Admin* berhasil mengubah data hewan.
Assumptions	**:** *Admin* mengubah isian *form* hewan dengan data yang *valid.*
Priority	**:** *Medium*.
Risk	**:** *Medium.*

Tabel 3. 38 Skenario Use Case Hapus Hewan.

Use Case Name **:** Hapus Hewan	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menghapus data hewan.	
Triggering Event **:** *Admin* menekan tombol hapus.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol hapus hewan.	*List* Hewan.
2. Sistem memproses penghapusan data hewan.	*List* Hewan, *Data* Hewan.
3. Sistem menampilkan pesan *"Data* hewan *telah berhasil dihapus"*.	*List* Hewan.
Preconditions **:** *Admin* telah memasuki halaman pengolahan hewan.	
Postconditions **:** *Admin* berhasil menghapus data hewan.	
Assumptions **:** Data hewan yang dihapus merupakan data yang tersimpan di *database*..	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 39 Skenario Use Case Tambah Tumbuhan.

Use Case Name **:** Tambah Tumbuhan	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menambahkan data tumbuhan baru.	
Triggering Event **:** *Admin* mengisi isian data hewan dan menekan tombol *submit.*	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***

1. *Admin* menekan tombol tambah tumbuhan.	*List* Tumbuhan.
2. Sistem menampilkan *form* tambah tumbuhan.	*Form* Tumbuhan.
3. *Admin* menginputkan isian *form* tumbuhan dan menekan tombol *submit.*	*Form* Tumbuhan.
4. Sistem memproses penambahan data tumbuhan.	*Form* Tumbuhan, *Data* Tumbuhan.
5. Sistem menampilkan pesan *"Tumbuhan baru berhasil ditambahkan"*.	*Form* Tumbuhan.
Preconditions **:** *Admin* telah memasuki halaman pengolahan tumbuhan.	
Postconditions **:** *Admin* berhasil menambahkan data tumbuhan baru.	
Assumptions **:** *Admin* menginputkan isian *form* tumbuhan dengan data yang *valid.*	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 40 Skenario Use Case Detail Tumbuhan.

Use Case Name **:** Detail Tumbuhan	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menampilkan detail data tumbuhan.	
Triggering Event **:** *Admin* menekan tombol detail..	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol detail tumbuhan.	*List* Tumbuhan.
2. Sistem menampilkan informasi detail tumbuhan.	*Detail* Tumbuhan.
Preconditions **:** *Admin* telah memasuki halaman pengolahan tumbuhan.	
Postconditions **:** Detail tumbuhan berhasil ditampilkan.	
Assumptions **:** Detail tumbuhan yang ditampilkan merupakan data yang sudah tersimpan	

di *database.*
Priority **:** *Medium*.
Risk **:** *Medium.*

Tabel 3. 41 Skenario Use Case Ubah Tumbuhan.

Use Case Name **:** Ubah Tumbuhan	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk mengubah data tumbuhan di *database.*	
Triggering Event **:** *Admin* mengubah data tumbuhan dan menekan tombol *submit.*	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol ubah tumbuhan.	*List* Tumbuhan.
2. Sistem menampilkan *form* ubah tumbuhan.	*Form* Tumbuhan.
3. *Admin* mengubah isian *form* tumbuhan dan menekan tombol *submit.*	*Form* Tumbuhan.
4. Sistem memproses perubahan data tumbuhan.	*Form* Tumbuhan, *Data* Tumbuhan.
5. Sistem menampilkan pesan "*Ubah data tumbuhan berhasil"*.	*Form* Tumbuhan.
Preconditions **:** *Admin* telah memasuki halaman pengolahan tumbuhan.	
Postconditions **:** *Admin* berhasil mengubah data tumbuhan.	
Assumptions **:** *Admin* mengubah isian *form* tumbuhan dengan data yang *valid.*	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 42 Skenario Use Case Hapus Tumbuhan.

Use Case Name **:** Hapus Tumbuhan	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menghapus data tumbuhan.	
Triggering Event **:** *Admin* menekan tombol hapus.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol hapus tumbuhan.	*List* Tumbuhan.
2. Sistem memproses penghapusan data tumbuhan.	*List* Tumbuhan, *Data* Tumbuhan.
3. Sistem menampilkan pesan *"Data tumbuhan telah berhasil dihapus"*.	*List* Tumbuhan.
Preconditions **:** *Admin* telah memasuki halaman pengolahan tumbuhan.	
Postconditions **:** *Admin* berhasil menghapus data tumbuhan.	
Assumptions **:** Data tumbuhan yang dihapus merupakan data yang tersimpan di *database*..	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 43 Skenario Use Case Tambah Fasilitas Umum.

Use Case Name **:** Tambah Fasilitas Umum	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menambahkan data fasilitas umum baru.	
Triggering Event **:** *Admin* mengisi isian data fasilitas umum dan menekan tombol *submit.*	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***

1. *Admin* menekan tombol tambah fasilitas umum.	*List* Fasilitas Umum.
2. Sistem menampilkan *form* tambah fasilitas umum.	*Form* Fasilitas Umum.
3. *Admin* menginputkan isian *form* fasilitas umum dan menekan tombol *submit.*	*Form* Fasilitas Umum.
4. Sistem memproses penambahan data fasilitas umum.	*Form* Fasilitas Umum, *Data* Fasilitas Umum.
5. Sistem menampilkan pesan *"Fasilitas umum baru berhasil ditambahkan"*.	*Form* Fasilitas Umum.
Preconditions **:** *Admin* telah memasuki halaman pengolahan fasilitas umum.	
Postconditions **:** *Admin* berhasil menambahkan data fasilitas umum baru.	
Assumptions **:** *Admin* menginputkan isian *form* fasilitas umum dengan data yang *valid.*	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 44 Skenario Use Case Detail Fasilitas Umum.

Use Case Name **:** Detail Fasilitas Umum	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menampilkan detail data fasilitas umum.	
Triggering Event **:** *Admin* menekan tombol detail..	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol detail fasilitas umum.	*List* Fasilitas Umum.
2. Sistem menampilkan informasi detail fasilitas umum.	*Detail* Fasilitas Umum.
Preconditions **:** *Admin* telah memasuki halaman pengolahan fasilitas umum.	

Postconditions	**:** Detail fasilitas umum berhasil ditampilkan.
Assumptions	**:** Detail fasilitas umum yang ditampilkan merupakan data yang sudah tersimpan di *database.*
Priority	**:** *Medium*.
Risk	**:** *Medium.*

Tabel 3. 45 Skenario Use Case Ubah Fasilitas Umum.

Use Case Name	**:** Ubah Fasilitas Umum
Area	**:** *Backend*
Actor(s)	**:** *Admin*
Description	**:** Fungsional untuk mengubah data fasilitas umum di *database*.
Triggering Event	**:** *Admin* mengubah data fasilitas umum dan menekan tombol *submit.*
Trigger Type	**:** *External*
Steps Performed (Main Path)	***Information for Steps***
6. *Admin* menekan tombol ubah fasilitas umum.	*List* Fasilitas Umum.
7. Sistem menampilkan *form* ubah fasilitas umum.	*Form* Fasilitas Umum.
8. *Admin* mengubah isian *form* fasilitas umum dan menekan tombol *submit.*	*Form* Fasilitas Umum.
9. Sistem memproses perubahan data fasilitas umum.	*Form* Fasilitas Umum, *Data* Fasilitas Umum.
10. Sistem menampilkan pesan "*Ubah data fasilitas umum berhasil*".	*Form* Fasilitas Umum.
Preconditions	**:** *Admin* telah memasuki halaman pengolahan fasilitas umum.
Postconditions	**:** *Admin* berhasil mengubah data fasilitas umum.
Assumptions	**:** *Admin* mengubah isian *form* fasilitas umum dengan data yang *valid.*
Priority	**:** *Medium*.
Risk	**:** *Medium.*

Tabel 3. 46 Skenario Use Case Hapus Fasilitas Umum.

Use Case Name **:** Hapus Fasilitas Umum	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menghapus data fasilitas umum.	
Triggering Event **:** *Admin* menekan tombol hapus.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol hapus fasilitas umum.	*List* Fasilitas Umum.
2. Sistem memproses penghapusan data fasilitas umum.	*List* Fasilitas Umum, *Data* Fasilitas Umum.
3. Sistem menampilkan pesan *"Data fasilitas umum telah berhasil dihapus"*.	*List* Fasilitas Umum.
Preconditions **:** *Admin* telah memasuki halaman pengolahan fasilitas umum.	
Postconditions **:** *Admin* berhasil menghapus data fasilitas umum.	
Assumptions **:** Data fasilitas umum yang dihapus merupakan data yang tersimpan di *database*..	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 47 Skenario Use Case Tambah POI.

Use Case Name **:** Tambah *POI*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menambahkan data *POI* baru.	
Triggering Event **:** *Admin* mengisi isian data *POI* dan menekan tombol *submit.*	
Trigger Type **:** *External*	

Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol tambah *POI*	*List POI.*
2. Sistem menampilkan *form* tambah *POI*	*Form* Fasilitas Umum.
3. *Admin* menginputkan isian *form POI* dan menekan tombol *submit.*	*Form POI.*
4. Sistem memproses penambahan data *POI.*	*Form* Fasilitas Umum, *Data* Fasilitas Umum.
5. Sistem menampilkan pesan *"POI baru berhasil ditambahkan"*.	*Form POI.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *POI*.	
Postconditions **:** *Admin* berhasil menambahkan data *POI* baru.	
Assumptions **:** *Admin* menginputkan isian *form POI* dengan data yang *valid.*	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 48 Skenario Use Case Detail POI.

Use Case Name **:** Detail *POI*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menampilkan detail data *POI.*	
Triggering Event **:** *Admin* menekan tombol detail..	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol detail *POI*	*List POI.*
2. Sistem menampilkan informasi detail *POI*.	*Detail POI.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *POI*.	
Postconditions **:** Detail *POI* berhasil ditampilkan.	
Assumptions **:** Detail *POI* yang ditampilkan merupakan data yang sudah tersimpan di	

database.	
Priority	**:** *Medium*.
Risk	**:** *Medium.*

Tabel 3. 49 Skenario Use Case Ubah POI.

Use Case Name	**:** Ubah *POI*
Area	**:** *Backend*
Actor(s)	**:** *Admin*
Description	**:** Fungsional untuk mengubah data *POI* di *database*.
Triggering Event	**:** *Admin* mengubah data *POI* dan menekan tombol *submit.*
Trigger Type	**:** *External*
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol ubah *POI*	*List POI.*
2. Sistem menampilkan *form POI*	*Form POI.*
3. *Admin* mengubah isian *form POI* dan menekan tombol *submit.*	*Form POI.*
4. Sistem memproses perubahan data fasilitas umum.	*Form POI, Data POI.*
5. Sistem menampilkan pesan "*Ubah data POI berhasil"*.	*Form POI.*
Preconditions	**:** *Admin* telah memasuki halaman pengolahan *POI*.
Postconditions	**:** *Admin* berhasil mengubah data *POI*.
Assumptions	**:** *Admin* mengubah isian *form POI* dengan data yang *valid.*
Priority	**:** *Medium*.
Risk	**:** *Medium.*

Tabel 3. 50 Skenario Use Case Hapus POI.

Use Case Name **:** Hapus *POI*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional untuk menghapus data *POI.*	
Triggering Event **:** *Admin* menekan tombol hapus.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol hapus *POI*	*List POI.*
2. Sistem memproses penghapusan data *POI.*	*List POI, Data POI.*
3. Sistem menampilkan pesan *"Data POI telah berhasil dihapus"*.	*List POI.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan *POI*.	
Postconditions **:** *Admin* berhasil menghapus data *POI*.	
Assumptions **:** Data *POI* yang dihapus merupakan data yang tersimpan di *database*..	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 51 Skenario Use Case Identifikasi Koordinat.

Use Case Name **:** Identifikasi Koordinat	
Area **:** *Frontend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional mengidentifikasi koordinat pengguna.	
Triggering Event **:** *Admin* menekan tombol identifikasi koordinat.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***

1. *Admin* menekan tombol identifikasi koordinat.	*Activity BonBinTour*
2. Sistem mengambil titik koordinat lokasi dengan *GPS*..	*Activity Cuurent Location Map V2*
3. Sistem menampilkan informasi koordinat pengguna .	*Activity Cuurent Location Map V2*
Preconditions **:** *Admin* telah memasuki halaman utama *frontend*.	
Postconditions **:** Sistem berhasil menampilkan informasi koordinat pengguna..	
Assumptions **:** *GPS* sudah dinyalakan.	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 52 Skenario Use Case Tambah Gambar.

Use Case Name **:** Tambah Gambar	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional menambah konten gambar..	
Triggering Event **:** *Admin* menginputkan isian form gambar, menekan tombol *submit.*	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol gambar.	*List Hewan/List Tumbuhan/List Fasilitas Umum.*
2. Admin menginputkan isian form tambah gambar dan menekan tombol *submit.*	*Form Gambar.*
3. Sistem meng-*upload* gambar dan menampilkan pesan *"Gambar telah berhasil diupload"* .	*Form Gambar.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan hewan/tumbuhan/fasilitas umum.	
Postconditions **:** Sistem berhasil meng-*upload* gambar ke *server.*	

Assumptions	**:** *Admin* mengisi isisan *form* gambar dengan data yang *valid*.
Priority	**:** *Medium*.
Risk	**:** *Medium.*

Tabel 3. 53 Skenario Use Case Tambah Suara.

Use Case Name	**:** Tambah Suara
Area	**:** *Backend*
Actor(s)	**:** *Admin*
Description	**:** Fungsional menambah konten suara.
Triggering Event	**:** *Admin* menginputkan isian form suara, menekan tombol *submit.*
Trigger Type	**:** *External*
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol suara.	*List Hewan/List Tumbuhan/List Fasilitas Umum.*
2. Admin menginputkan isian form tambah suara dan menekan tombol *submit.*	*Form* Suara.
3. Sistem meng-*upload* suara dan menampilkan pesan *"Suara telah berhasil diupload"* .	*Form* Suara.
Preconditions	**:** *Admin* telah memasuki halaman pengolahan hewan/tumbuhan/fasilitas umum.
Postconditions	**:** Sistem berhasil meng-*upload* suara ke *server.*
Assumptions	**:** *Admin* mengisi isisan *form* suara dengan data yang *valid*.
Priority	**:** *Medium*.
Risk	**:** *Medium.*

Tabel 3. 54 Skenario Use Case Tambah Video.

Use Case Name **:** Tambah Video	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional menambah konten video.	
Triggering Event **:** *Admin* menginputkan isian form video, menekan tombol *submit.*	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol video.	*List Hewan/List Tumbuhan/List Fasilitas Umum.*
2. Admin menginputkan isian form tambah video dan menekan tombol *submit.*	*Form* Video.
3. Sistem meng-*upload* video dan menampilkan pesan *"Video telah berhasil diupload"* .	*Form* Video.
Preconditions **:** *Admin* telah memasuki halaman pengolahan hewan/tumbuhan/fasilitas umum.	
Postconditions **:** Sistem berhasil meng-*upload* video ke *server.*	
Assumptions **:** *Admin* mengisi isisan *form* video dengan data yang *valid*.	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 55 Skenario Use Case Tambah Model.

Use Case Name **:** Tambah *Model*	
Area **:** *Backend*	
Actor(s) **:** *Admin*	
Description **:** Fungsional menambah konten *model.*	
Triggering Event **:** *Admin* menginputkan isian form *model,* menekan tombol *submit.*	

Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. *Admin* menekan tombol *model*.	*List Hewan/List Tumbuhan/List Fasilitas Umum.*
2. Admin menginputkan isian form tambah *model* dan menekan tombol *submit.*	*Form Model.*
3. Sistem meng-*upload model* dan menampilkan pesan *"Model telah berhasil diupload"* .	*Form Model.*
Preconditions **:** *Admin* telah memasuki halaman pengolahan hewan/tumbuhan/fasilitas umum.	
Postconditions **:** Sistem berhasil meng-*upload model* ke *server*.	
Assumptions **:** *Admin* mengisi isisan *form model* dengan data yang *valid*.	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 56 Skenario Use Case Tampil Hewan Sekitar.

Use Case Name **:** Tampil Hewan Sekitar	
Area **:** *Frontend*	
Actor(s) **:** Pengunjung	
Description **:** Fungsional menampilkan *AR POI* keberadaan hewan di sekitar pengunjung.	
Triggering Event **:** Pengunjung menekan tombol "Lihat Hewan".	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. Pengunjung menekan tombol "Lihat Hewan"	*Activity BonBinTour.*
2. Sistem membuka *AR Browser*	*Activity AR Hewan.*
3. Sistem mengambil koordinat *POI* di *database* dan menampilkan *AR* di sekitar	*AREL XML Helper.*

pengunjung.	
Preconditions **:** Pengunjung telah memasuki halaman utama *frontend.*	
Postconditions **:** Sistem berhasil menamppilkan *augmented reality* keberadaan hewan di sekitar pengunjung..	
Assumptions **:** *GPS* dan kamera telah diaktifkan.	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 57 Skenario Use Case Tampil Tumbuhan Sekitar.

Use Case Name **:** Tampil Tumbuhan Sekitar	
Area **:** *Frontend*	
Actor(s) **:** Pengunjung	
Description **:** Fungsional menampilkan *AR POI* keberadaan hewan di sekitar pengunjung.	
Triggering Event **:** Pengunjung menekan tombol ”Lihat Tumbuhan”.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. Pengunjung menekan tombol ”Lihat Tumbuhan”	*Activity BonBinTour.*
2. Sistem membuka *AR Browser*	*Activity AR Tumbuhan.*
3. Sistem mengambil koordinat *POI* di *database* dan menampilkan *AR* di sekitar pengunjung.	*AREL XML Helper.*
Preconditions **:** Pengunjung telah memasuki halaman utama *frontend.*	
Postconditions **:** Sistem berhasil menamppilkan *augmented reality* keberadaan tumbuhan di sekitar pengunjung..	
Assumptions **:** *GPS* dan kamera telah diaktifkan.	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 58 Skenario Use Case Tampil Fasilitas Umum Sekitar.

Use Case Name **:** Tampil Fasilitas Umum Sekitar	
Area **:** *Frontend*	
Actor(s) **:** Pengunjung	
Description **:** Fungsional menampilkan *AR POI* keberadaan fasilitas umum di sekitar pengunjung.	
Triggering Event **:** Pengunjung menekan tombol ”Lihat Fasilitas Umum”.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. Pengunjung menekan tombol ”Lihat Fasilitas Umum”	*Activity BonBinTour.*
2. Sistem membuka *AR Browser*	*Activity AR Fasilitas Umum.*
3. Sistem mengambil koordinat *POI* di *database* dan menampilkan *AR* di sekitar pengunjung.	*AREL XML Helper.*
Preconditions **:** Pengunjung telah memasuki halaman utama *frontend.*	
Postconditions **:** Sistem berhasil menamppilkan *augmented reality* keberadaan fasilitas umum di sekitar pengunjung..	
Assumptions **:** *GPS* dan kamera telah diaktifkan.	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 59 Skenario Use Case Lihat Detail.

Use Case Name **:** Lihat Detail
Area **:** *Frontend*
Actor(s) **:** Pengunjung
Description **:** Fungsional menampilkan informasi detail *POI.*
Triggering Event **:** Pengunjung menekan tombol lihat detail.

Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. Pengunjung menekan tombol lihat detail.	*Activity Scan Marker.*
2. Sistem mengambil data detail *POI* dari *database.*	*AREL XML Helper.*
3. Sistem menampilkan data detail *POI.*	*AREL XML Helper.*
Preconditions **:** Pengunjung telah memasuki halaman *AR Scan Marker.*	
Postconditions **:** Sistem berhasil menampilkan konten detail *POI.*	
Assumptions **:** *GPS* dan kamera telah diaktifkan.	
Priority **:** *Medium*.	
Risk **:** *Medium.*	

Tabel 3. 60 Skenario Use Case Putar Video.

Use Case Name **:** Putar Video	
Area **:** *Frontend*	
Actor(s) **:** Pengunjung	
Description **:** Fungsional memuatar/mendownload video *POI.*	
Triggering Event **:** Pengunjung menekan tombol putar video.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. Pengunjung menekan tombol putar video.	*Activity Scan Marker.*
2. Pengunjung mengkonfirmasi akses *download/streaming.*	*AREL XML Helper.*
3. Sistem melakukan *download/streaming* video.	*AREL XML Helper.*
Preconditions **:** Pengunjung telah memasuki halaman *AR Scan Marker.*	
Postconditions **:** Sistem berhasil menampilkan konten video *POI.*	

Assumptions	**:** *GPS* dan kamera telah diaktifkan.
Priority	**:** *Medium*.
Risk	**:** *Medium.*

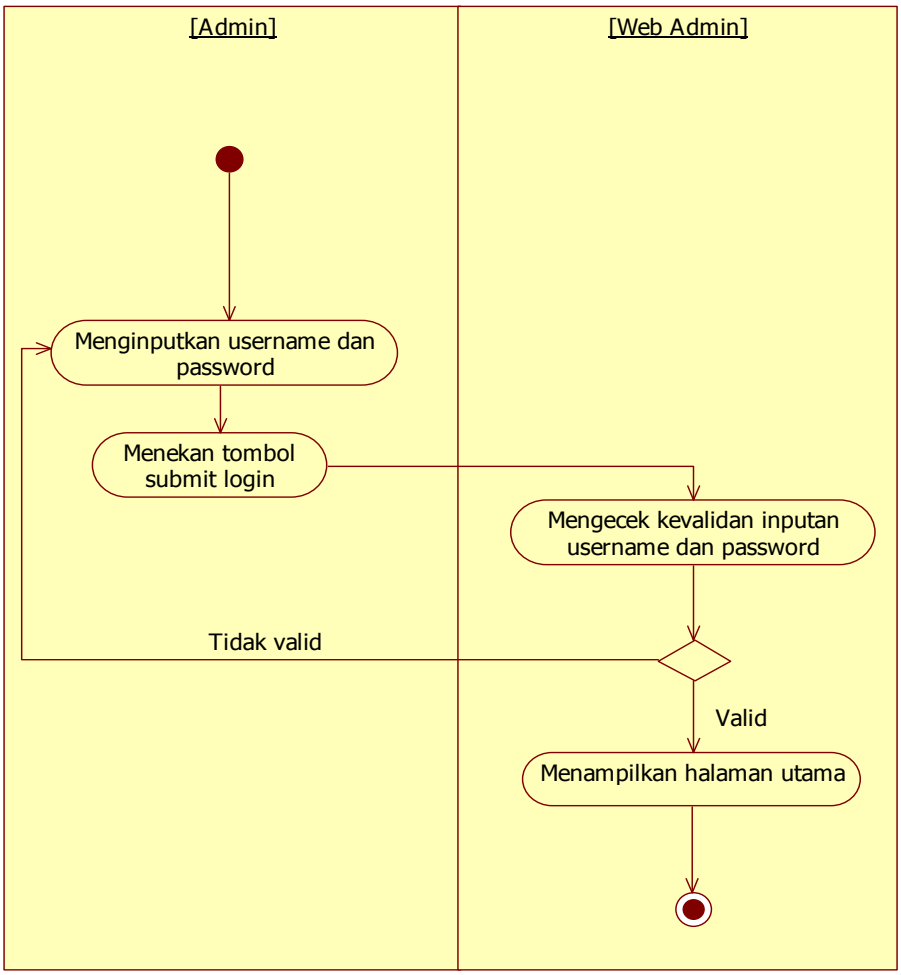
Tabel 3. 61 Skenario Use Case Putar Suara.

Use Case Name	**:** Putar Suara
Area	**:** *Frontend*
Actor(s)	**:** Pengunjung
Description	**:** Fungsional memuatar/mendownload suara *POI.*
Triggering Event	**:** Pengunjung menekan tombol putar suara.
Trigger Type	**:** *External*
Steps Performed (Main Path)	***Information for Steps***
1. Pengunjung menekan tombol putar suara.	*Activity Scan Marker.*
2. Pengunjung mengkonfirmasi akses *download/streaming*.	*AREL XML Helper.*
3. Sistem melakukan *download/streaming* suara.	*AREL XML Helper.*
Preconditions	**:** Pengunjung telah memasuki halaman *AR Scan Marker.*
Postconditions	**:** Sistem berhasil menampilkan konten suara *POI.*
Assumptions	**:** *GPS* dan kamera telah diaktifkan.
Priority	**:** *Medium*.
Risk	**:** *Medium.*

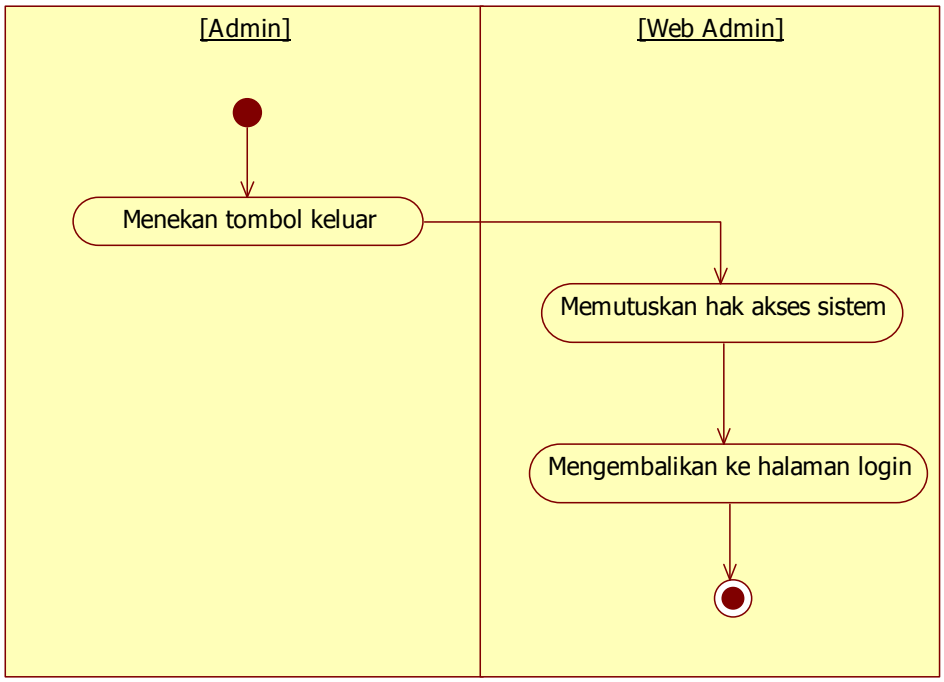
Tabel 3. 62 Skenario Use Case Tampil Model.

Use Case Name **:** Tampil Model	
Area **:** *Frontend*	
Actor(s) **:** Pengunjung	
Description **:** Fungsional memuatar/mendownload suara *POI.*	
Triggering Event **:** Pengunjung menekan tombol putar suara.	
Trigger Type **:** *External*	
Steps Performed (Main Path)	***Information for Steps***
1. Pengunjung menekan tombol tampil model.	*Activity Scan Marker.*
2. Sistem mengidentifikasi lokasi *GPS* pengunjung.	*AREL XML Helper.*
3. Sistem me-*load* data koordinat *POI* dan menampilkan model 3D pada koordinat *POI.*	*AREL XML Helper.*
Preconditions **:** Pengunjung telah memasuki halaman *AR Scan Marker.*	
Postconditions **:** Sistem berhasil menampilkan konten model 3D *POI.*	
Assumptions **:** *GPS* dan kamera telah diaktifkan.	
Priority **:** *Medium.*	
Risk **:** *Medium.*	

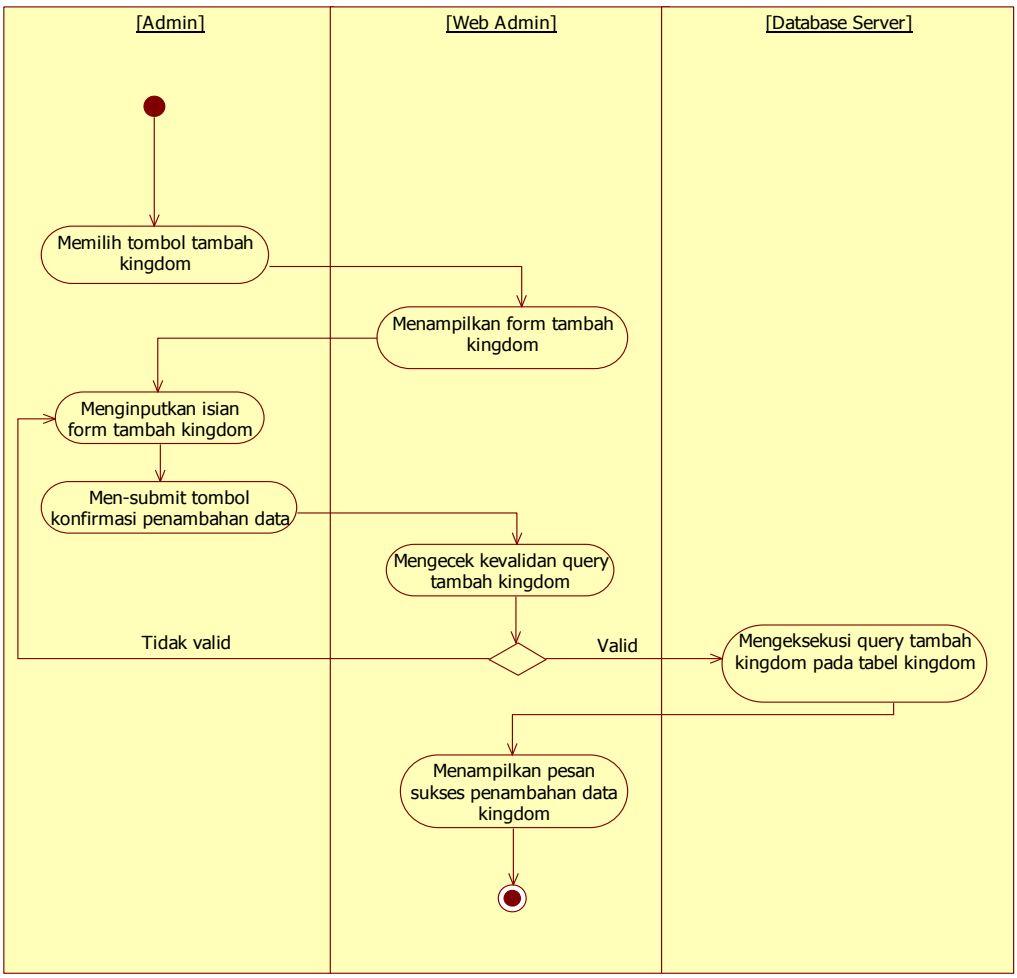
B. Activity Diagram



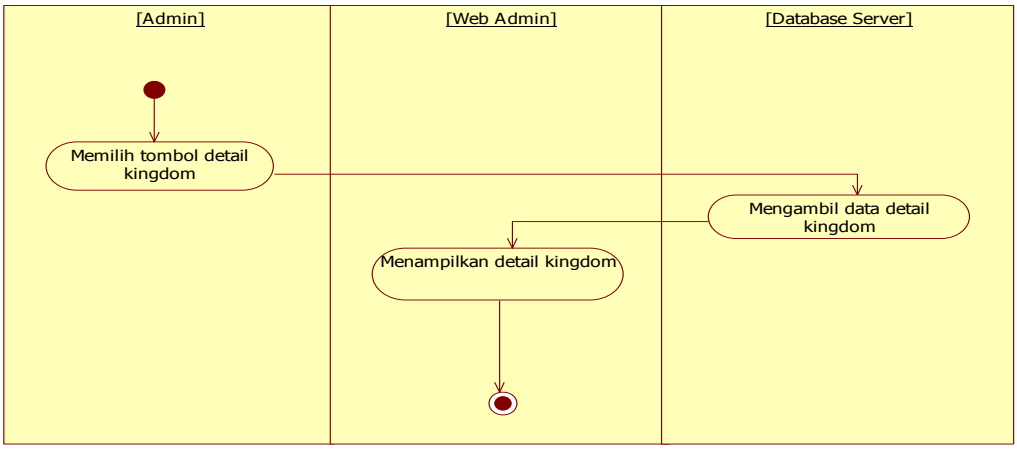
Gambar 3. 7 Activity Diagram Login.



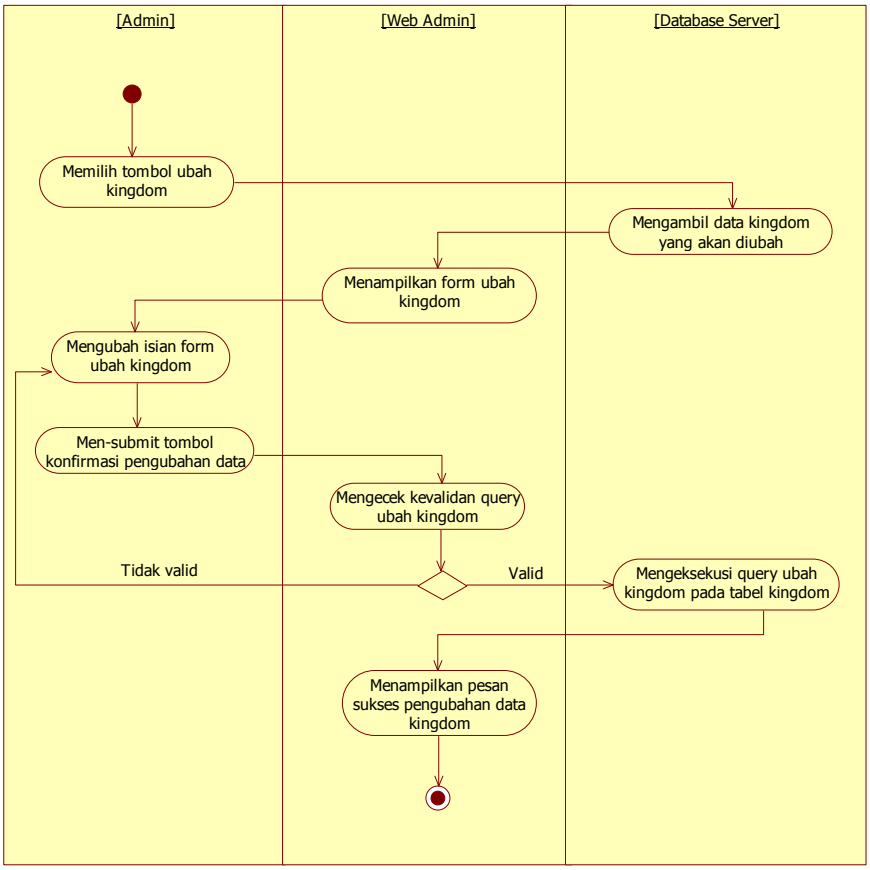
Gambar 3. 8 Activity Diagram Logout.



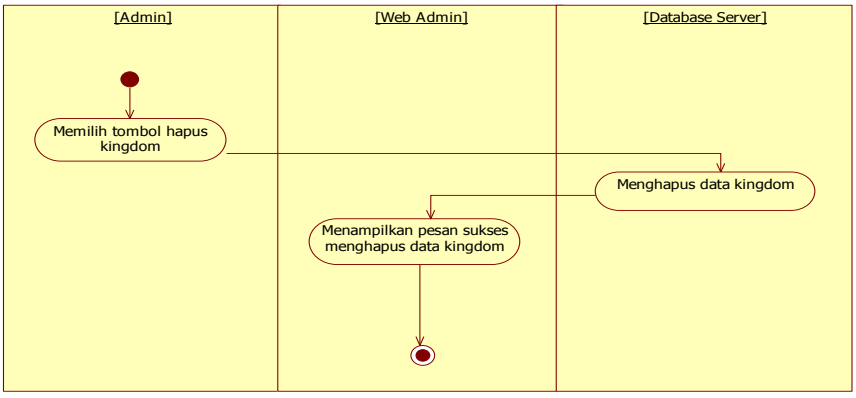
Gambar 3. 9 Activity Diagram Tambah Kingdom.



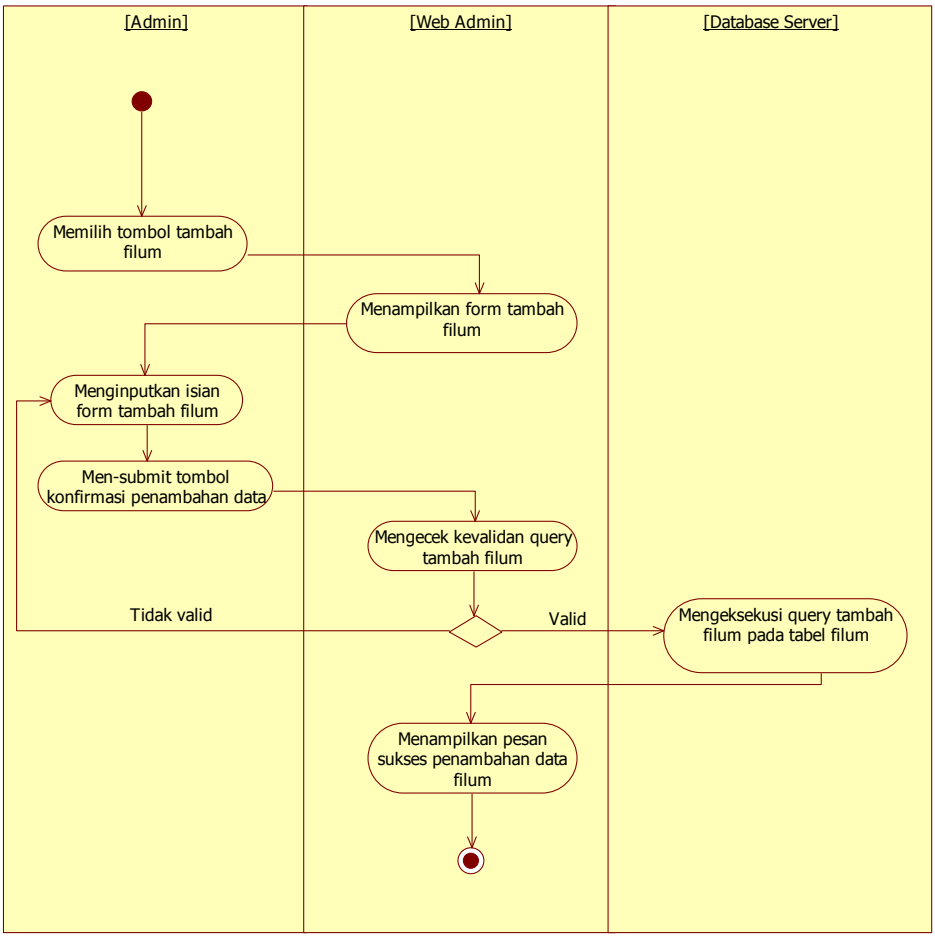
Gambar 3. 10 Activity Diagram Detail Kingdom



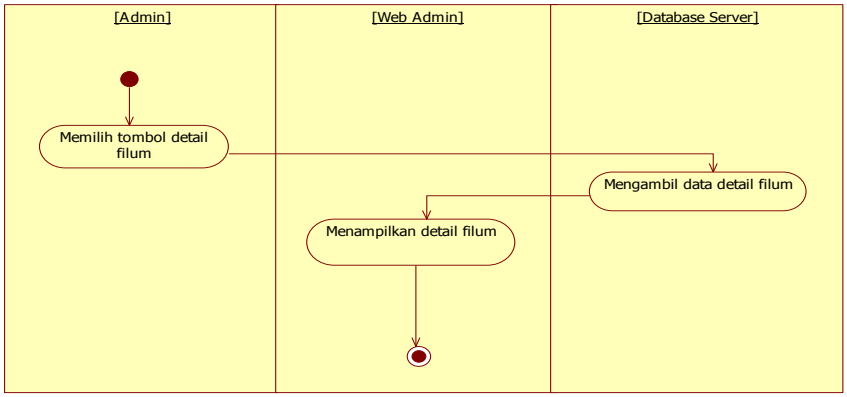
Gambar 3. 11 Activity Diagram Ubah Kingdom.



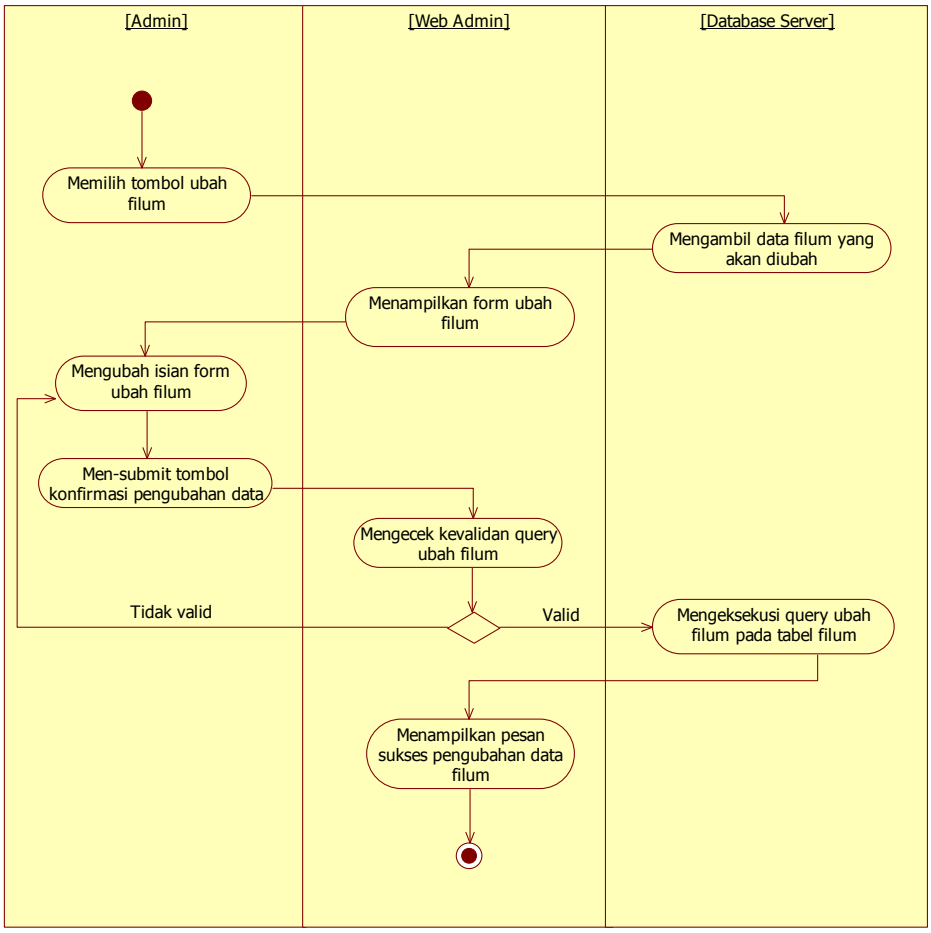
Gambar 3. 12 Activity Diagram Hapus Kingdom



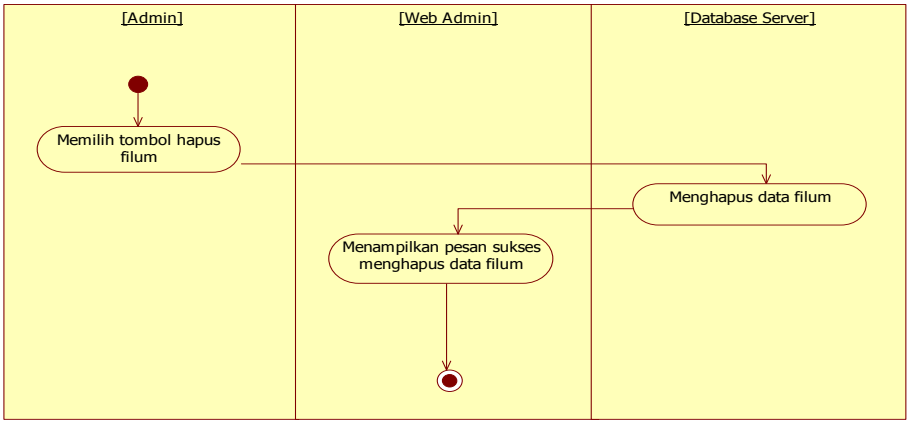
Gambar 3. 13 Activity Diagram Tambah Filum.



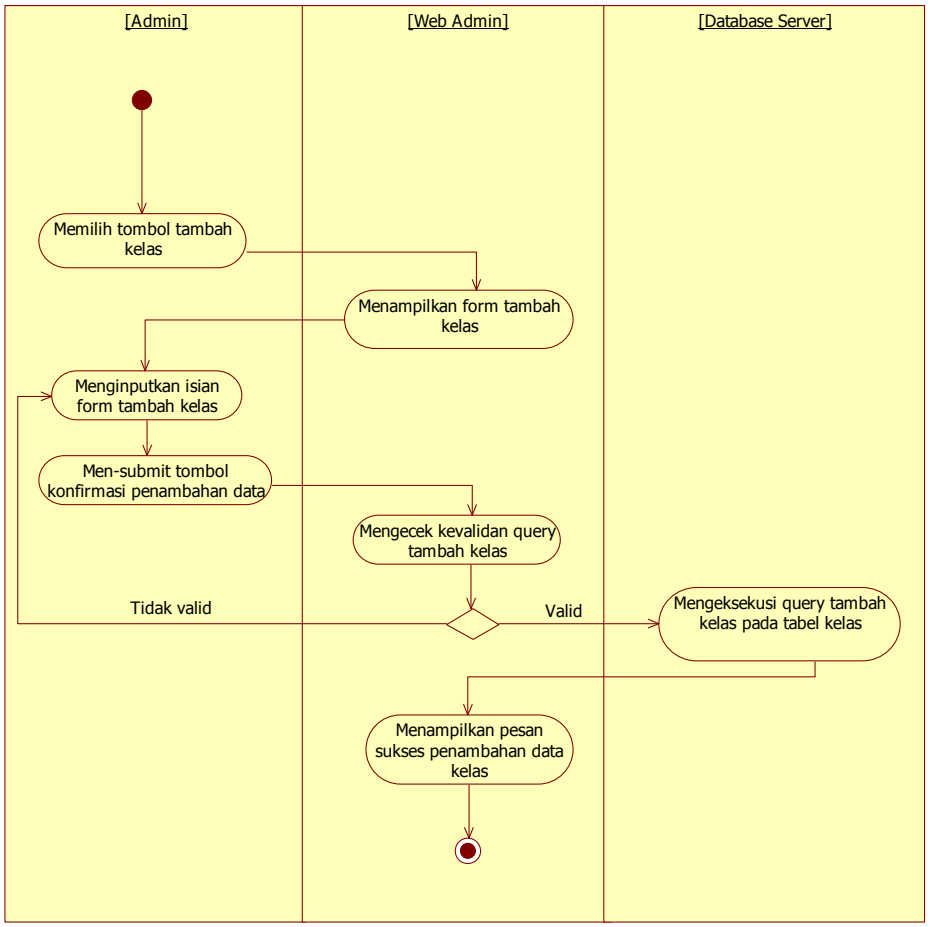
Gambar 3. 14 Activity Diagram Detail Filum



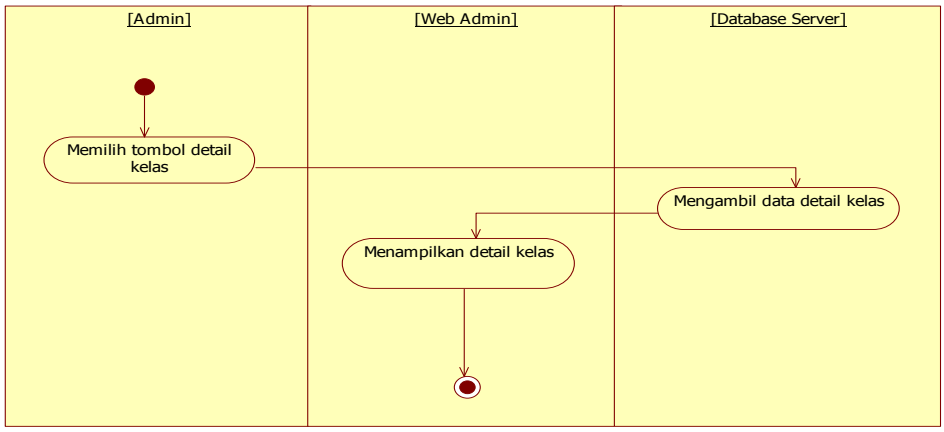
Gambar 3. 15 Activity Diagram Ubah Filum.



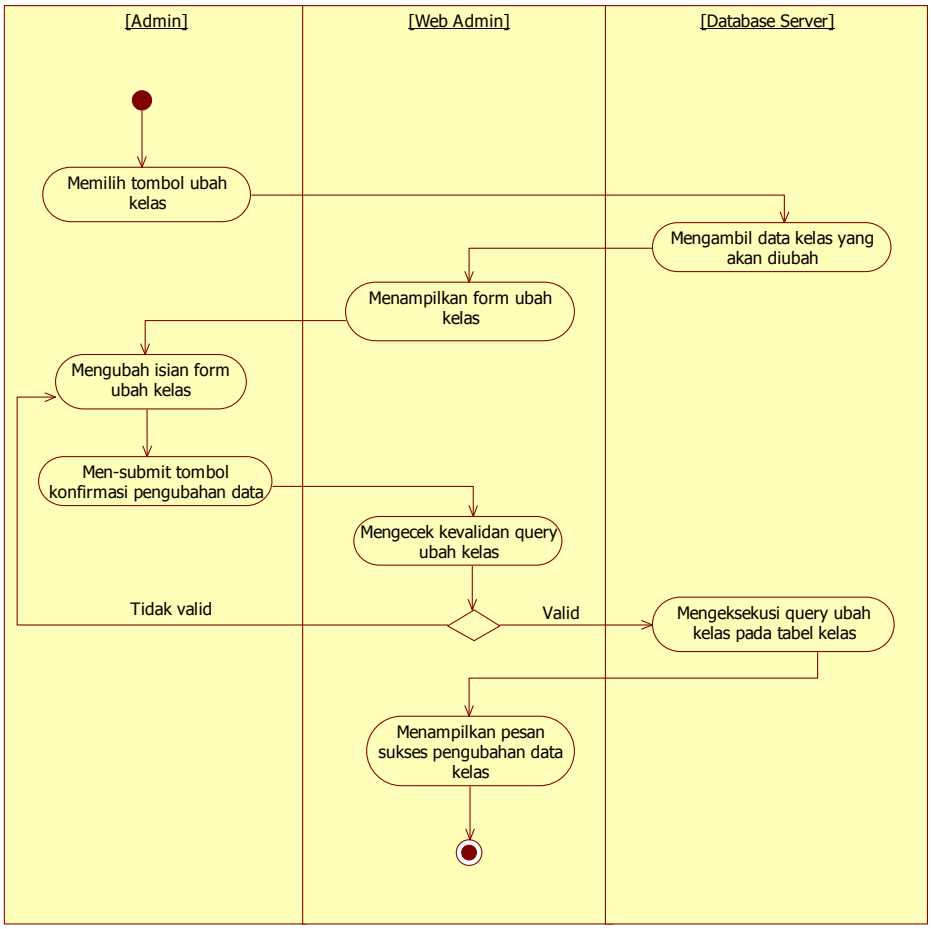
Gambar 3. 16 Activity Diagram Hapus Filum



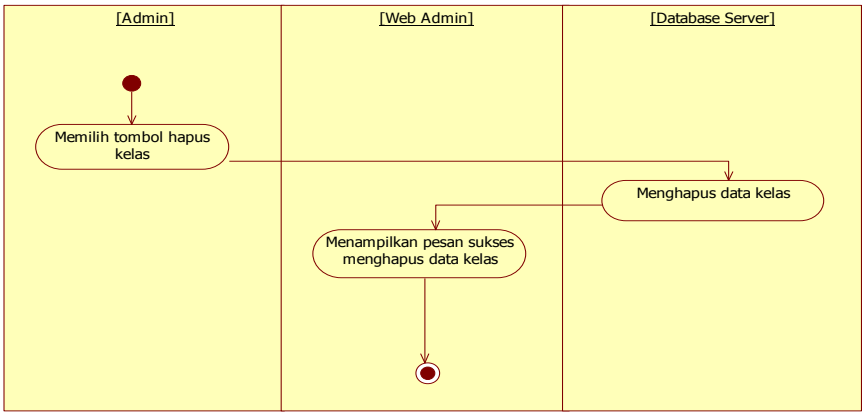
Gambar 3. 17 Activity Diagram Tambah Kelas.



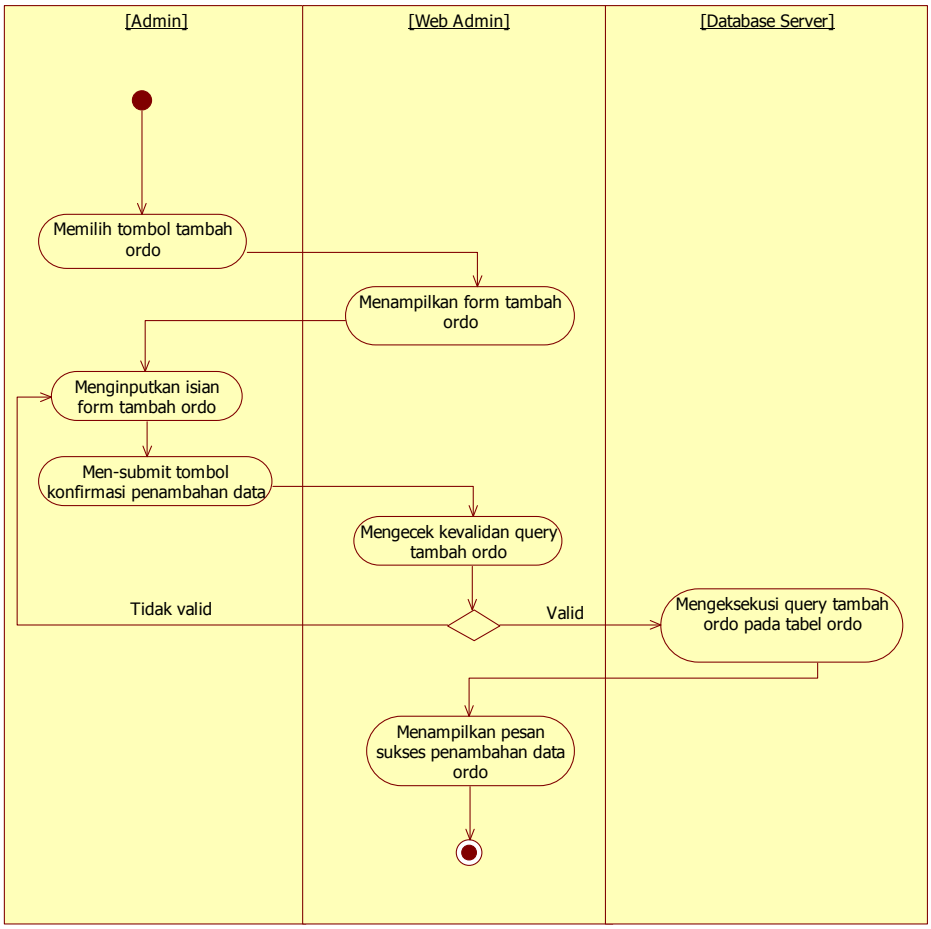
Gambar 3. 18 Activity Diagram Detail Kelas



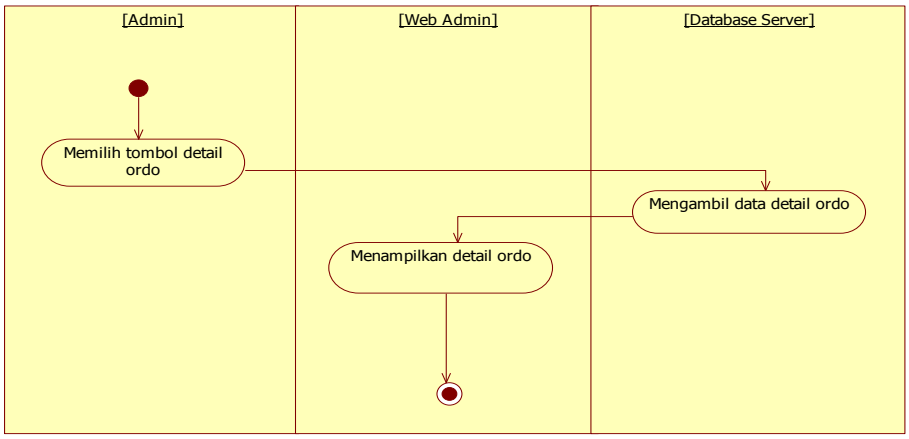
Gambar 3. 19 Activity Diagram Ubah Kelas.



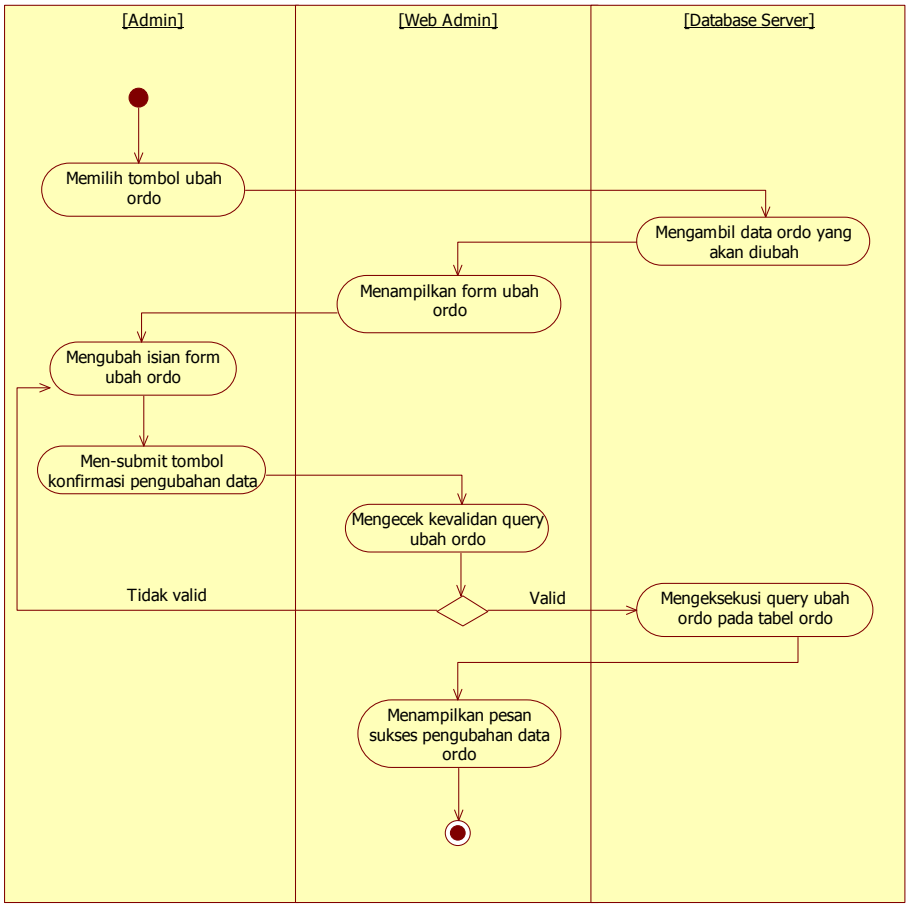
Gambar 3. 20 Activity Diagram Hapus Kelas



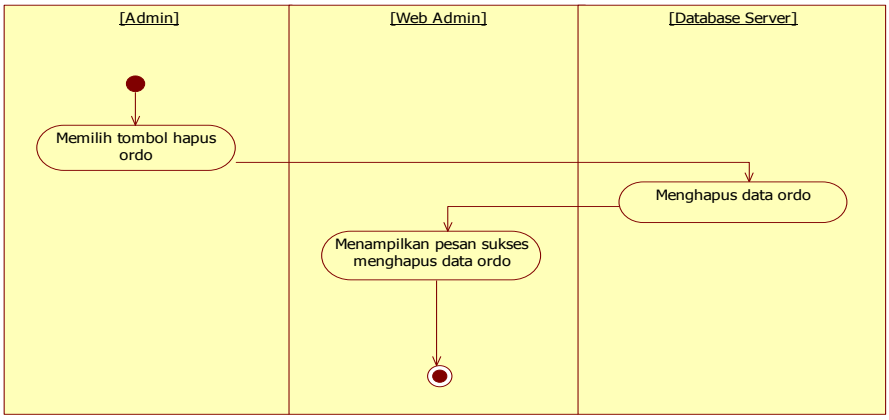
Gambar 3. 21 Activity Diagram Tambah Ordo.



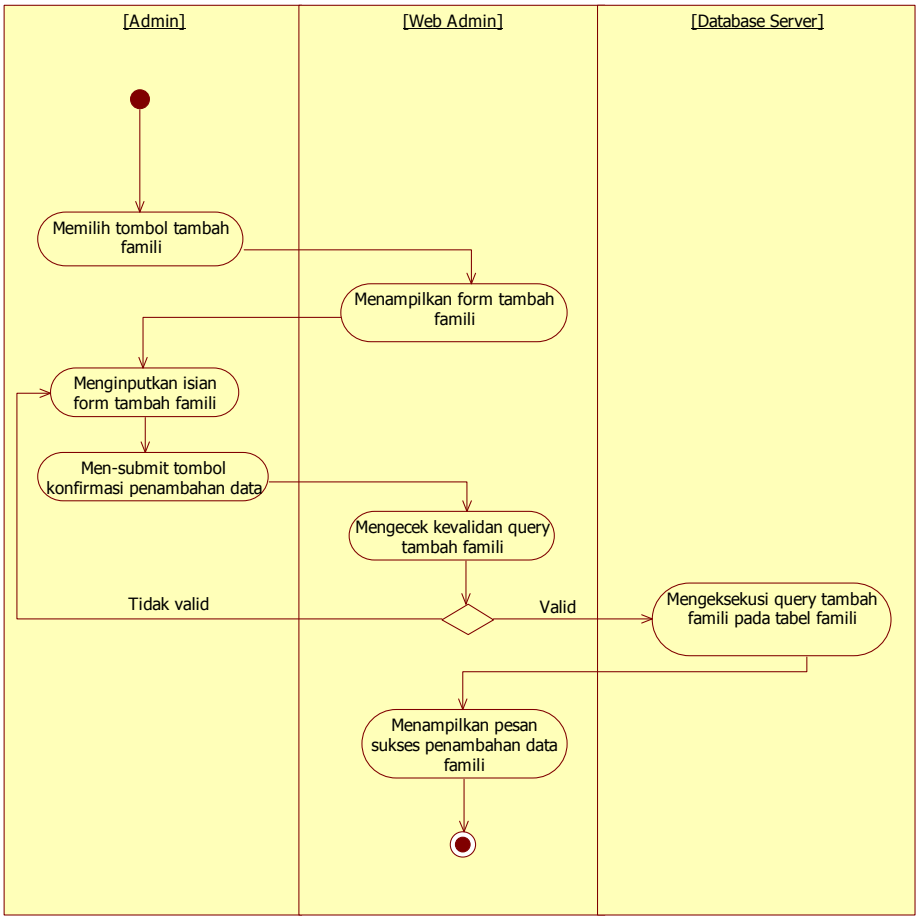
Gambar 3. 22 Activity Diagram Detail Ordo



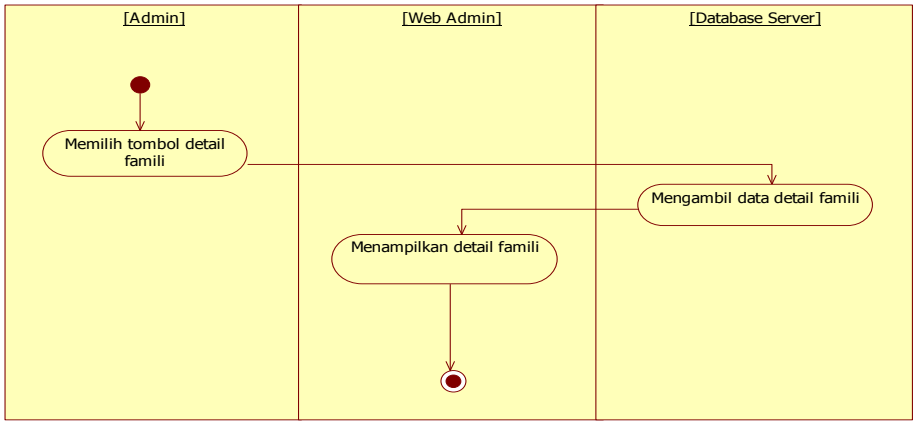
Gambar 3. 23 Activity Diagram Ubah Ordo.



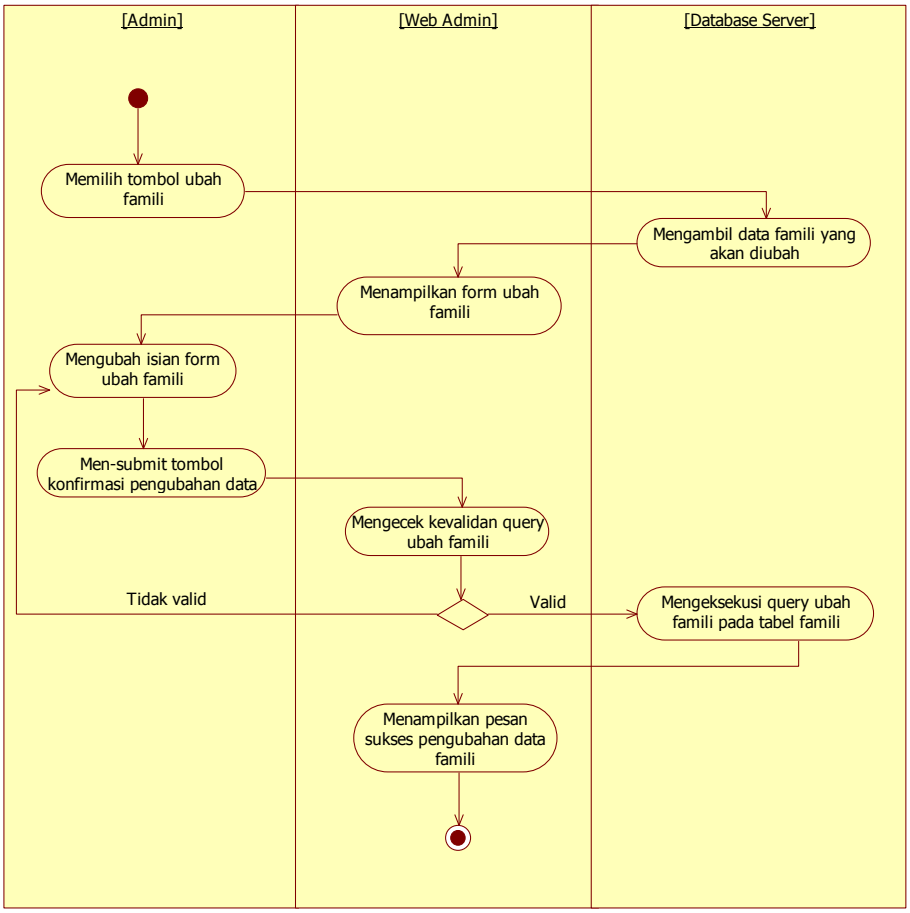
Gambar 3. 24 Activity Diagram Hapus Ordo



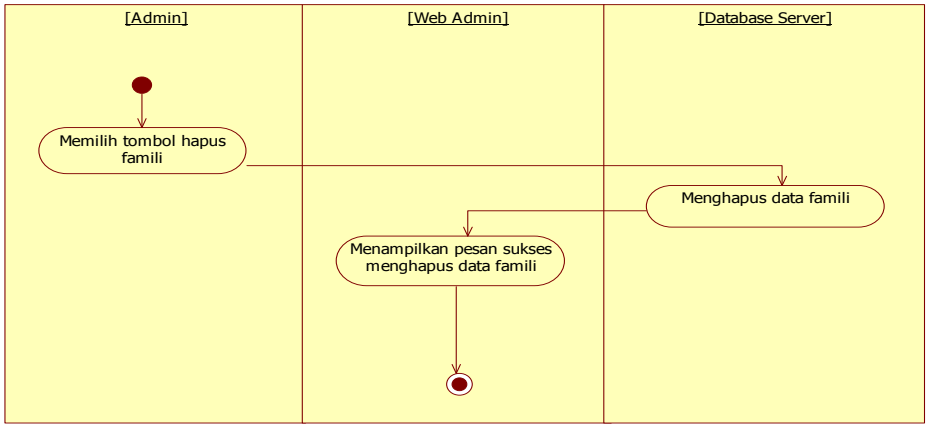
Gambar 3. 25 Activity Diagram Tambah Famili.



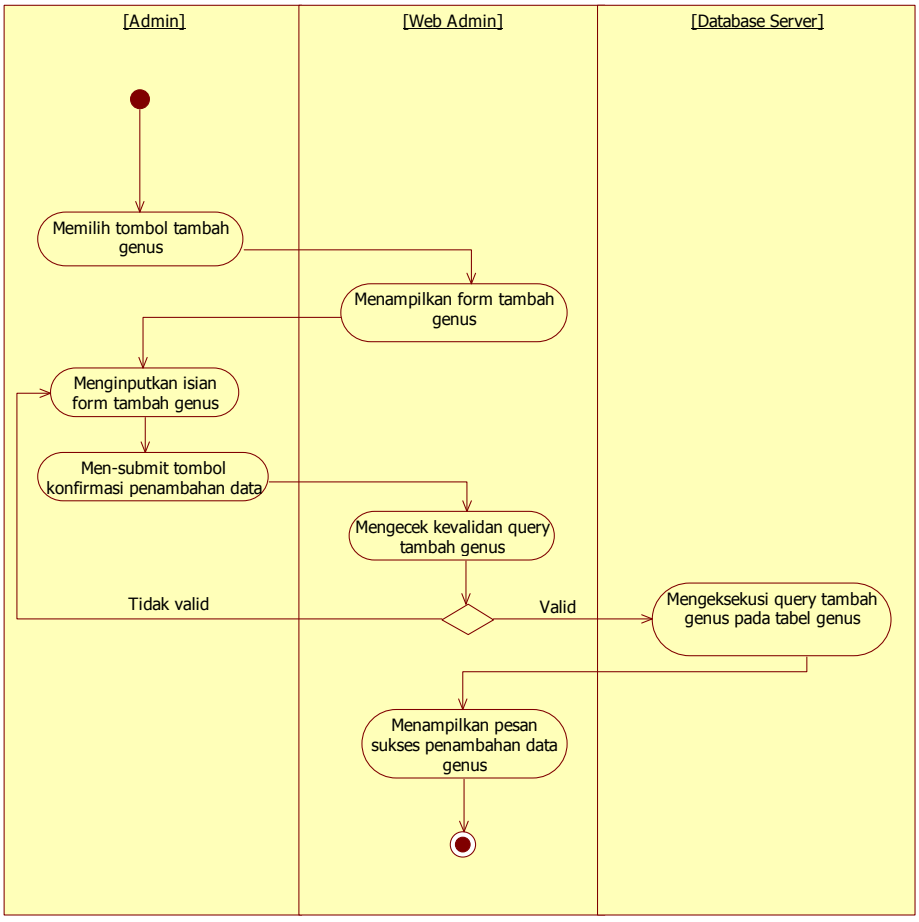
Gambar 3. 26 Activity Diagram Detail Famili



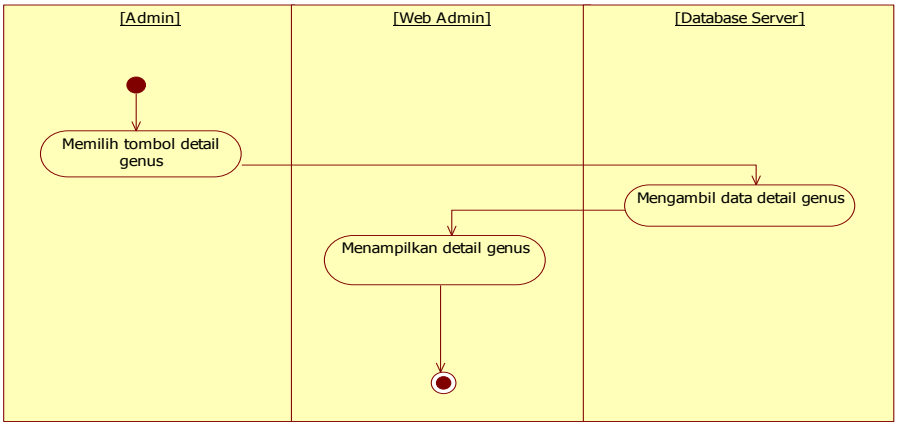
Gambar 3. 27 Activity Diagram Ubah Famili.



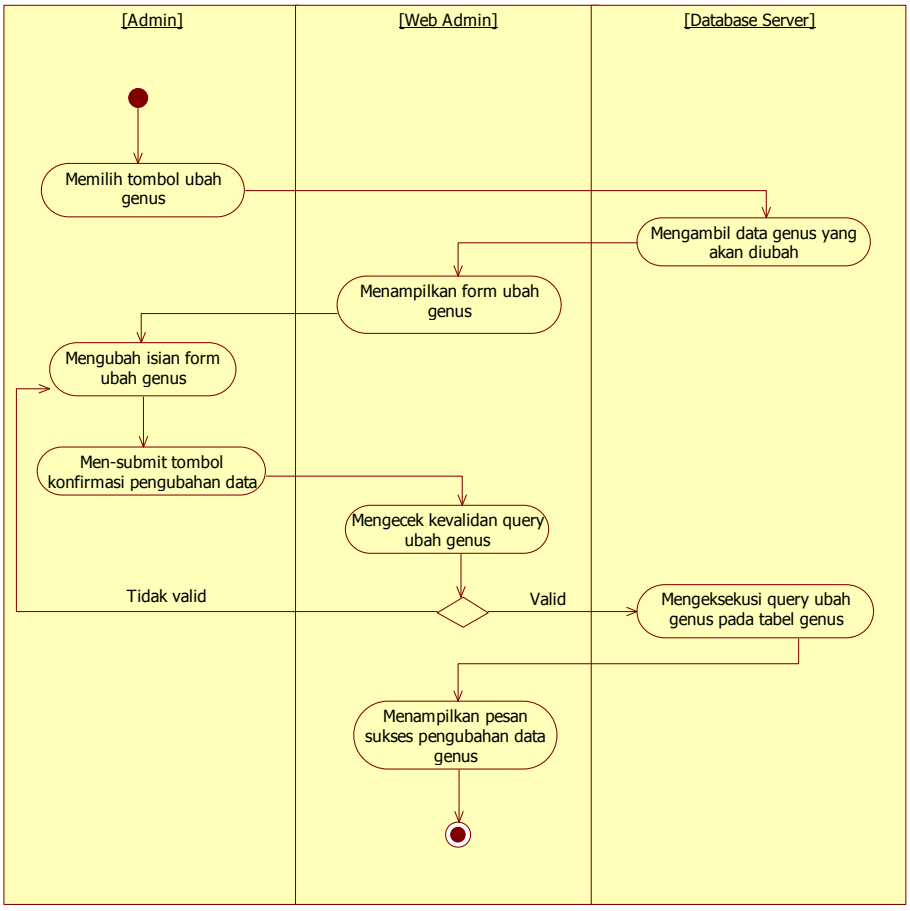
Gambar 3. 28 Activity Diagram Hapus Famili



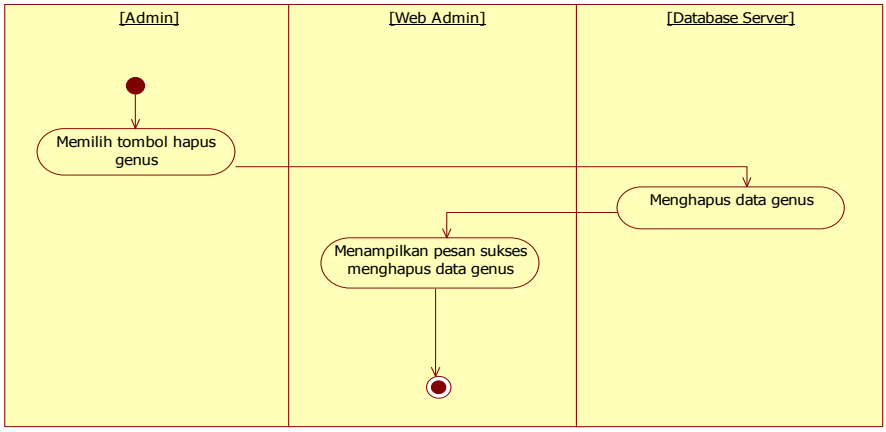
Gambar 3. 29 Activity Diagram Tambah Genus.



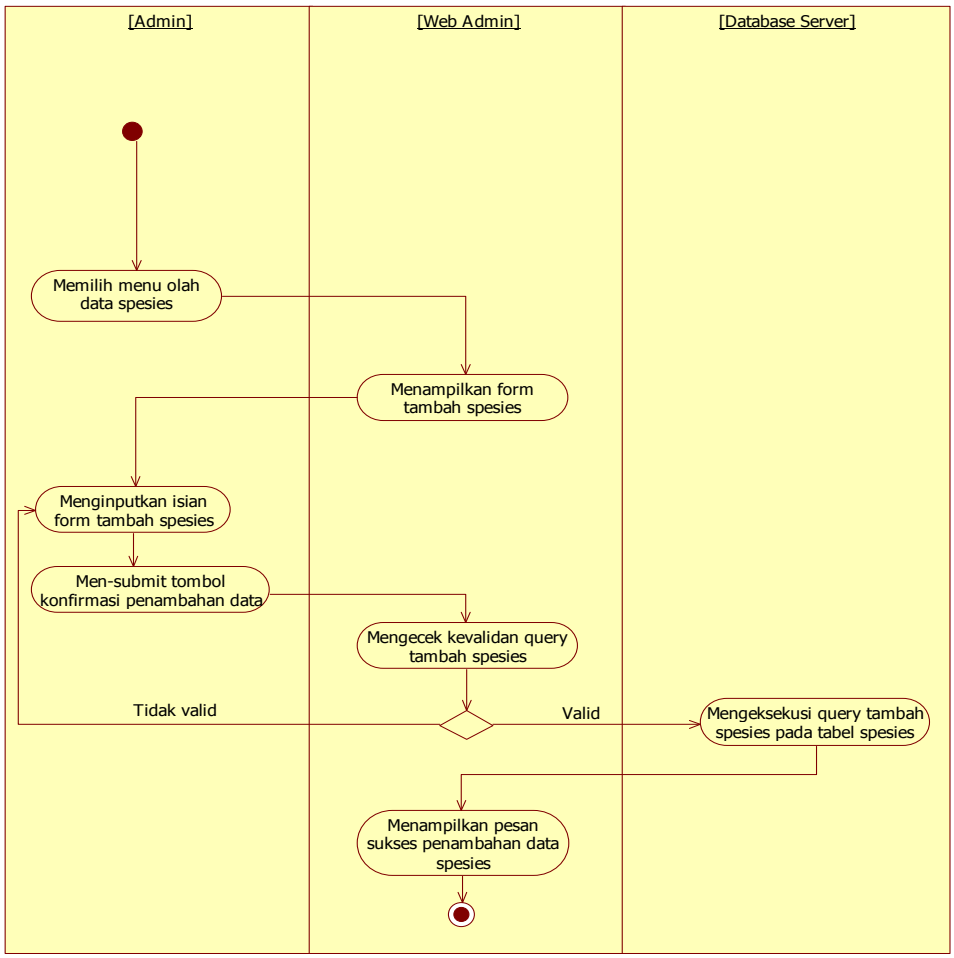
Gambar 3. 30 Activity Diagram Detail Genus



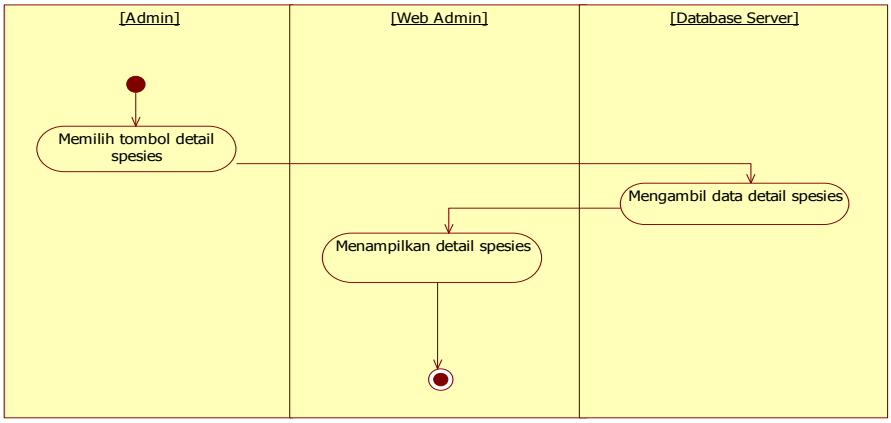
Gambar 3. 31 Activity Diagram Ubah Genus.



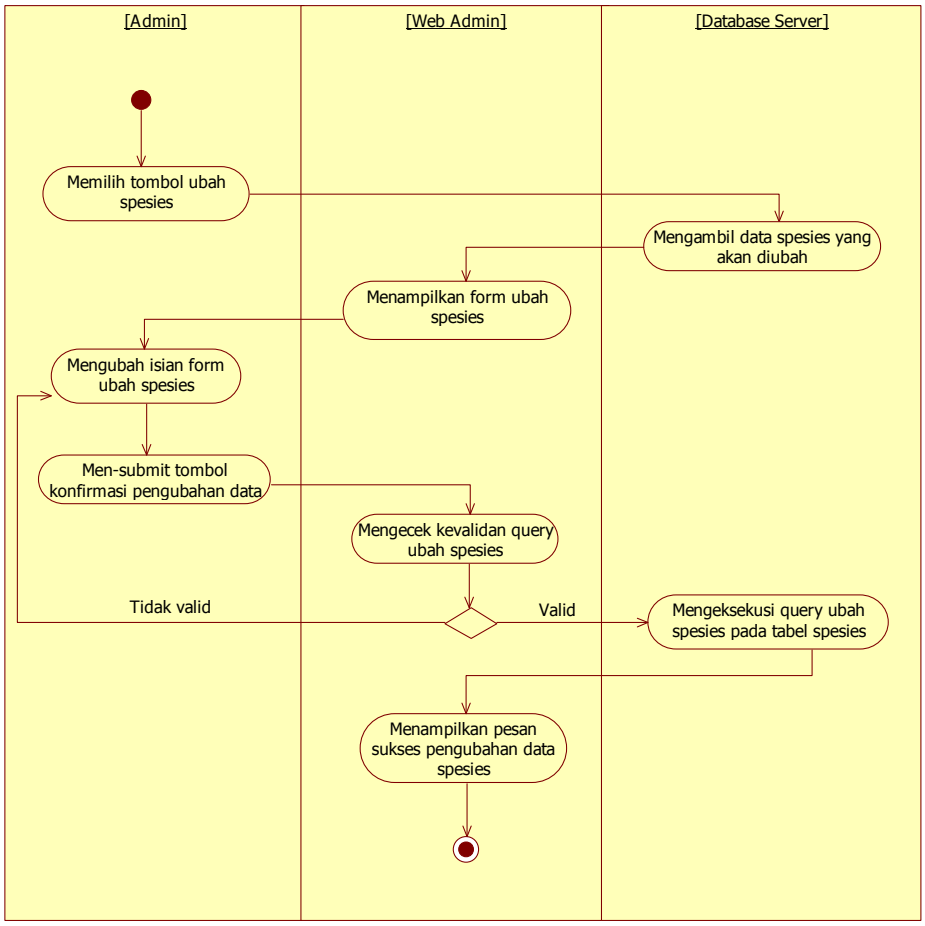
Gambar 3. 32 Activity Diagram Hapus Genus



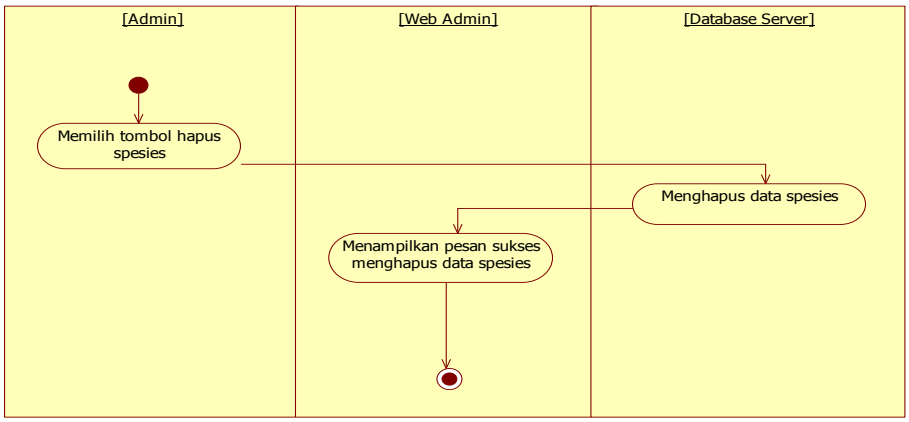
Gambar 3. 33 Activity Diagram Tambah Spesies.



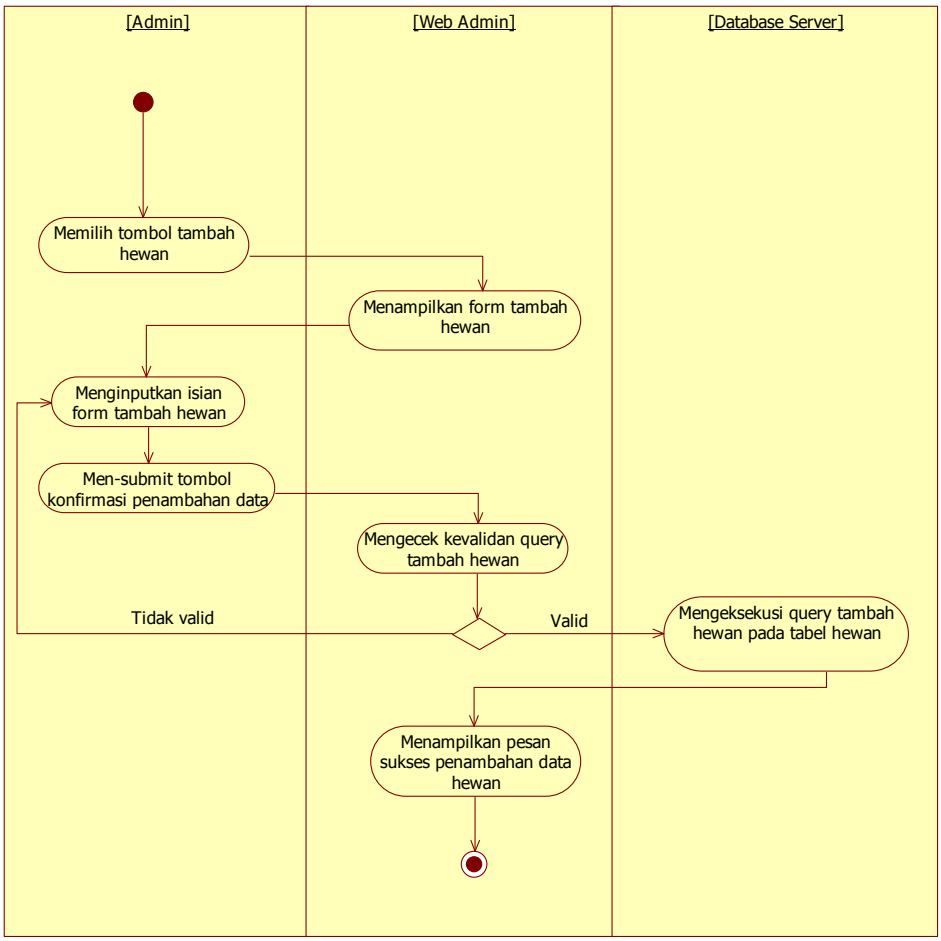
Gambar 3. 34 Activity Diagram Detail Spesies



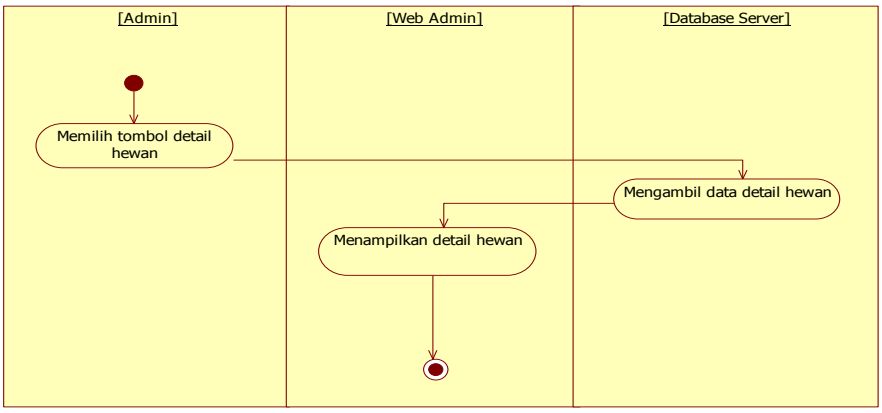
Gambar 3. 35 Activity Diagram Ubah Spesies.



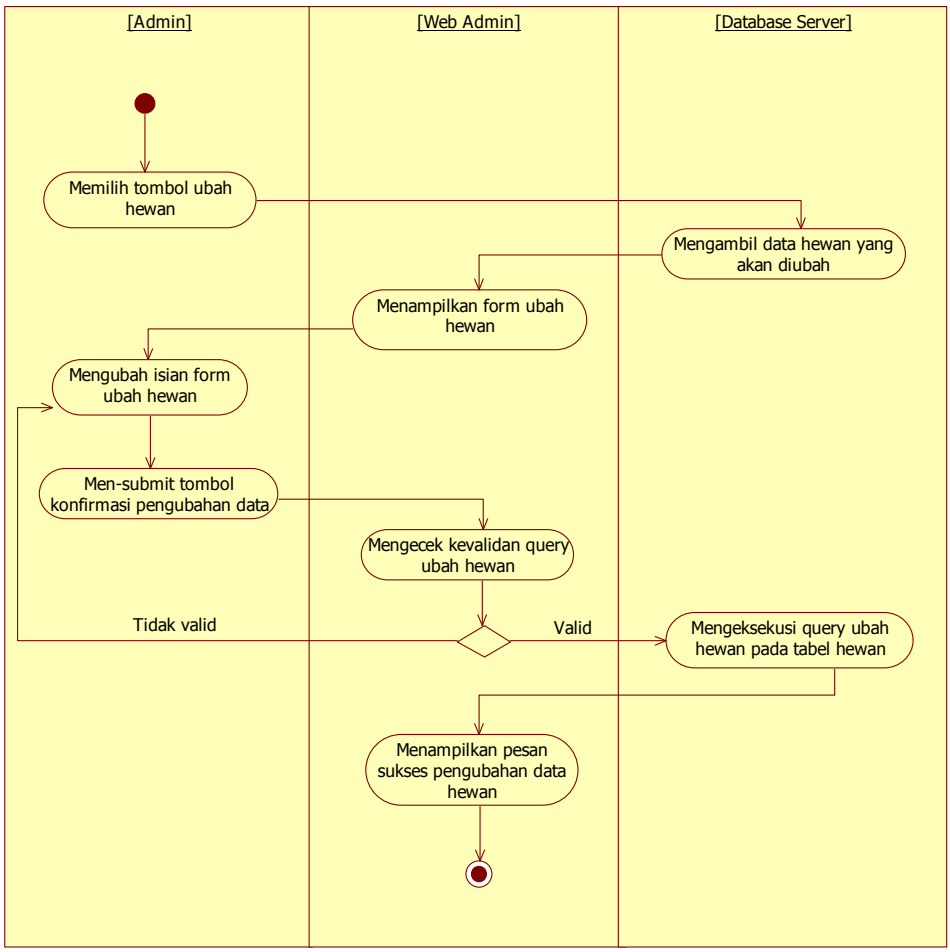
Gambar 3. 36 Activity Diagram Hapus Spesies



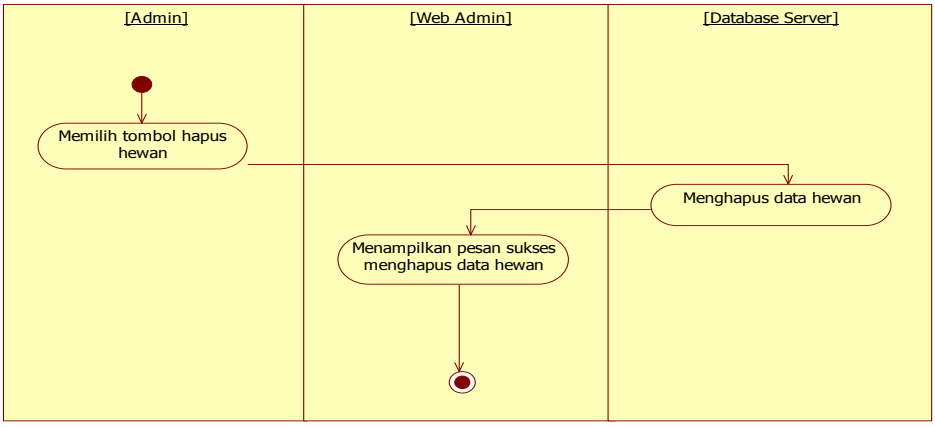
Gambar 3. 37 Activity Diagram Tambah Hewan.



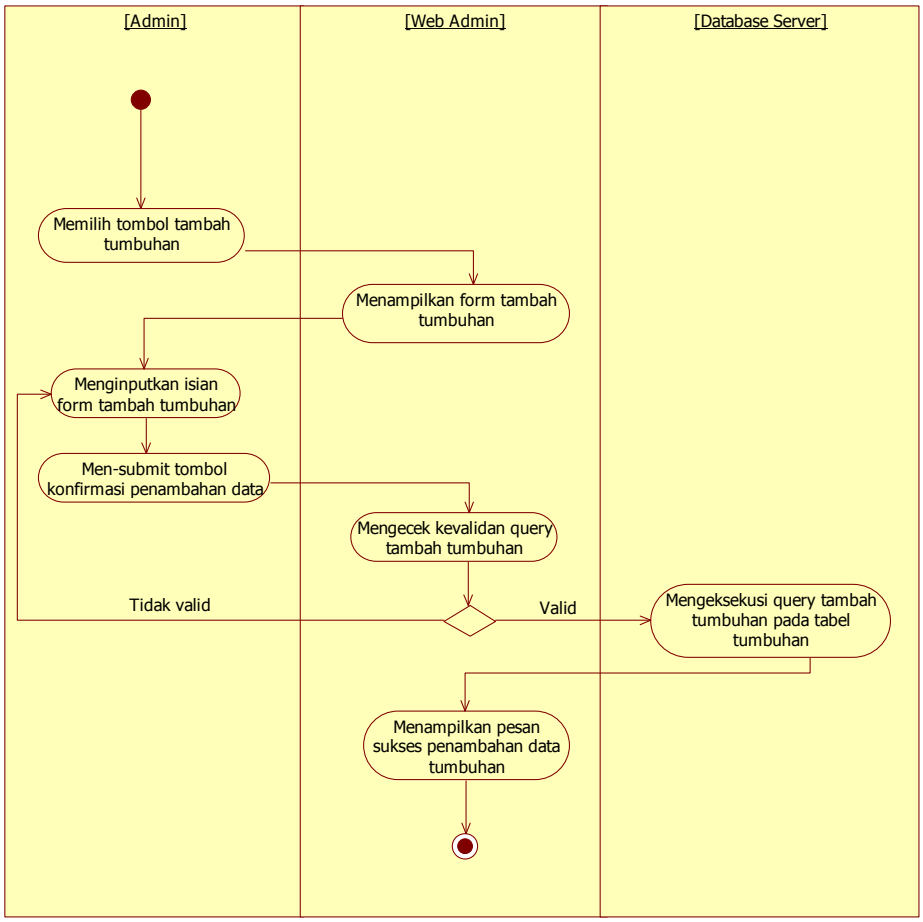
Gambar 3. 38 Activity Diagram Detail Hewan



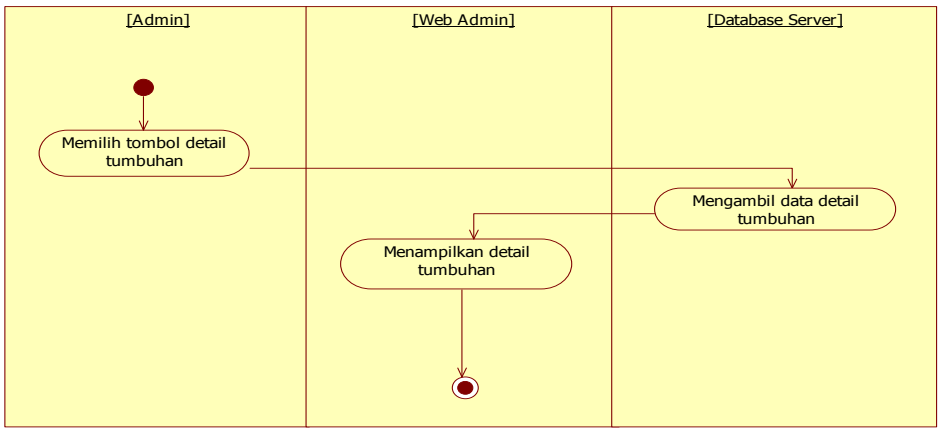
Gambar 3. 39 Activity Diagram Ubah Hewan.



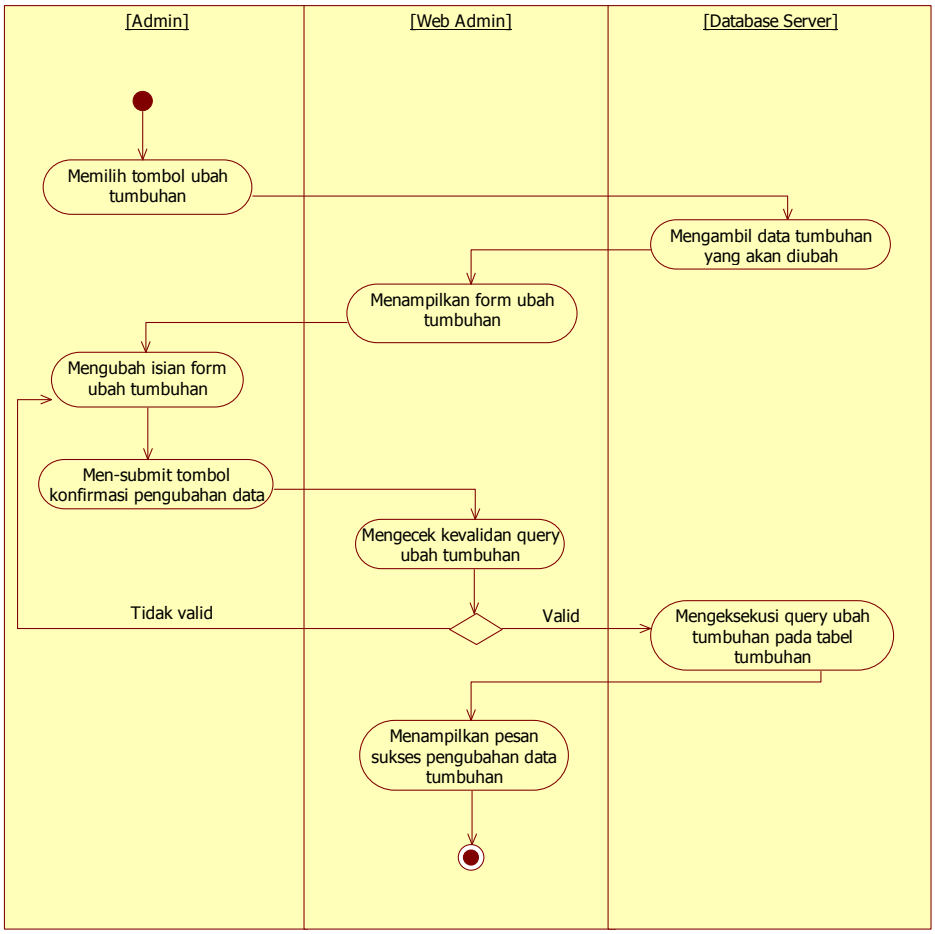
Gambar 3. 40 Activity Diagram Hapus Hewan



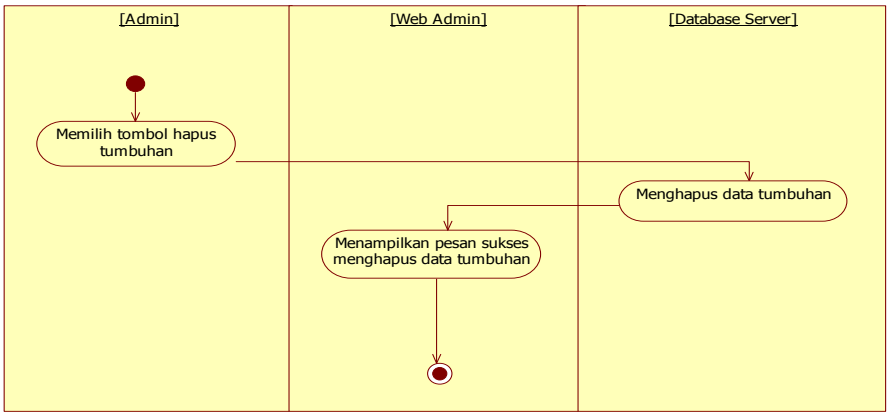
Gambar 3. 41 Activity Diagram Tambah Tumbuhan.



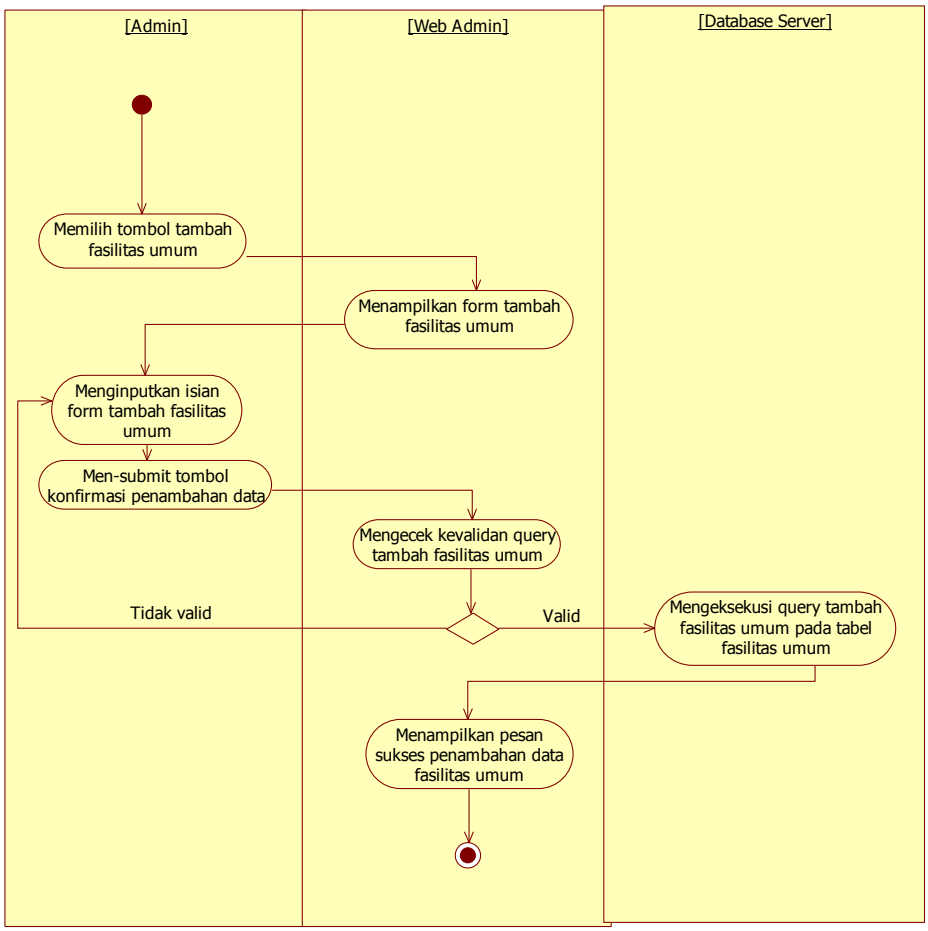
Gambar 3. 42 Activity Diagram Detail Tumbuhan



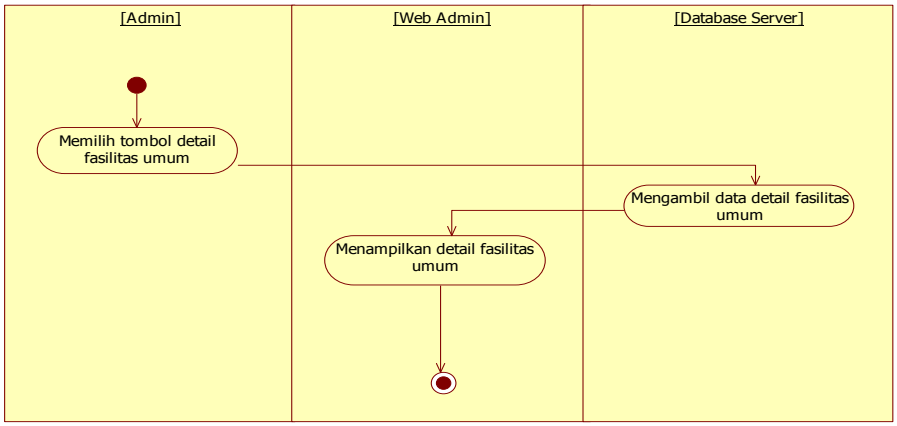
Gambar 3. 43 Activity Diagram Ubah Tumbuhan.



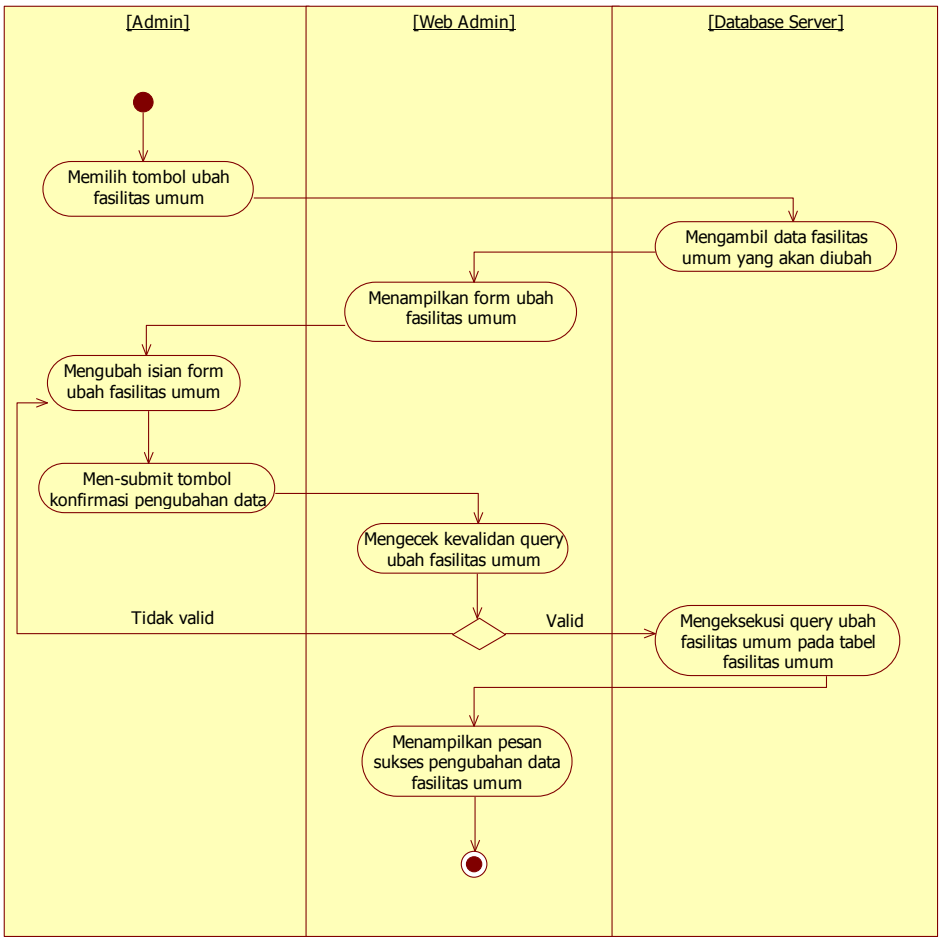
Gambar 3. 44 Activity Diagram Hapus Tumbuhan



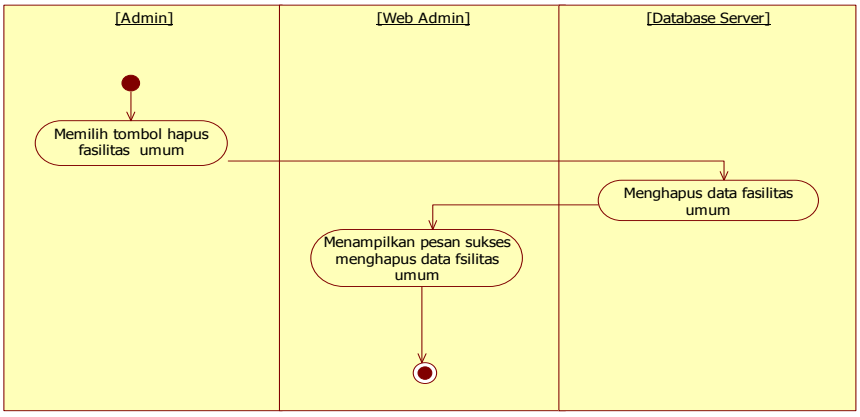
Gambar 3. 45 Activity Diagram Tambah Fasilitas Umum.



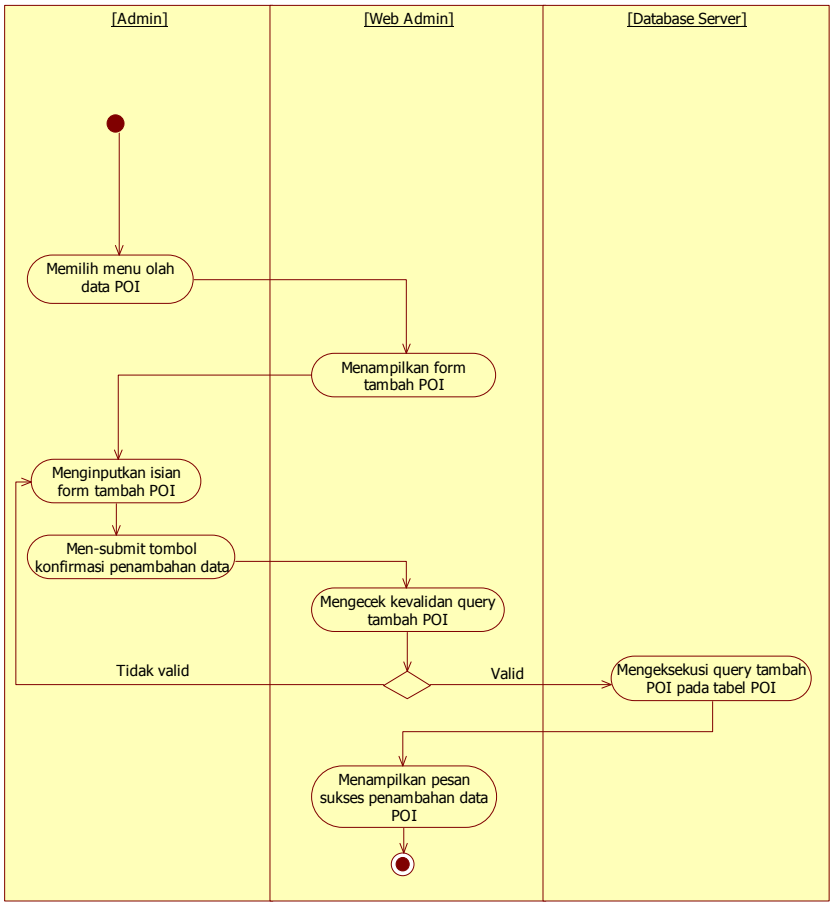
Gambar 3. 46 Activity Diagram Detail Fasilitas Umum



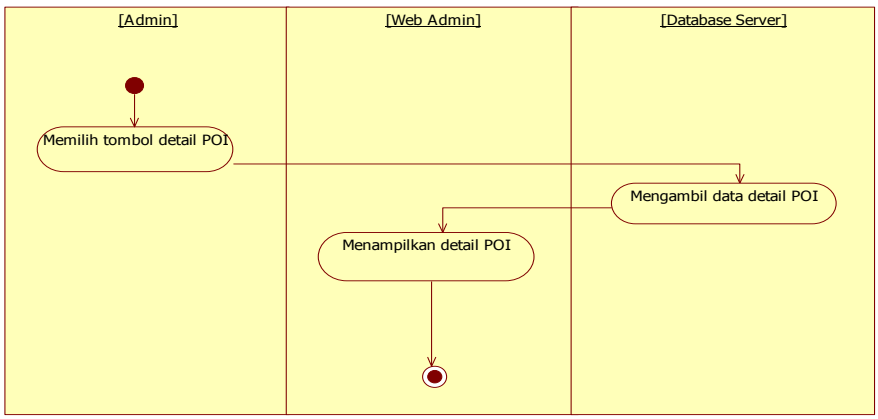
Gambar 3. 47 Activity Diagram Ubah Fasilitas Umum.



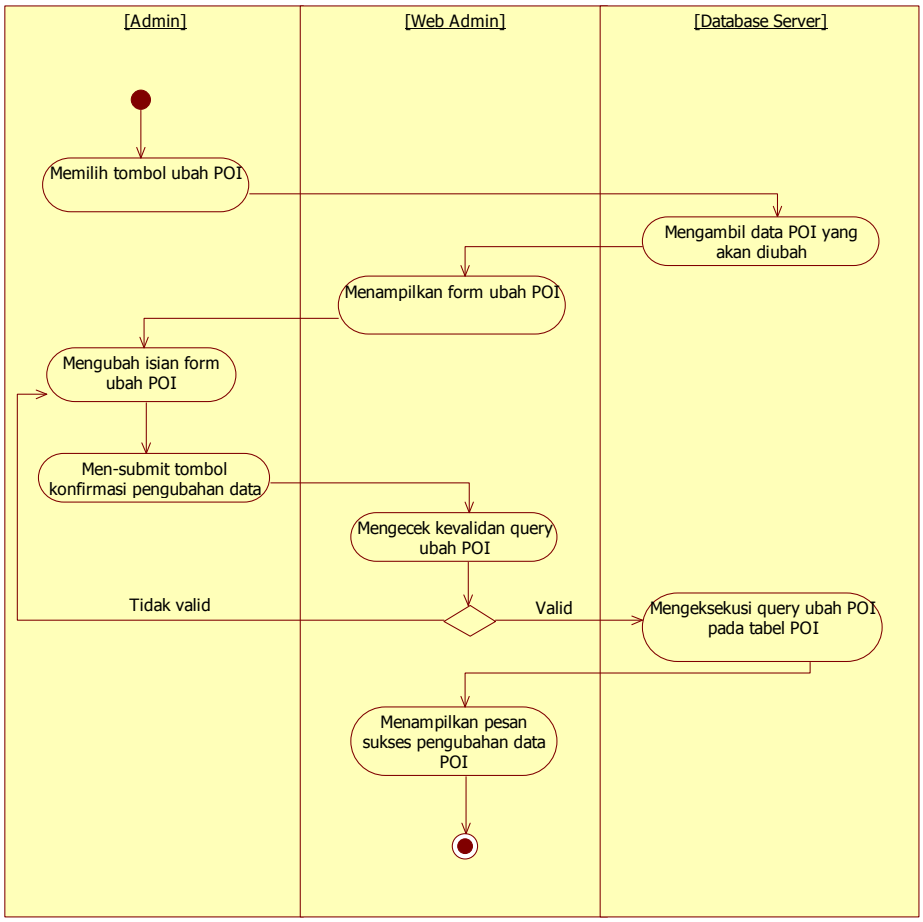
Gambar 3. 48 Activity Diagram Hapus Fasilitas Umum.



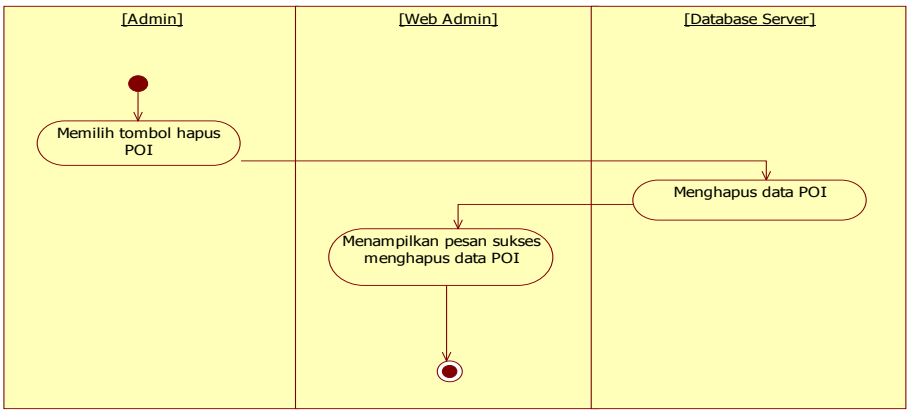
Gambar 3. 49 Activity Diagram Tambah POI.



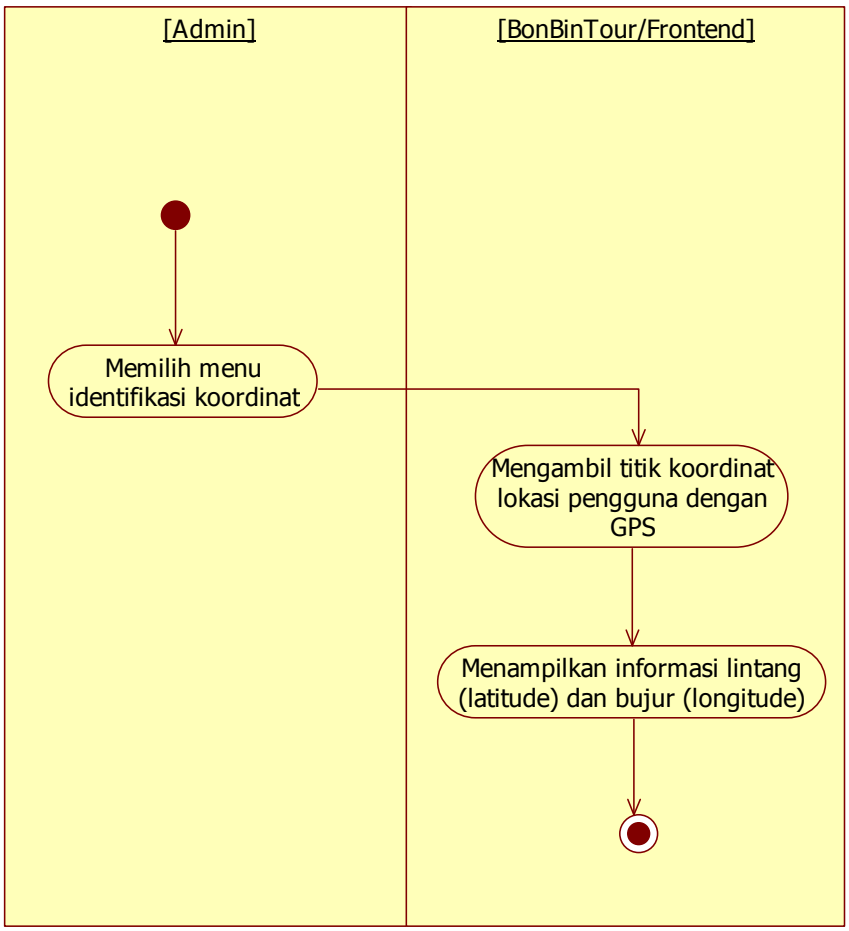
Gambar 3. 50 Activity Diagram Detail POI



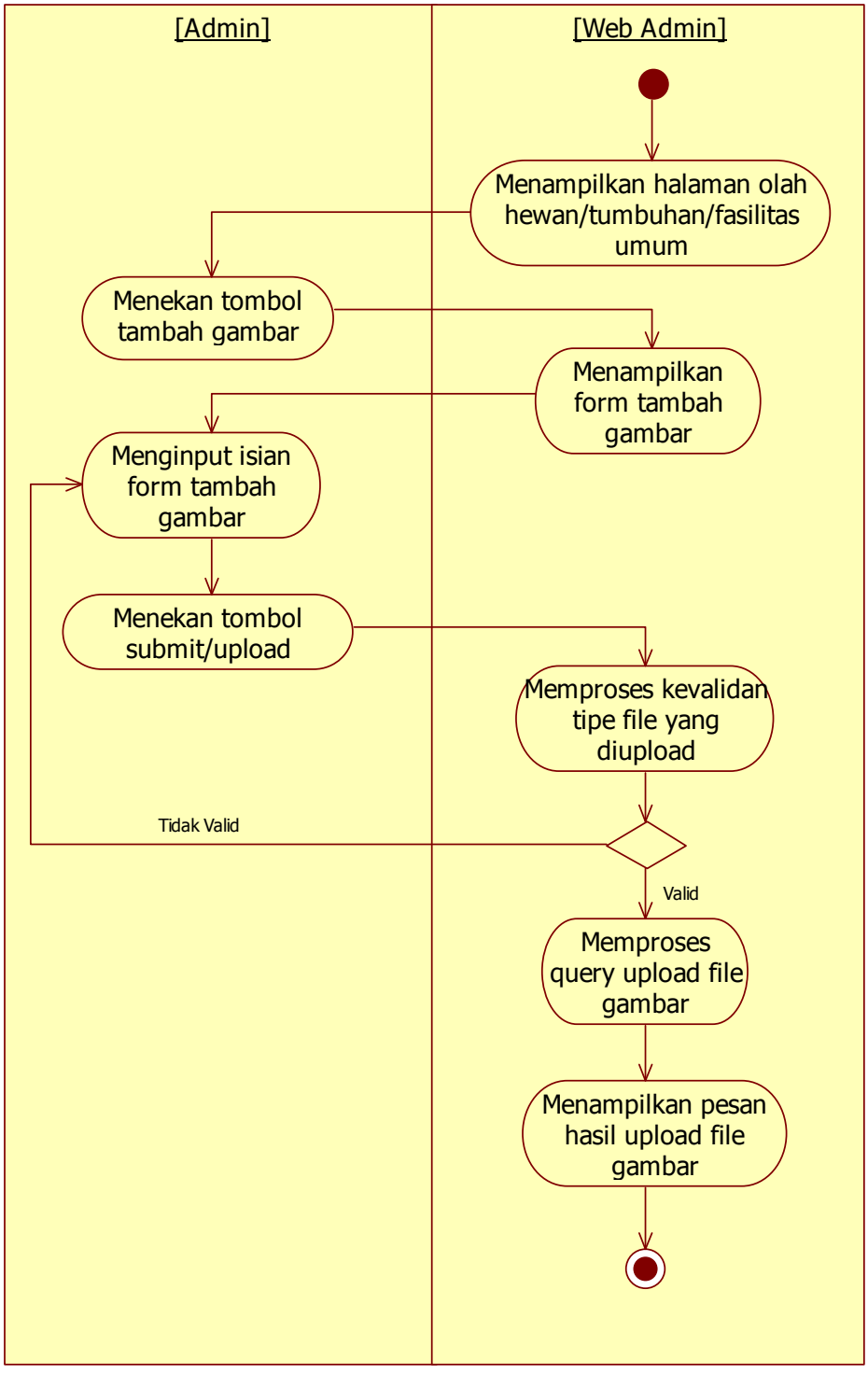
Gambar 3. 51 Activity Diagram Ubah POI.



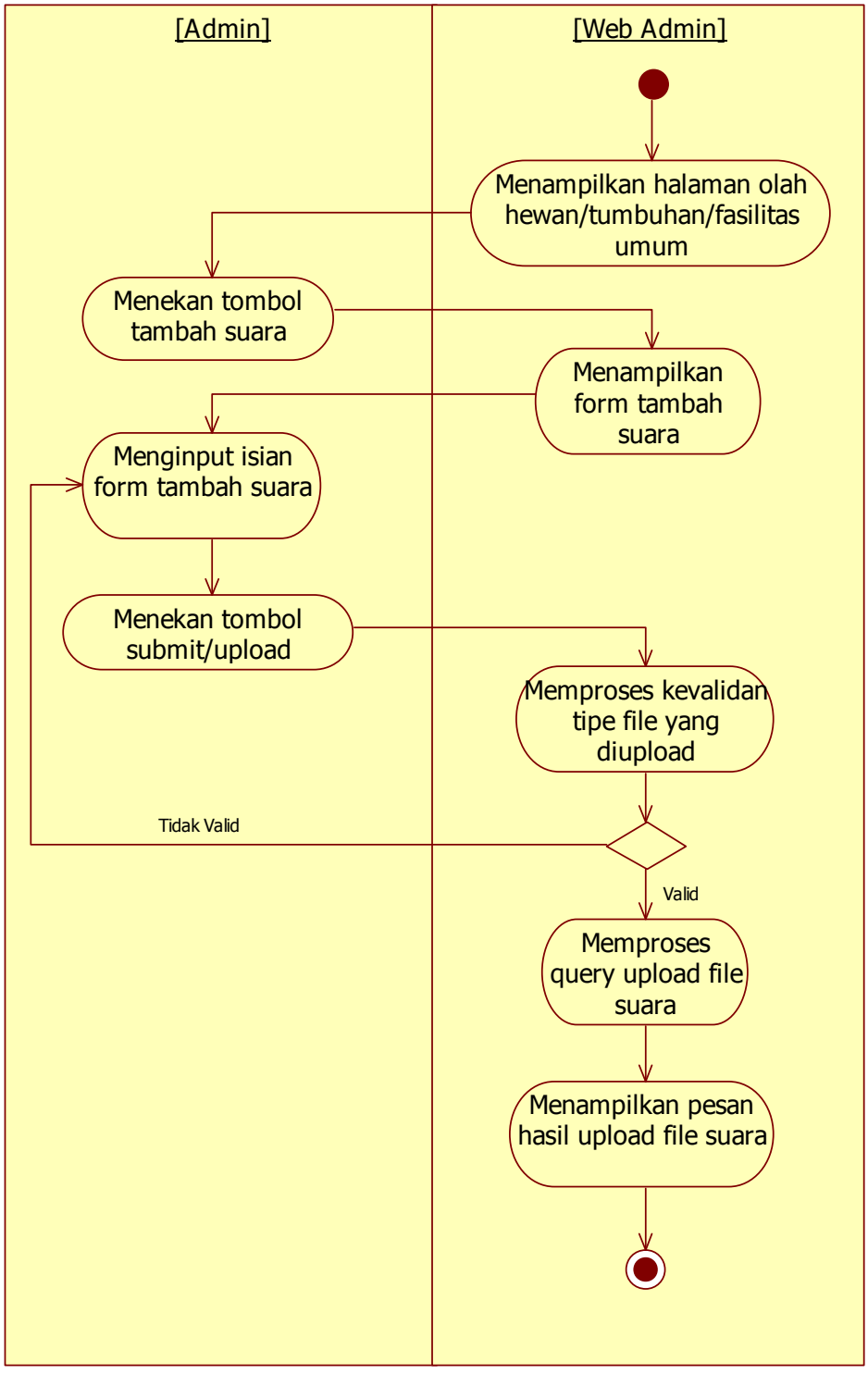
Gambar 3. 52 Activity Diagram Hapus POI.



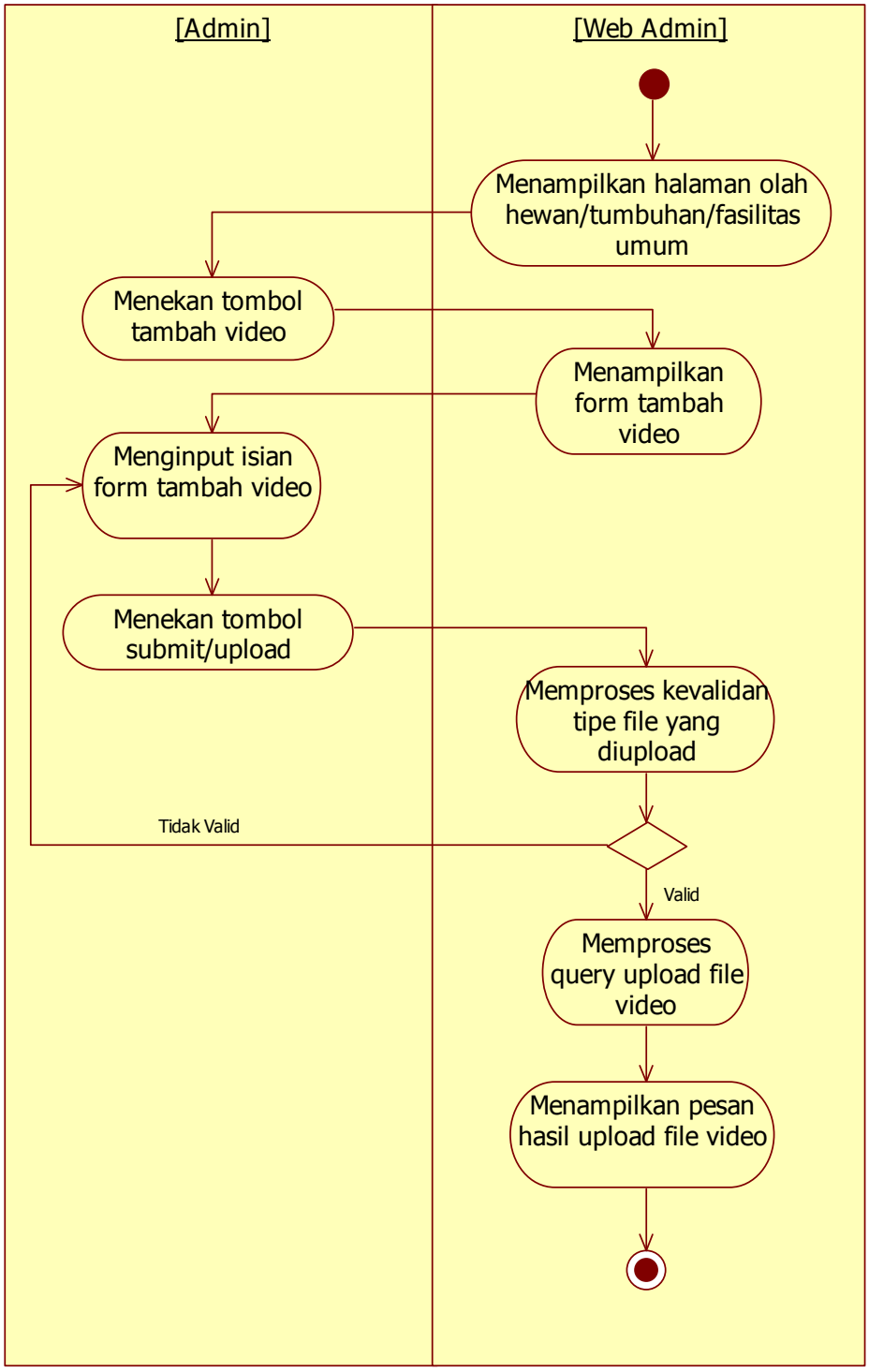
Gambar 3. 53 Activity Diagram Identifikasi Koordinat.



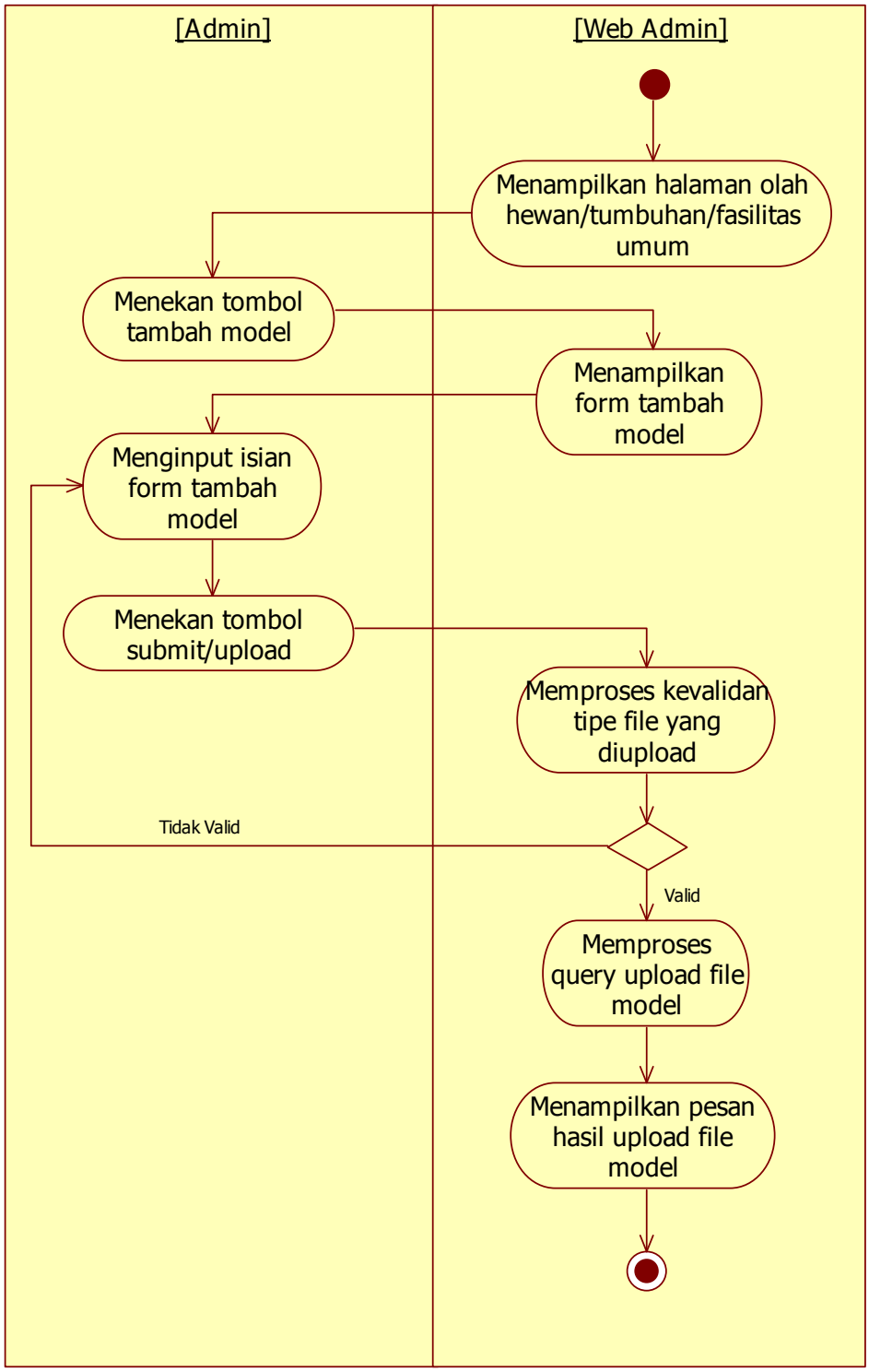
Gambar 3. 54 Activity Diagram Tambah Gambar.



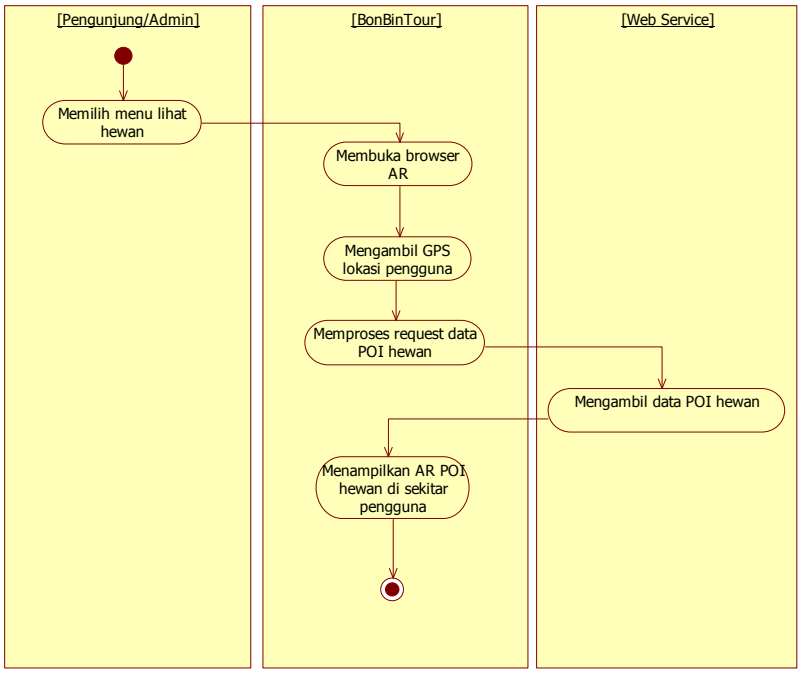
Gambar 3. 55 Activity Diagram Tambah Suara.



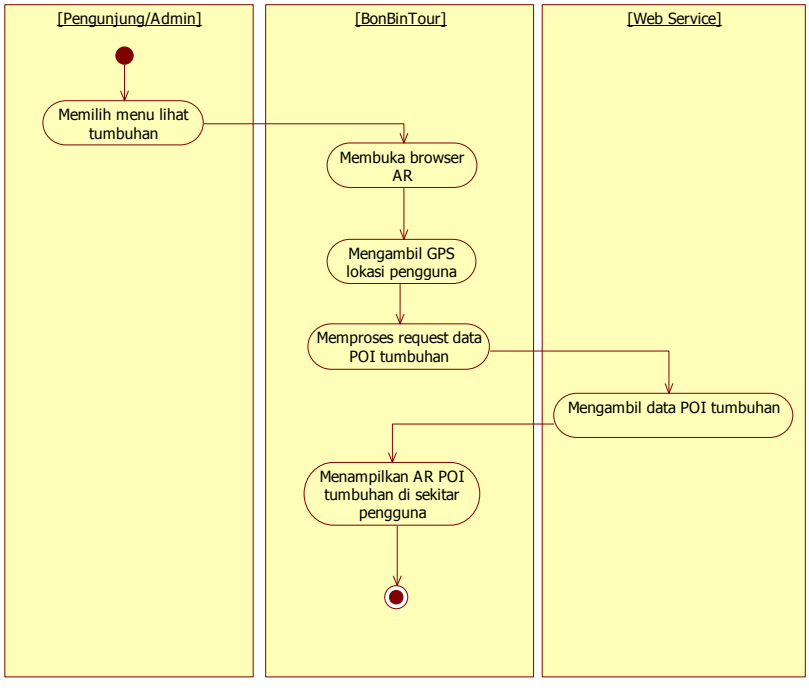
Gambar 3. 56 Activity Diagram Tambah Video.



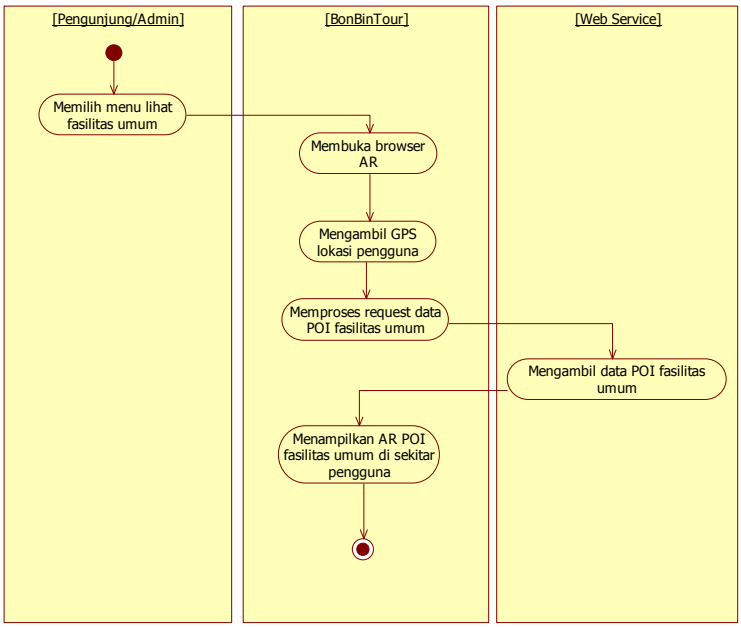
Gambar 3. 57 Activity Diagram Tambah Model.



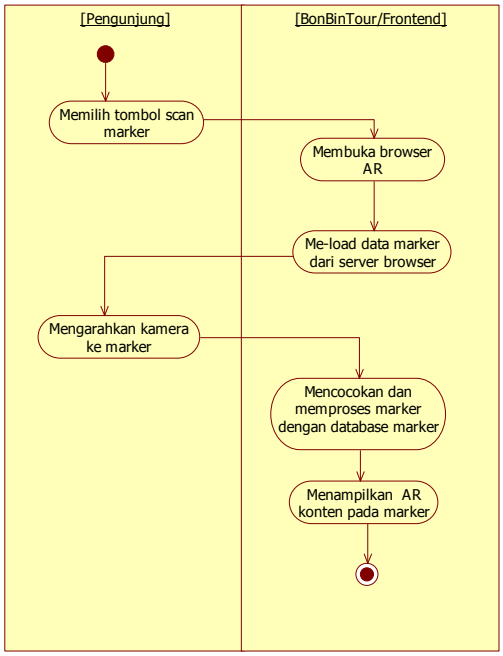
Gambar 3. 58 Activity Diagram Tampil Hewan Sekitar.



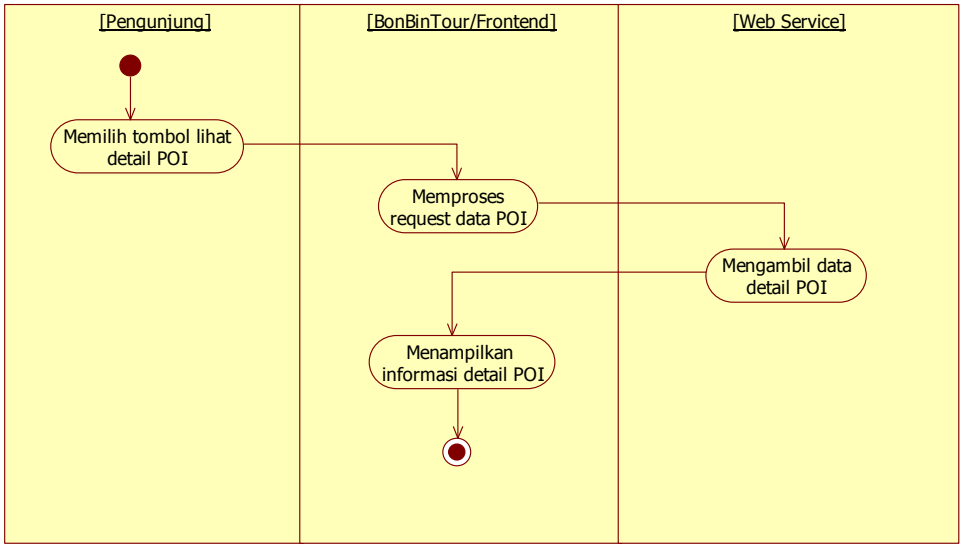
Gambar 3. 59 Activity Diagram Tampil Tumbuhan Sekitar.



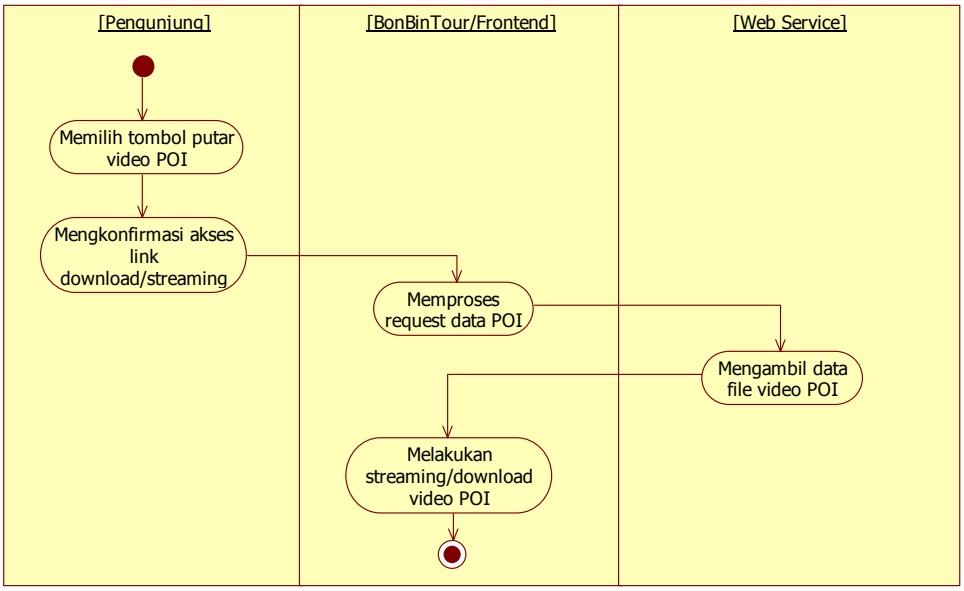
Gambar 3. 60 Activity Diagram Tampil Fasilitas Umum Sekitar.



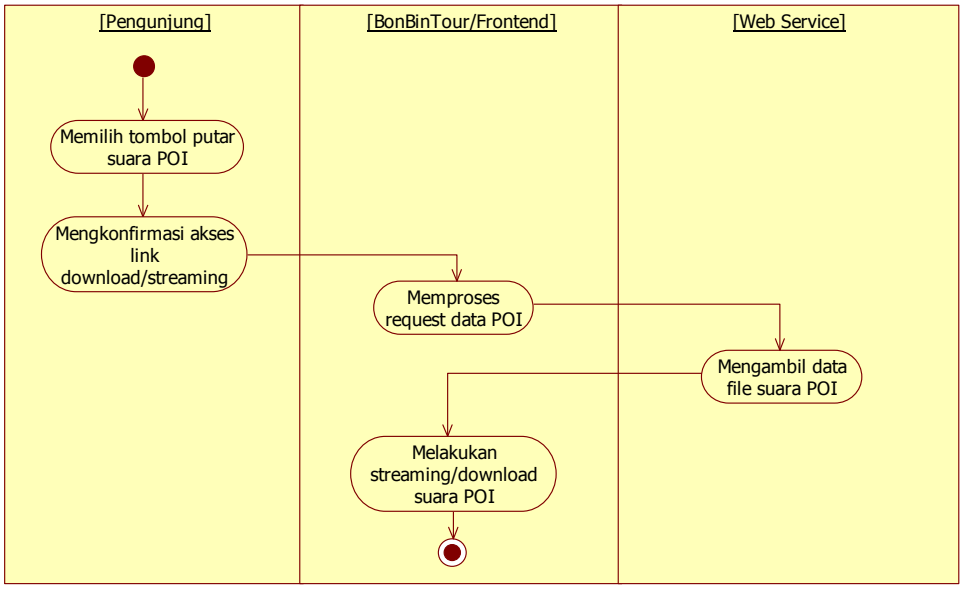
Gambar 3. 61 Activity Diagram Scan Marker.



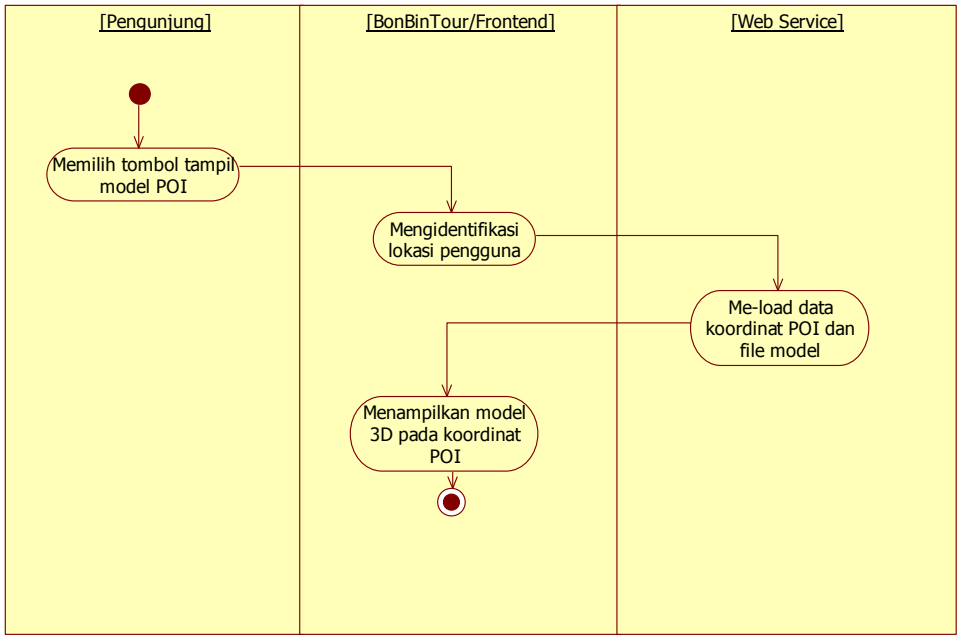
Gambar 3. 62 Activity Diagram Lihat Detail.



Gambar 3. 63 Activity Diagram Putar Video.



Gambar 3. 64 Activity Diagram Putar Suara.



Gambar 3. 65 Activity Diagram Tampil Model.

C. Class Diagram

Untuk lebih jelas mengenai pembahasan ini lihat diagram use case yang terlampir pada lampiran A.

Berikut pemaparan deskripsi *class diagram* yang dibangun :

Tabel 3. 63 Deskripsi Class Diagram.

No	*Class*	*Method*	Jenis Kelas	Deskripsi
1.	*Login*	*verifikasi(), logout()*	*Control*	*Class* yang mengontrol hak akses ke dalam sistem.
2.	*v_login*	-	*Boundary*	*Class interface form login.*
3.	*Kingdom*	*tambahKingdom(), detail(), ubahKingdom(), hapus(), _set_fields(), _set_rules()*	*Control*	*Class* yang mengontrol pengolahan data kingdom.
4.	*Mdl_Kingdom*	*list_all(), count_all(), get_by_id(), save(), update(), delete()*	*Entity*	*Class* yang berhubungan dengan data *kingdom.*.
5.	*list_kingdom*	-	*Boundary*	*Class interface list kingdom*
6.	*ubah_kingdom*	-	*Boundary*	*Class interface form kingdom*
7.	*detail_kingdom*	-	*Boundary*	*Class interface detail kingdom*
8.	*Filum*	*tambahFilum(), detail(), ubahFilum(), hapus(), _set_fields(), _set_rules()*	*Control*	*Class* yang mengontrol pengolahan data *filum.*
9.	*Mdl_Filum*	*list_all(), count_all(), get_by_id(), save(), update(), delete()*	*Entity*	*Class* yang berhubungan dengan data *filum.*.

10.	*list_filum*	-	*Boundary*	*Class interface list filum*
11.	*ubah_filum*	-	*Boundary*	*Class interface form filum*
12.	*detail_filum*	-	*Boundary*	*Class interface detail filum*
13.	*Kelas*	*tambahKelas(), detail(), ubahKelas(), hapus(), _set_fields(), _set_rules()*	*Control*	*Class* yang mengontrol pengolahan data *kelas*.
14.	*Mdl_Kelas*	*list_all(), count_all(), get_by_id(), save(), update(), delete()*	*Entity*	*Class* yang berhubungan dengan data *kelas.*.
15.	*list_kelas*	-	*Boundary*	*Class interface list kelas*
16.	*ubah_kelas*	-	*Boundary*	*Class interface form kelas*
17.	*detail_kelas*	-	*Boundary*	*Class interface detail kelas*
18.	*Ordo*	*tambahOrdo(), detail(), ubahOrdo(), hapus(), _set_fields(), _set_rules()*	*Control*	*Class* yang mengontrol pengolahan data *ordo*.
19.	*Mdl_Ordo*	*list_all(), count_all(), get_by_id(), save(), update(), delete()*	*Entity*	*Class* yang berhubungan dengan data *ordo.*.
20.	*list_ordo*	-	*Boundary*	*Class interface list ordo*
21.	*ubah_ordo*	-	*Boundary*	*Class interface form ordo*
22.	*detail_ordo*	-	*Boundary*	*Class interface*

				detail ordo
23.	*Famili*	*tambahFamili(), detail(), ubahFamili(), hapus(), _set_fields(), _set_rules()*	*Control*	*Class* yang mengontrol pengolahan data *famili*.
24.	*Mdl_Famili*	*list_all(), count_all(), get_by_id(), save(), update(), delete()*	*Entity*	*Class* yang berhubungan dengan data *famili*..
25.	*list_famili*	-	*Boundary*	*Class interface list famili*
26.	*ubah_famili*	-	*Boundary*	*Class interface form famili*
27.	*detail_famili*	-	*Boundary*	*Class interface detail famili*
28.	*Genus*	*tambahGenus(), detail(), ubahGenus(), hapus(), _set_fields(), _set_rules()*	*Control*	*Class* yang mengontrol pengolahan data *genus*.
29.	*Mdl_Genus*	*list_all(), count_all(), get_by_id(), save(), update(), delete()*	*Entity*	*Class* yang berhubungan dengan data *genus*..
30.	*list_genus*	-	*Boundary*	*Class interface list genus*
31.	*ubah_genus*	-	*Boundary*	*Class interface form genus*
32.	*detail_genus*	-	*Boundary*	*Class interface detail genus*
33.	*Spesies*	*tambahSpesies(), detail(), ubahSpesies(), hapus(), _set_fields(), _set_rules()*	*Control*	*Class* yang mengontrol pengolahan data *spesies*.
34.	*Mdl_Spesies*	*list_all(), count_all(), get_by_id(), save(), update(), delete()*	*Entity*	*Class* yang berhubungan dengan data

				spesies..
35.	*list_spesies*	-	*Boundary*	*Class interface list spesies*
36.	*ubah_spesies*	-	*Boundary*	*Class interface form spesies*
37.	*detail_spesies*	-	*Boundary*	*Class interface detail spesies*
38.	*Hewan*	*tambahHewan(), detail(), ubahHewan(), hapus(), _set_fields(), _set_rules()*	*Control*	*Class* yang mengontrol pengolahan data *hewan*.
39.	*Mdl_Hewan*	*list_all(), count_all(), get_by_id(), save(), update(), delete()*	*Entity*	*Class* yang berhubungan dengan data *hewan*..
40.	*list_hewan*	-	*Boundary*	*Class interface list hewan*
41.	*ubah_hewan*	-	*Boundary*	*Class interface form hewan*
42.	*detail_hewan*	-	*Boundary*	*Class interface detail hewan*
43.	*Tumbuhan*	*tambahTumbuhan(), detail(), ubahTumbuhan(), hapus(), _set_fields(), _set_rules()*	*Control*	*Class* yang mengontrol pengolahan data *tumbuhan*.
44.	*Mdl_Tumbuhan*	*list_all(), count_all(), get_by_id(), save(), update(), delete()*	*Entity*	*Class* yang berhubungan dengan data *tumbuhan*..
45.	*list_tumbuhan*	-	*Boundary*	*Class interface list tumbuhan*
46.	*ubah_tumbuhan*	-	*Boundary*	*Class interface form tumbuhan*

47.	*detail_tumbuhan*	-	*Boundary*	*Class interface detail tumbuhan*
48.	*Fasilitas_Umum*	*tambahFasilitas_Umum() , detail(), ubahFasilitas_Umum(), hapus(), _set_fields(), _set_rules()*	*Control*	*Class* yang mengontrol pengolahan data *fasilitas umum.*
49.	*Mdl_Fasilitas_Umum*	*list_all(), count_all(), get_by_id(), save(), update(), delete()*	*Entity*	*Class* yang berhubungan dengan data *fasilitas umum..*
50.	*list_fasilitas_umum*	-	*Boundary*	*Class interface list fasilitas umum*
51.	*ubah_fasilitas_umum*	-	*Boundary*	*Class interface form fsilitas umum*
52.	*detail_fasilitas_umum*	-	*Boundary*	*Class interface detail fasilitas umum*
53.	*POI*	*tambahPOI(), detail(), ubahPOI(), hapus(), _set_fields(), _set_rules()*	*Control*	*Class* yang mengontrol pengolahan data *POI*
54.	*Mdl_POI*	*list_all(), count_all(), get_by_id(), save(), update(), delete()*	*Entity*	*Class* yang berhubungan dengan data *POI*
55.	*list_poi*	-	*Boundary*	*Class interface list fasilitas umum*
56.	*ubah_poi*	-	*Boundary*	*Class interface form POI*
57.	*detail_poi*	-	*Boundary*	*Class interface detail POI*

58.	Suara	*tambahSuara()*	*Control*	*Class* yang mengontrol proses *upload* file suara.
59.	*v_suara*	-	*Boundary*	*Class interface form upload suara*.
60.	Video	*tambahVideo()*	*Control*	*Class* yang mengontrol proses *upload* file video.
61.	*v_video*	-	*Boundary*	*Class interface form upload video*.
62.	Gambar	*tambahGambar()*	*Control*	*Class* yang mengontrol proses *upload* file gambar.
63.	*v_gambar*	-	*Boundary*	*Class interface form upload gambar*.
64.	Model	*tambahModel()*	*Control*	*Class* yang mengontrol proses *upload* file model.
65.	*v_model*	-	*Boundary*	*Class interface form upload model*.
66.	*ARELXMLHelper*	*createLocationBasedModel(), create360Object(), createLocationBasedPOI(), createScreenFixedModel3D(), outputObject(), createSceneOptions()*	*Control*	*Class* yang mengontrol proses pertukaran data *frontend* dan *backend*.
67.	*ActivityARView*	*onCreate(), onStart(), onLocationSensorChanged(), showProgressBar()*	*Control*	*Class* yang mengontrol proses *AR browser.*
68.	*ActivityARHewan*	*onCreate(), launchView(), startChannel(),*	*Control*	*Class* yang mengontrol proses menampilkan

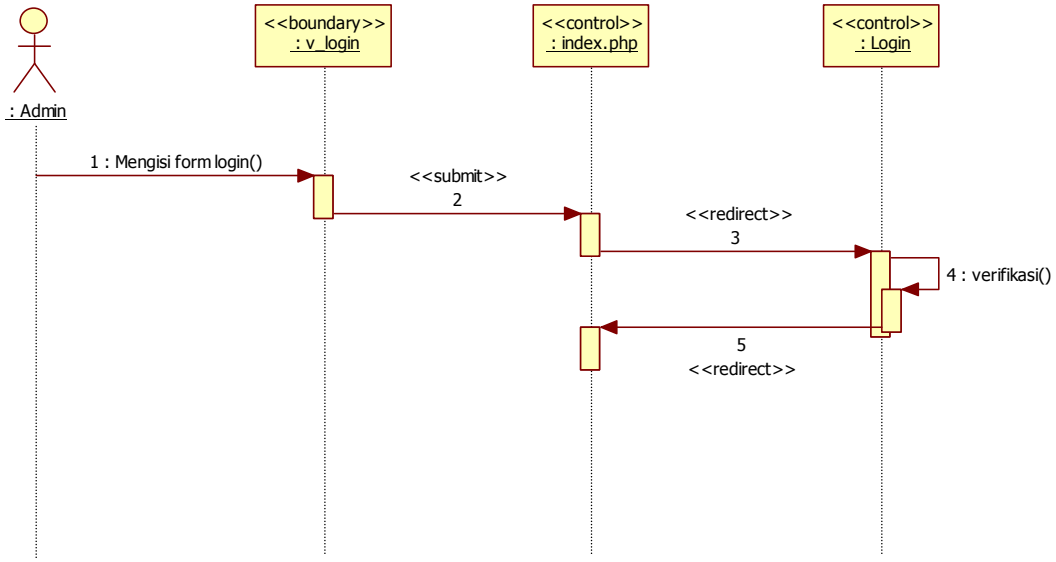
		junaioStarterTask()		*AR Hewan* sekitar.
69.	*ActivityARTumbuhan*	*onCreate(), launchView(), startChannel(), junaioStarterTask()*	*Control*	*Class* yang mengontrol proses menampilkan *AR Tumbuhan* sekitar.
70.	*ActivityARFasilitasUmum*	*onCreate(), launchView(), startChannel(), junaioStarterTask()*	*Control*	*Class* yang mengontrol proses menampilkan *AR Fasilitas Umum* sekitar.
71.	*ActivityScanMarker*	*onCreate(), launchView(), startChannel(), junaioStarterTask()*	*Control*	*Class* yang mengontrol proses identifikasi *marker*.
72.	*ActivityCurrentLocationMapV2*	*onCreate(), onMyLocation()*	*Control*	*Class* yang mengontrol proses identifikasi koordinat.
73.	*ActivityBonBinTour*	*onCreate(), onOptionItemSelected(), selectedItem(), onPostCreate(), onConfigurationChanged(), setTitle()*	*Control*	*Class* yang mengontrol keseluruhan aksi yang terdapat pada sisi *frontend*.
74.	*SherlockFragmentActivity*	-	*Control*	*Class* yang mengontrol tampilan antarmuka *frontend*.
75.	*OnMyLocationChangeListener*	-	*Control*	*Class* yang mengontrol pergerakan posisi pengguna dalam peta.
76.	*MetaioCloudPlugin*	-	*Control*	*Class* yang mengontrol *AR*.

77.	*index.php*	-	*Control*	*Class* yang terdepan untuk mengakses *controller* pada konsep *MVC.*
78.	*index.xml*	-	*Control*	*Class* yang mengontrol pertukaran data *XML* untuk proses *scan marker*..
79.	*ActivityPOIHewan*	*onCreate(), toTitleCase(), processResponse (), populateListView (),*	*Control*	*Class* yang mengontrol proses menampilkan detail info *POI* hewan di sisi *frontend.*
80.	*ServerRequestPOIHewan*	*onCreate(), sendGetRequest(), sendPostRequest()*	*Control*	*Class* yang mengontrol dalam menghubungkan *server* dengan *client* melalui *parsing JSON* di sisi *frontend*.
81.	*POIHewan*	*getIdHewanPOIHewan (), setIdHewanPOIHewan ()*	*Control*	*Class* yang mengontrol proses penampungan informasi *POI* hewan di sisi *frontend.*
82.	*ListAdapterPOIHewan*	*getView()*	*Control*	*Class* yang mengontrol dalam proses menampilkan informasi detail *POI* hewan di sisi *frontend*.
83.	*ActivityPOITumbuhan*	*onCreate(), toTitleCase(), processResponse (), populateListView (),*	*Control*	*Class* yang mengontrol proses

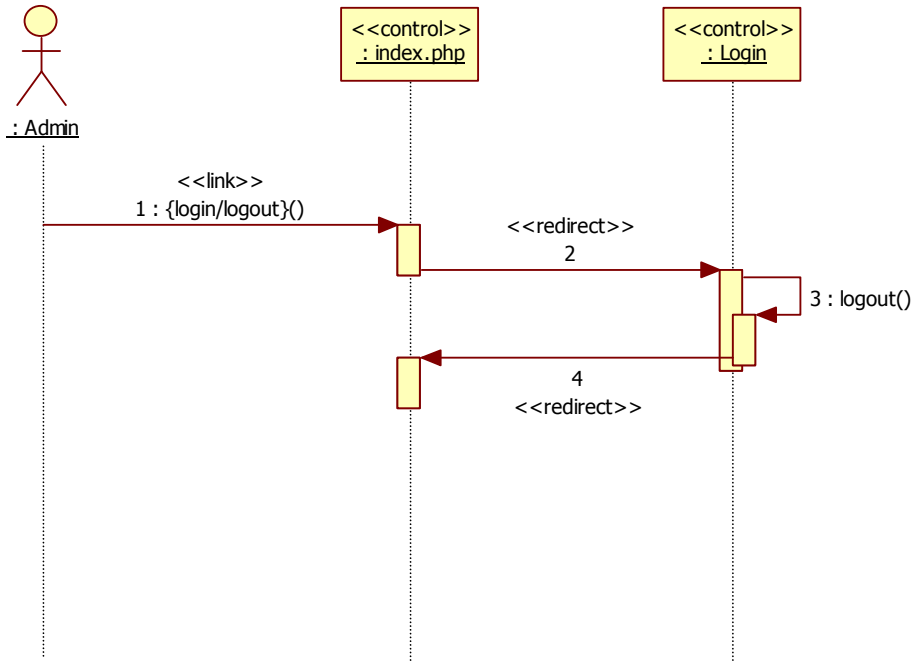
				menampilkan detail info *POI* tumbuhan di sisi *frontend*.
84.	*ServerRequestPOITumbuhan*	*onCreate(), sendGetRequest(), sendPostRequest()*	*Control*	*Class* yang mengontrol dalam menghubungkan *server* dengan *client* melalui *parsing JSON* di sisi *frontend*.
85.	*POITumbuhan*	*getIdTumbuhanPOITumbuhan(), setIdTumbuhanPOITumbuhan()*	*Control*	*Class* yang mengontrol proses penampungan informasi *POI* tumbuhan di sisi *frontend*.
86.	*ListAdapterPOITumbuhan*	*getView()*	*Control*	*Class* yang mengontrol dalam proses menampilkan informasi detail *POI* tumbuhan di sisi *frontend*.
87.	*ActivityPOIFasilitasUmum*	*onCreate(), toTitleCase(), processResponse (), populateListView (),*	*Control*	*Class* yang mengontrol proses menampilkan detail info *POI* fasilitas umum di sisi *frontend*.
88.	*ServerRequestPOIFasilitasUmum*	*onCreate(), sendGetRequest(), sendPostRequest()*	*Control*	*Class* yang mengontrol dalam menghubungkan *server* dengan *client* melalui *parsing JSON* di sisi *frontend*.

89.	*POIFasilitasUmum*	*getIdFasilitasUmumPOIFasilitasUmum(), setIdFasilitasUmumPOIFasilitasUmum()*	*Control*	*Class* yang mengontrol proses penampungan informasi *POI* fasilitas umum di sisi *frontend.*
90.	*ListAdapterPOIFasilitasUmum*	*getView()*	*Control*	*Class* yang mengontrol dalam proses menampilkan informasi detail *POI* fasilitas umum di sisi *frontend*.
91.	*ActivityVideoPlayer*	*onCreate()*	*Control*	*Class* yang mengontrol pemutaran video *POI*.
92.	*ActivityVideoPlayerView*	*onCreate()*	*Control*	*Class* yang mengontrol konektifitas video *streaming POI*.

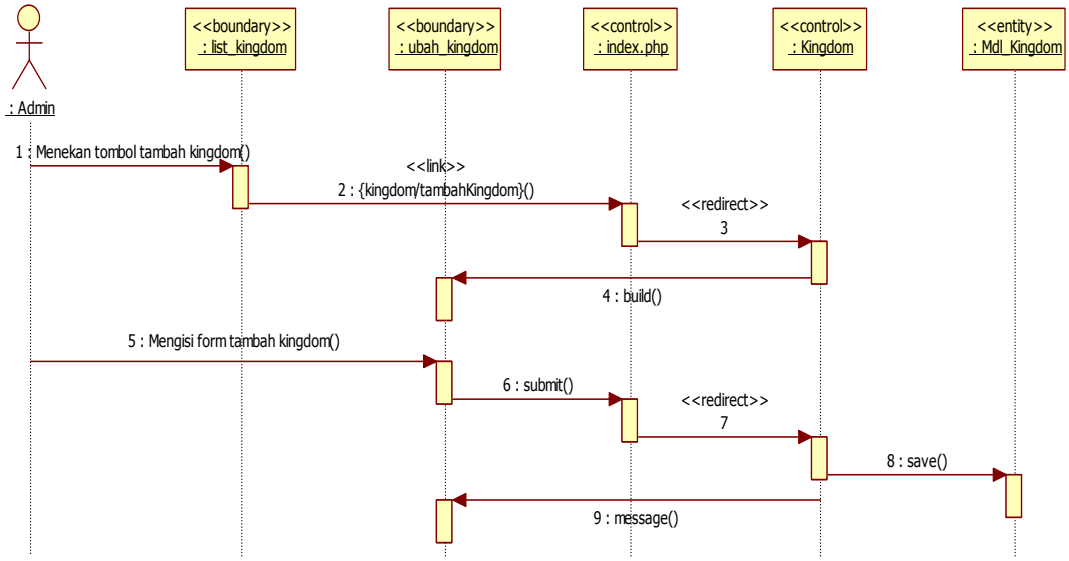
D. Sequence Diagram



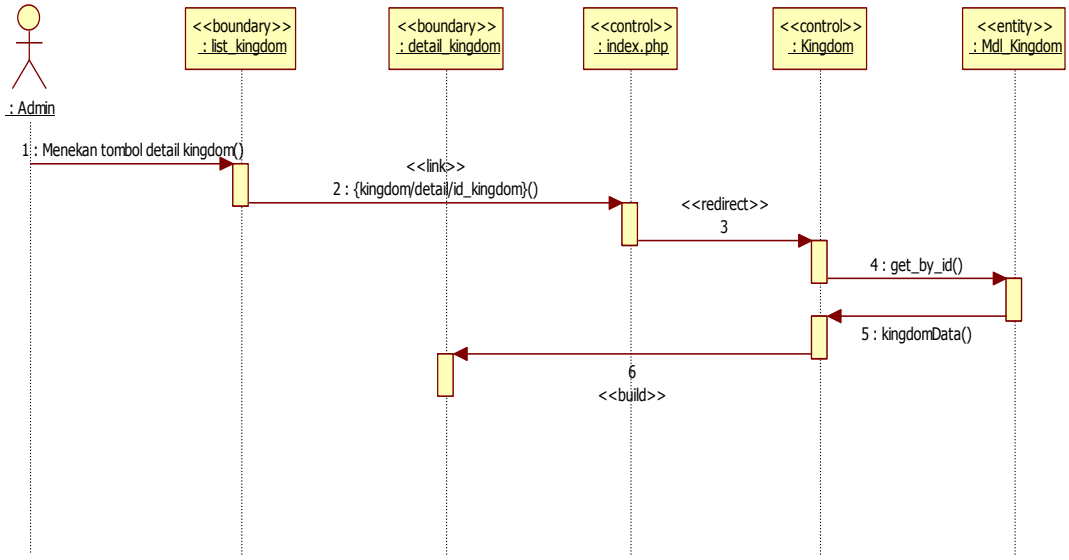
Gambar 3. 66 Sequence Diagram Login.



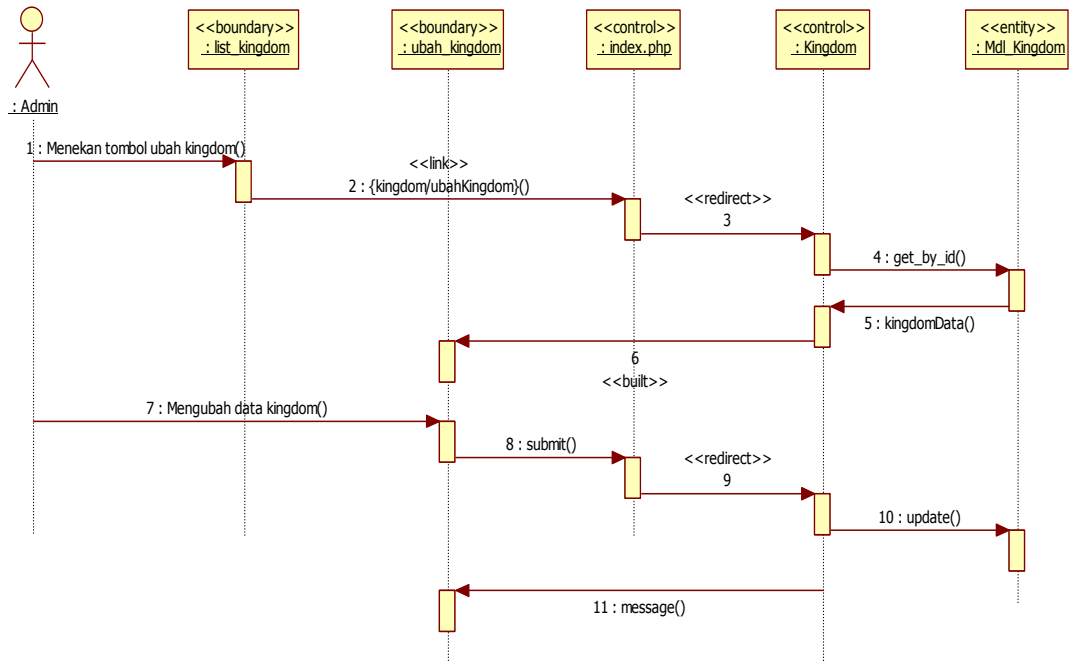
Gambar 3. 67 Sequence Diagram Logout.



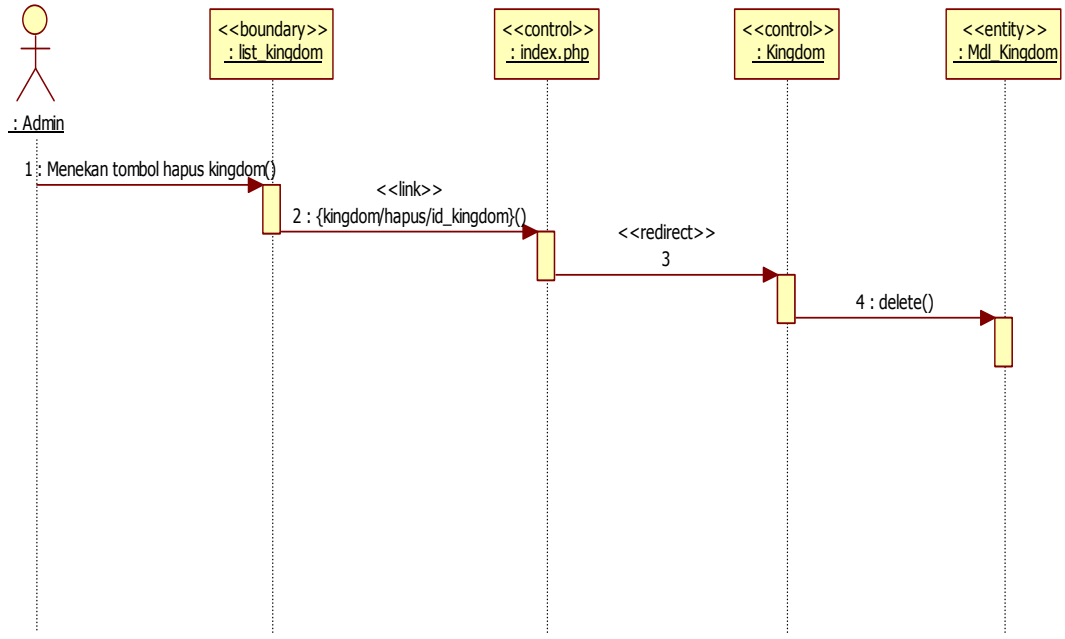
Gambar 3. 68 Sequence Diagram Tambah Kingdom.



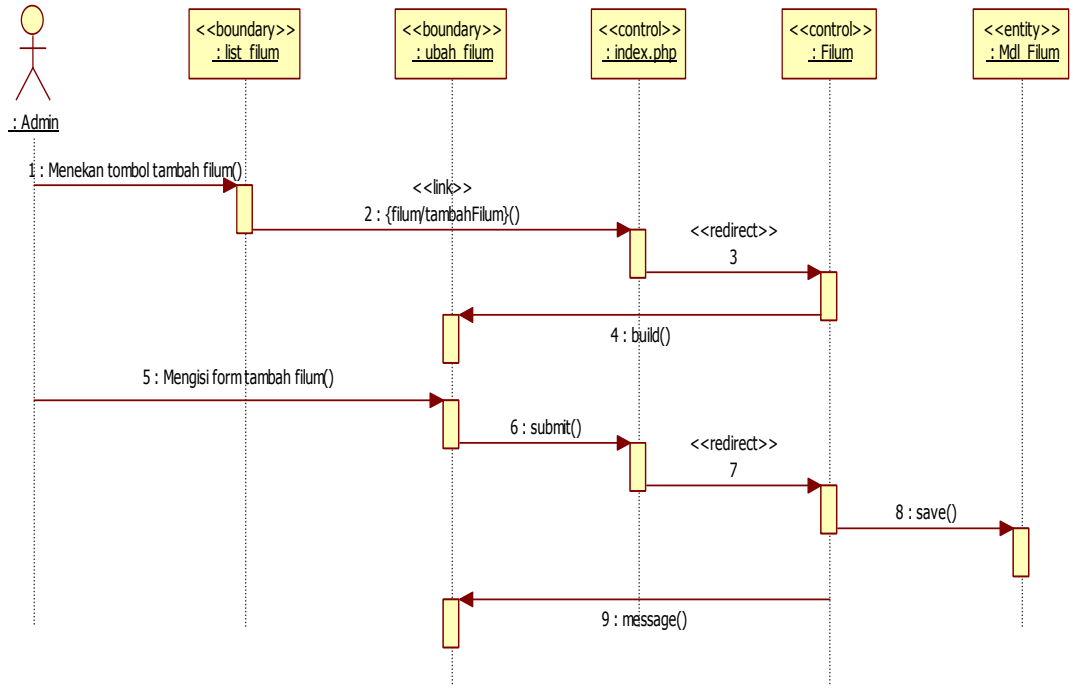
Gambar 3. 69 Sequence Diagram Detail Kingdom.



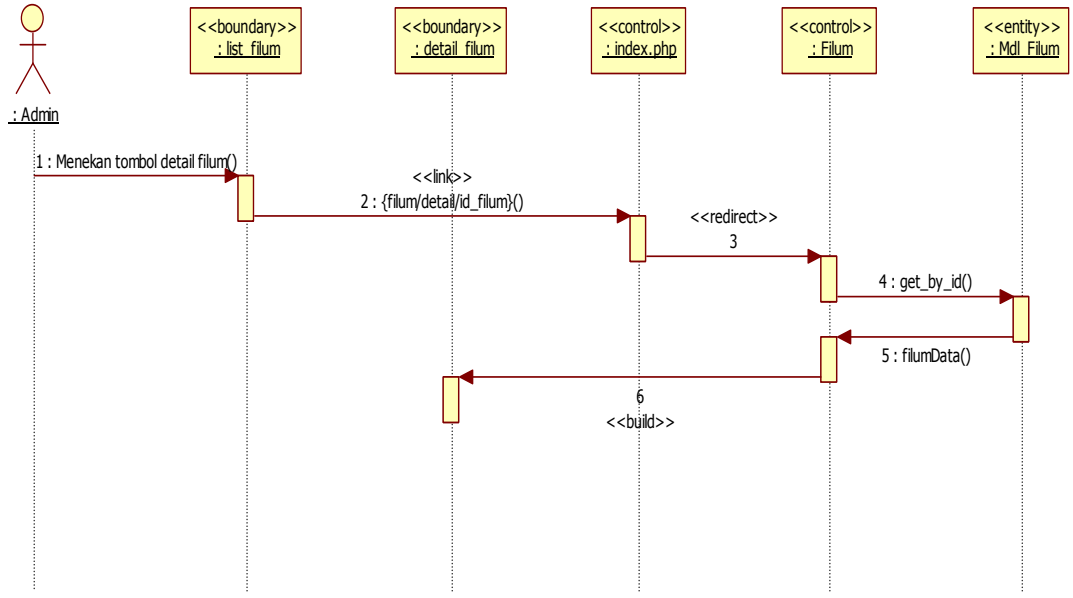
Gambar 3. 70 Sequence Diagram Ubah Kingdom.



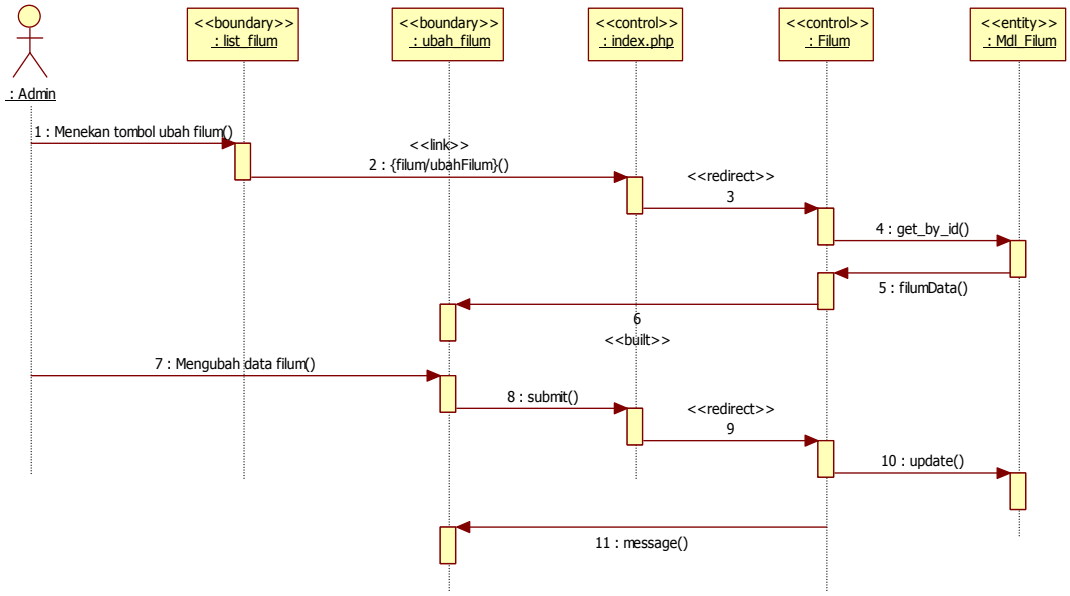
Gambar 3. 71 Sequence Diagram Hapus Kingdom.



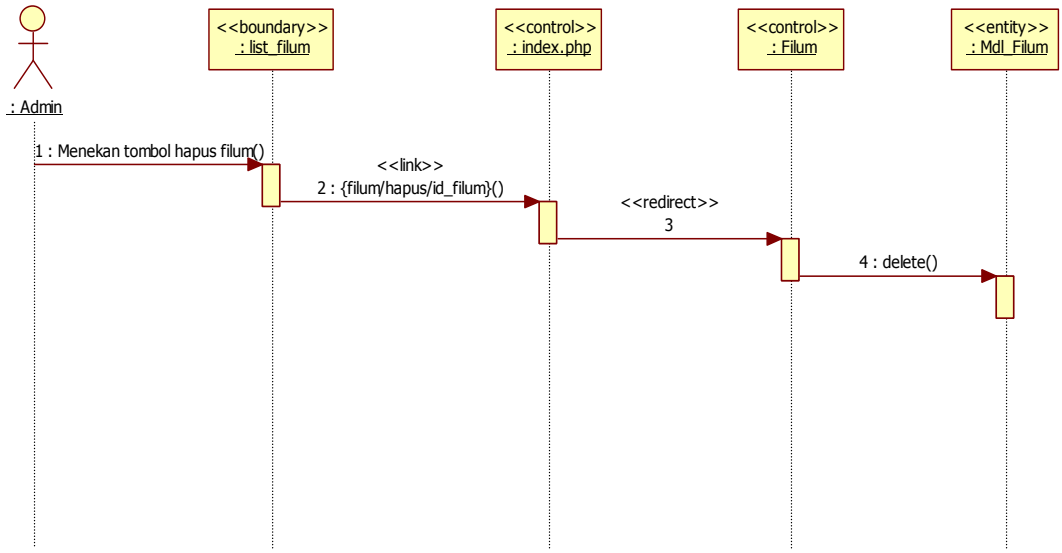
Gambar 3. 72 Sequence Diagram Tambah Filum.



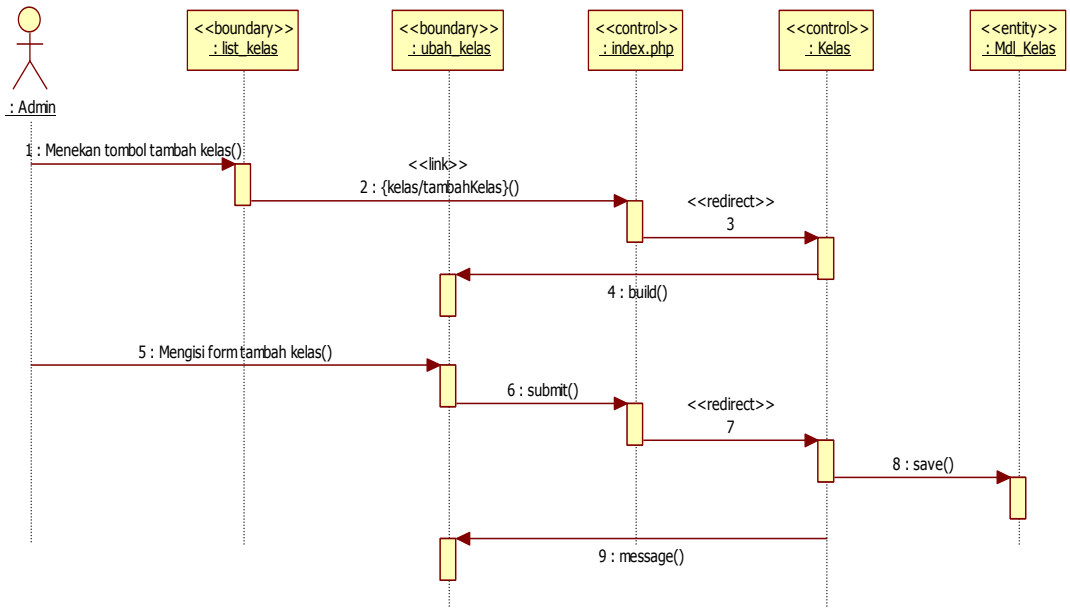
Gambar 3. 73 Sequence Diagram Detail Filum.



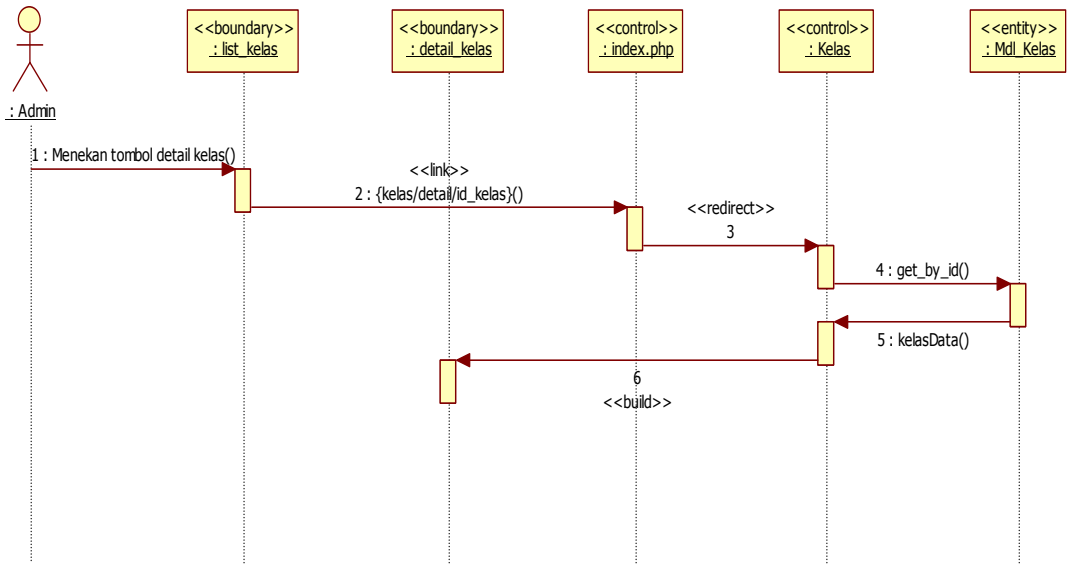
Gambar 3. 74 Sequence Diagram Ubah Filum.



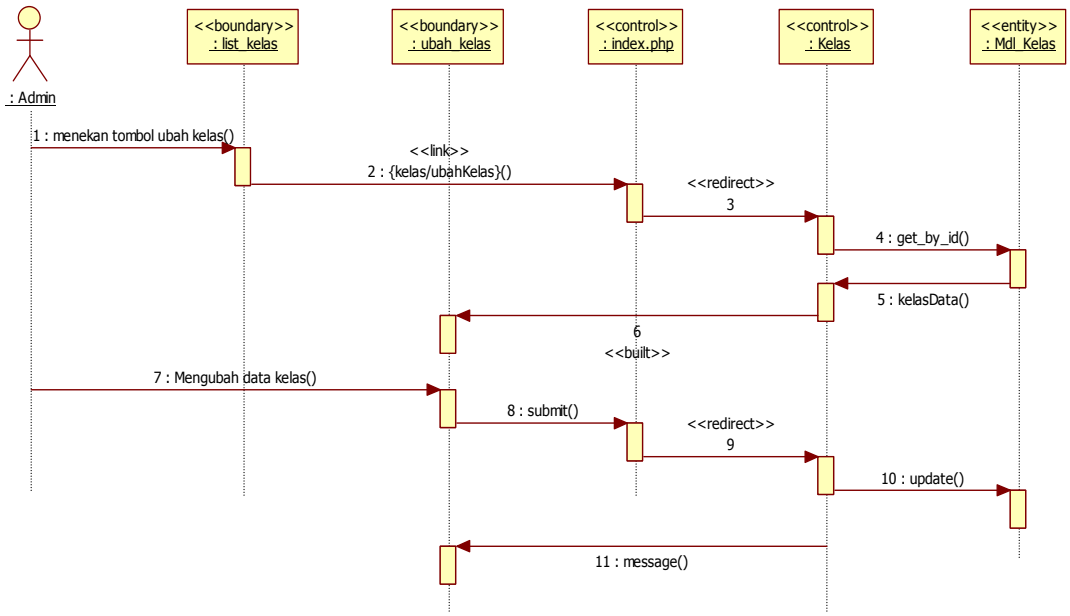
Gambar 3. 75 Sequence Diagram Hapus Filum.



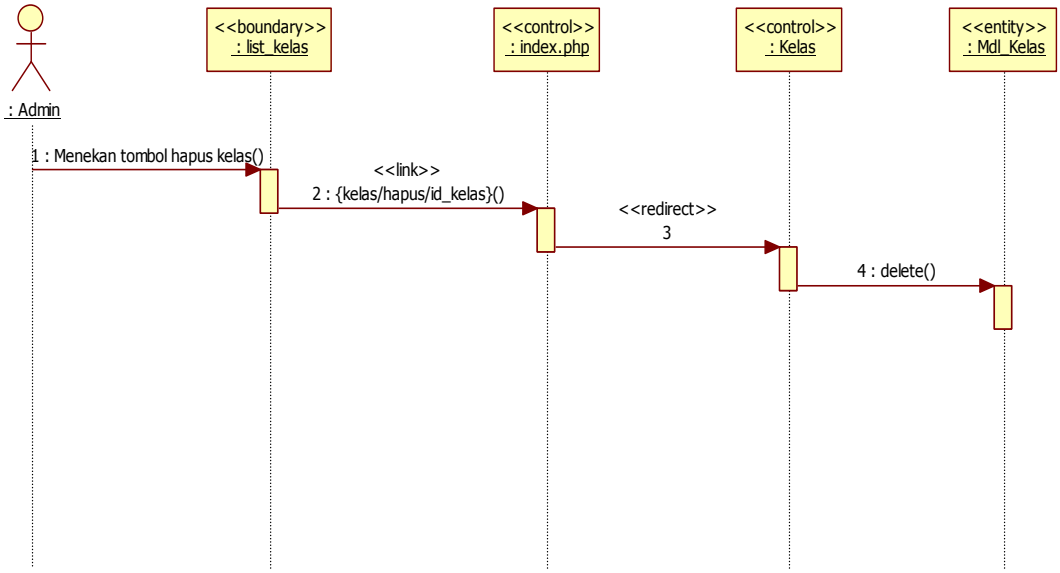
Gambar 3. 76 Sequence Diagram Tambah Kelas.



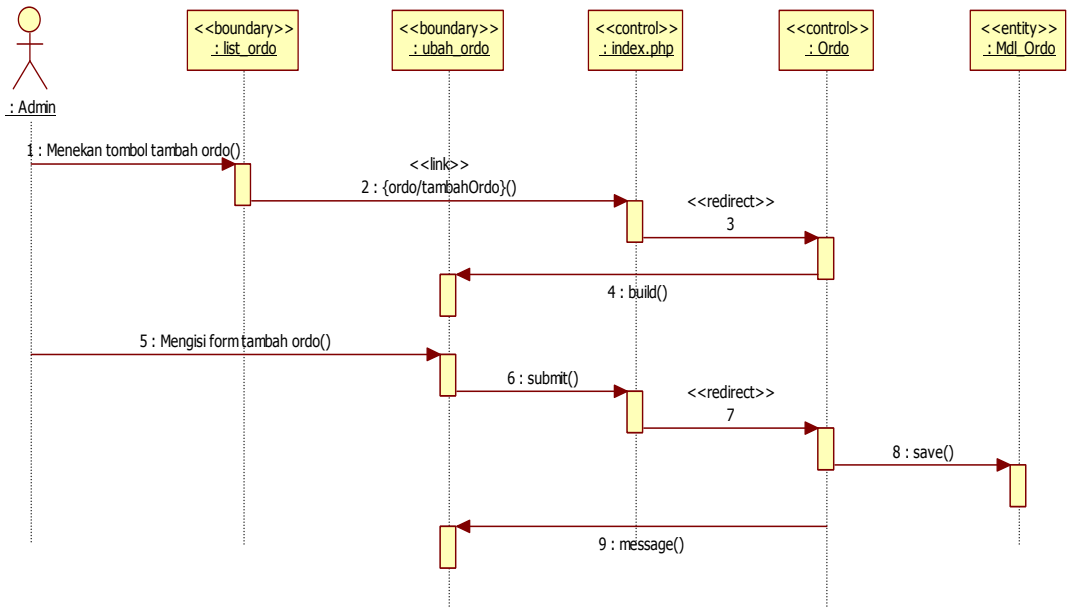
Gambar 3. 77 Sequence Diagram Detail Kelas.



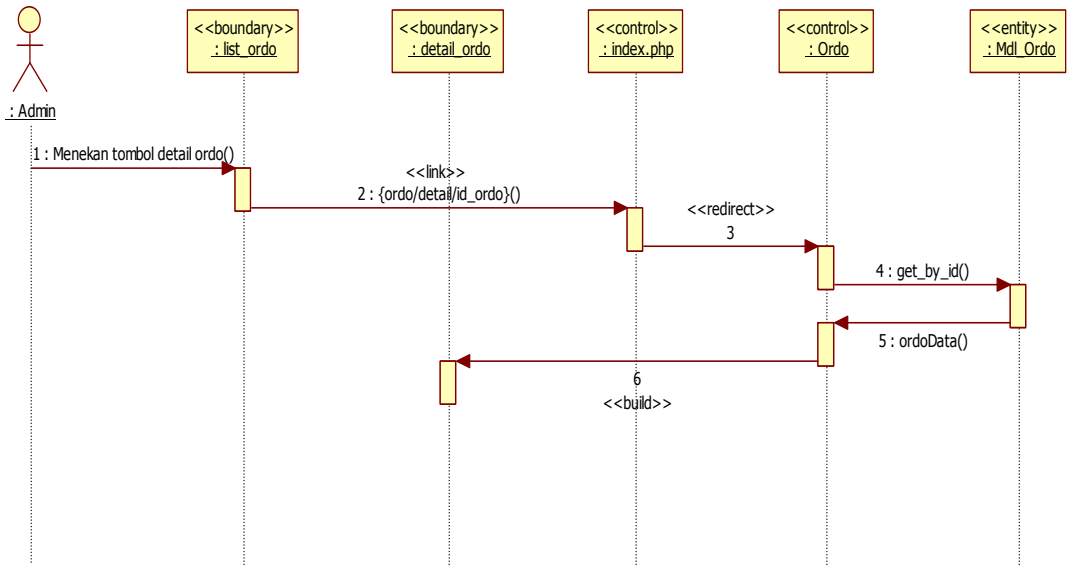
Gambar 3. 78 Sequence Diagram Ubah Kelas.



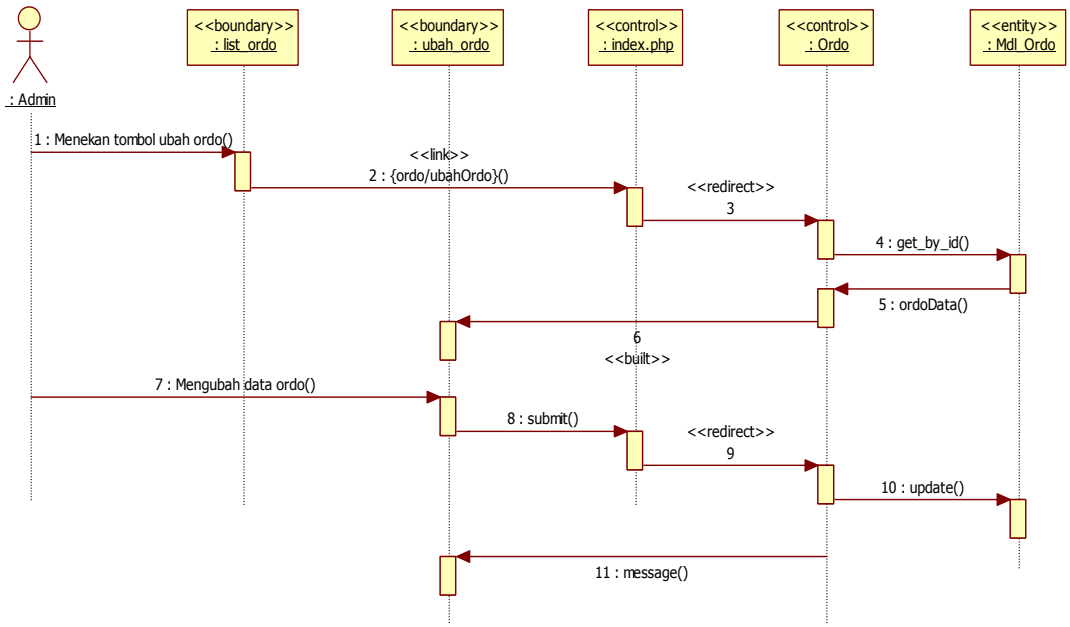
Gambar 3. 79 Sequence Diagram Hapus Kelas.



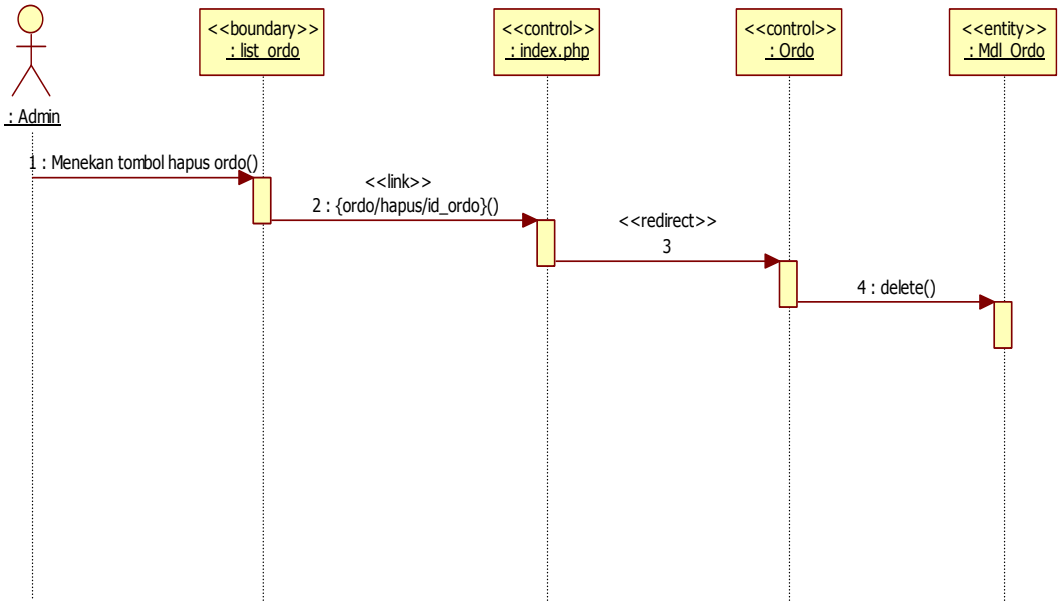
Gambar 3. 80 Sequence Diagram Tambah Ordo.



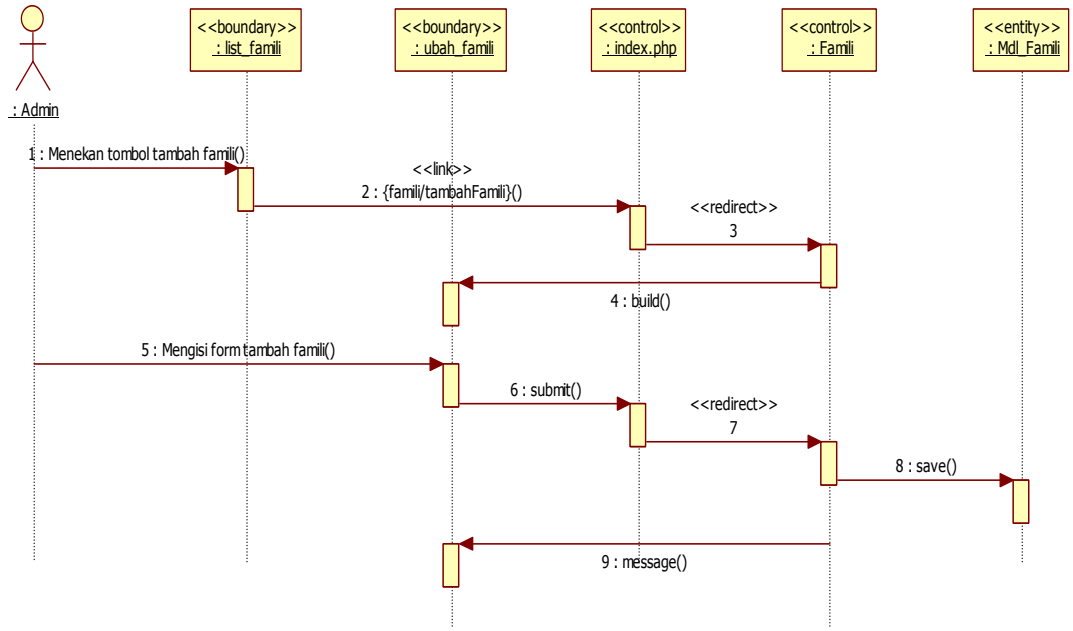
Gambar 3. 81 Sequence Diagram Detail Ordo.



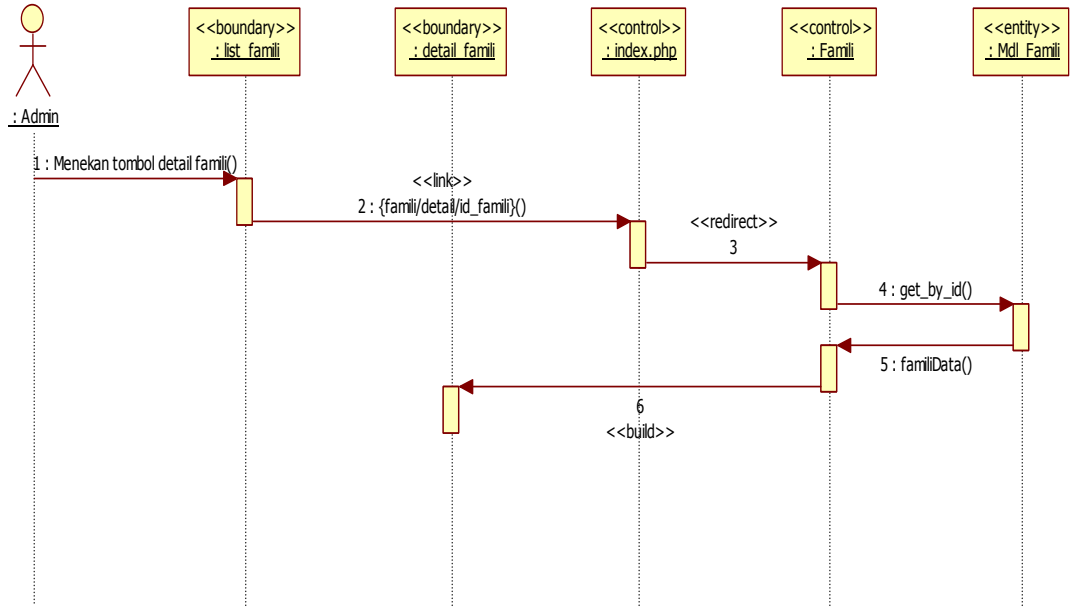
Gambar 3. 82 Sequence Diagram Ubah Ordo.



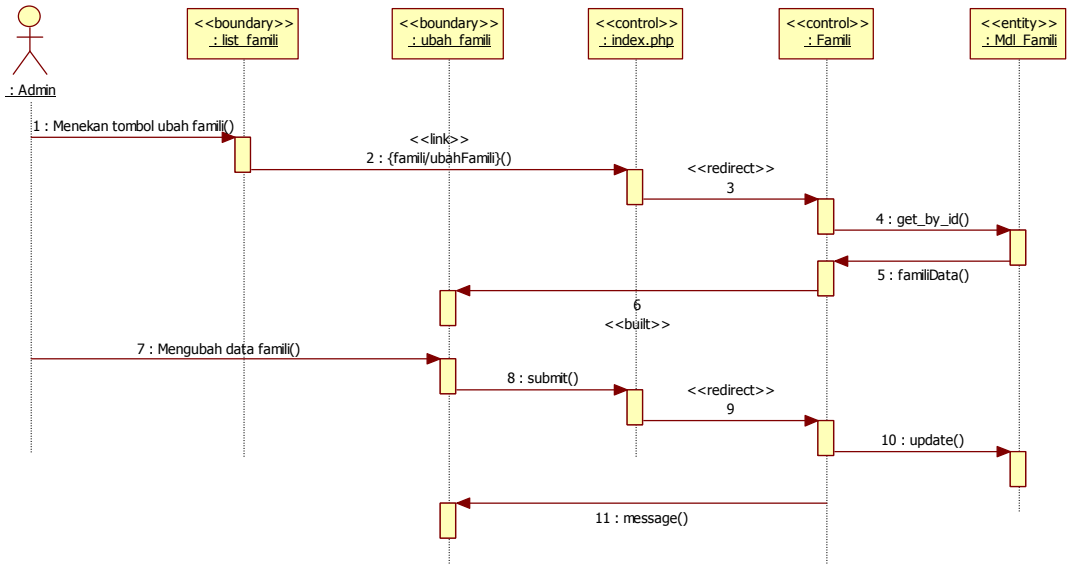
Gambar 3. 83 Sequence Diagram Hapus Ordo.



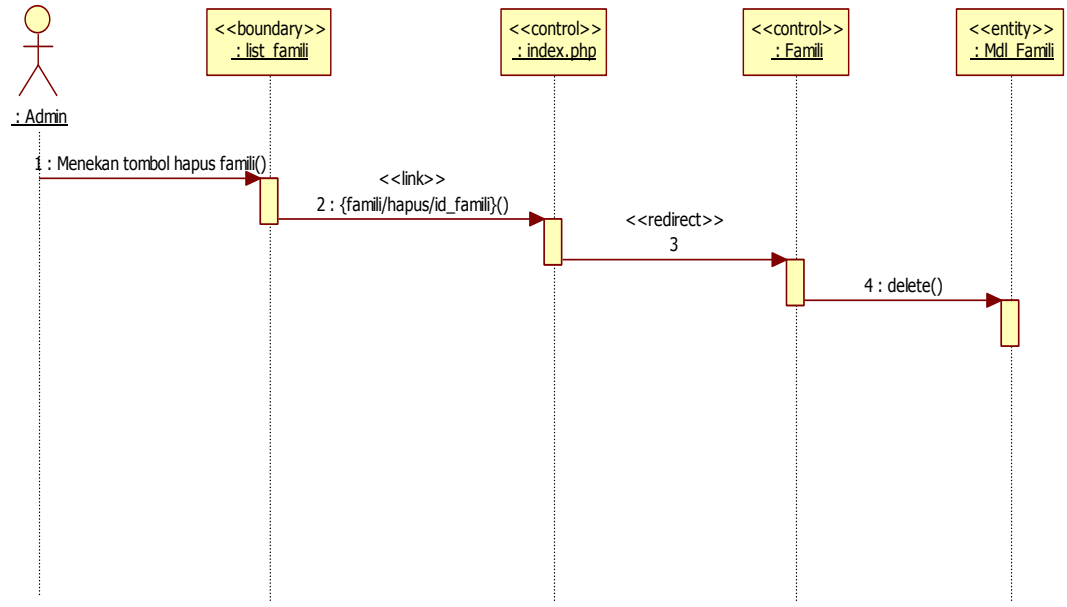
Gambar 3. 84 Sequence Diagram Tambah Famili.



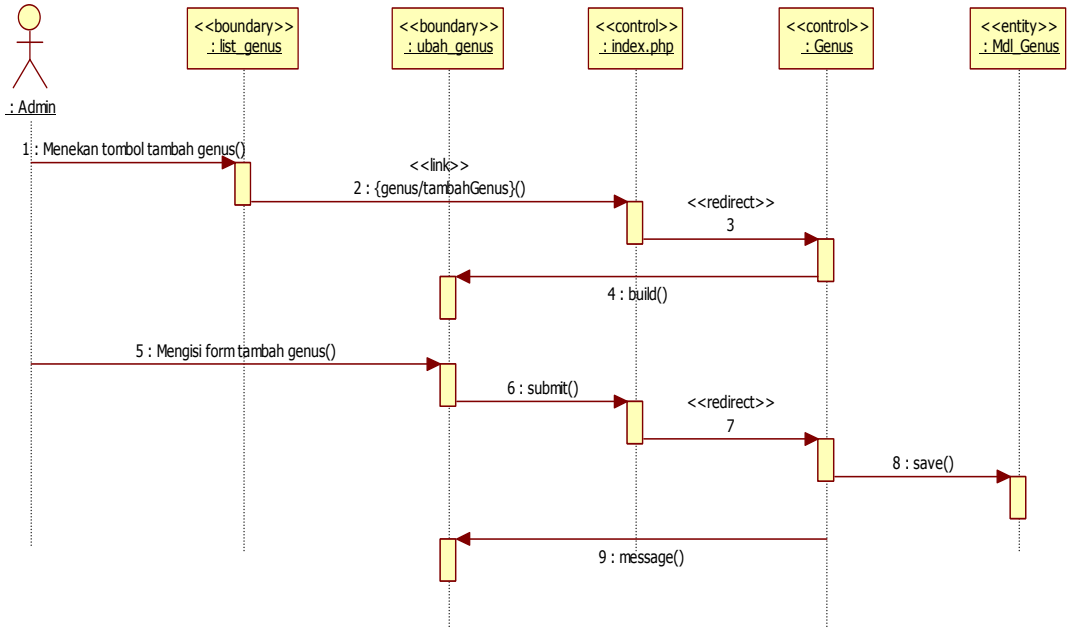
Gambar 3. 85 Sequence Diagram Detail Famili.



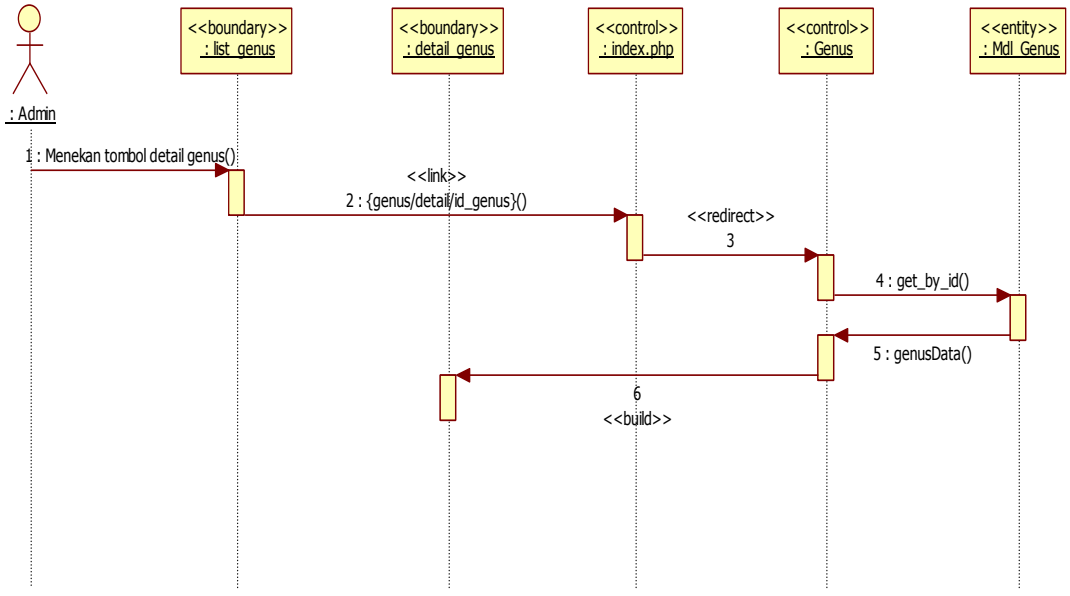
Gambar 3. 86 Sequence Diagram Ubah Famili.



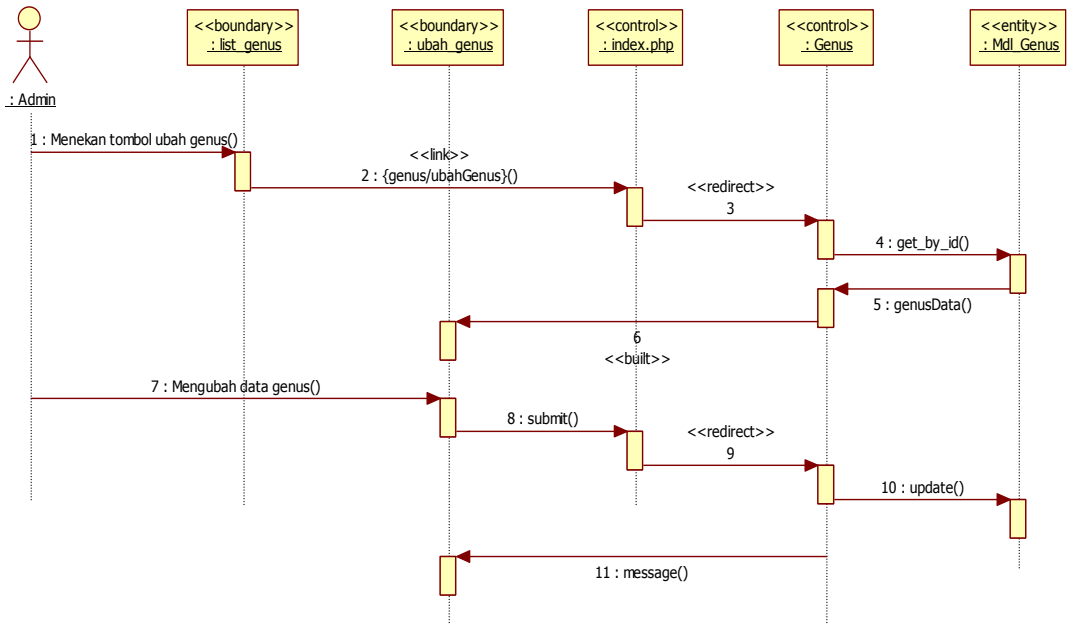
Gambar 3. 87 Sequence Diagram Hapus Famili.



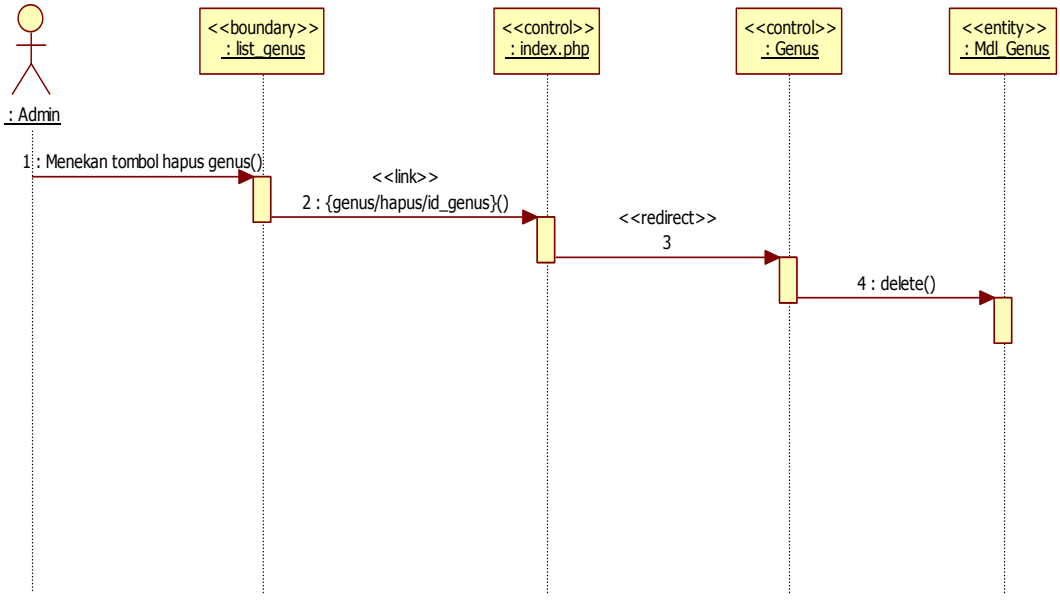
Gambar 3. 88 Sequence Diagram Tambah Genus.



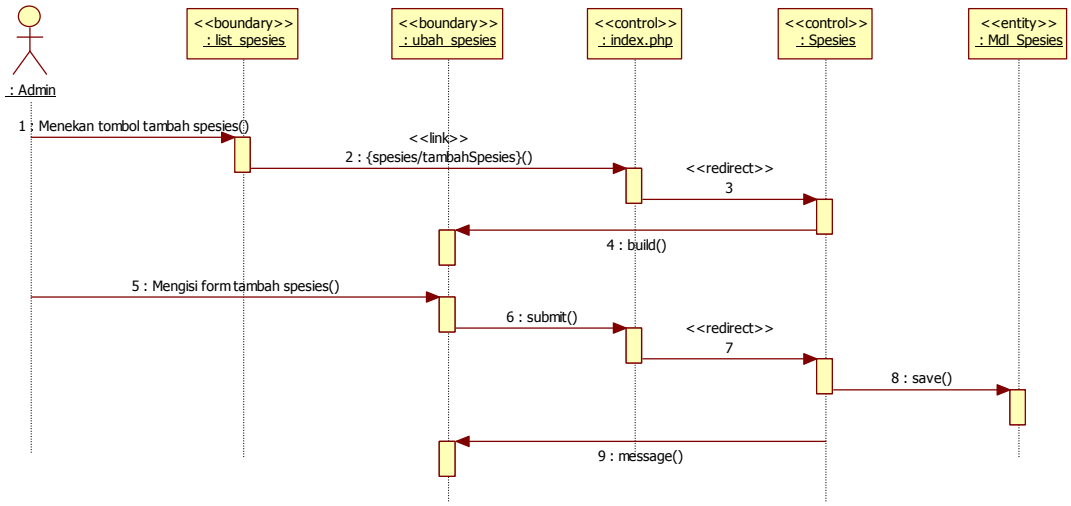
Gambar 3. 89 Sequence Diagram Detail Genus.



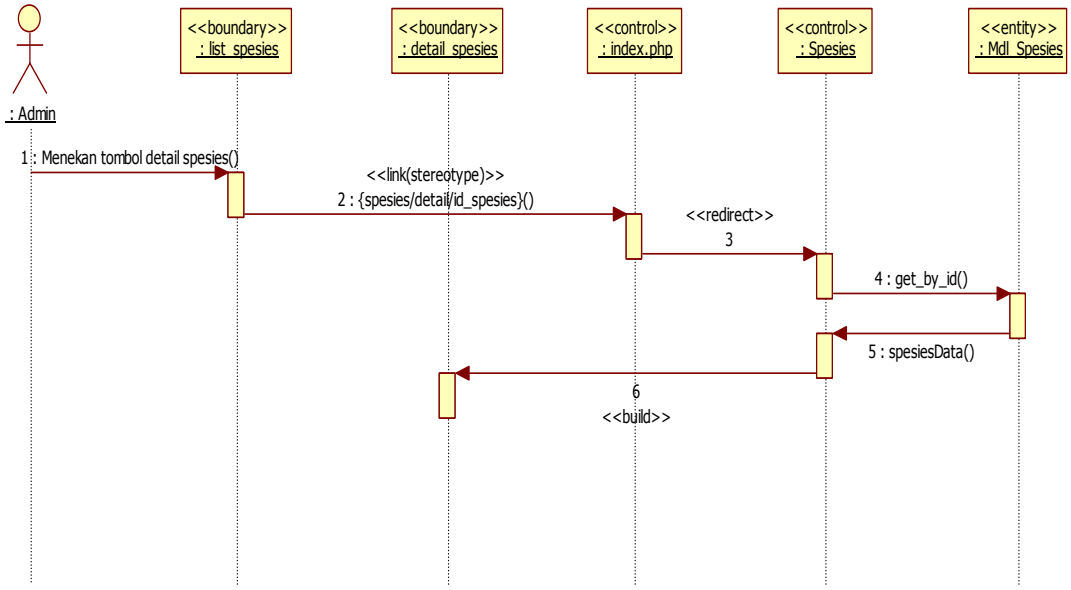
Gambar 3. 90 Sequence Diagram Ubah Genus.



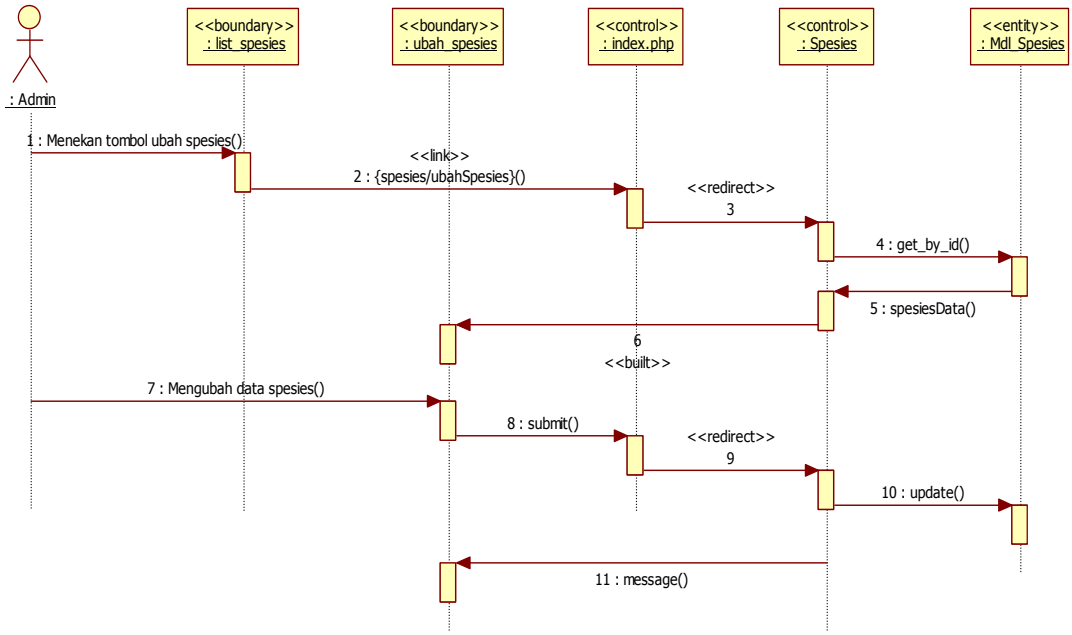
Gambar 3. 91 Sequence Diagram Hapus Genus.



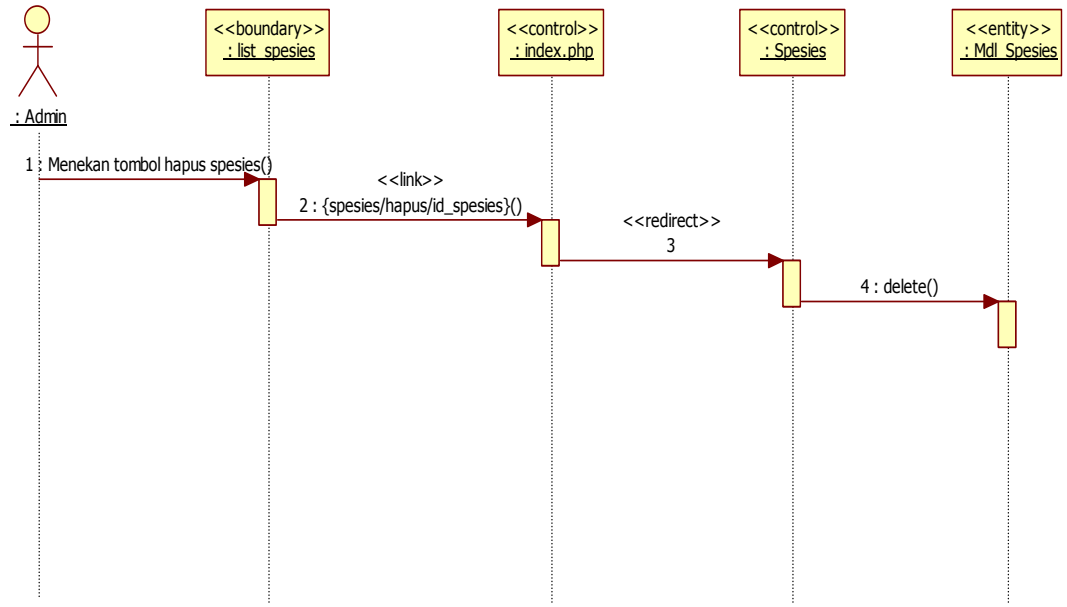
Gambar 3. 92 Sequence Diagram Tambah Spesies.



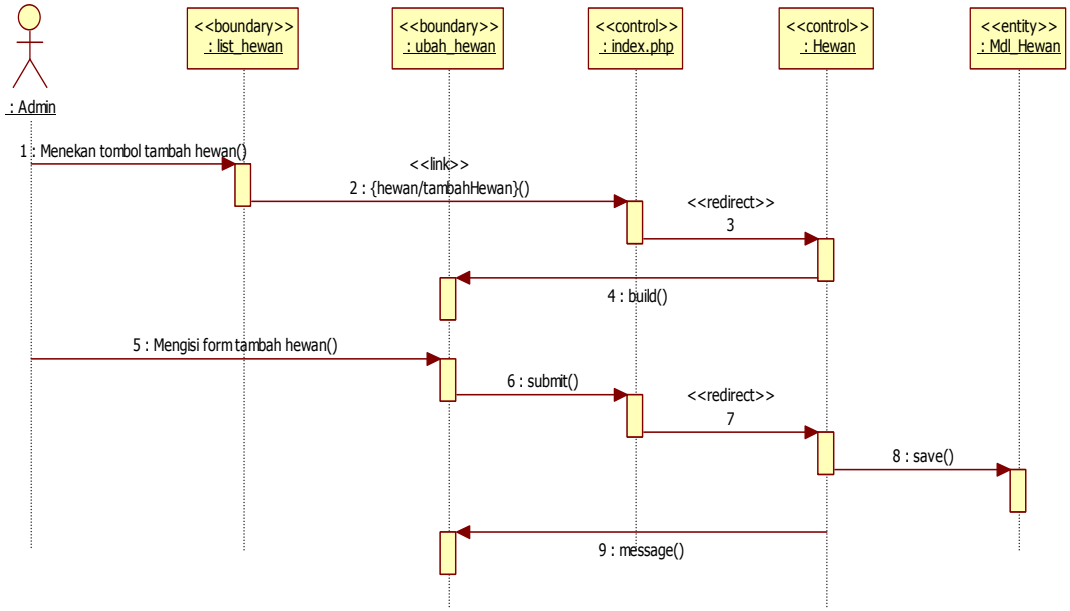
Gambar 3. 93 Sequence Diagram Detail Spesies.



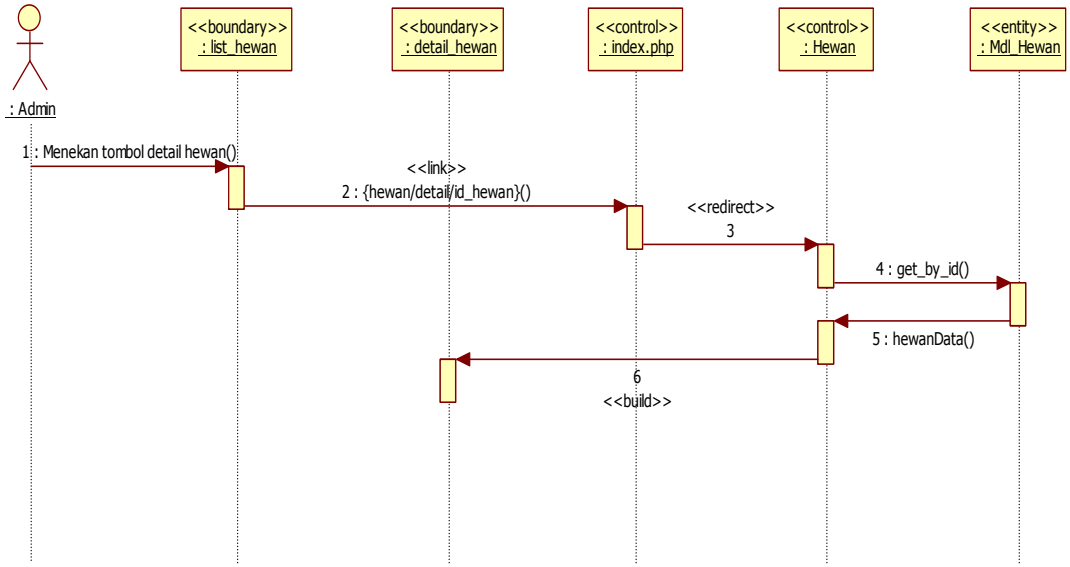
Gambar 3. 94 Sequence Diagram Ubah Spesies.



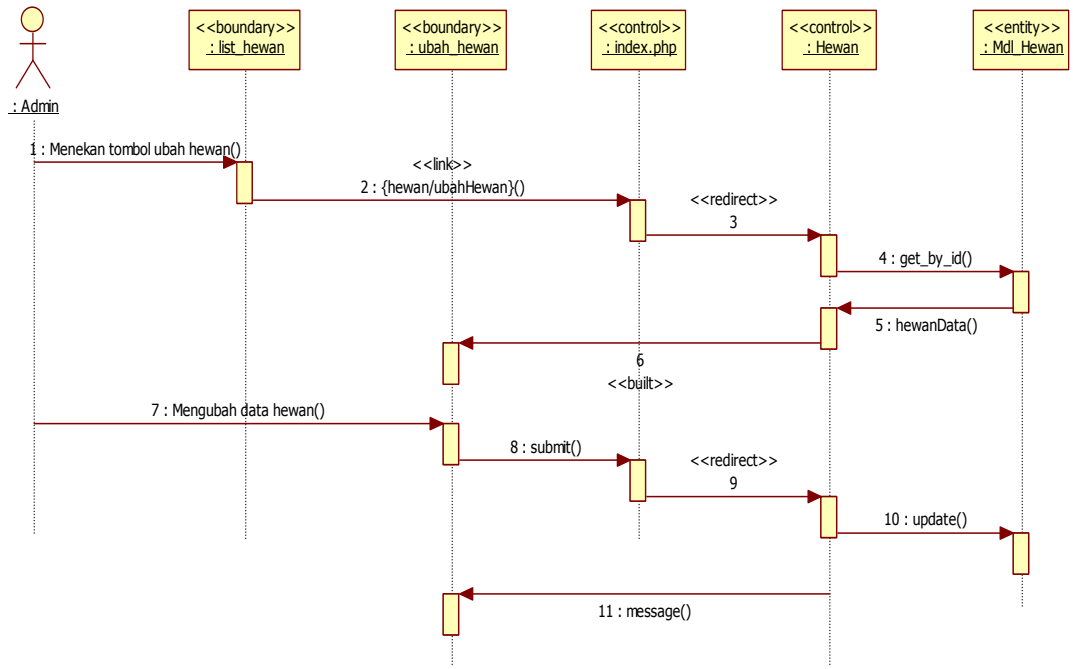
Gambar 3. 95 Sequence Diagram Hapus Spesies.



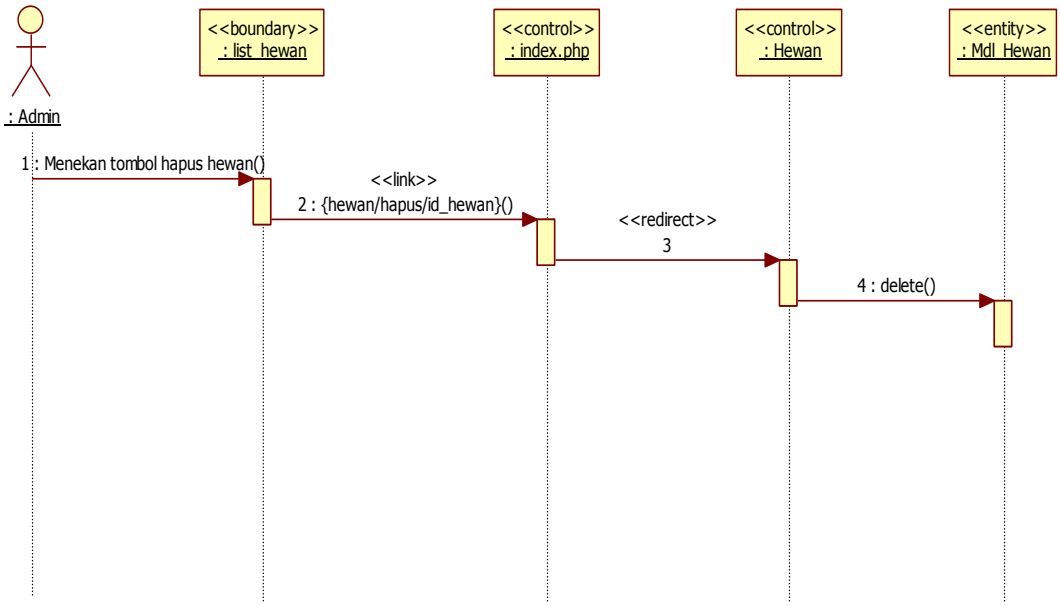
Gambar 3. 96 Sequence Diagram Tambah Hewan.



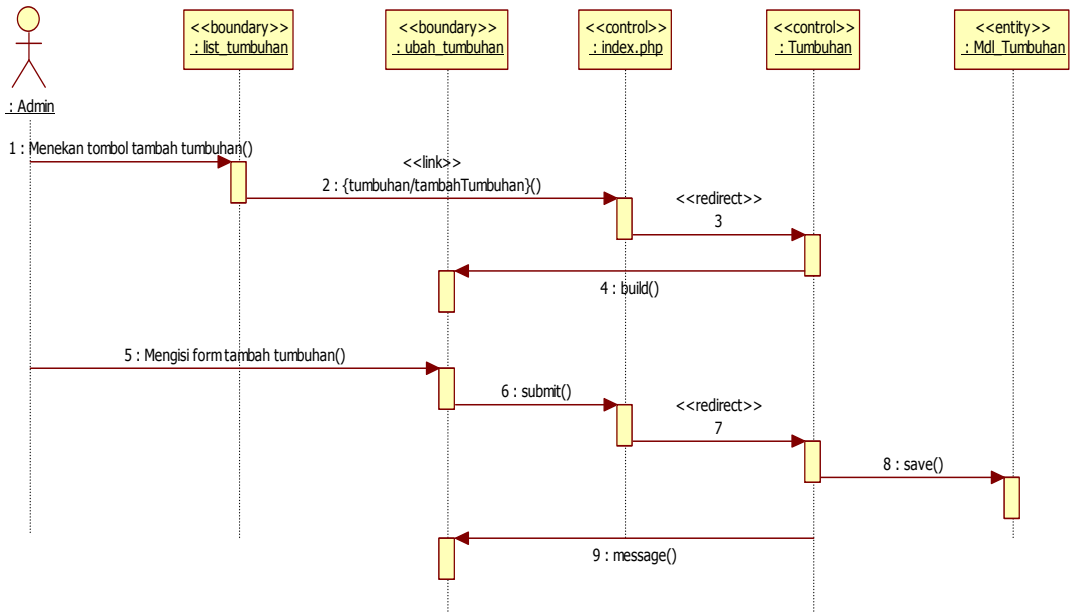
Gambar 3. 97 Sequence Diagram Detail Hewan.



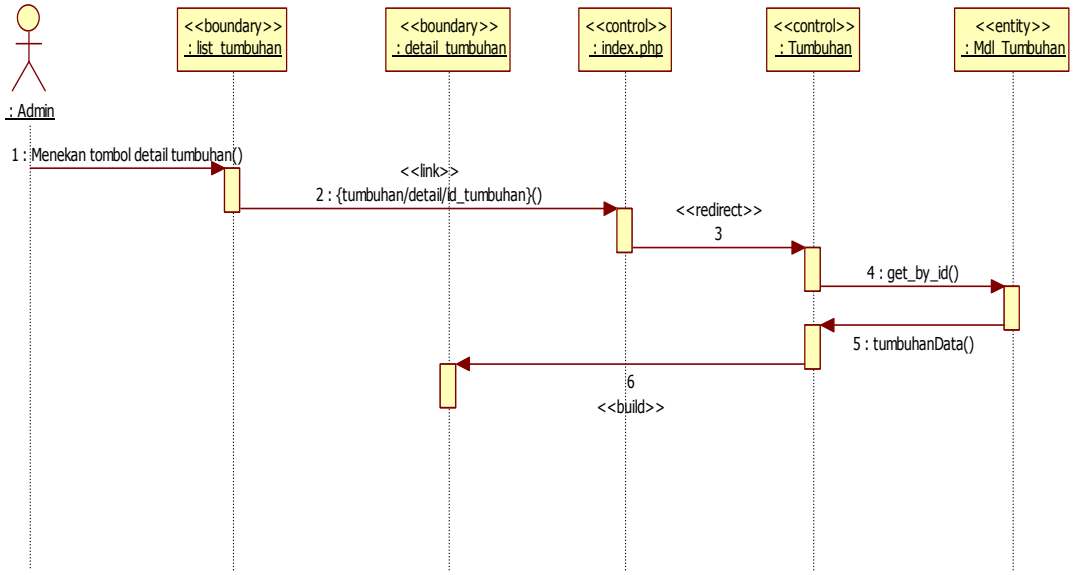
Gambar 3. 98 Sequence Diagram Ubah Hewan.



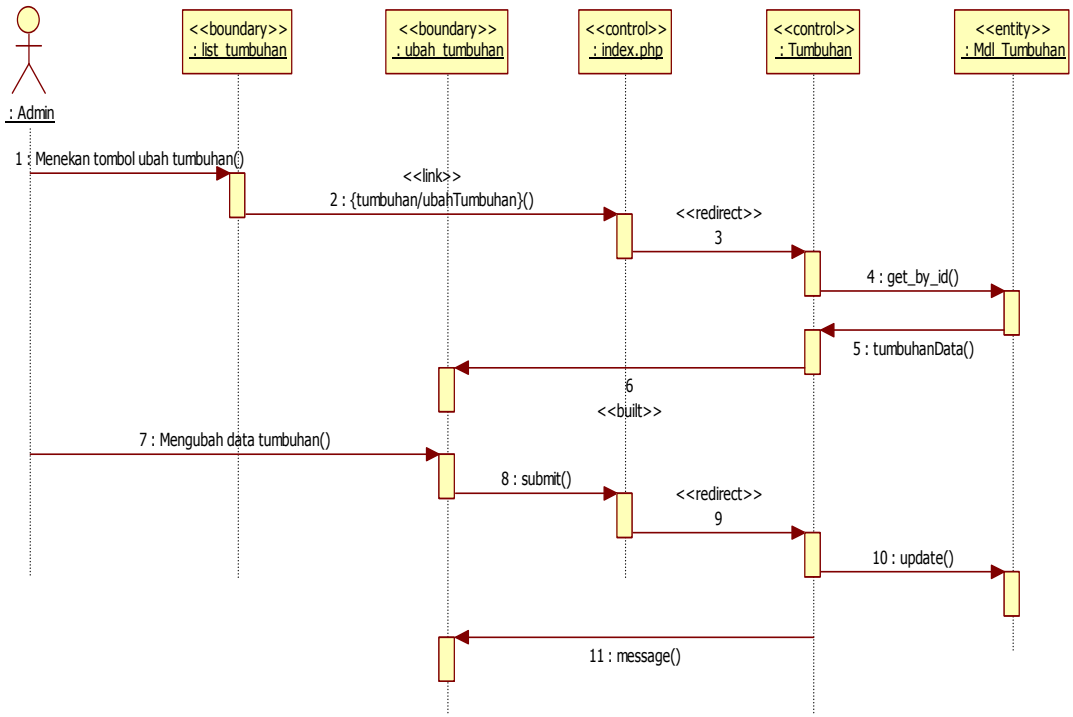
Gambar 3. 99 Sequence Diagram Hapus Hewan.



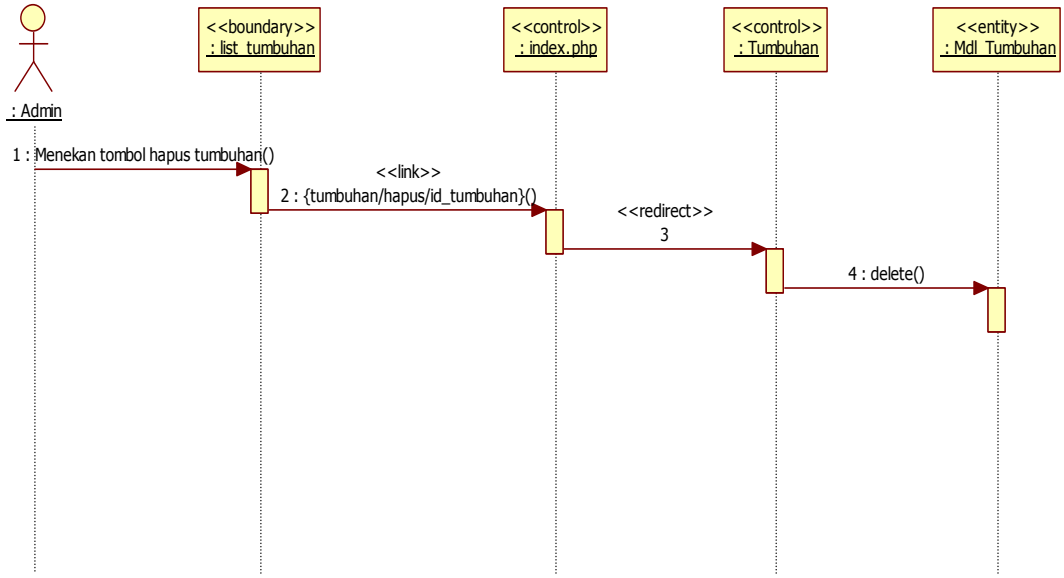
Gambar 3. 100 Sequence Diagram Tambah Tumbuhan.



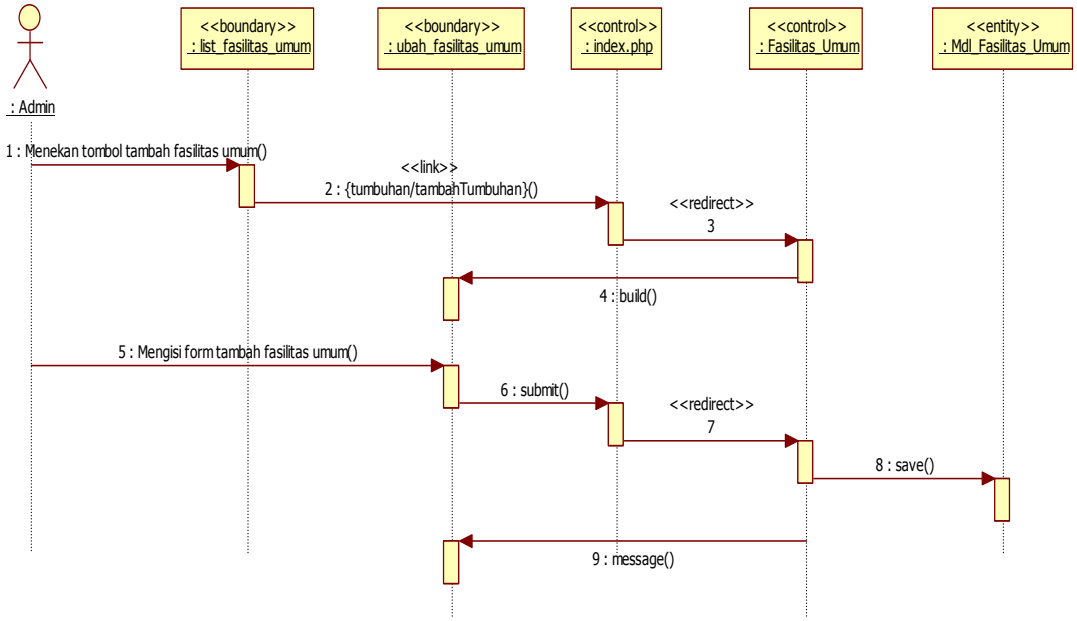
Gambar 3. 101 Sequence Diagram Detail Tumbuhan.



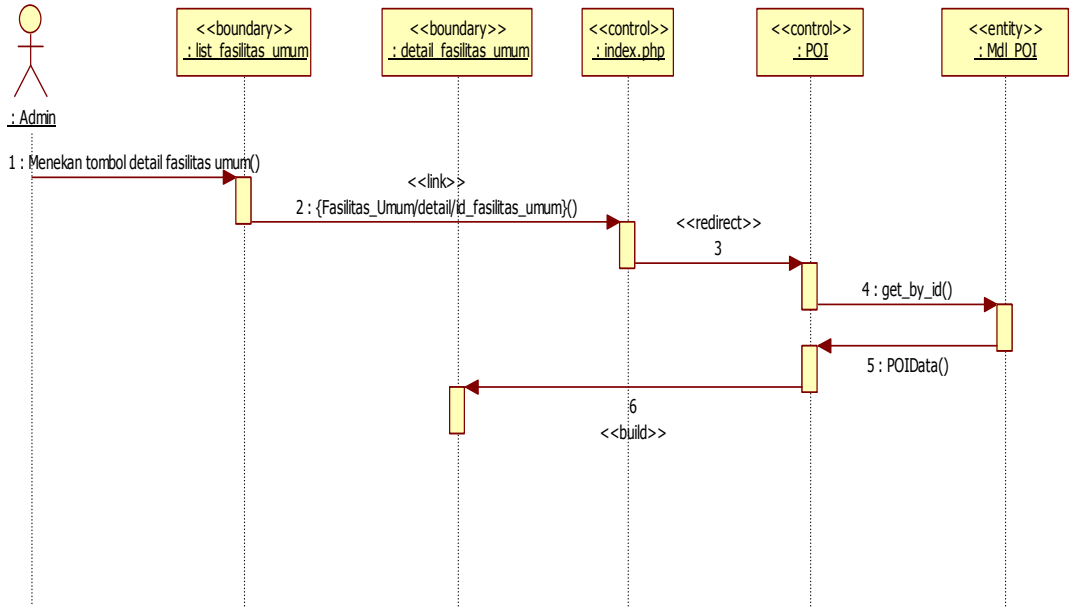
Gambar 3. 102 Sequence Diagram Ubah Tumbuhan.



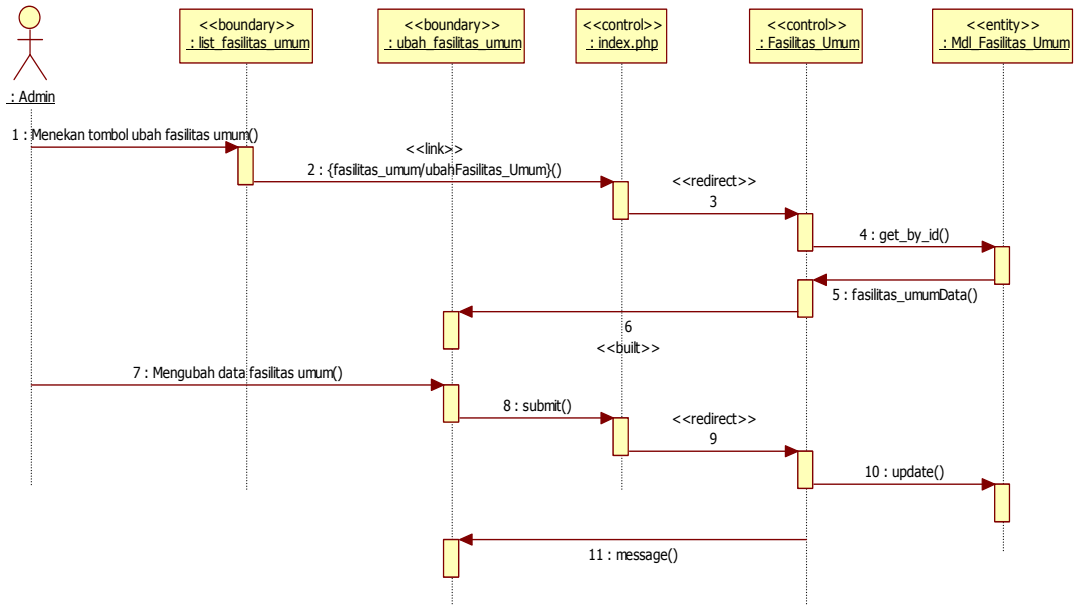
Gambar 3. 103 Sequence Diagram Hapus Tumbuhan.



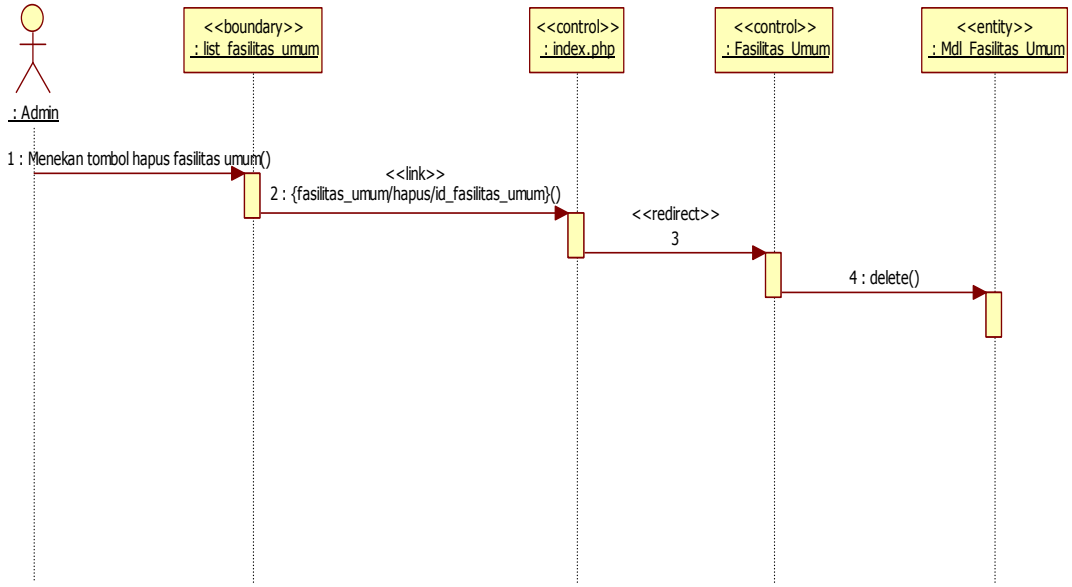
Gambar 3. 104 Sequence Diagram Tambah Fasilitas Umum.



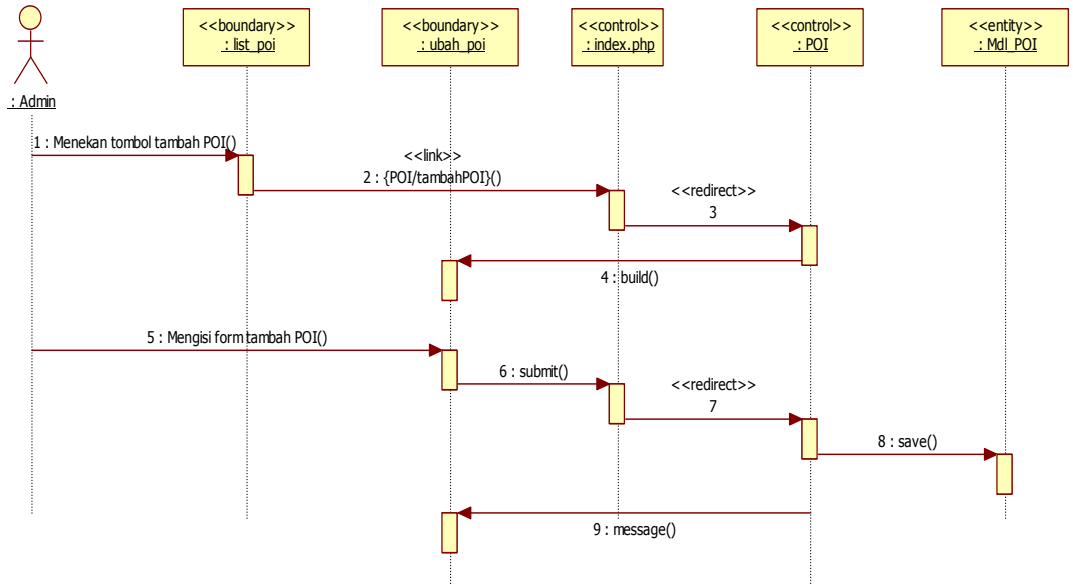
Gambar 3. 105 Sequence Diagram Detail Fasilitas Umum.



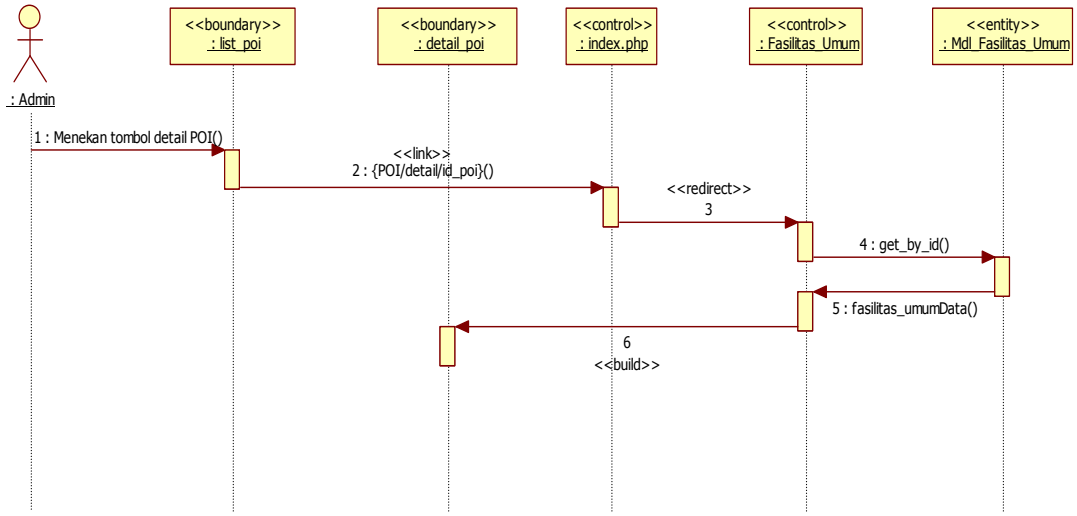
Gambar 3. 106 Sequence Diagram Ubah Fasilitas Umum.



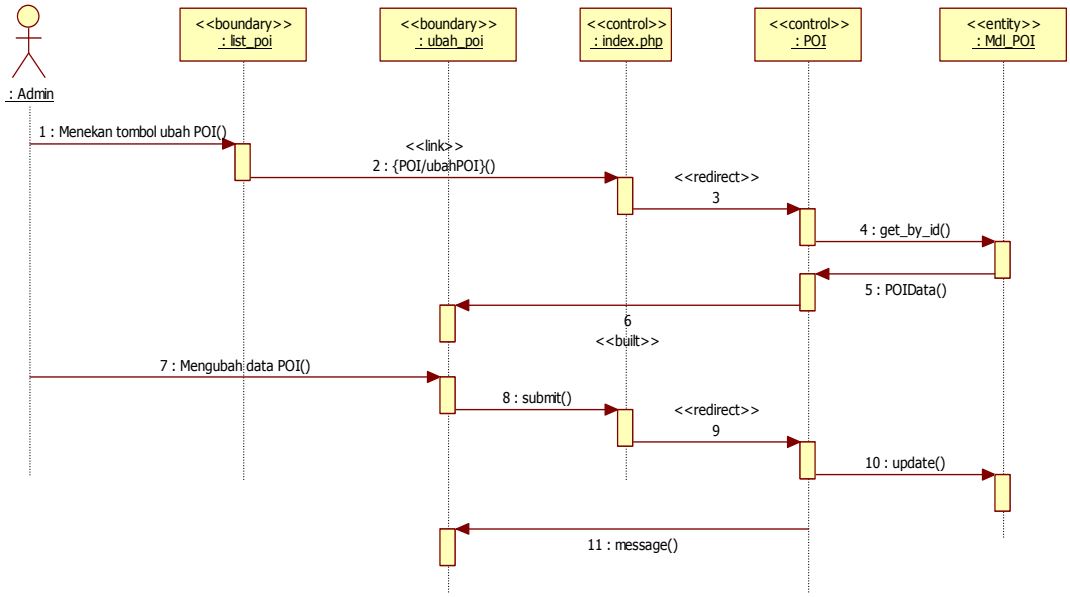
Gambar 3. 107 Sequence Diagram Hapus Fasilitas Umum.



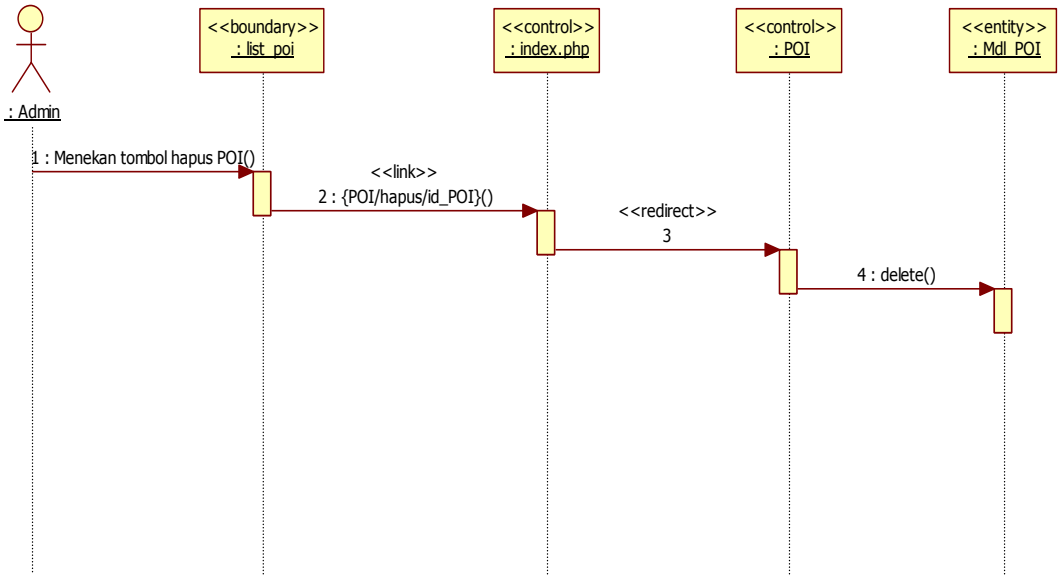
Gambar 3. 108 Sequence Diagram Tambah POI.



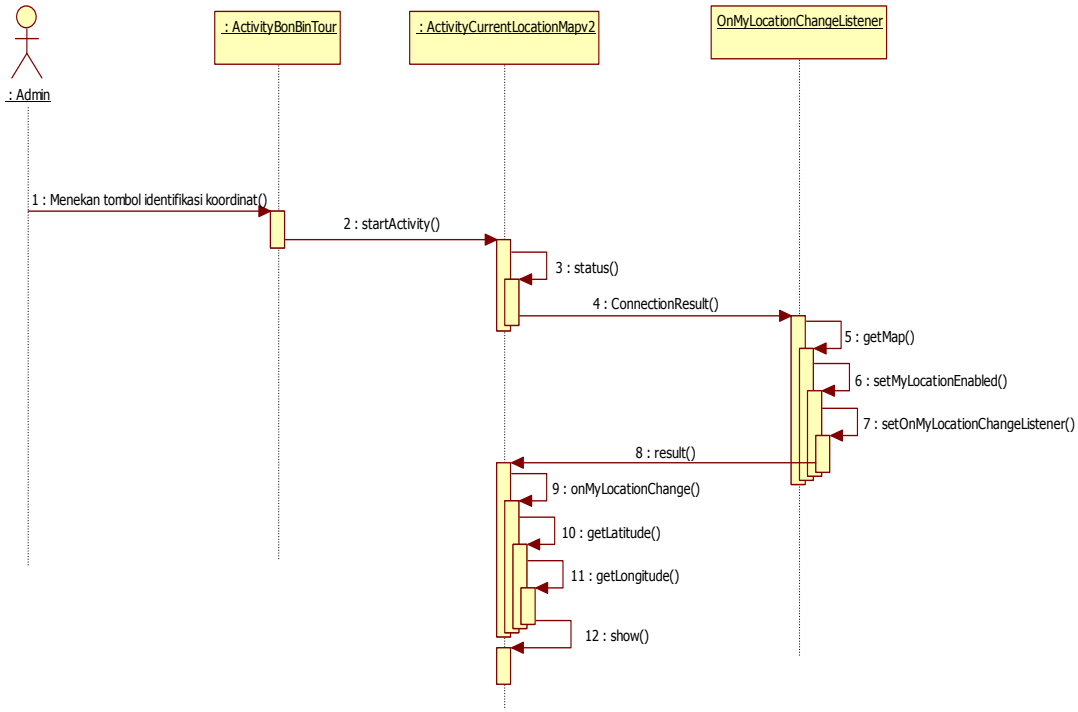
Gambar 3. 109 Sequence Diagram Detail POI.



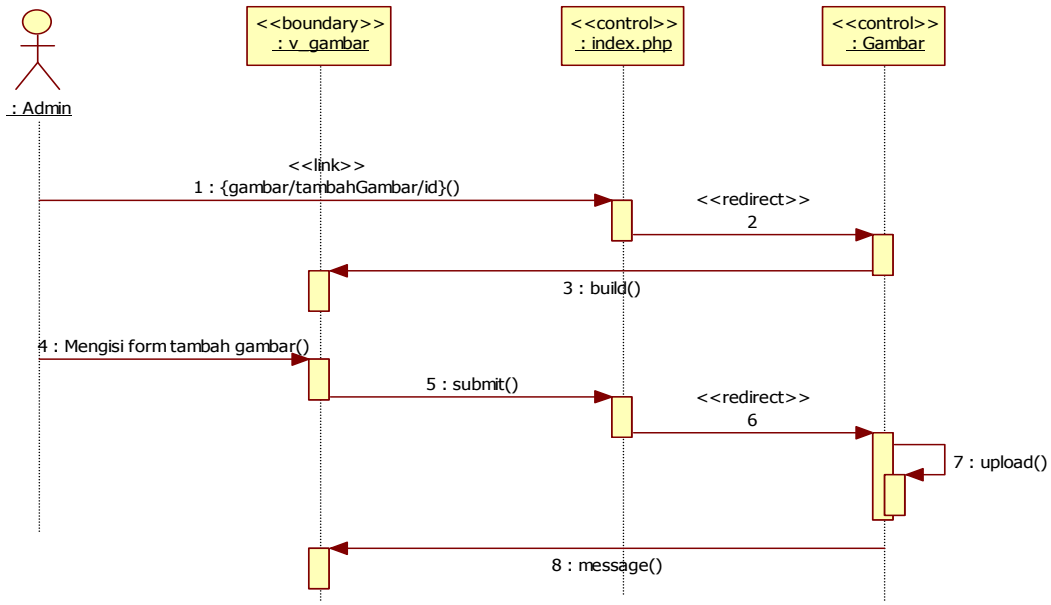
Gambar 3. 110 Sequence Diagram Ubah POI.



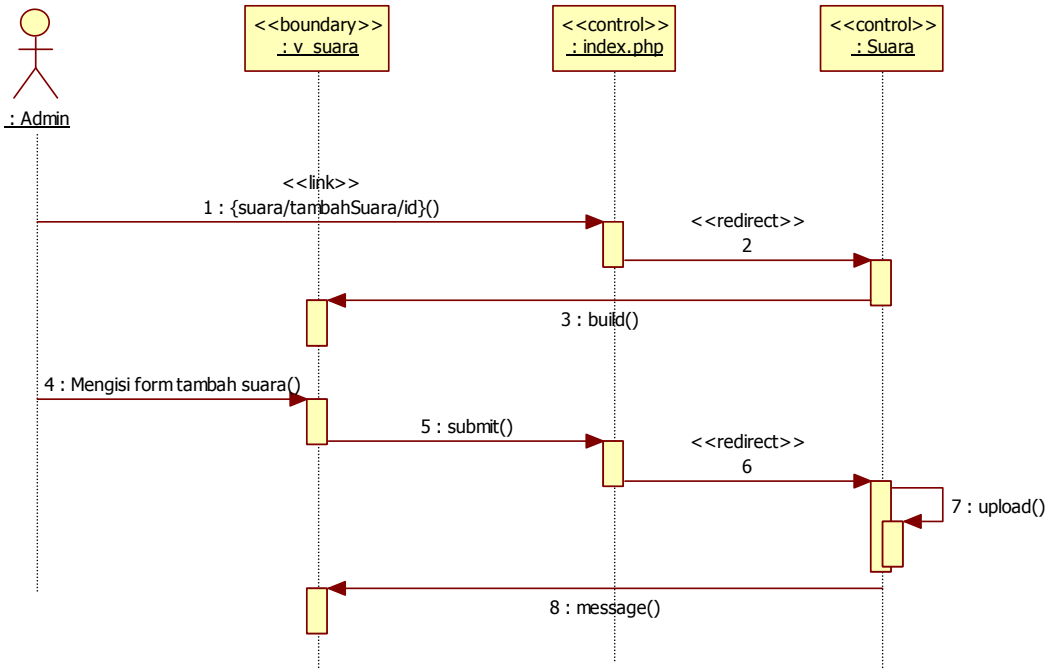
Gambar 3. 111 Sequence Diagram Hapus POI.



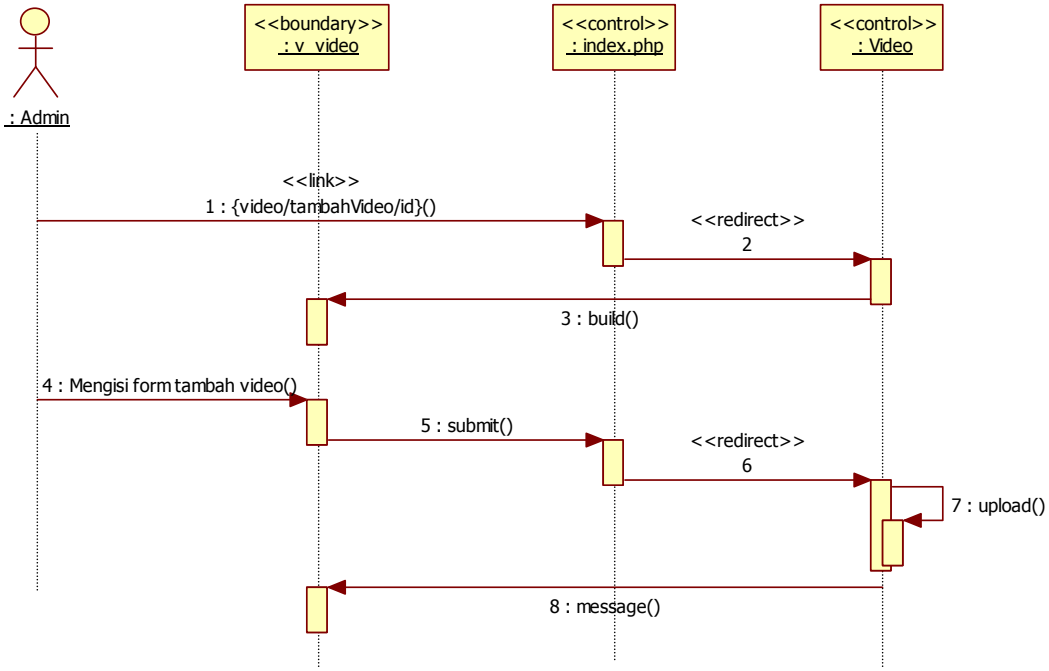
Gambar 3. 112 Sequence Diagram Identifikasi Koordinat.



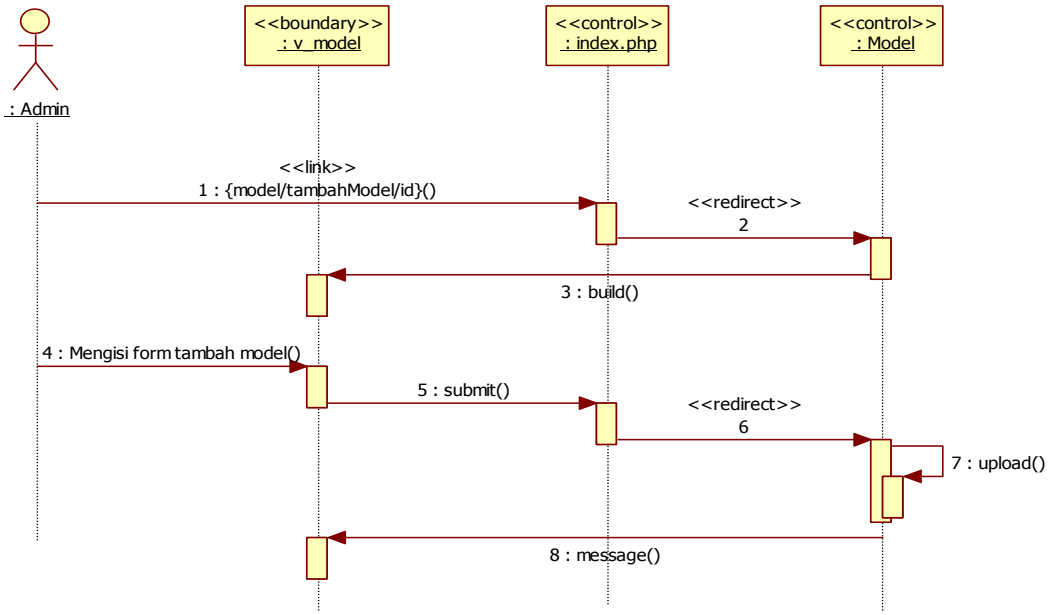
Gambar 3. 113 Sequence Diagram Tambah Gambar.



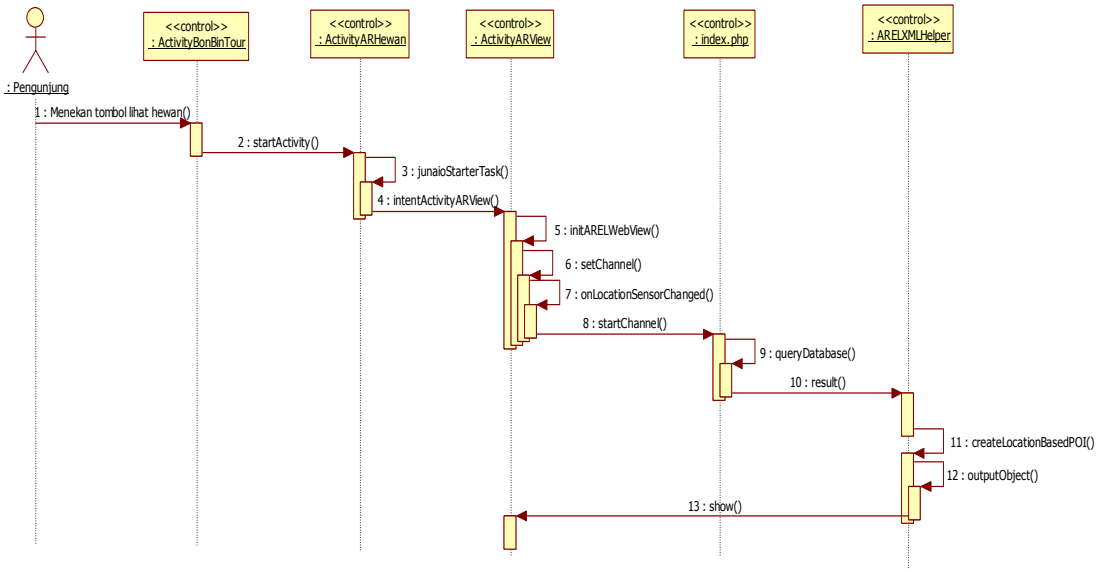
Gambar 3. 114 Sequence Diagram Tambah Suara.



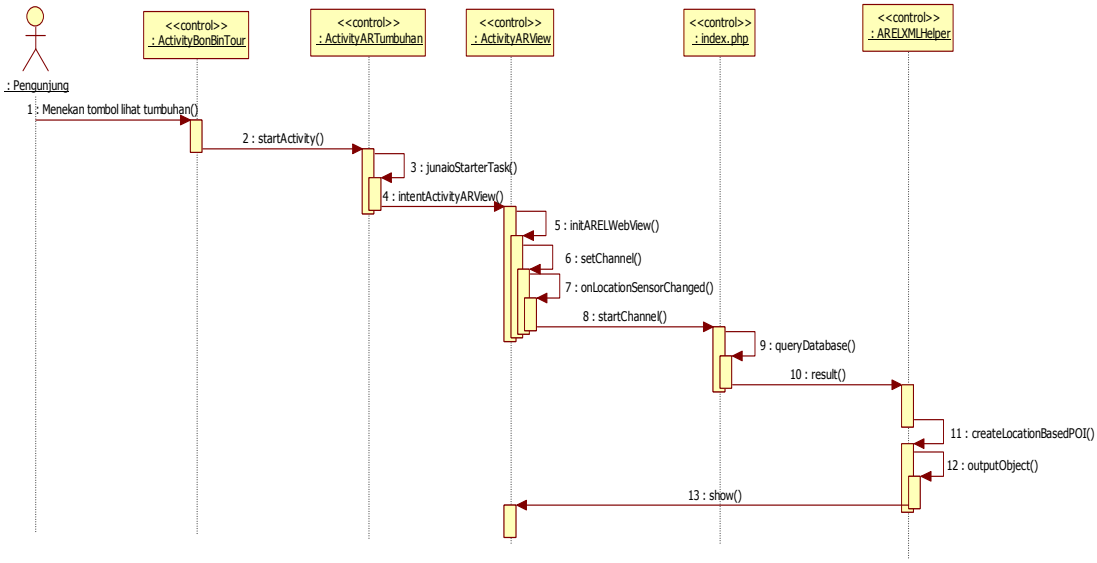
Gambar 3. 115 Sequence Diagram Tambah Video.



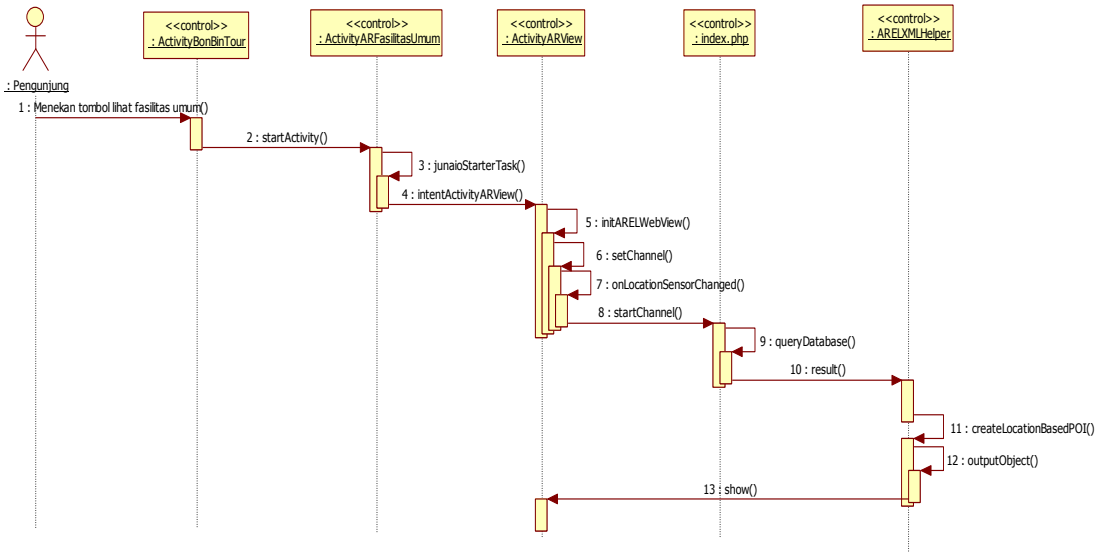
Gambar 3. 116 Sequence Diagram Tambah Model.



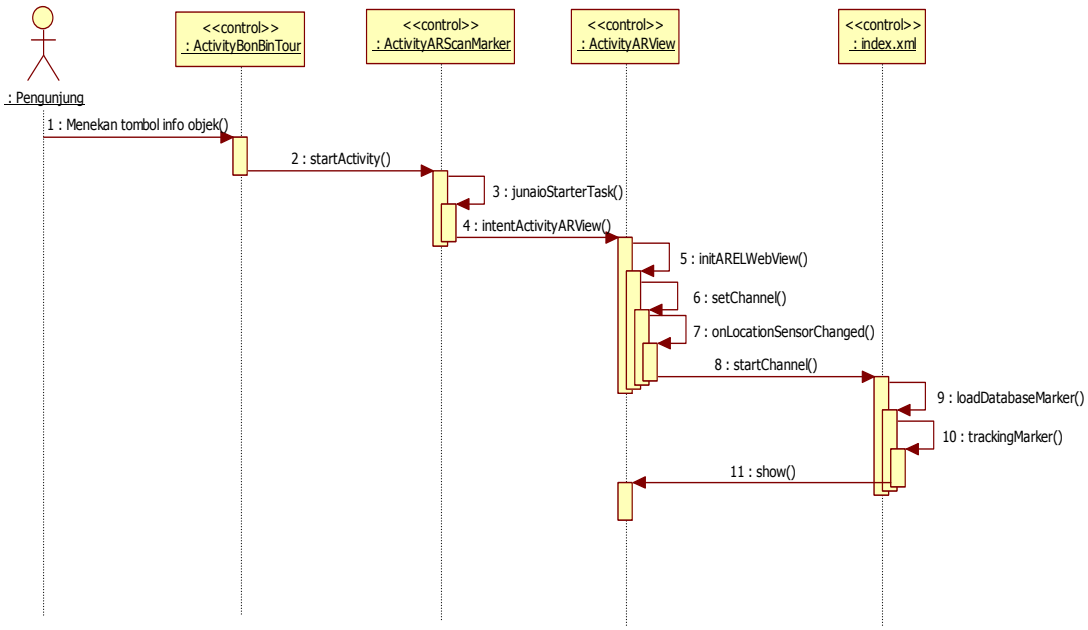
Gambar 3. 117 Sequence Diagram Tampil Hewan Sekitar.



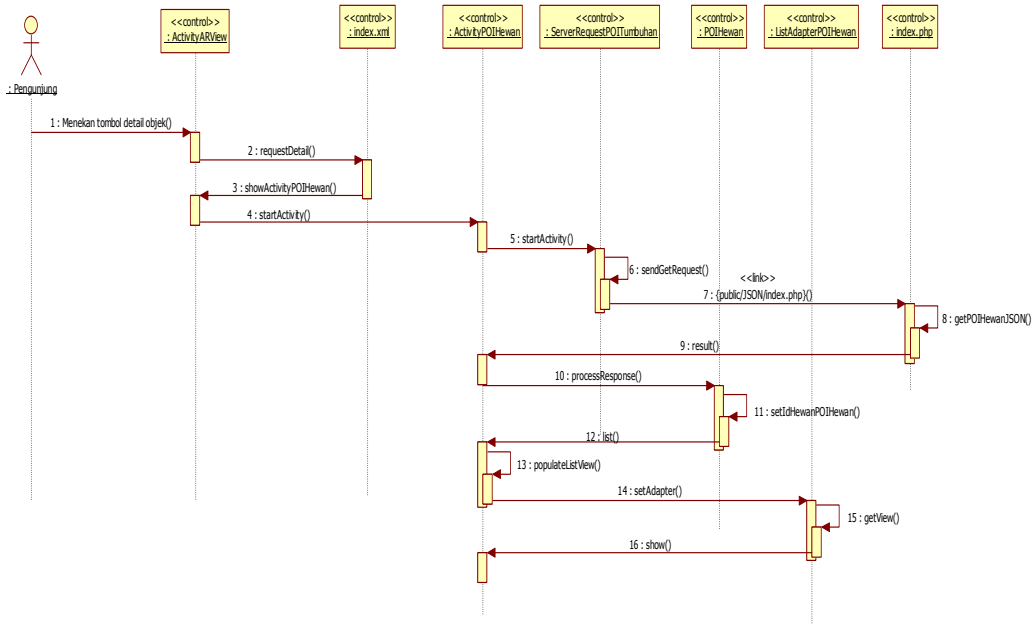
Gambar 3. 118 Sequence Diagram Tampil Tumbuhan Sekitar.



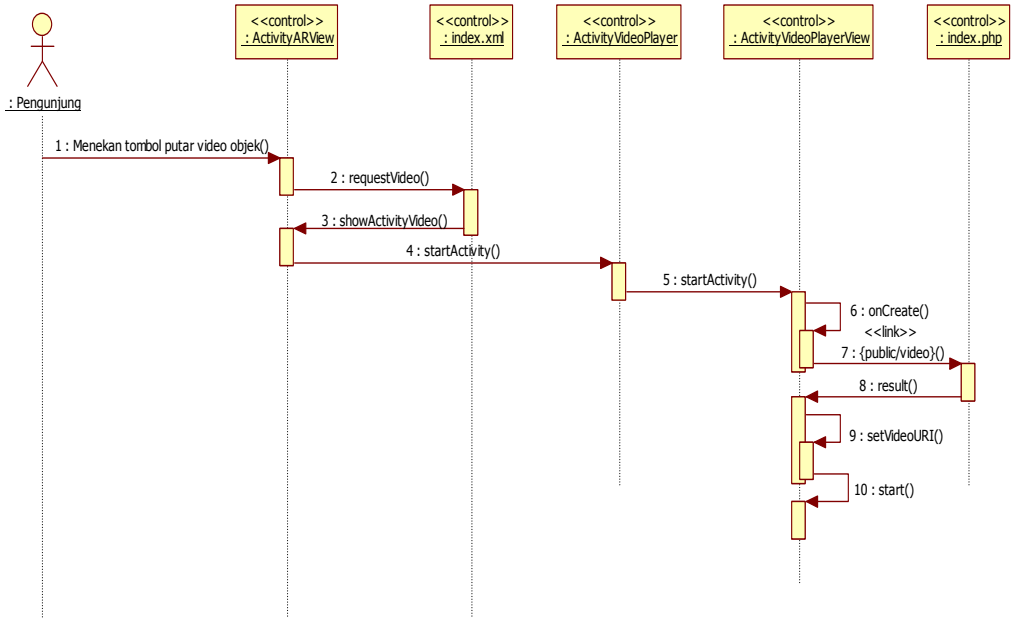
Gambar 3. 119 Sequence Diagram Tampil Fasilitas Umum Sekitar.



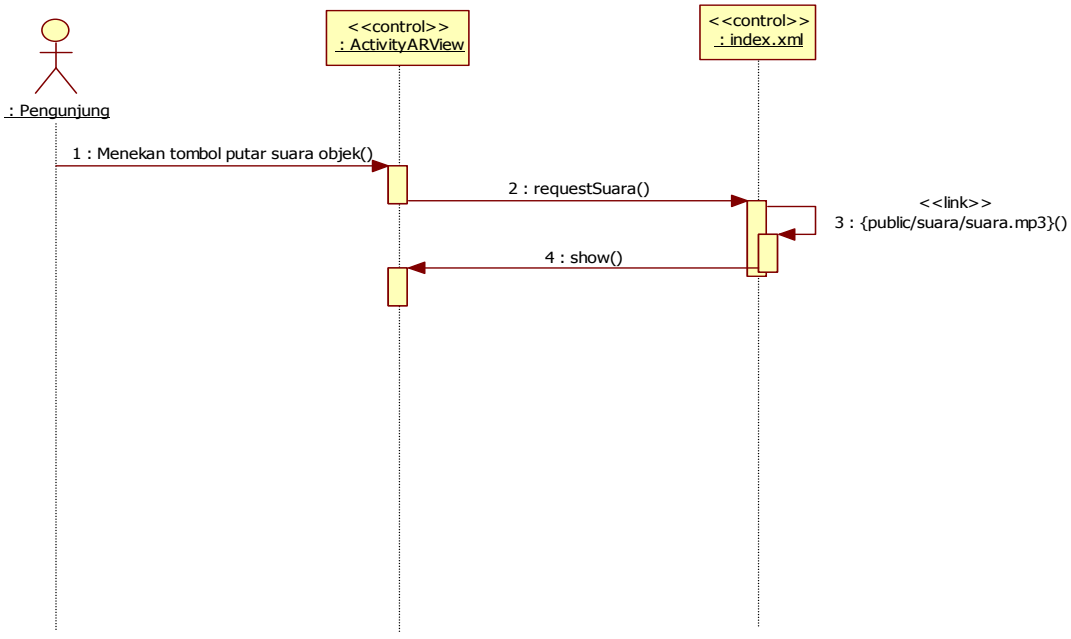
Gambar 3. 120 Sequence Diagram Scan Marker.



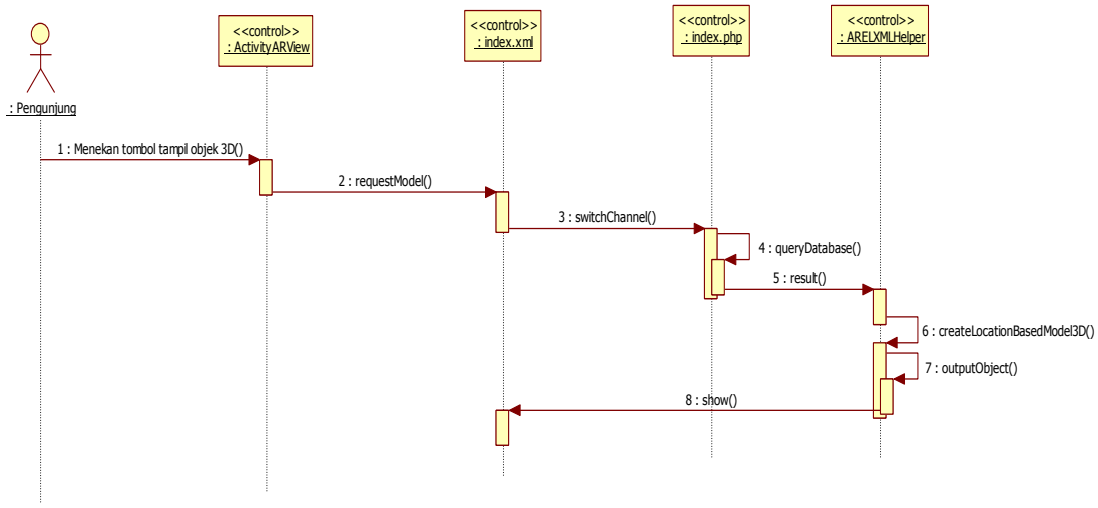
Gambar 3. 121 Sequence Diagram Lihat Detail.



Gambar 3. 122 Sequence Diagram Putar Video.



Gambar 3. 123 Sequence Diagram Putar Suara.

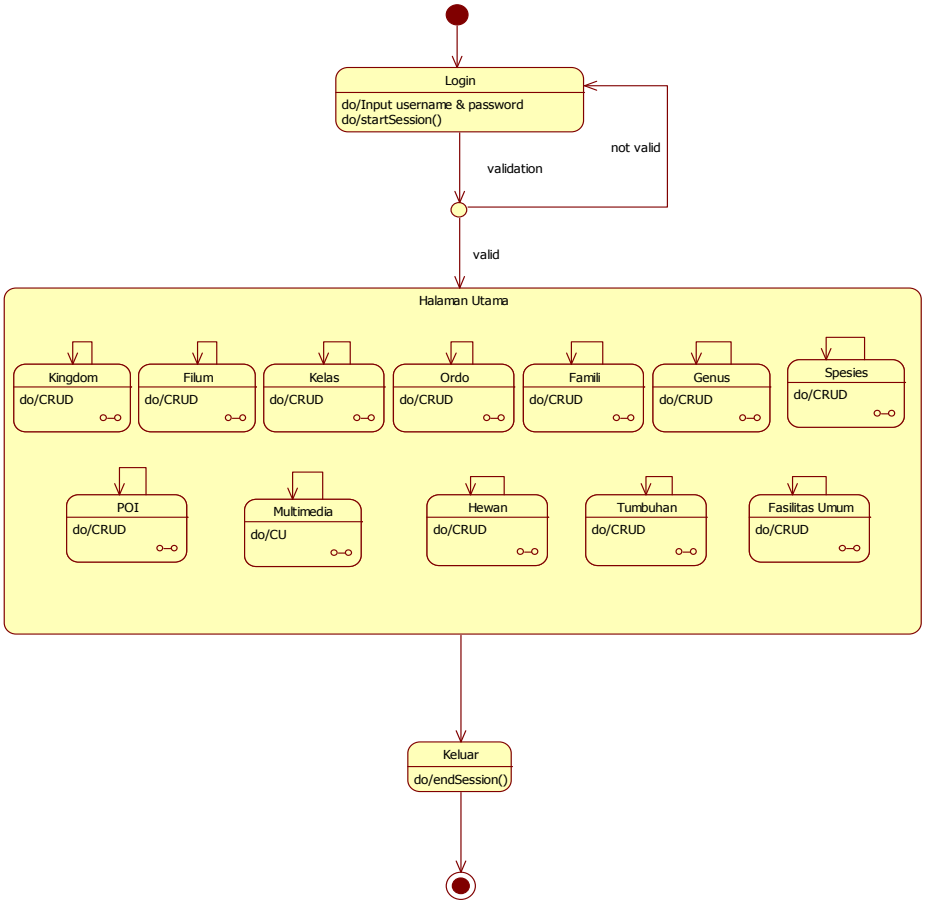


Gambar 3. 124 Sequence Diagram Tampil Model.

E. Statechart Diagram

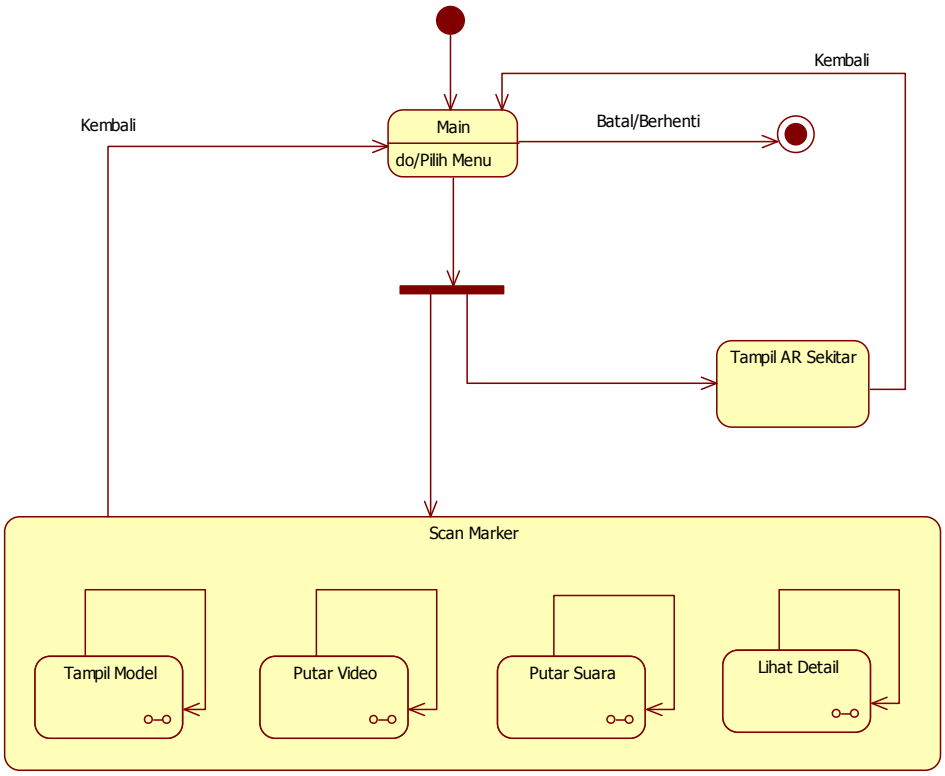
Analisis *statechart diagram* terbagi menjadi 2 bagian yaitu *statechart diagram backend* dan *statechart diagram frontend*, berikut penjelasannya :

a) Statechart Diagram Backend



Gambar 3. 125 Statechart Diagram Backend.

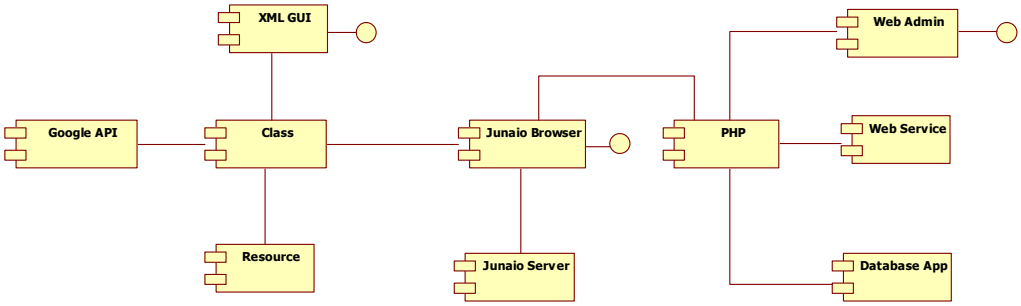
b) Statechart Diagram Frontend



Gambar 3. 126 Statechart Diagram Frontend.

F. Component Diagram

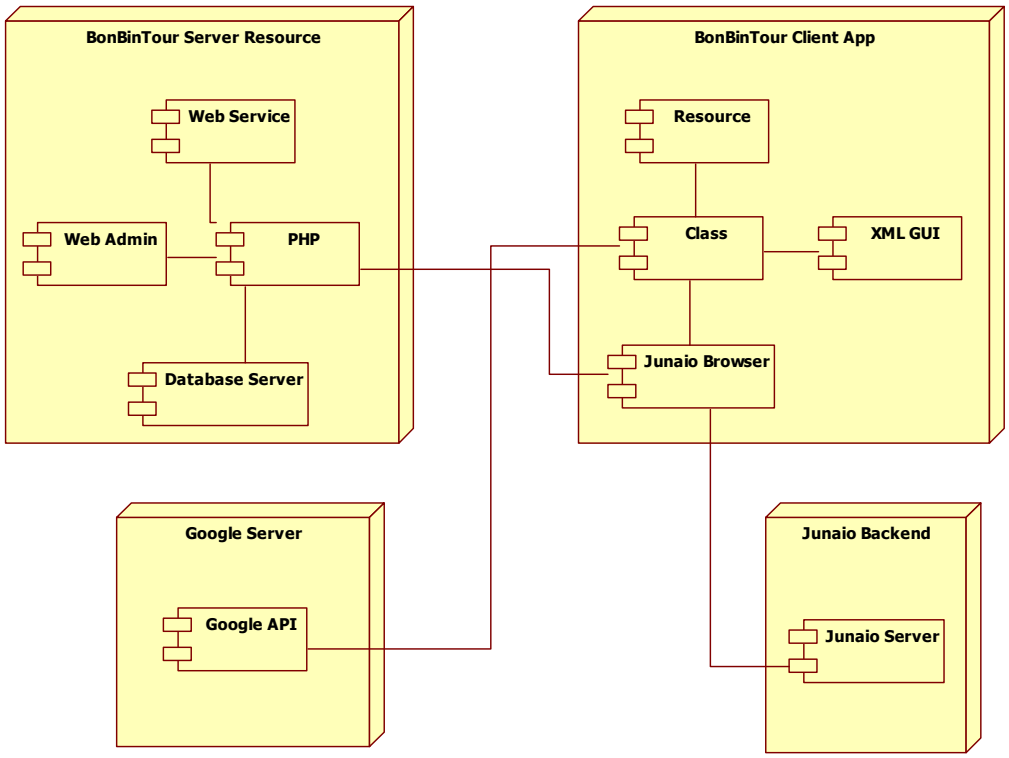
Berikut analisis *component diagram* pada sistem yang dibangun :



Gambar 3. 127 Component Diagram Sistem.

G. Deployment Diagram

Berikut analisis *deployment diagram pada sistem yang dibangun :*



Gambar 3. 128 Deployment Diagram Sistem.

3.2. Perancangan Sistem

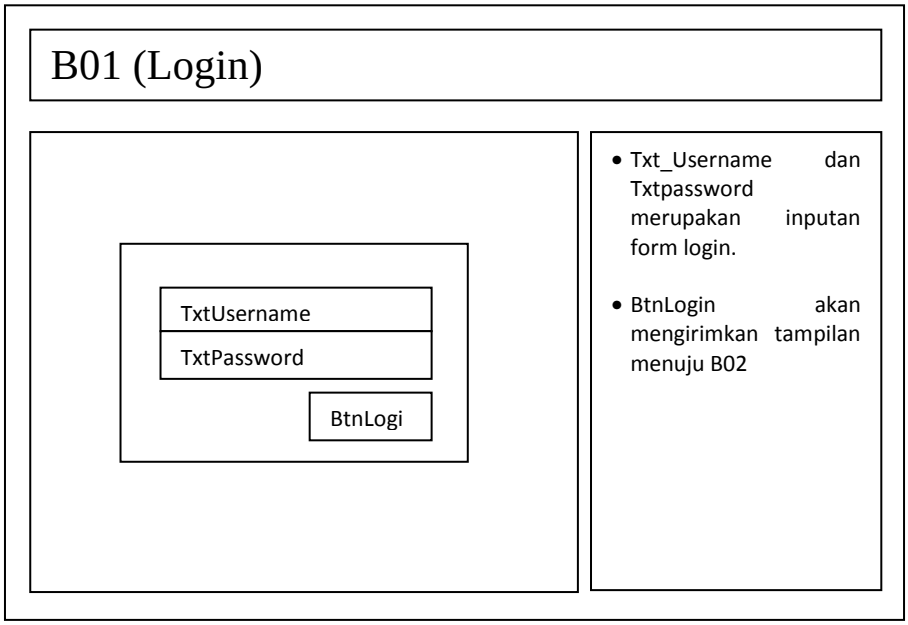
Perancangan sistem adalah suatu proses yang menggambarkan bagaimana suatu sistem dibangun untuk memenuhi kebutuhan pada fase analisis. Adapun tahapan yang dilakukan dalam perancangan sistem ini membahas mengenai perancangan antarmuka dan jaringan semantik.

3.2.1. Perancangan Antarmuka

1. Antarmuka *Backend*

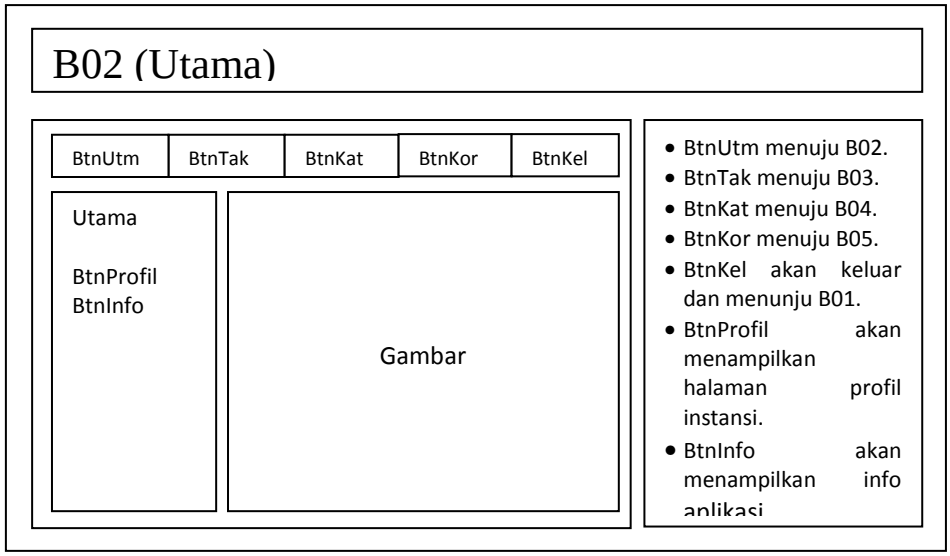
Agar dapat memudahkan proses pembuatan aplikasi maka terlebih dahulu membuat rancangan *design* tampilannya. Pada perancangan antar muka *backend application* ini meliputi :

1. Antarmuka *Login*



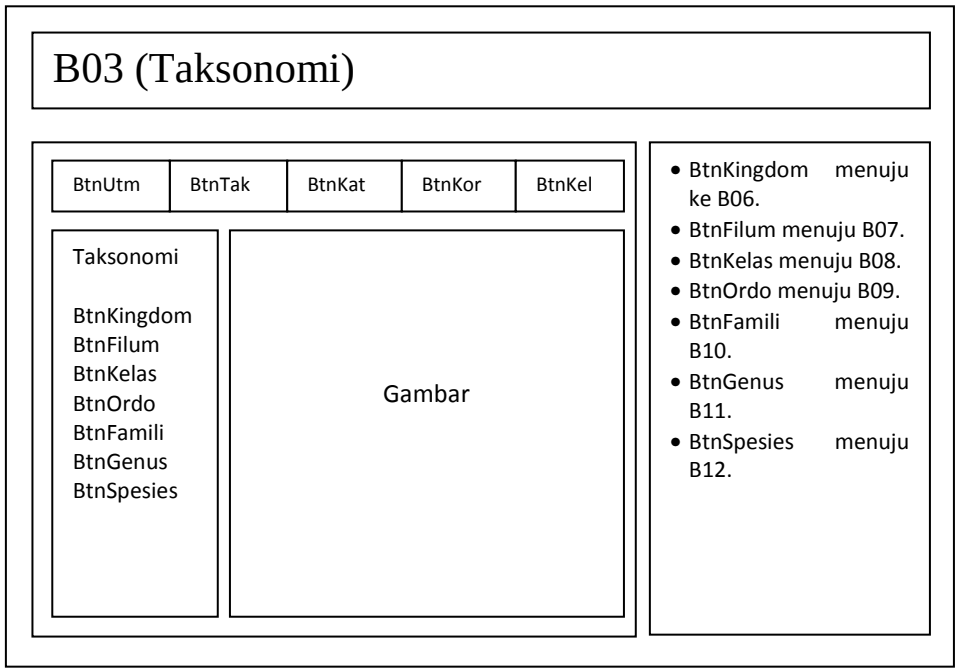
Gambar 3. 129 Antarmuka Menu Login.

2. Antarmuka Halaman Utama



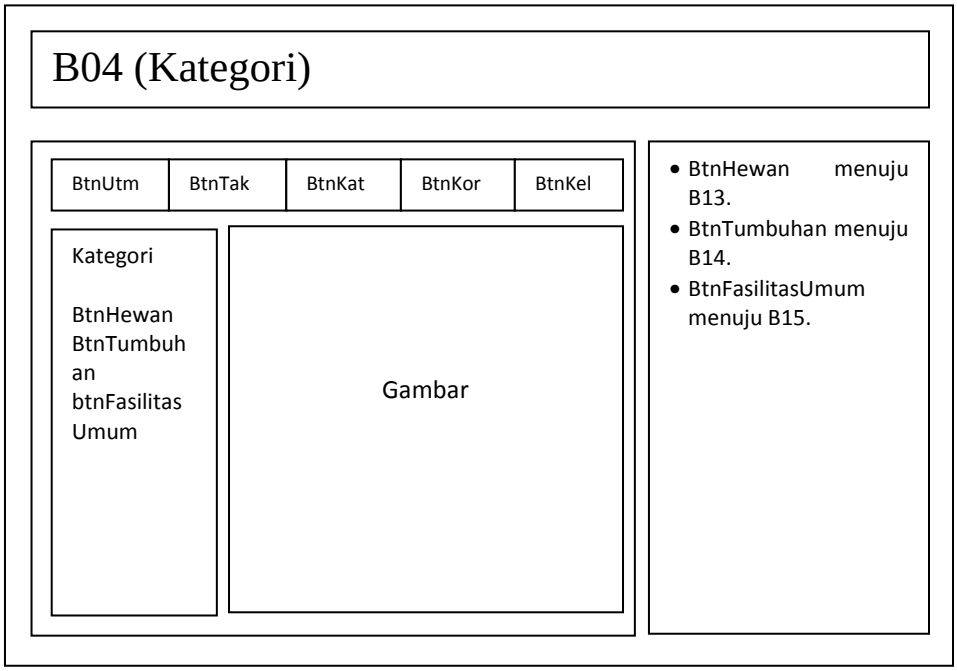
Gambar 3. 130 Antarmuka Halaman Utama.

3. Antarmuka Halaman Taksonomi



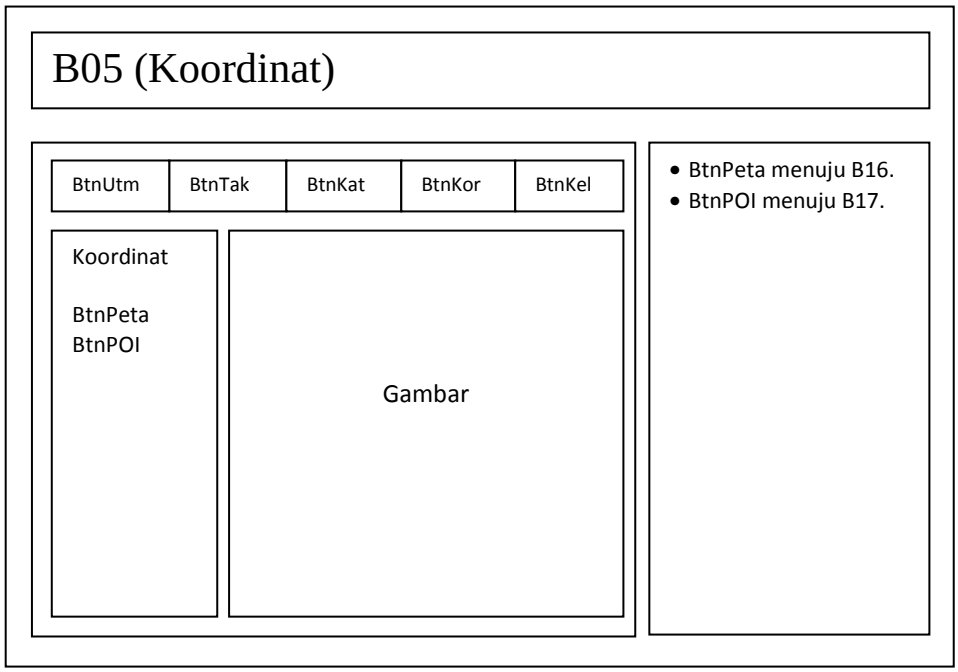
Gambar 3. 131 Antarmuka Halaman Taksonomi.

4. Antarmuka Halaman Kategori



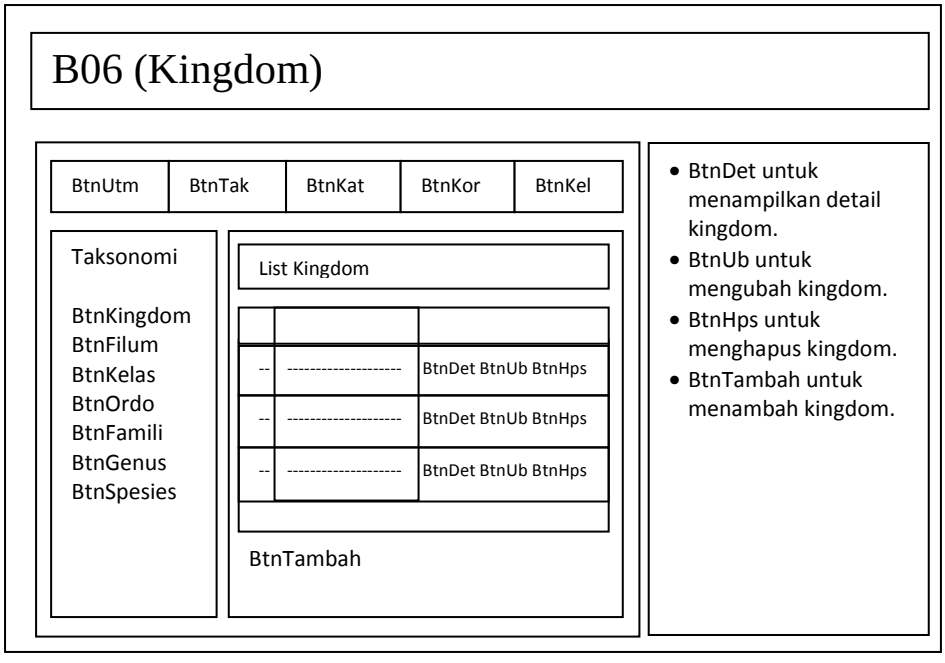
Gambar 3. 132 Antarmuka Kategori.

5. Antarmuka Halaman Koordinat



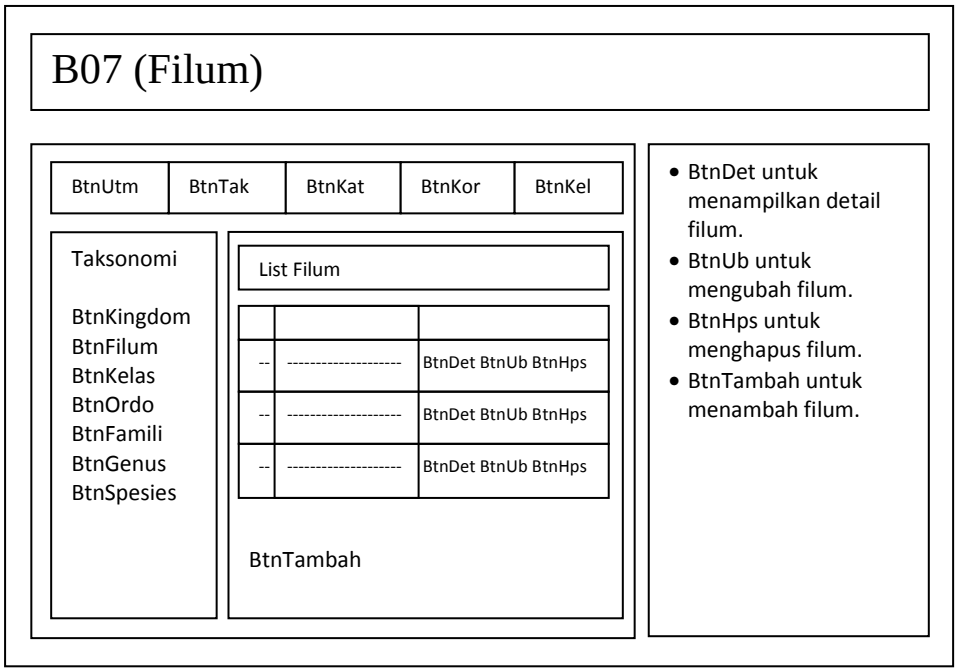
Gambar 3. 133 Antarmuka Koordinat.

6. Antarmuka Halaman Kingdom.



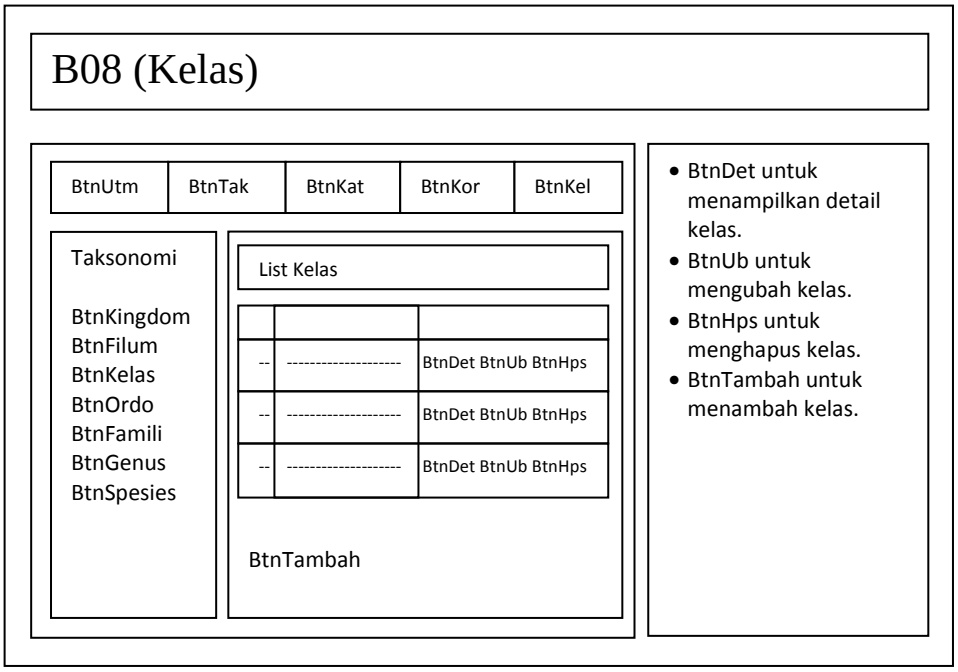
Gambar 3. 134 Antarmuka Kingdom.

7. Antarmuka Halaman Filum.



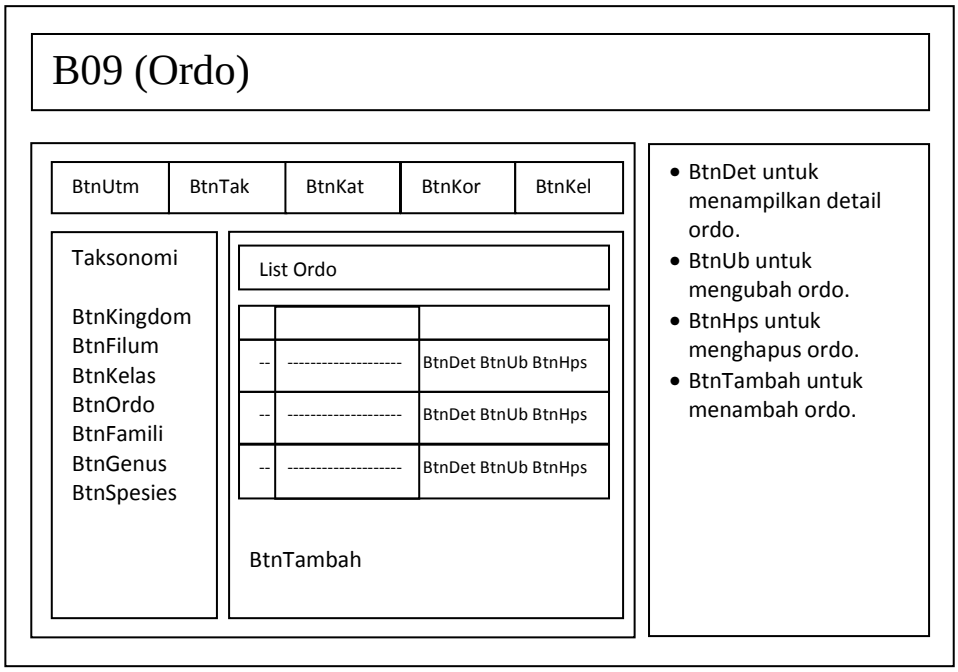
Gambar 3. 135 Antarmuka Filum.

8. Antarmuka Halaman Kelas.



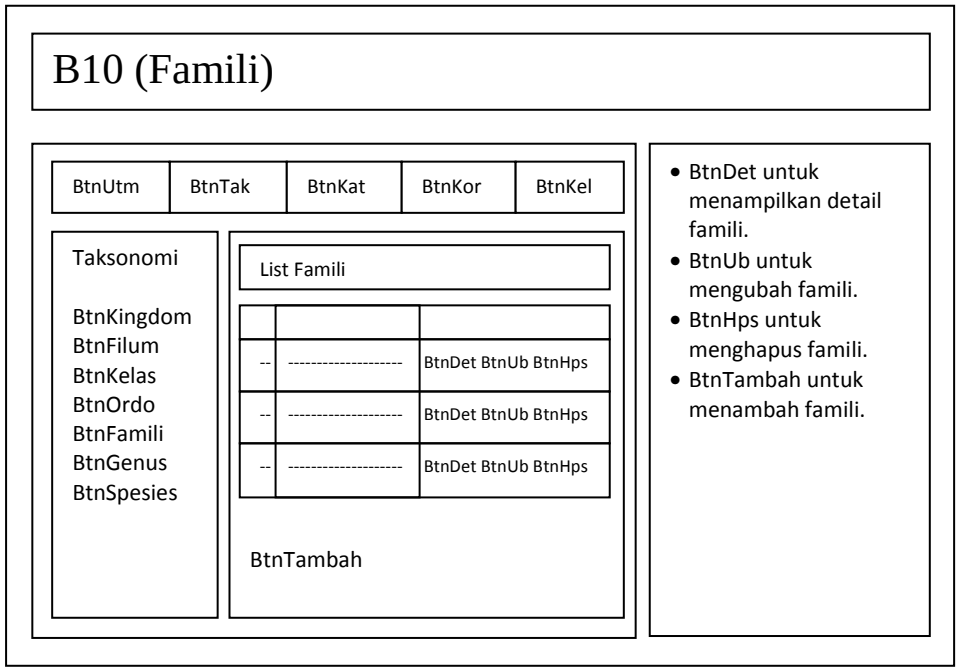
Gambar 3. 136 Antarmuka Kelas.

9. Antarmuka Halaman Ordo.



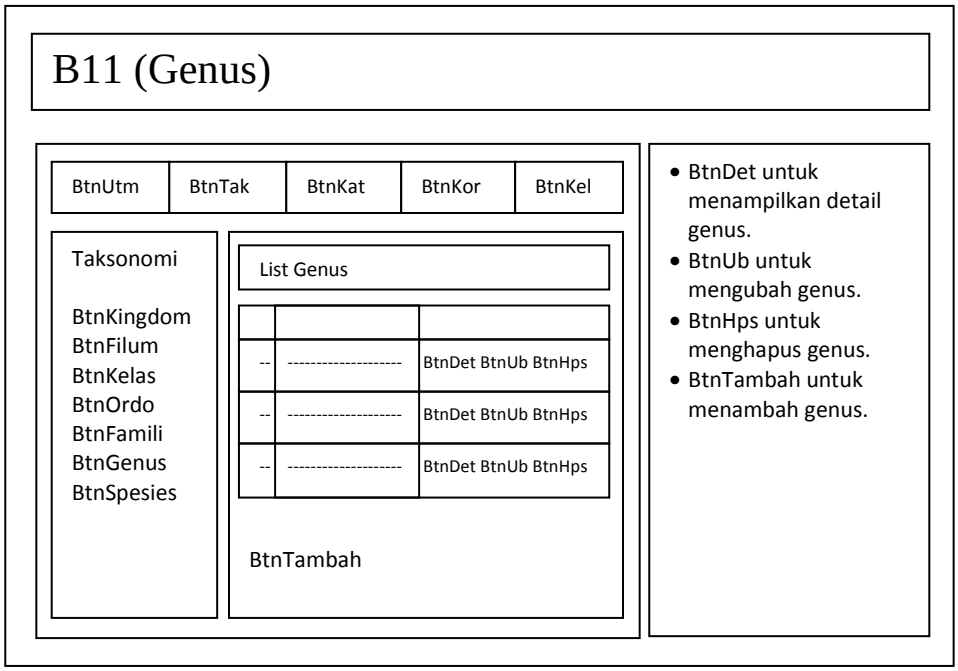
Gambar 3. 137 Antarmuka Ordo.

10. Antarmuka Halaman Famili.



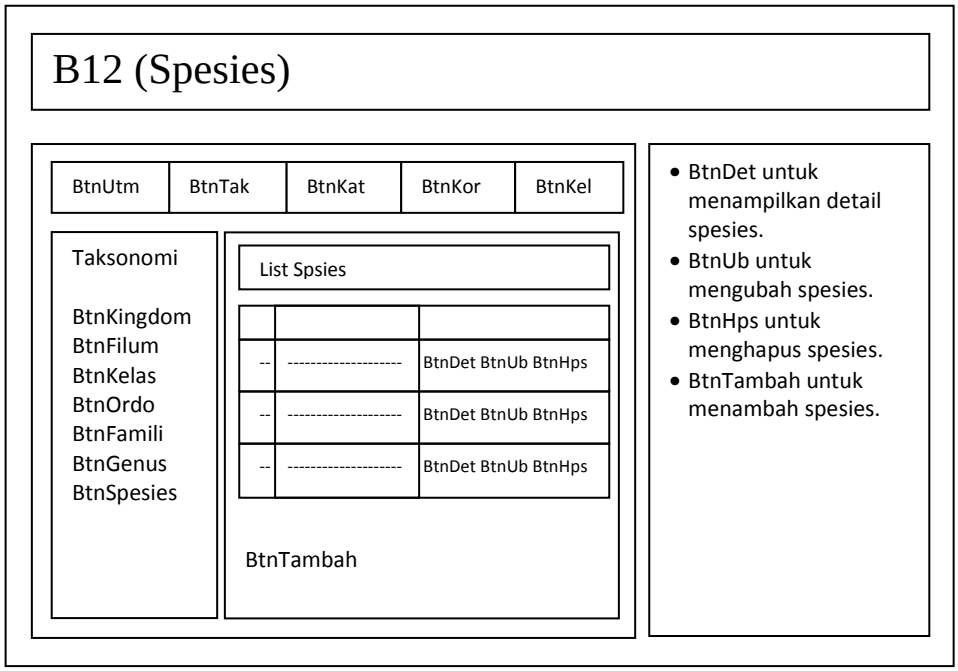
Gambar 3. 138 Antarmuka Famili.

11. Antarmuka Halaman Genus.



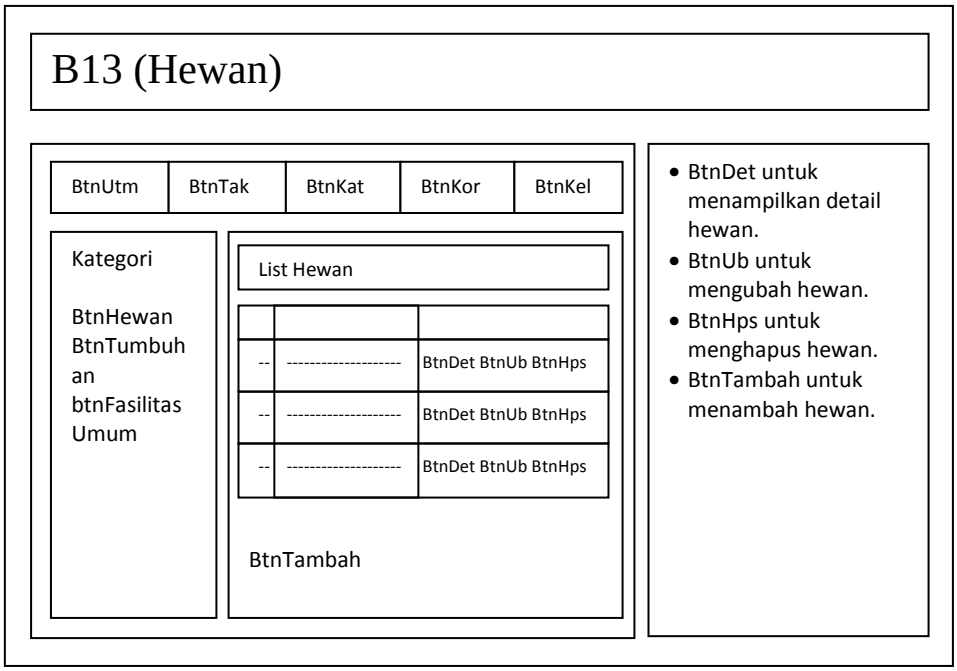
Gambar 3. 139 Antarmuka Genus.

12. Antarmuka Halaman Spesies.



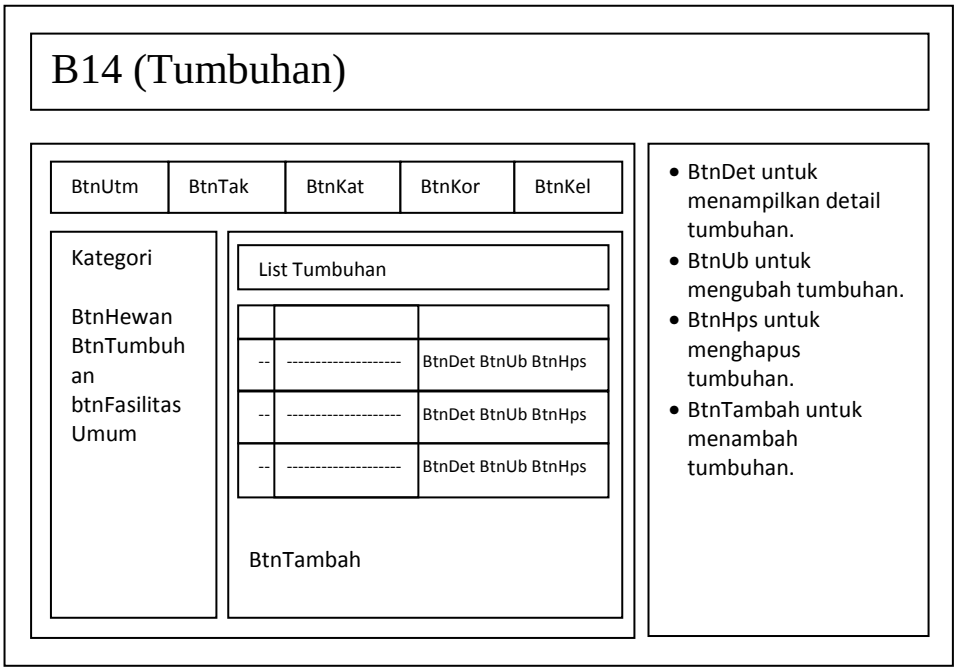
Gambar 3. 140 Antarmuka Spesies.

13. Antarmuka Halaman Hewan.



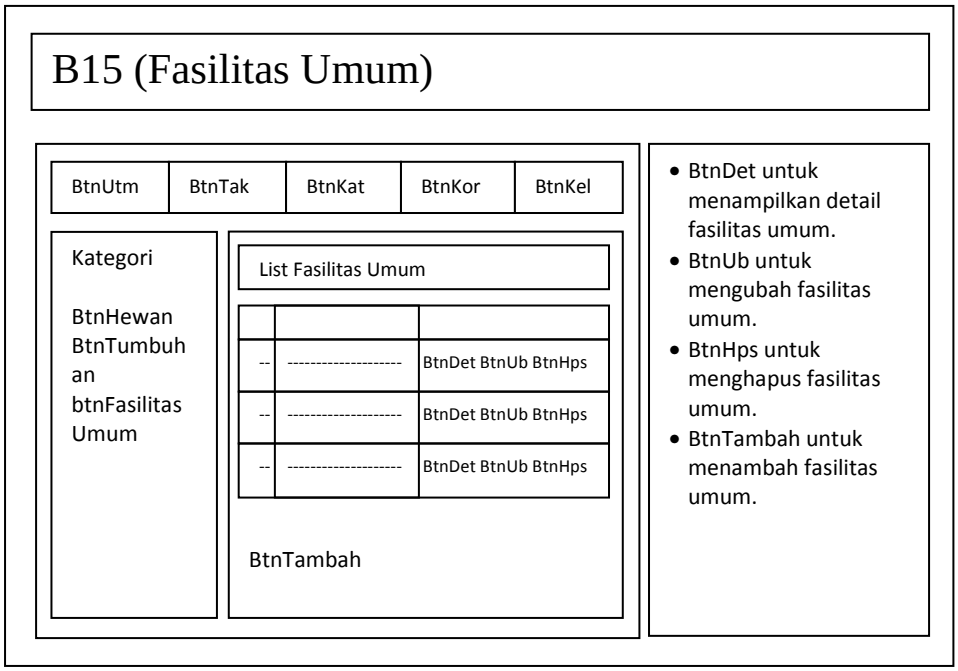
Gambar 3. 141 Antarmuka Hewan.

14. Antarmuka Halaman Tumbuhan.



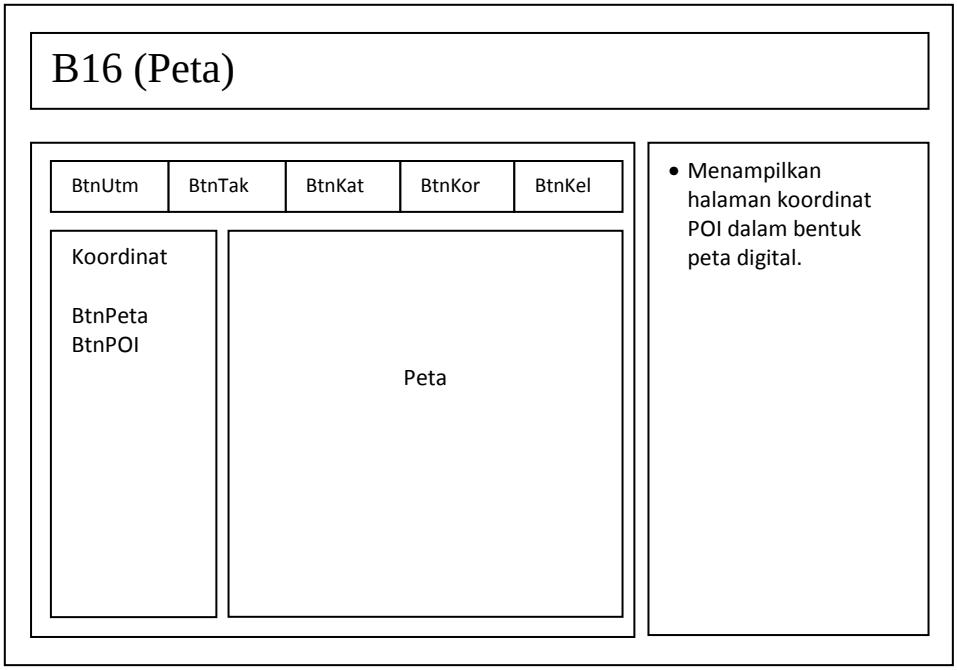
Gambar 3. 142 Antarmuka Tumbuhan.

15. Antarmuka Halaman Fasilitas Umum.



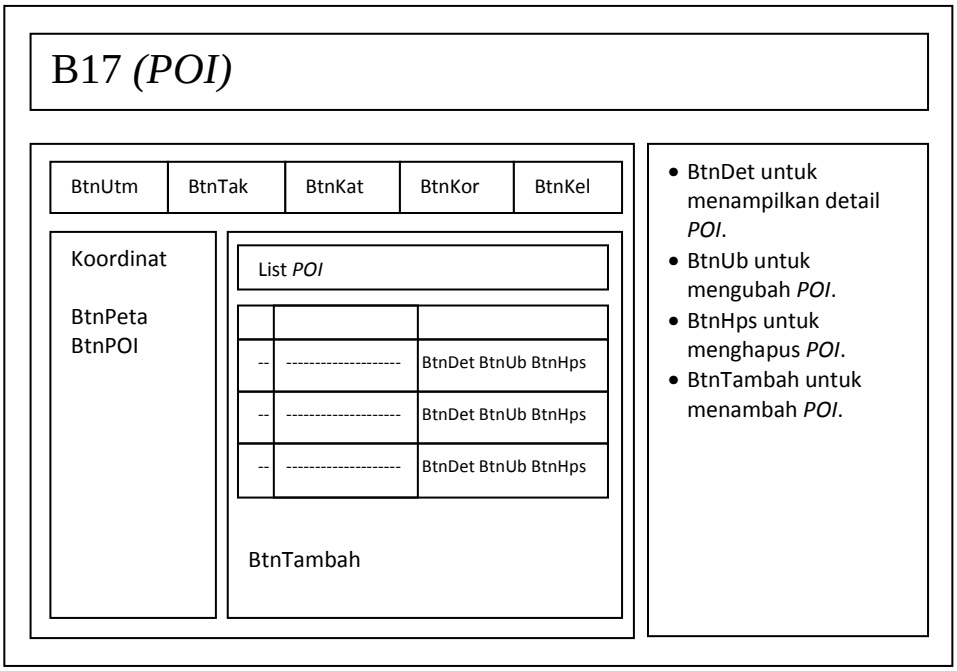
Gambar 3. 143 Antarmuka Fasilitas Umum.

16. Antarmuka Halaman Peta.



Gambar 3. 144 Antarmuka POI.

17. Antarmuka Halaman *POI*.

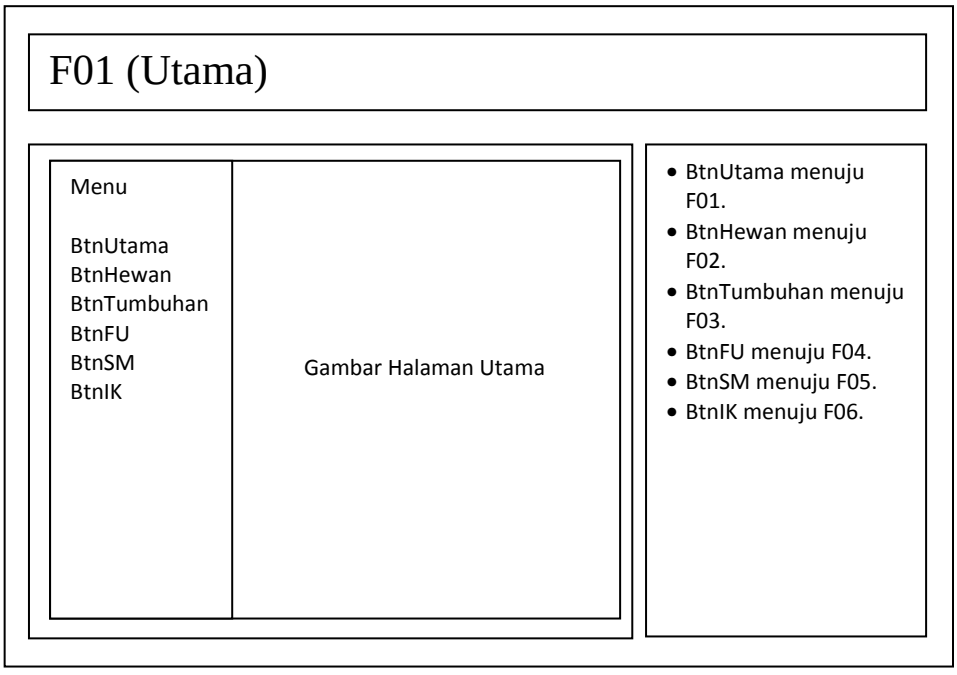


Gambar 3. 145 Antarmuka POI.

2. Antarmuka *Frontend*

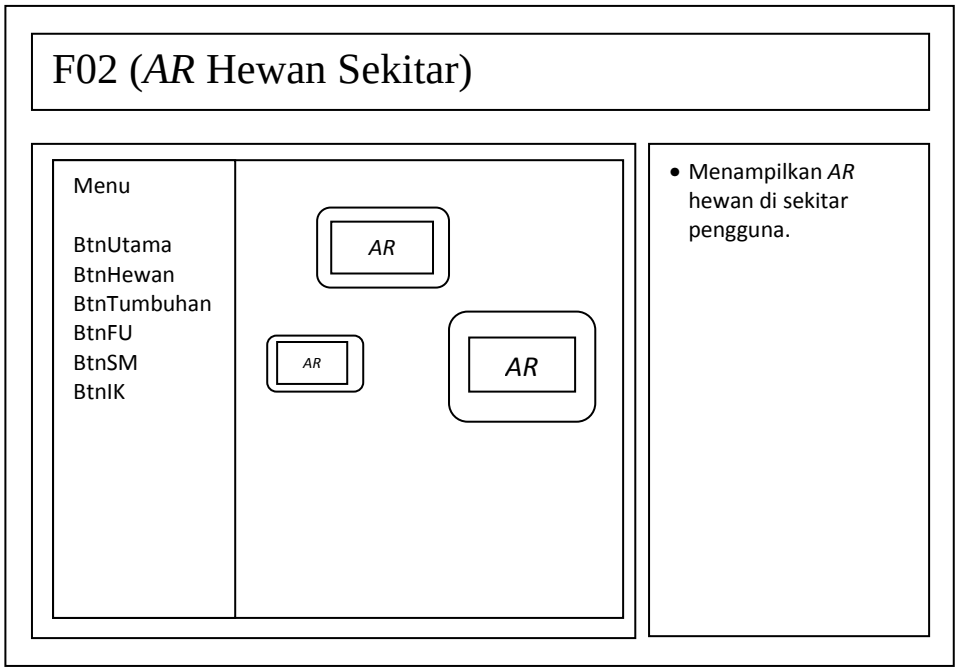
Berikut perancangan antarmuka aplikasi pada sisi *frontend :*

1. Antarmuka Halaman Utama.



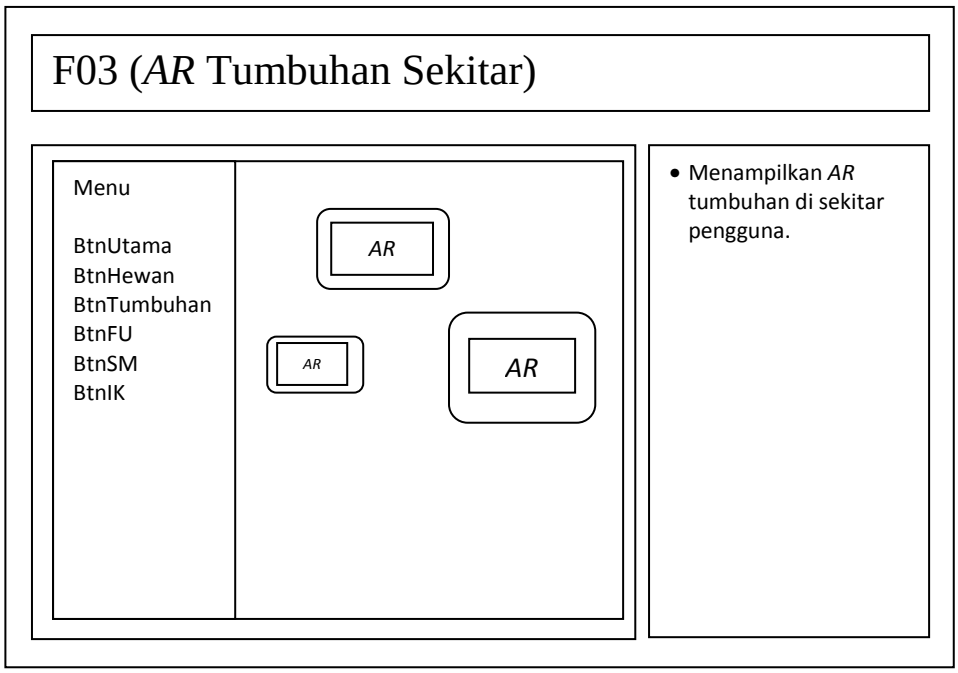
Gambar 3. 146 Antarmuka Utama.

2. Antarmuka Halaman *AR* Hewan Sekitar.



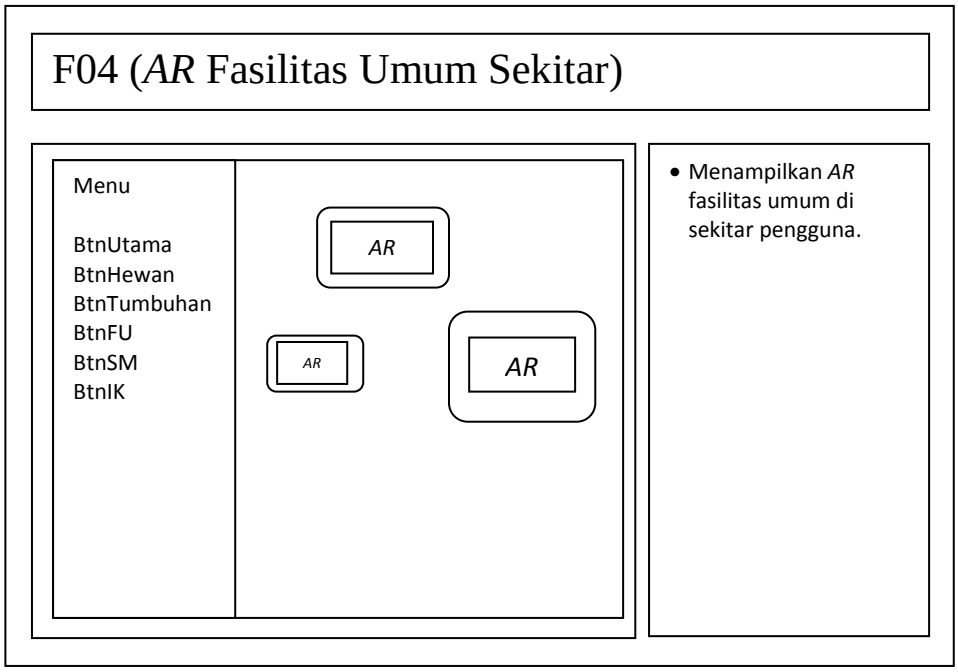
Gambar 3. 147 Antarmuka AR Hewan Sekitar.

3. Antarmuka Halaman *AR* Tumbuhan Sekitar.



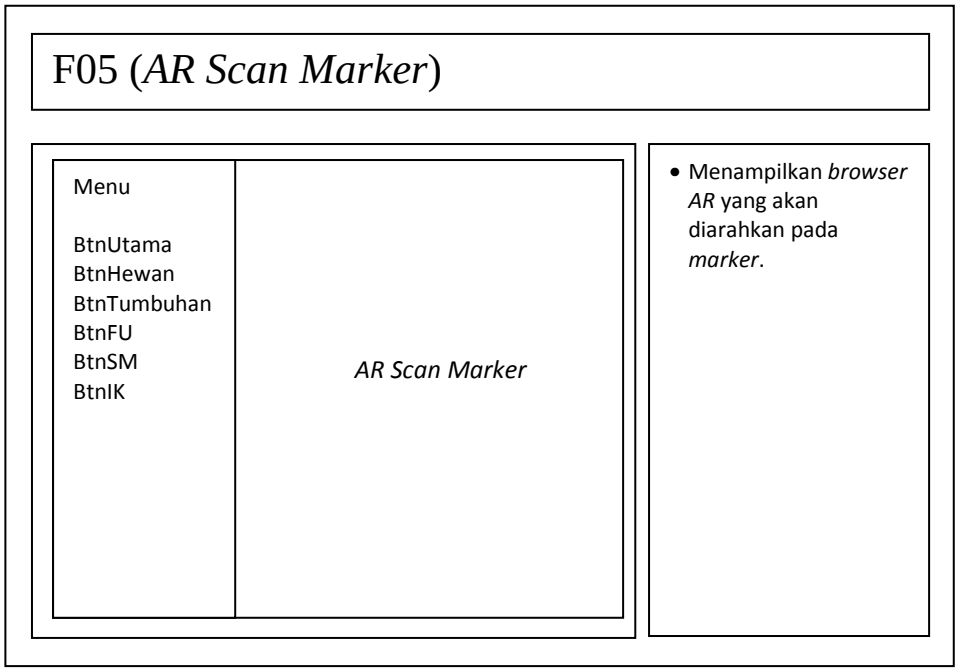
Gambar 3. 148 Antarmuka AR Tumbuhan Sekitar.

4. Antarmuka Halaman *AR* Fasilitas Umum Sekitar.



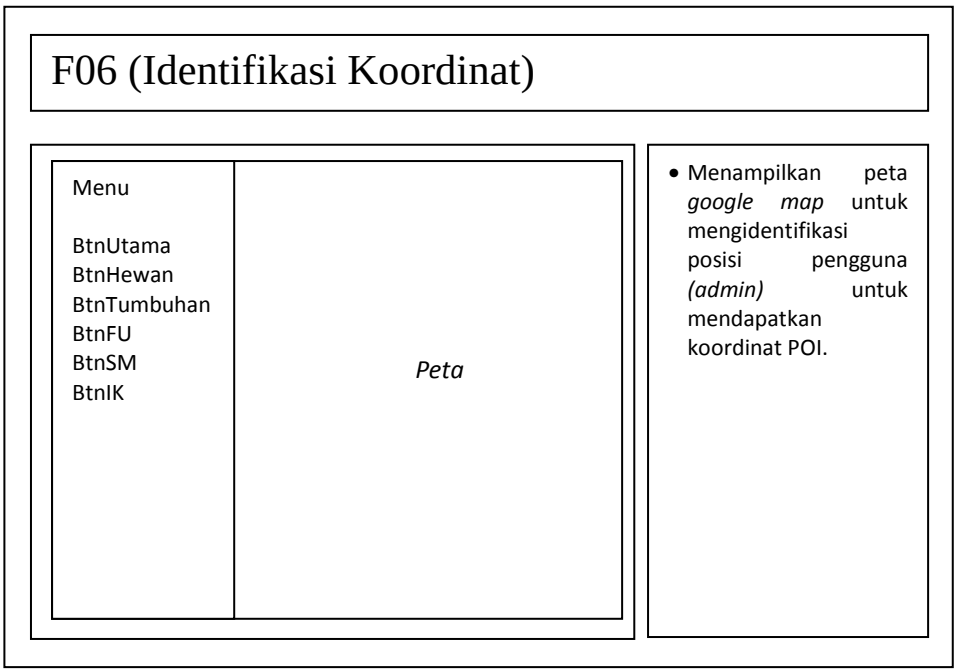
Gambar 3. 149 Antarmuka AR Fasilitas Umum Sekitar.

5. Antarmuka Halaman *AR Scan Marker*.



Gambar 3. 150 Antarmuka AR Scan Marker.

6. Antarmuka Halaman Identifikasi Koordinat.

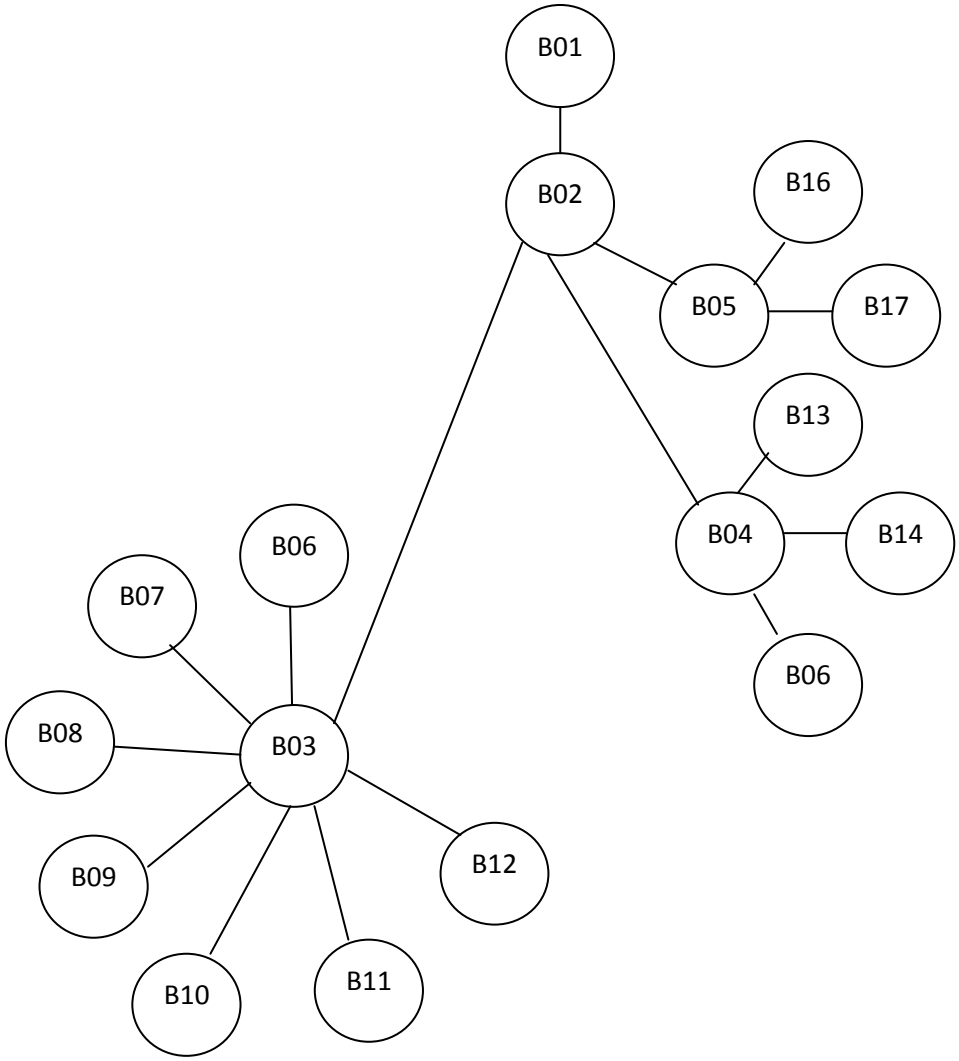


Gambar 3. 151 Antarmuka Identifikasi Koordinat.

3.2.2. Jaringan Semantik

a. Jaringan Semantik *Backend*

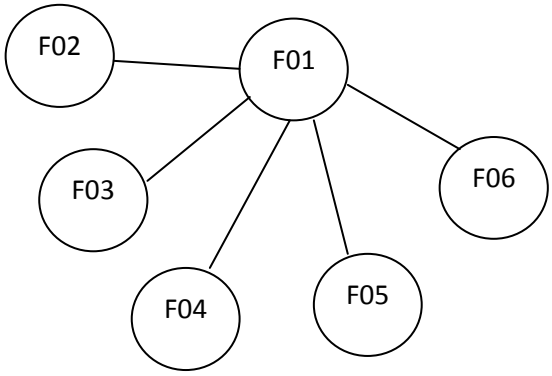
Berikut perancangan jaringan semantik aplikasi pada sisi *backend* :



Gambar 3. 152 Jaringan Semantik Backend.

b. Jaringan Semantik *Frontend*

Berikut perancangan jaringan semantik aplikasi pada sisi *frontend* :



Gambar 3. 153 Jaringan Semantik Frontend.