BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis dan Perancangan Sistem

Analisis dan perancangan sistem memerlukan tahapan yang sistematis untuk mendapatkan aplikasi yang baik dan bersesuaian dengan kegunaan dan tujuannya. Tahap awal dari analisis adalah menganalisis masalah, analisis sistem yang sedang berjalan, deskripsi sistem, dan kebutuhan-kebutuhan sistem mulai dari kebutuhan non fungsional, kebutuhan fungsional, hingga kebutuhan pengguna. Sedangkan untuk tahap perancangan aplikasi yaitu perancangan data, perancangan menu, perancangan antarmuka, dan perancangan prosedural.

3.1.1 Analisis Masalah

Analisis masalah yang dilakukan adalah melihat dan mengidentifikasikan permasalahan atau kendala di dalam penelitian yang dilakukan. Masalah dapat didefinisikan sebagai suatu pertanyaan yang diinginkan untuk dipecahkan. Perlu ada analisis masalah melalui rumusan masalah yang sudah ditentukan sebelumnya. Dari hal rumusan masalah, permasalahan yang terjadi adalah:

1. Sulitnya Dinas Kelautan dan Perikanan menyalurkan informasi berita, pengumuman, harga dan lokasi kepada nelayan.
2. Sulitnya nelayan menyampaikan keluhan kepada Dinas Kelautan dan Perikanan.
3. Sulitnya nelayan memperoleh informasi tentang cuaca, harga, berita, pengumuman dan lokasi.

Berdasarkan masalah yang ada, maka diperlukan pembangunan sistem yang akan menyelesaikan masalah yang ada. Sistem ini terdiri dari dua bagian, sistem yang pertama adalah aplikasi *front end*, aplikasi *front end* yang memberikan informasi berita, informasi harga ikan, informasi prakiraan cuaca dan memfasilitasi kegiatan penyampaian keluhan dengan dinas kelautan dan perikanan jawa barat. Sistem yang kedua adalah aplikasi *back end*, aplikasi *back end* dibangun untuk mengelola kegiatan pendistribusian informasi dan kegiatan

penyampaian keluhan dengan Dinas Kelautan dan Perikanan, mulai dari memasukkan informasi. Dengan adanya sistem yang terdiri dari *front end* dan *back end* ini, kegiatan pendistribusian informasi tidak lagi konvensional, informasi langsung disampaikan pada nelayan, juga kegiatan penyampaian keluhan dengan dinas kelautan dan perikanan akan lebih fleksibel karena dapat dilakukan kapan pun dan dimana pun.

3.1.2 Analisis Sistem yang Sedang Berjalan

Sistem yang berjalan pada saat ini adalah sistem yang konvensional. Kegiatan Dinas Kelautan dan Perikanan bagian sumber daya seperti penyampaian informasi kepada setiap nelayan dilakukan dengan cara yang konvensional, yaitu dengan cara penyuluh langsung datang ketempat nelayan. Kegiatan untuk sharing atau bertukar pikiran dengan nelayan lain yang ditempat berbeda masih belum bisa dan juga untuk berkomunikasi dinas kelautan dan perikanan dilakukan dengan mendatangi atau tenaga penyuluh langsung datang ke tempat nelayan tersebut.

1. **Proses Bisnis Yang Sedang Berjalan**

   Proses bisnis yang sedang berjalan adalah proses bisnis yang digunakan dalam pelaksanaan kegiatan peyuluhan sebelum dibuatnya aplikasi M-Nelayan. Proses bisnis yang sedang berjalan merupakan suatu proses yang diperlukan untuk dianalisis untuk mendapatkan proses bisnis yang baru setelah pengimplementasian sistem.

Tabel 3.1 Proses Bisnis yang Sedang Berjalan

No	Kegiatan	Proses bisnis yang berjalan
1	Informasi Cuaca, harga ikan, lokasi ikan	Penyuluh dari dinas kelautan datang langsung kelapangan untuk menyampaikan infomasi tersebut kepada setiap nelayan.
2	Keluhan	Tenaga penyuluhan langsung datang ke nelayan untuk mendengarkan keluhan dari nelayan.

2. Alur Pendistribusian Data atau Informasi yang sedang berjalan

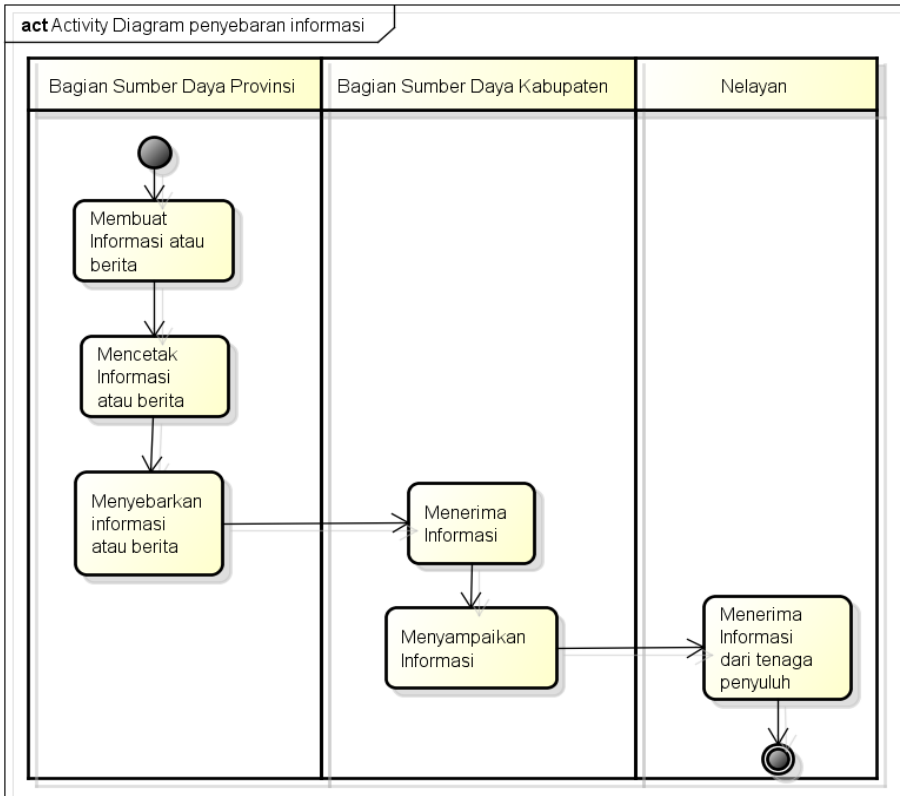
Alur pendistribusian data atau informasi yang sedang berjalan saat ini adalah data yang didapat dari Dinas Kelautan dan Perikanan yang akan diberikan kepada nelayan. Pendistribusian data dimulai dari Dinas Kelautan dan Perikanan melalui tenaga penyuluhan lalu ke nelayan.

3.1.2.1 *Activity Diagram* Sistem yang sedang Berjalan

Activity diagram merupakan diagram yang memodelkan aliran kerja atau *workflow* dari urutan aktifitas dalam suatu proses. *Activity diagram* sistem yang berjalan terdiri dari *activity diagram* penyebaran informasi, *activity diagram* penyampaian keluhan.

1. *Activity Diagram* Penyebaran Informasi

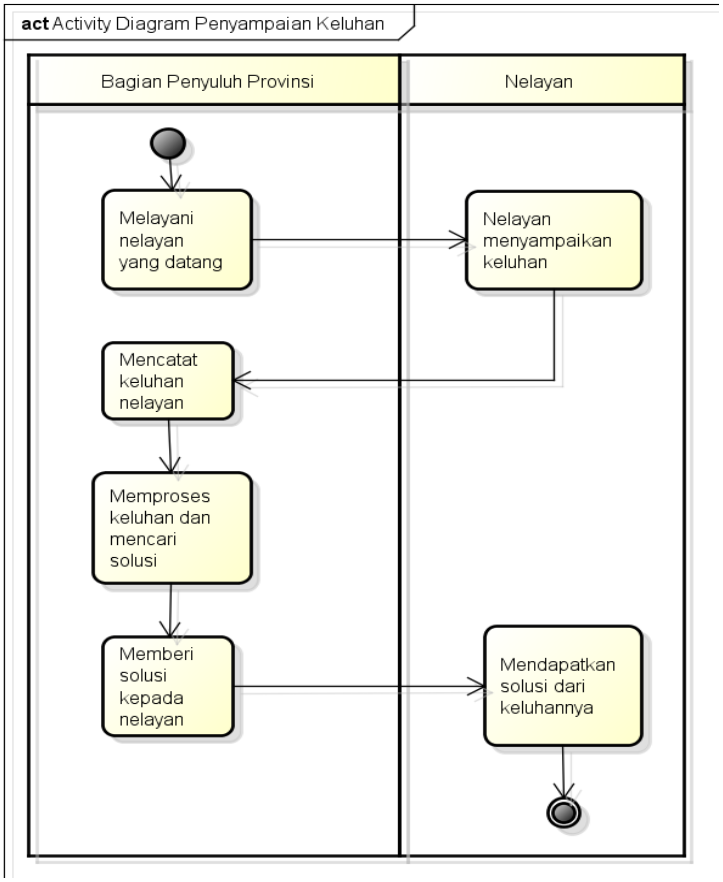
Activity diagram penyebaran informasi menjelaskan alur kerja sistem penyebaran informasi tentang cuaca, harga ikan dan lokasi ikan pada sistem yang sedang berjalan.



Gambar 3.1 *Activity Diagram* Penyebaran Informasi

2. *Activity Diagram* Penyampaian Keluhan

Activity diagram penyampaian keluhan menjelaskan alur kerja sistem kegiatan komunikasi pada sistem yang sedang berjalan.



Gambar 3.2 *Activity Diagram* Penyampaian Keluhan

3.1.3 Deskripsi Sistem

Deskripsi sistem membahas tentang detail sistem yang akan dibangun, dimulai dari proses bisnis baru, pendistribusian data atau informasi yang baru, dan detail arsitektur sistem yang dibangun.

1. Proses Bisnis yang akan dibangun

Proses bisnis baru yang dibangun adalah proses bisnis yang digunakan dalam menyampaikan informasi yang diimplementasikan dengan aplikasi *mobile* yang akan dibangun. Proses bisnis baru merupakan suatu proses yang

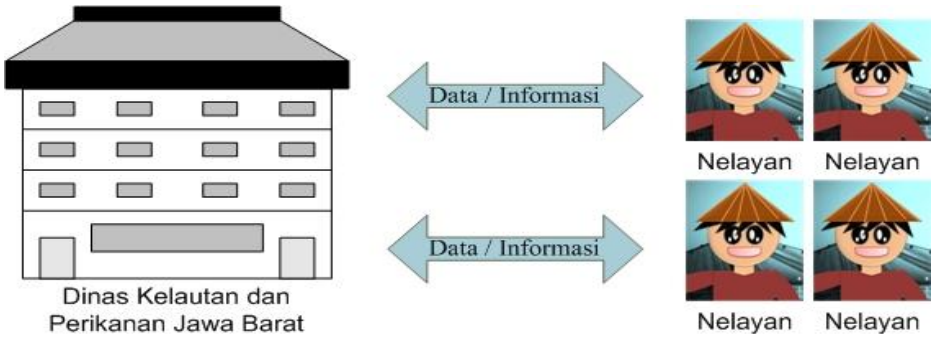
diperlukan karena telah dibuatnya sistem yang baru dalam menyampaikan informasi.

Tabel 3.2 Proses Bisnis yang akan Dibangun

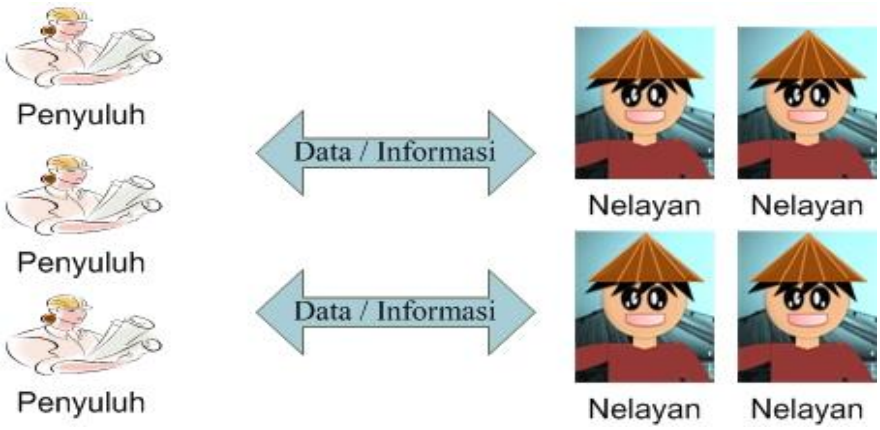
NO	Kegiatan	Proses bisnis yang akan dibangun
1	Informasi cuaca, informasi harga ikan, informasi lokasi ikan dll	Informasi disimpan kedalam sistem, kemudian akan ditampilkan kedalam aplikasi android.
2	Penyampaian keluhan	Sarana untuk menyampaikan keluhan dan tukar pikiran dengan nelayan yang lain dapat dilakukan melalui aplikasi M-Nelayan ini.

1. Alur Pendistribusian Data atau Informasi M-Nelayan

Alur pendistribusian data atau informasi ini adalah alur pendistribusian data atau informasi yang baru setelah adanya sistem aplikasi M-Nelayan. Alur pendistribusian data atau informasi yang baru ini digunakan dalam pengimplementasian aplikasi M-Nelayan.



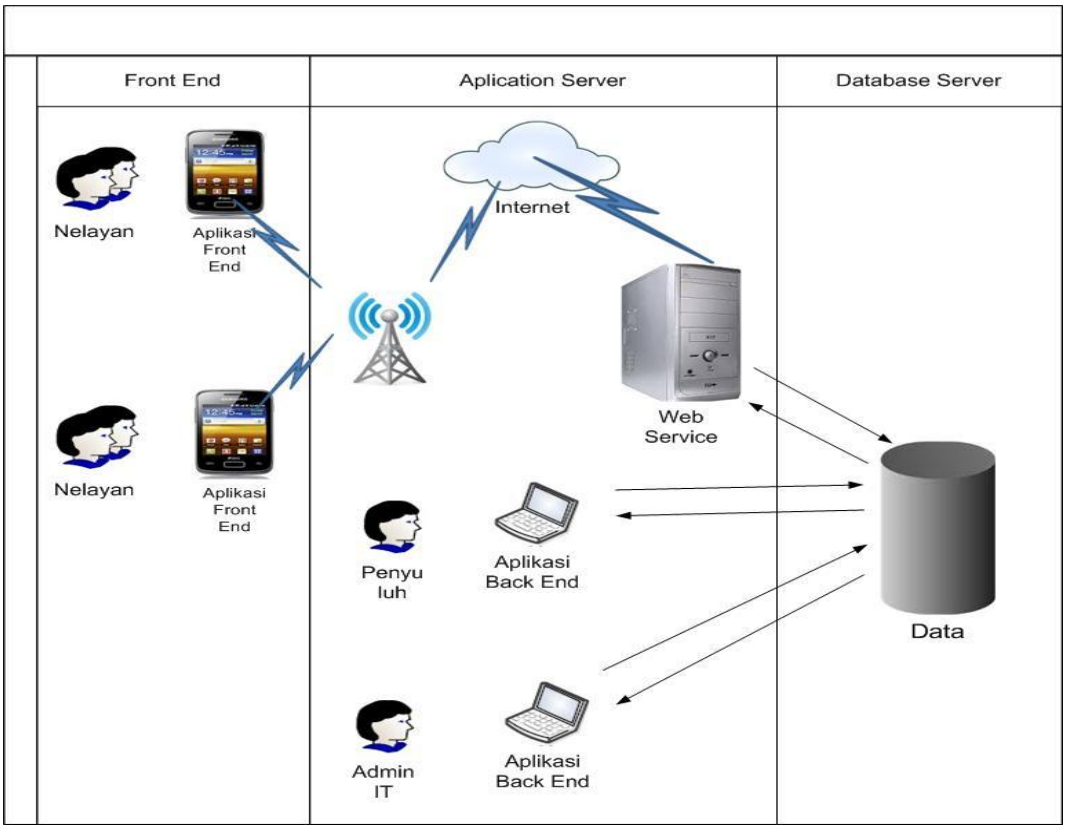
Gambar 3.3 Alur Pendistribusian Data atau Informasi M-Nelayan



Gambar 3.4 Alur Penyampaian Pesan Keluhan M-Nelayan

3. Arsitektur Sistem

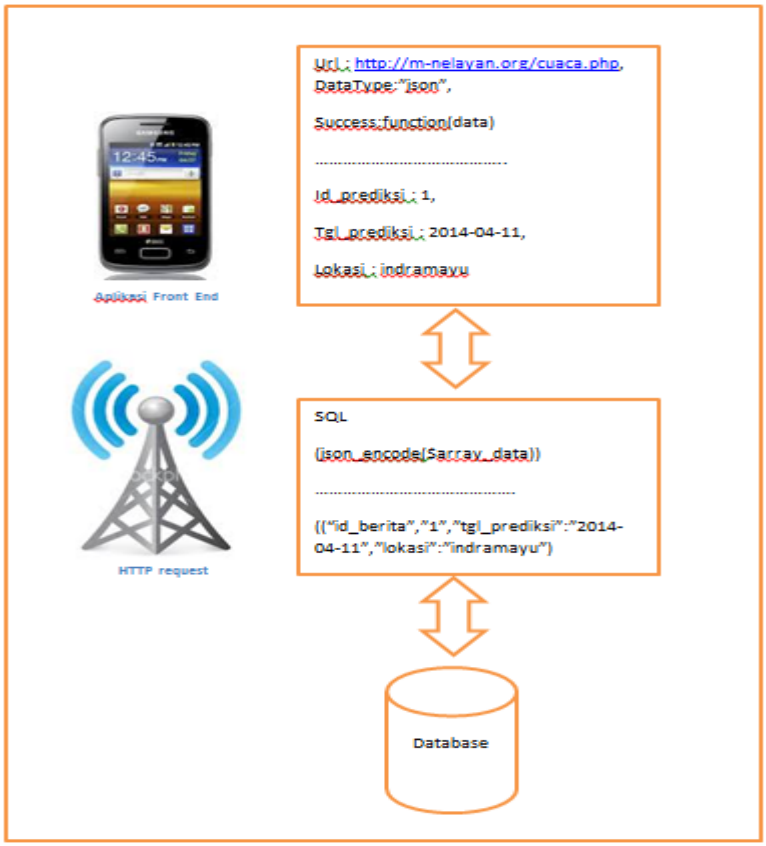
Arsitektur fisik sistem yang digunakan dalam penelitian adalah menggunakan model arsitektur aplikasi *two tier* dimana arsitektur fisik sistem ini mempunyai tiga komponen utama yaitu *client* (*frontend*), *application server* (*backend* dan *Web Service*) dan *database server*.



Gambar 3.5 Arsitektur Sistem

3.1.3.1 *Aplikasi Front End*

Aplikasi *front end* merupakan aplikasi yang berfungsi sebagai antarmuka antara pengguna dan sistem. *Front end* digunakan untuk menampilkan informasi kelautan, memberikan layanan pendataan nelayan, sarana untuk tukar pikiran dengan nelayan lain dan tim penyuluhan dari dinas kelautan dan perikanan. Pengguna aplikasi *front end* adalah seluruh nelayan yang ada di Jawa Barat dengan hak akses dapat menggunakan seluruh fitur yang ada di dalam aplikasi *front end*. Aplikasi *front end* me-*request* data informasi kelautan, data diskusi, dan data media yang ada didalam database server, *request* diproses oleh web service dan data dijadikan JSON, lalu aplikasi *front end* mem-*parsing* data JSON yang telah di sediakan tersebut dan menampilkannya di aplikasi *front end*.



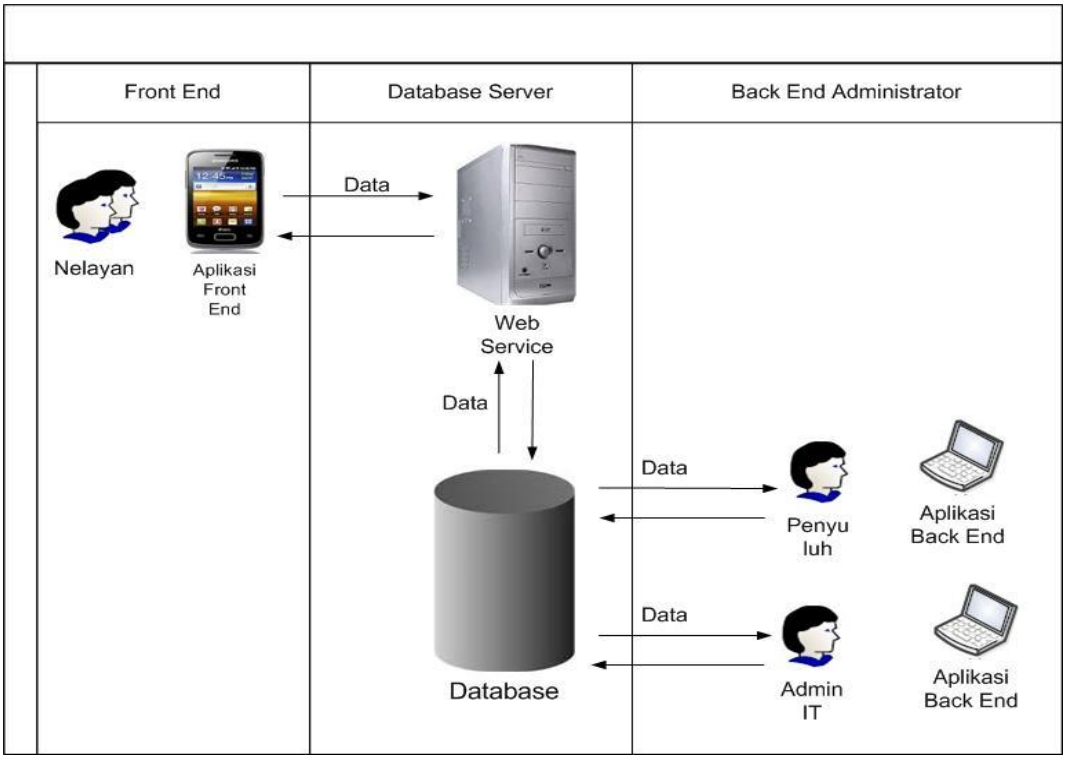
Gambar 3.6 Request Data JSON

3.1.3.2 *Aplikasi Back End*

Aplikasi *back end* merupakan pengolahan data yang diimplementasikan pada sistem berbasis web. *Back end* terdiri dari web admin, web *service* dan database server. Pengguna atau pengelola aplikasi *back end* adalah bagian sumber daya yaitu sebagai admin yang memiliki hak akses untuk meng CRUD informasi. Pengelola yang kedua bagian bidang penyuluh dinas kelautan dan perikanan provinsi jawa barat diberi hak akses untuk mengelola bagian sarana komunikasi antara nelayan yang lain dengan dinas kelautan. Web admin digunakan untuk mengolah seluruh layanan aplikasi mulai dari mengirimkan informasi, mengelola data nelayan dan menjawab pertanyaan. Web *service* sebagai fasilitas sistem yang digunakan untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem ini melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan web service, dalam hal ini sistem *back end* aplikasi *mobile* M-Nelayan. Web *service* yang digunakan menggunakan PHP dan JSON. PHP yang menyediakan layanan web *service* yang akan memproses *request* kemudian menghasilkan JSON sebagai hasil dari pemrosesan *request*. *Database Server* yang digunakan pada aplikasi ini adalah MySQL. Data Informasi lokasi ikan dan data penyampaian keluhan yang tersimpan didalam *database server* di *request* kemudian diproses oleh web *service* yang disediakan oleh PHP dan kemudian menjadikan data tersebut dalam bentuk JSON. Data JSON yang dihasilkan tersebut yang disediakan untuk di-*request* oleh aplikasi *front end.*

3.1.3.3 *Aliran Data*

Data mengalir melalui internet, data yang disediakan untuk aplikasi *front end* dan aplikasi *back end* disimpan di database server.



Gambar 3.7 Aliran Data

Penjelasan dari alur data sistem yang akan dibangun adalah sebagai berikut:

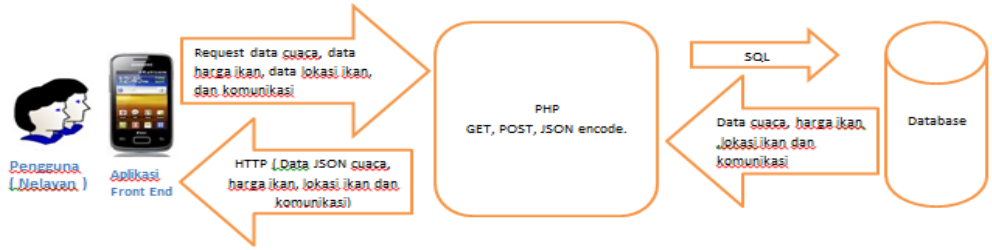
1. Aplikasi *front end*

Aplikasi *front end* dalam hal ini merupakan analisis fungsionalitas aplikasi untuk mengakses data dari database server melalui web service. Aplikasi *front end* ini dibangun diatas *platform* Android dan bekerja dengan cara mem-parsing data dari web service untuk diolah pada aplikasi *front end*. Data cuaca, data harga ikan, dan data lokasi yang kemudian diproses oleh web service dan dijadikan data dalam bentuk *JSON* di parsing dan kemudian ditampilkan di aplikasi *front end*.

2. Web Service

Web *service* memudahkan beberapa aplikasi atau komponennya untuk saling berhubungan dengan aplikasi lain dalam sebuah organisasi maupun diluar organisasi dengan menggunakan standar yang tidak terikat *platform* (*platform-neutral)* dan tidak terikat akan bahasa pemrograman yang digunakan (*language-neutral*). Web *service* ini berfungsi untuk menghubungkan aplikasi *front end* dan aplikasi *back end*. Cara kerja web service ini yaitu dengan mengambil data dari database server dan kemudian mengkonversikan data tersebut kedalam format

pertukaran data dalam hal ini JSON. Data cuaca, data harga ikan, dan data lokasi ikan yang tersimpan didalam database server di *request* kemudian di proses oleh web service yang disediakan oleh PHP dan kemudian menjadikan data tersebut dalam bentuk JSON. Data JSON yang dihasilkan tersebut yang disediakan untuk di-*request* oleh aplikasi *front end*.



Gambar 3.8 Alur Request Data

3. Protokol HTTP

Protokol HTTP merupakan protokol jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif, dan menggunakan hipermedia. Pada penelitian ini HTTP digunakan pada aplikasi *mobile* dan web administrator sebagai protokol yang dapat mendistribusikan data yang bersumber dari database server. Pendistribusian data cuaca, data harga ikan, data lokasi ikan dan data komunikasi dilakukan melalui protokol http. Data yang tersimpan didalam database server di *request* kemudian diproses oleh web service dan kemudian menjadikan data tersebut dalam bentuk JSON, proses pendistribusian data tersebut dilakukan melalui protokol http.

3.1.4 Analisis dan Kebutuhan Non-Fungsional

Analisis dan kebutuhan non-fungsional menggambarkan kebutuhan sistem yang menitik-beratkan pada perilaku yang dimiliki oleh sistem sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan, yaitu meliputi analisis dan kebutuhan perangkat

keras, analisis dan kebutuhan perangkat lunak, serta analisis dan kebutuhan user.

3.1.4.1 *Analisis dan Kebutuhan Perangkat Keras*

Analisis kebutuhan perangkat keras merupakan analisis kebutuhan yang harus dipenuhi dalam perancangan sistem. Analisis kebutuhan perangkat keras terdiri dari :

1. **Perangkat Keras Aplikasi yang Ada di Dinas**

   Dinas Kelautan dan Perikanan Provinsi Jawa Barat pada bagian sumber daya saat ini mempunyai 3 unit komputer dan 2 unit printer dengan spesifikasi sebagai berikut :

   a) Processor : Intel Pentium Dual Core@ 2.0 Ghz
   b) Ram : 1024 Mb
   c) Harddisk : 160 Gb
   d) Monitor : LCD 15”
   e) Mouse dan Keyboard
   f) Printer

2. **Perangkat Keras Aplikasi Back End yang di butuhkan**

Kebutuhan perangkat keras minimal aplikasi *back end* dijelaskan pada tabel dibawah ini.

Tabel 3.3 Kebutuhan Perangkat Keras Minimal untuk Aplikasi Back End

No	Perangkat Keras	Spesifikasi
1	Processor	2.0 GHz
2	RAM	512 GB
3	Harddisk	80 GB
4	Mouse	Standar
5	Keyboard	Standar
6	Monitor	Standar
7	Network Interface Card	10 Mbit/s

3. **Perangkat Keras Aplikasi *Front End***

Kebutuhan perangkat keras minimal aplikasi *front end* dijelaskan pada tabel dibawah ini.

Tabel 3.4 Kebutuhan Perangkat Keras Minimal untuk Aplikasi Front End

No	Perangkat Keras	Spesifikasi
1	Layar	Resolusi 240x320 pixels
2	Memori	100 MB
3	CPU	300 Mhz
4	WLAN WiFi	802.11 b/g/n
5	Jaringan	3G UMTS
6	Camera	2 MP, 1600x1200 pixel

3.1.4.2 Analisis dan Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak merupakan analisis kebutuhan yang harus dipenuhi dalam perancangan sistem. Analisis kebutuhan perangkat lunak terdiri dari :

1. Perangkat lunak Aplikasi *Front End*

Kebutuhan perangkat lunak minimal aplikasi *front end* dijelaskan pada dibawah ini.

Tabel 3.5 Kebutuhan Perangkat Lunak Minimal untuk Aplikasi Front End

NO	Perangkat Lunak	Spesifikasi
1	Sistem Operasi *Mobile*	Android 2.2 ke atas

2. Perangkat Lunak Aplikasi *Back End*

Kebutuhan perangkat lunak minimal aplikasi *back end* dijelaskan pada tabel dibawah ini.

Tabel 3.6 Kebutuhan Perangkat Lunak Minimal untuk Aplikasi Back End

NO	Perangkat Lunak	Spesifikasi
1	Sistem Operasi Desktop	Windows 7
2	IDE	Eclipse Galileo dan Dreamweaver
3	JDK	JDK v 1.6

4	SDK	Android SDK Windows
5	ADT	ADT v.10.0.0.
6	Browser	Mozila Firefox

3.1.4.3 *Analisis dan Kebutuhan Pengguna*

Analisis pengguna merupakan analisis yang dilakukan untuk pengguna yang nanti-nya akan menggunakan sistem, analisis ini meliputi siapa pengguna sistem dan hak akses pengguna terhadap sistem. Analisis pengguna aplikasi *back end* dan aplikasi *front end* dijelaskan pada tabel dibawah ini.

Tabel 3.7 Analisis Pengguna yang Sedang Berjalan

No.	Tipe	Tanggung Jawab	Tingkat Pendidikan	Tingkat Keteram pilan
1	Seksi Kelembagaan dan Pengelolaan Sumberdaya	Mengolah data tentang kelautan dan perikanan.	Minimal Strata I	Bisa mengikuti petunjuk yang ada pada dinas kelautan.
2	Penyuluh	Menyampaikan informasi ke nelayan.	Minimal SMA	Bisa mengikuti petunjuk yang ada pada dinas kelautan.

Tabel 3.8 Analisis Pengguna Aplikasi Back End

No.	Tipe	Tanggung Jawab	Hak Akses	Tingkat Pendidikan	Tingkat Keteram pilan	Pengal aman
1	Admin IT	Mengolah data tentang	Meng-create akun Penyuluh	Minimal Strata I	Bisa mengikuti petunjuk	Min. mampu mengope

		kelautan dan perikanan pada web server.	dan meng-update informasi setiap hari.		yang ada pada sistem.	rasikan web server.
2	Penyuluh	Mengolah data.	Mengelola data keluhan dari nelayan.	Minimal SMA	Bisa mengikuti petunjuk yang ada pada sistem.	Min. mampu mengoperasikan web server.

Tabel 3.9 Analisis Pengguna Aplikasi Front End

No.	Tipe	Tanggung Jawab	Hak Akses	Tingkat Pendidikan	Tingkat Keterampilan	Pengalaman
1	Nelayan	-	Melihat Informasi tentang kelautan, penyampaian keluhan dengan penyuluh.	Min. SMP	Bisa mengikuti petunjuk yang ada pada sistem.	Min. mampu mengoperasikan *mobile* berbasis android.

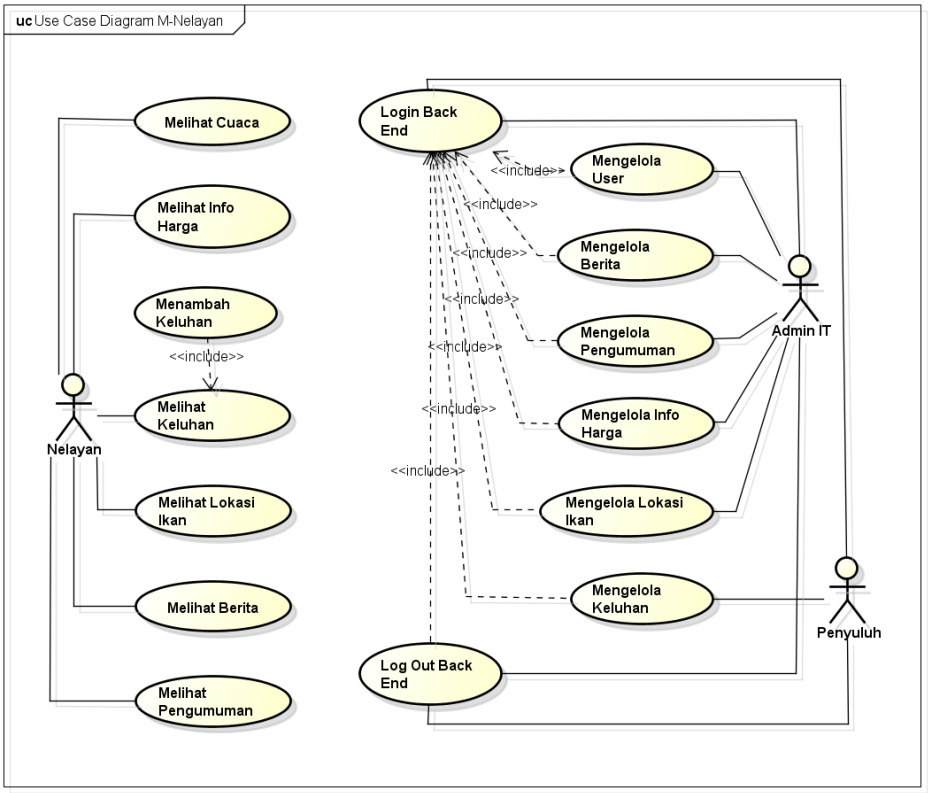
3.1.5 Analisis dan Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sistem yang akan dibangun sehingga sistem dapat berjalan dengan baik. Analisis dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis menggunakan

UML antara lain *use case* diagram, *use case* skenario, *activity* diagram, *sequence* diagram, dan *class* diagram.

3.1.5.1 *Use case Diagram*

Use case diagram merupakan kontruksi untuk mendeskripsikan hubungan yang terjadi antara aktor dengan aktivitas yang terdapat pada sistem, *use case* menjelaskan proses apa saja yang ada dalam sistem dan bagaimana hubungannya dengan aktor. *Use case* diagram aplikasi M-Nelayan merupakan gabungan dari *use case* aplikasi *front end* dan *use case* aplikasi *back end*.



Gambar 3.9 Use Case Diagram

3.1.5.2 Definisi *Use Case*

Definisi *usecase* berfungsi untuk menjelaskan proses yang terdapat pada setiap Use Case. Definisinya dapat dilihat pada table dibawah ini.

Tabel 3.10 Definisi *Use Case*

No	*Use Case*	Deskripsi
1	Melihat Cuaca	Proses untuk melihat informasi cuaca

		yang di berikan oleh yahoo.
2	Melihat Info Harga	Proses untuk melihat daftar harga ikan terbaru setiap hari.
3	Melihat Keluhan	Proses untuk melihat semua keluhan yang telah dikirim oleh nelayan.
4	Menambah Keluhan	Proses untuk menyampaikan keluhan yang sedang dialami nelayan kepada Dinas Kelauatan dan Perikanan.
5	Melihat Lokasi Ikan	Proses untuk melihat lokasi ikan terbaru setiap 2 hari sekali.
6	Melihat Berita	Proses untuk melihat kumpulan berita yang ditampilkan ke *mobile*.
7	Melihat Pengumuman	Proses untuk melihat pengumuman yang ditampilkan ke *mobile*.
8	Login Back End	Halaman awal aplikasi yang mengharuskan seluruh pengguna menuliskan *username* dan *password* dengan benar. Di dalam proses ini terdapat pengecekan *username* dan *password*.
9	Mengelola User	Proses ini untuk menambah, mengubah dan menghapus data pengguna.
10	Mengelola Berita	Proses dimana Admin mengelola data tentang berita yang akan disampaikan.
11	Mengelola Pengumuman	Proses dimana Admin mengelola data tentang pengumuman yang akan di sampaikan.
12	Mengelola Info Harga	Proses dimana Admin mengelola data tentang harga ikan setiap hari.
13	Mengelola Lokasi Ikan	Proses dimana Admin mengelola data tentang lokasi ikan perdua hari.
14	Mengelola Keluhan	Proses dimana Penyuluh membaca keluhan dari nelayan dan memproses

		untuk mengambil tindakan terhadap keluhan nelayan.
15	Log Out *Back End*	Proses untuk keluar dari aplikasi M-nelayan.

3.1.5.3 Skenario *Use Case*

Skenario *use case* merupakan deskripsi urutan langkah-langkah dalam proses bisnis baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. Skenario *use case* dijelaskan di setiap tabel skenario yang menjelaskan masing-masing *use case*.

Tabel 3.11 Skenario Use Case Melihat Cuaca

Use Case	Melihat Cuaca
Dekripsi	Berisi informasi cuaca yang terupdate dari yahoo.
Aktor	Nelayan
Kondisi Awal	Halaman utama aplikasi *front end*
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Memilih menu cuaca	
	2. Menampilkan list cuaca berdasarkan tanggal terbaru.
3. Memilih list cuaca terbaru	
	4. Menampilkan cuaca yang dipilih.
Kondisi Akhir	Tampilan informasi cuaca

Tabel 3.12 Skenario Use Case Melihat Info Harga

Use Case	Melihat Info Harga
Dekripsi	Berisi tentang informasi harga ikan
Aktor	Nelayan
Kondisi Awal	Menu utama aplikasi
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Memilih menu info harga	
	2. Menampilkan list harga ikan yang ingin di lihat
3. Memilih list info harga terbaru	
	4. Menampilkan harga ikan yang dipilih
Kondisi Akhir	Tampilan harga ikan yang diinginkan

Tabel 3.13 Skenario Use Case Menambah Keluhan

Use Case	Menambah Keluhan
Dekripsi	Proses mengirim atau menyampaikan keluhan kepada Dinas Kelautan dan Perikanan
Aktor	Nelayan
Kondisi Awal	Halaman utama keluhan
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Memilih button tambah	
	2. Menampilkan form tambah keluhan
3. Mengisi form tambah keluhan dan menekan tombol simpan.	
	4. Mengecek data sudah benar (jika data kurang valid maka akan kembali ke no.4)
Kondisi Akhir	Menampilkan halaman keluhan

Tabel 3.14 Skenario Use Case Melihat Keluhan

Use Case	Melihat Keluhan
Dekripsi	Berisi tentang keluhan-keluhan yang disampaikan nelayan
Aktor	Nelayan
Kondisi Awal	Halaman utama aplikasi
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Memilih Menu Keluhan	
	2. Menampilkan halaman keluhan dan list keluhan yang sudah ada dari nelayan.
Kondisi Akhir	Menampilkan halaman keluhan

Tabel 3.15 Skenario Use Case Melihat Lokasi Ikan

Use Case	Melihat Lokasi Ikan
Dekripsi	Berisi informasi lokasi ikan yang banyak berkumpul dan untuk mengetahui lebih cepat keberadaan ikan.
Aktor	Nelayan
Kondisi Awal	Halaman utama aplikasi
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Memilih menu lokasi ikan	
	2. Menampilkan list peta lokasi ikan berdasarkan tanggal terbaru.
3. Memilih list menu peta lokasi ikan.	
	4. Menampilkan peta lokasi ikan yang dipilih.
Kondisi Akhir	Tampilan informasi lokasi ikan

Tabel 3.16 Skenario Use Case Melihat Berita

Use Case	Melihat Berita
Dekripsi	Berisi informasi tentang kelautan dan perikanan
Aktor	Nelayan
Kondisi Awal	Menu utama aplikasi
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Memilih menu berita	
	2. Menampilkan list berita apa yang ingin di lihat
3. Memilih list berita	
	4. Menampilkan berita yang dipilih
Kondisi Akhir	Tampilan informasi yang diinginkan

Tabel 3.17 Skenario Use Case Melihat Pengumuman

Use Case	Melihat Pengumuman
Dekripsi	Berisi tentang pengumuman dari dinas kelautan dan perikanan
Aktor	Nelayan
Kondisi Awal	Menu utama aplikasi
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Memilih menu pengumuman	
	2. Menampilkan list pengumuman apa yang ingin dilihat
3. Memilih list pengumuman	
	4. Menampilkan pengumuman yang dipilih
Kondisi Akhir	Tampilan informasi yang diinginkan

Tabel 3.18 Skenario Use Case Login Back End

Use Case	Login Back End
Dekripsi	Form login yang berisi username dan password pengguna yang mau masuk ke sistem.
Aktor	Penyuluh, Admin IT
Kondisi Awal	Halaman login penyuluh, Admin IT
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Mengisi Username dan Password	
	2. Memeriksa apakah username dan password benar
	3. Melanjutkan ke halaman utama aplikasi
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
1. Mengisi username dan password yang salah	
	2. Memeriksa kebenaran username dan password yang diinputkan
	3. Menampilkan koreksi kesalahan login
4. Mengisi kembali username dan password yang benar	
	5. Memeriksa kebenaran username dan password yang diinputkan
	6. Melanjutkan kehalaman utama aplikasi
Kondisi Akhir	Menampilkan Halaman Utama Aplikasi

Tabel 3.19 Skenario Use Case Mengelola User

Use Case	Mengelola User
Dekripsi	Mengelola user yang sudah ada
Aktor	Admin IT
Kondisi Awal	Halaman Utama Admin
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Memilih menu mengelola user	
	2. Menampilkan seluruh nama user yang ada
Kondisi Akhir	Menampilkan seluruh nama user yang ada
Skenario Edit User	
1) Memilih button edit	
	2) Menampilkan form edit
3) Merubah akun user dan menekan tombol update untuk di simpan	
	4) Mengecek data sudah benar (jika data kurang valid maka akan kembali ke no.2)
Kondisi Akhir	Menampilkan seluruh nama user

Tabel 3.20 Skenario Use Case Mengelola Berita

Use Case	Mengelola Berita
Dekripsi	Mengolah informasi yang akan disampaikan kepada nelayan
Aktor	Admin IT
Kondisi Awal	Halaman Utama Admin
Skenario Normal menampilkan isi berita	
Aksi Aktor	Reaksi Sistem
1. Memilih menu berita	
	2. Menampilkan Seluruh list berita
Kondisi Akhir	Menampilkan seluruh list berita
Skenario Menambah Berita	
1) Memilih button tambah berita	
	2) Menampilkan form tambah berita
3) Mengisi berita dan menekan tombol simpan berita	
	4) Mengecek data sudah benar (jika data kurang valid maka akan kembali ke no.2)
Skenario Edit Berita	
1) Memilih button edit berita	
	2) Menampilkan form edit berita
3) Merubah berita dan menekan tombol update untuk di simpan	
	4) Mengecek data sudah benar (jika data kurang valid maka akan kembali ke no.2)
Kondisi Akhir	Menampilkan seluruh isi berita

Tabel 3.21 Skenario Use Case Mengelola Pengumuman

Use Case	Mengelola Pengumuman
Dekripsi	Mengelola pengumuman yang akan disampaikan kepada nelayan
Aktor	Admin IT
Kondisi Awal	Halaman Utama Admin
Skenario Normal menampilkan list pengumuman	
Aksi Aktor	Reaksi Sistem
1. Memilih menu pengumuman	
	2. Menampilkan Seluruh list pengumuman
Kondisi Akhir	Menampilkan seluruh list pengumuman
Skenario Menambah pengumuman	
1) Memilih button tambah pengumuman	
	2) Menampilkan form tambah pengumuman
3) Mengisi pengumuman dan menekan tombol simpan pengumuman	
	4) Mengecek data sudah benar (jika data kurang valid maka akan kembali ke no.2)
Skenario Edit Pengumuman	
1) Memilih button edit pengumuman	
	2) Menampilkan form edit pengumuman
3) Merubah pengumuman dan menekan tombol update untuk di simpan	
	4) Mengecek data sudah benar (jika data kurang valid maka akan kembali ke no.2)
Kondisi Akhir	Menampilkan seluruh isi pengumuman

Tabel 3.22 Skenario Use Case Mengelola Harga Ikan

Use Case	Mengelola Info Harga
Dekripsi	Mengelola info harga yang akan disampaikan kepada nelayan
Aktor	Admin IT
Kondisi Awal	Halaman Utama Admin
Skenario Normal menampilkan list harga	
Aksi Aktor	Reaksi Sistem
1. Memilih mengelola harga	
	2. Menampilkan Seluruh list info harga
Kondisi Akhir	Menampilkan seluruh list info harga
Skenario Menambah info harga	
1) Memilih button tambah info harga	
	2) Menampilkan form tambah info harga
3) Mengisi data info harga dan menekan tombol simpan	
	4) Mengecek data sudah benar (jika data kurang valid maka akan kembali ke no.2)
Skenario Edit Info Harga	
1) Memilih button edit info harga	
	2) Menampilkan form edit info harga
3) Merubah info harga dan menekan tombol update untuk di simpan	
	4) Mengecek data sudah benar (jika data kurang valid maka akan kembali ke no.2)
Kondisi Akhir	Menampilkan seluruh isi info harga

Tabel 3.23 Skenario Use Case Mengelola Lokasi Ikan

Use Case	Mengelola Lokasi Ikan
Dekripsi	Mengelola informasi tentang lokasi yang banyak ikan.
Aktor	Admin IT
Kondisi Awal	Halaman Utama Admin
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Memilih menu lokasi	
	2. Menampilkan list lokasi ikan
Kondisi Akhir	Menampilkan lokasi ikan yang sudah ada.
Skenario Edit lokasi ikan	
1) Memilih button edit	
	2) Menampilkan form edit lokasi
3) Merubah form lokasi dan menekan tombol update untuk disimpan	
	4) Mengecek data sudah benar (jika data kurang valid maka akan kembali ke no.2)
Skenario Tambah Lokasi	
1) Memilih button tambah lokasi	
	2) Menampilkan form tambah lokasi
3) Mengisi form lokasi dan menekan tombol simpan.	
	4) Mengecek data sudah benar (jika data kurang valid maka akan kembali ke no.2)
Kondisi Akhir	Menampilkan seluruh isi lokasi ikan

Tabel 3.24 Skenario Use Case Mengelola Keluhan

Use Case	Mengelola Keluhan
Dekripsi	Mengelola Keluhan yang disampaikan Nelayan
Aktor	Penyuluh
Kondisi Awal	Halaman Utama Penyuluh
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Memilih Mengelola Keluhan	
	2. Menampilkan keluhan yang sudah dikirim
3. Memilih salah satu keluhan yang akan di baca	
	4. Menampilkan keluhan yang akan dibaca
Kondisi Akhir	Menampilkan keluhan yang dipilih
Skenario Membaca Keluhan	
1. Memilih button baca	
	2. Menampilkan keluhan yang dipilih
3. Mencari solusi untuk memecahkan keluhan nelayan	
	4. Menampilkan keluhan yang sudah dibaca
Kondisi Akhir	Menampilkan keluhan yang dipilih

Tabel 3.25 Skenario Use Case Logout Back End

Use Case	Logout Back End
Dekripsi	Keluar dari sistem atau aplikasi
Aktor	Penyuluhan,Admin IT
Kondisi Awal	Halaman menu Admin dan Aplikasi
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Memilih Menu Log Out	
	2. Menampilkan Halaman Login.
Kondisi Akhir	Menampilkan Halaman Login.

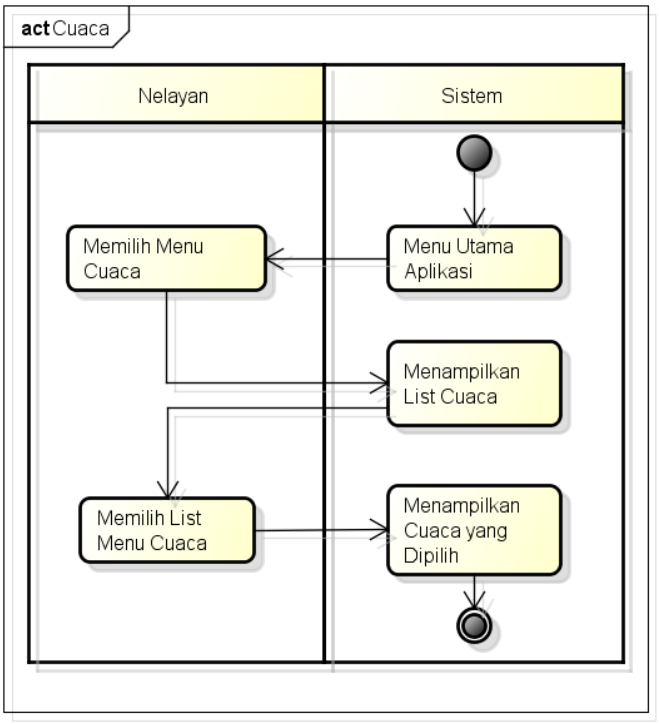
3.1.5.4 *Activity* Diagram

Activity diagram merupakan diagram yang memodelkan aliran kerja atau *workflow* dari urutan aktifitas dalam suatu proses yang mengacu pada *use*

case diagram yang ada. Berikut ini penjelasan dari masing-masing *activity* diagram.

1. *Activity* Diagram Melihat Cuaca

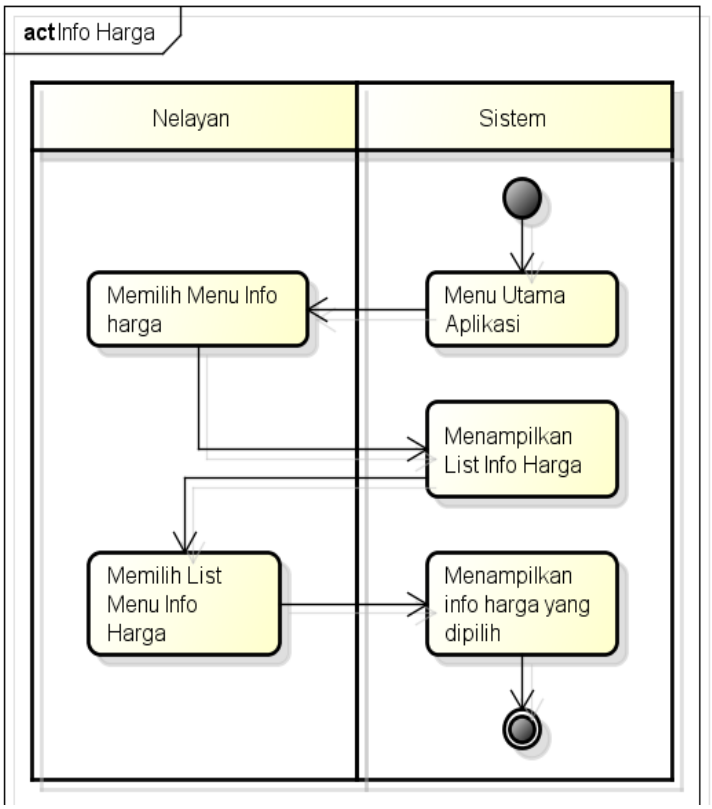
Activity diagram melihat cuaca memodelkan aliran kerja atau *workflow* dari urutan proses melihat cuaca pada aplikasi *front end.*



Gambar 3.10 *Activity* Diagram Melihat Cuaca

2. *Activity* Diagram Melihat Info Harga

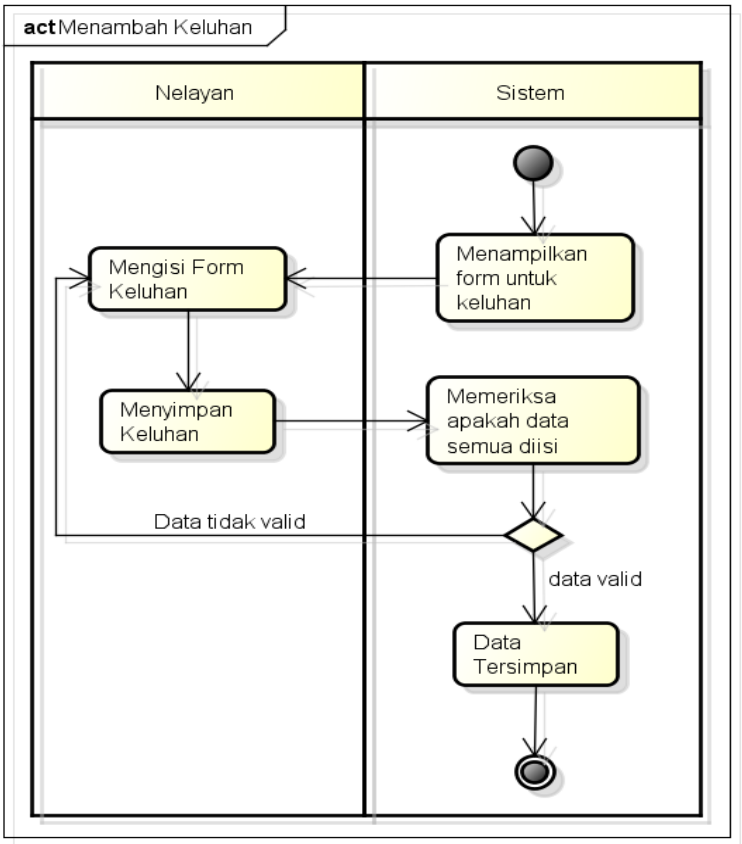
Activity diagram melihat info harga memodelkan aliran kerja atau *workflow* dari urutan proses melihat info hargapada aplikasi *front end*.



Gambar 3.11 *Activity* Diagram Melihat Info Harga

3. *Activity* Diagram Menambah Keluhan

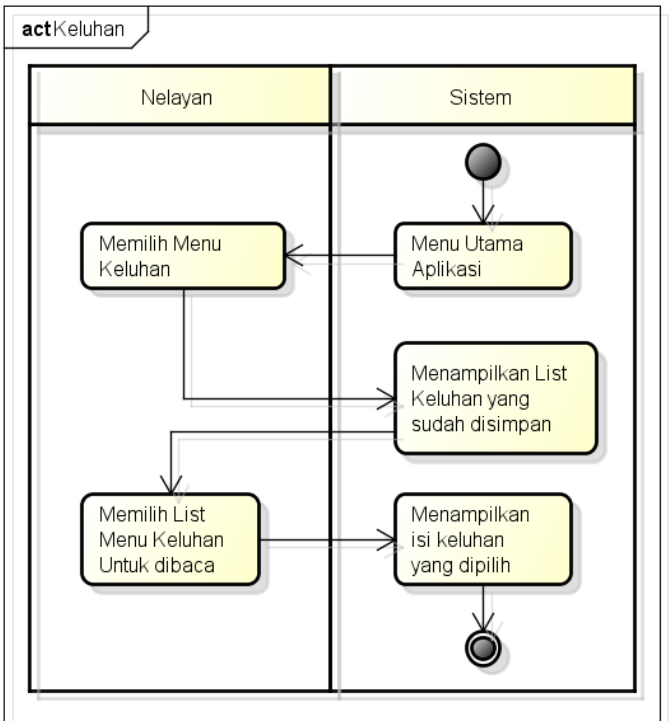
Activity diagram menambah keluhan memodelkan aliran kerja atau *workflow* dari urutan proses menambah keluhanpada aplikasi front end.



Gambar 3.12 *Activity* Diagram Menambah Keluhan

4. *Activity* Diagram Melihat Keluhan

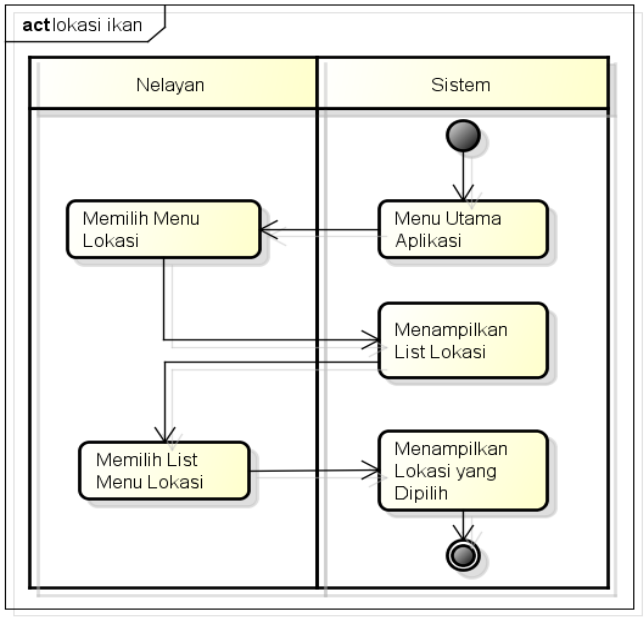
Activity diagram melihat keluhan memodelkan aliran kerja atau *workflow* dari urutan proses melihat keluhan pada aplikasi *front end*.



Gambar 3.13 *Activity* Diagram Melihat Keluhan

5. *Activity* Diagram Melihat Lokasi Ikan

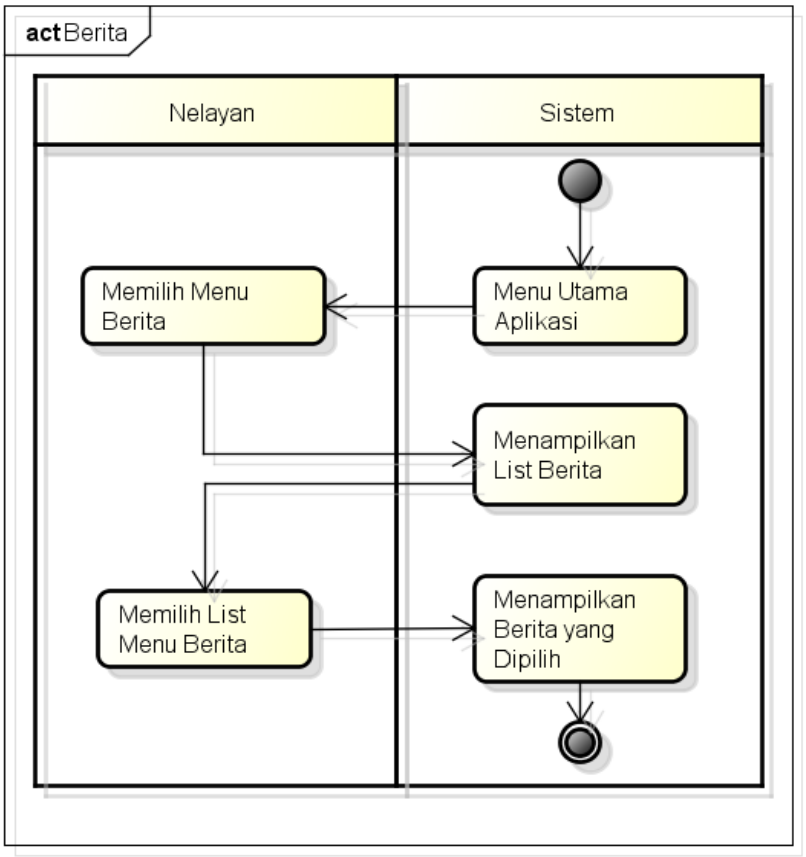
Activity diagram melihat lokasi ikan memodelkan aliran kerja atau *workflow* dari urutan proses melihat lokasi ikan pada aplikasi *front end*.



Gambar 3.14 *Activity* Diagram Melihat Lokasi Ikan

6. *Activity* Diagram Melihat Berita

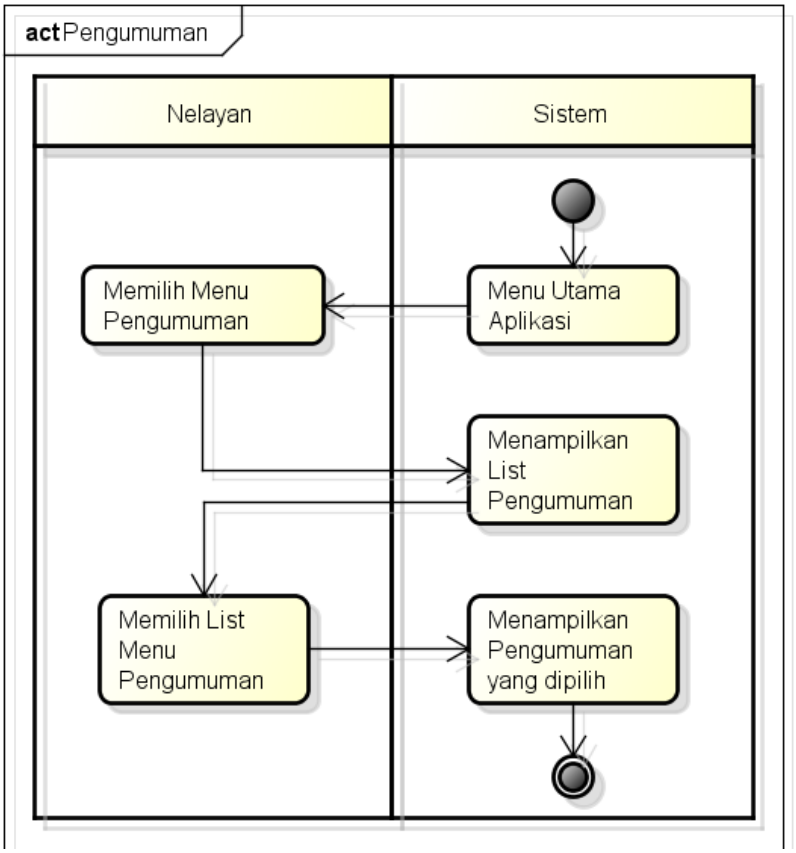
Activity diagram melihat berita memodelkan aliran kerja atau *workflow* dari urutan proses melihat berita pada aplikasi *front end.*



Gambar 3.15 *Activity* Diagram Melihat Berita

7. *Activity* Diagram Melihat Pengumuman

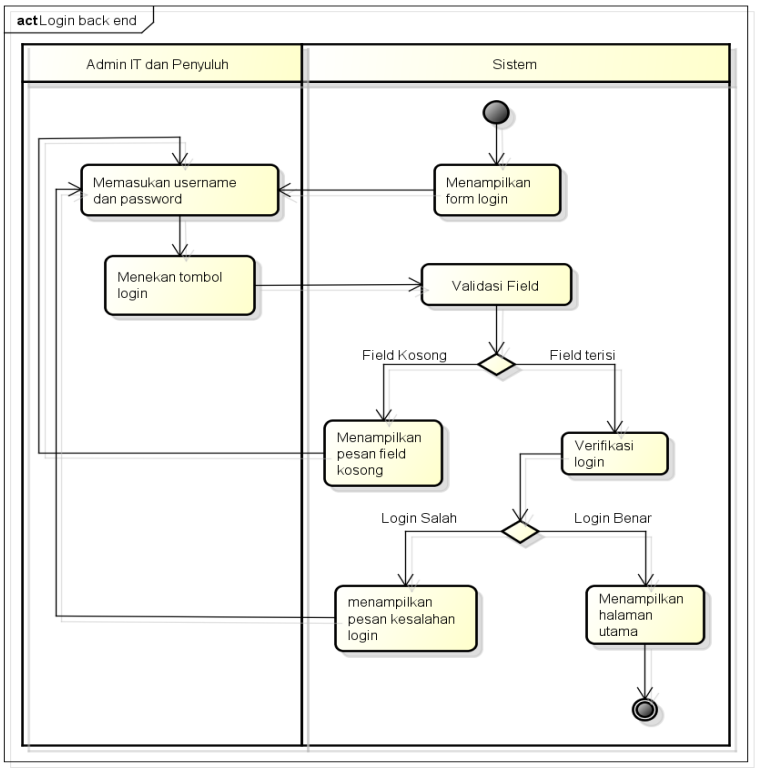
Activity diagram melihat pengumumanmemodelkan aliran kerja atau *workflow* dari urutan proses melihat pengumuman pada aplikasi *front end.*



Gambar 3.16 *Activity* Diagram Melihat Pengumuman

8. *Activity* Diagram Login Back End

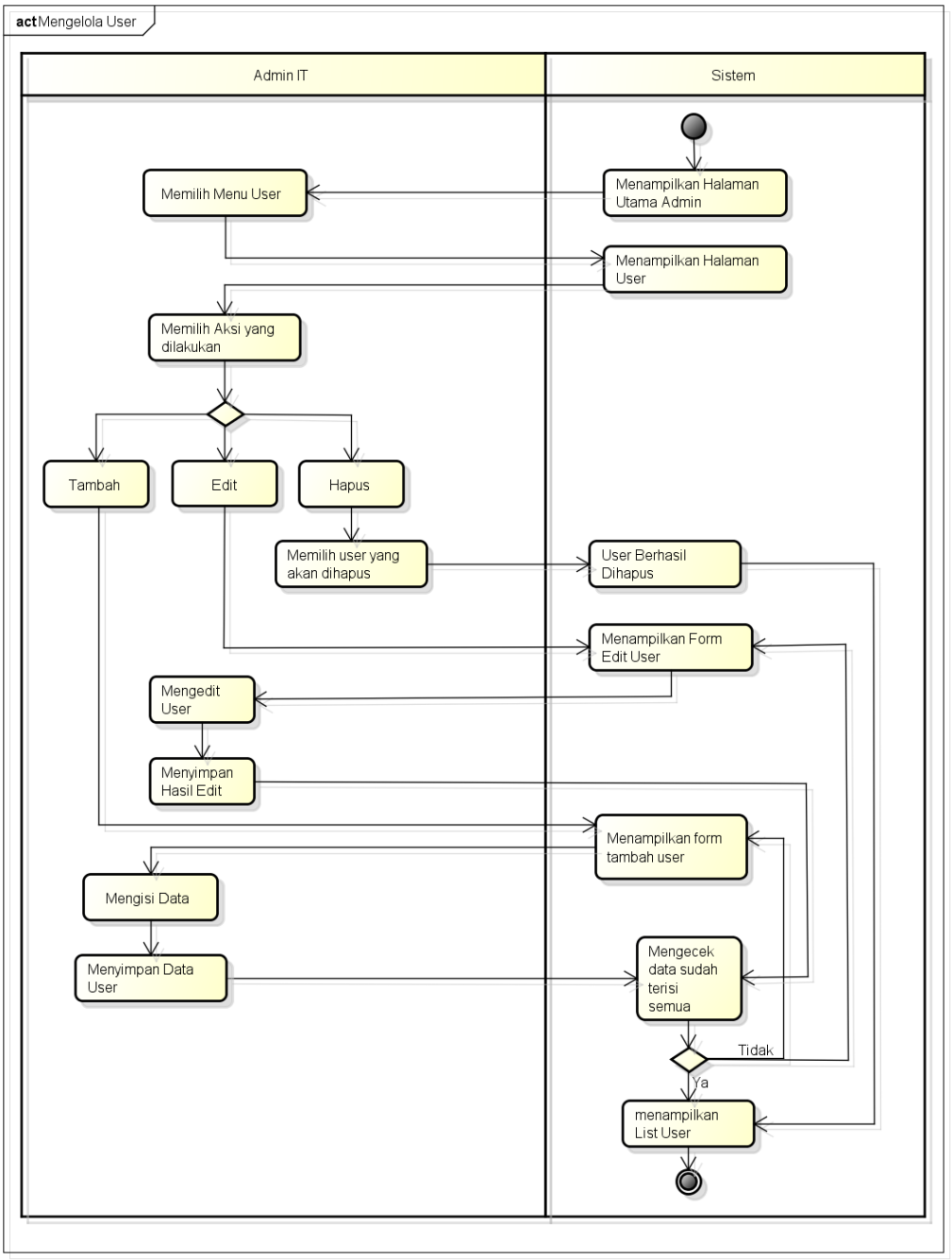
Activity diagram login *back end* memodelkan aliran kerja atau *workflow* dari urutan proses login pada aplikasi *back end.*



Gambar 3.17 *Activity* Diagram Login Back End

9. *Activity* Diagram Mengelola User

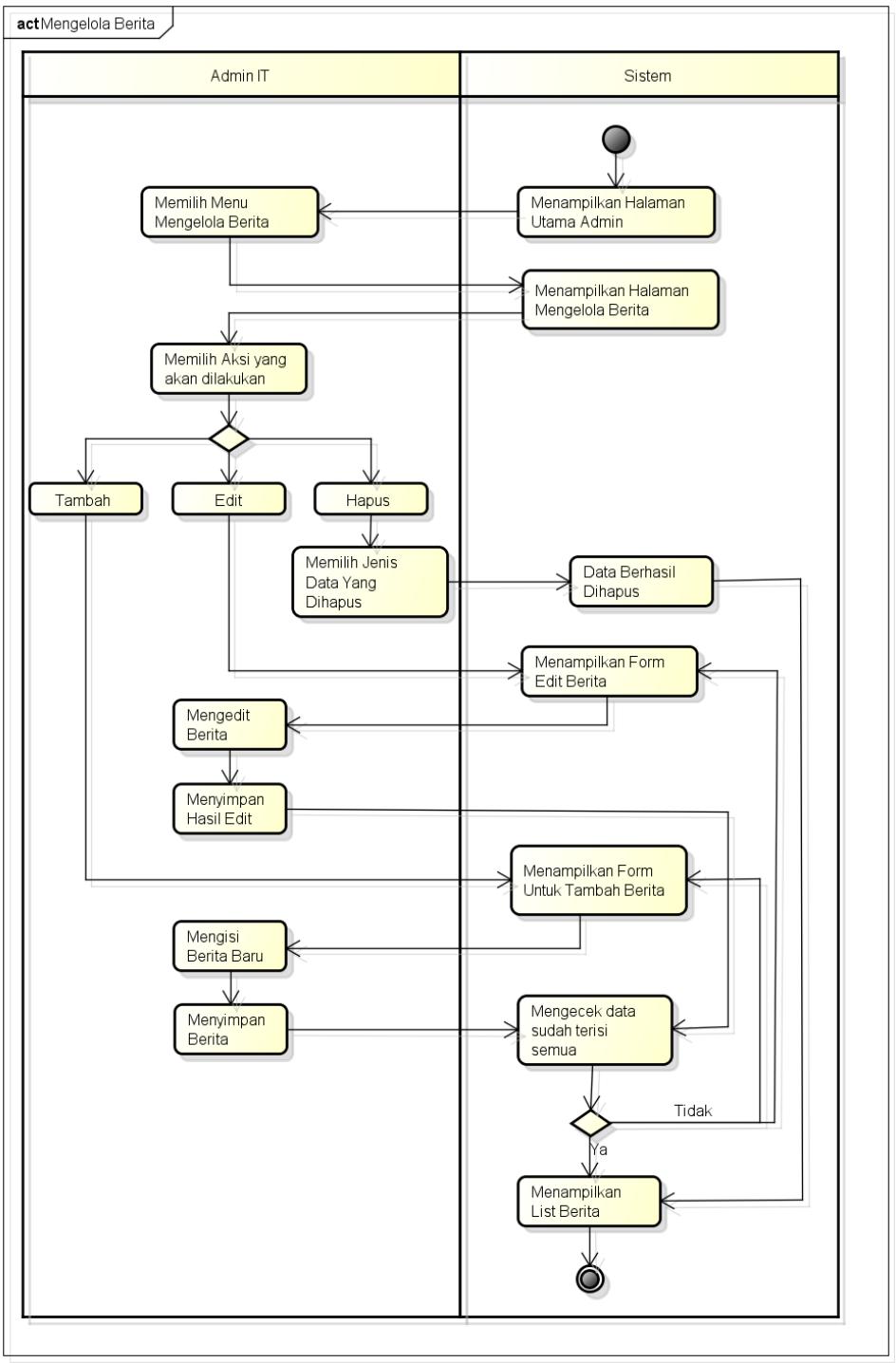
Activity diagram mengelola user memodelkan aliran kerja atau *workflow* dari urutan proses mengelola user pada aplikasi *back end*.



Gambar 3.18 *Activity* Diagram Mengelola User

10. *Activity* Diagram Mengelola Berita

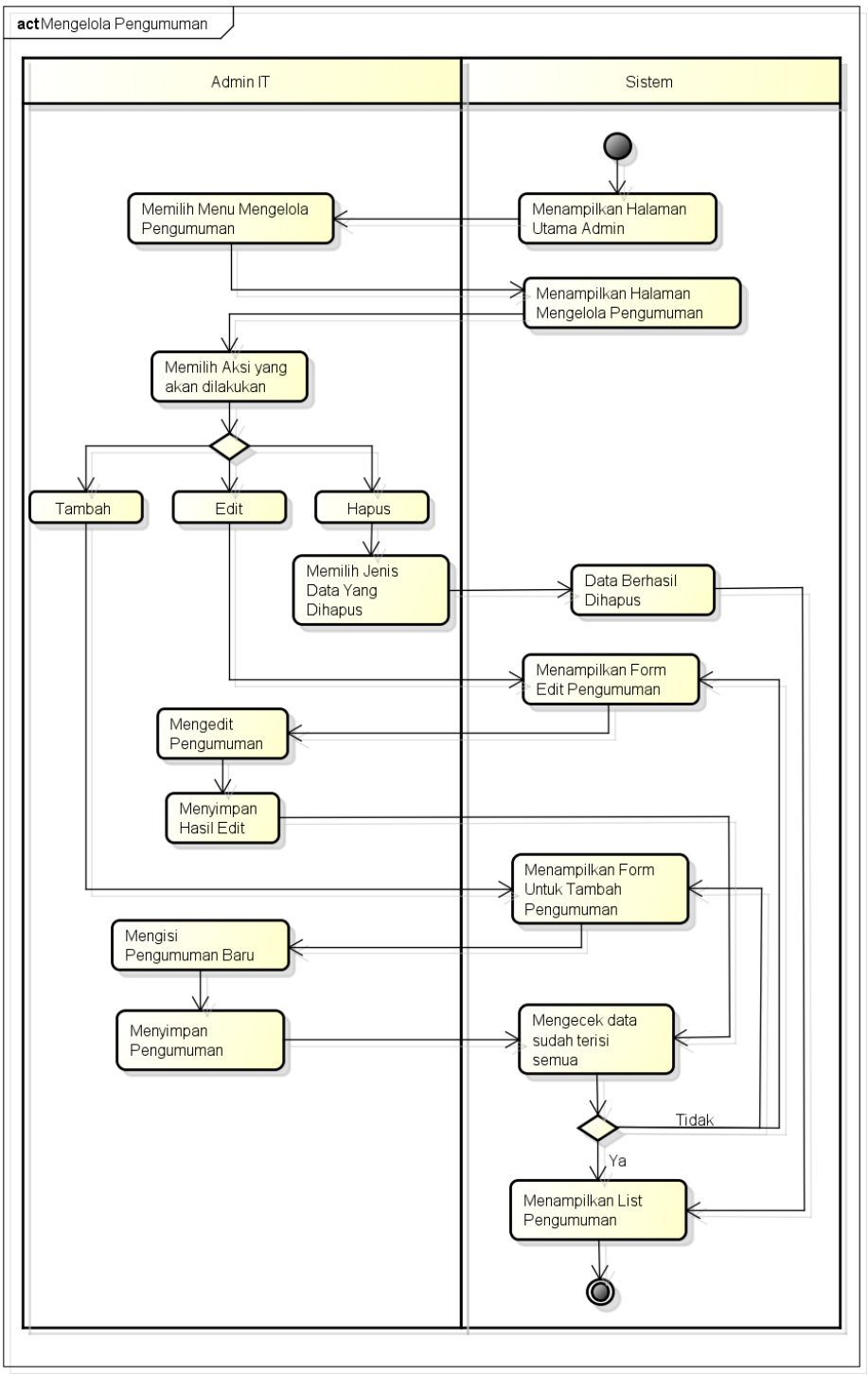
Activity diagram mengelola berita memodelkan aliran kerja atau *workflow* dari urutan proses mengelola berita.



Gambar 3.19 *Activity* Diagram Mengelola Berita

11. *Activity* Diagram Mengelola Pengumuman

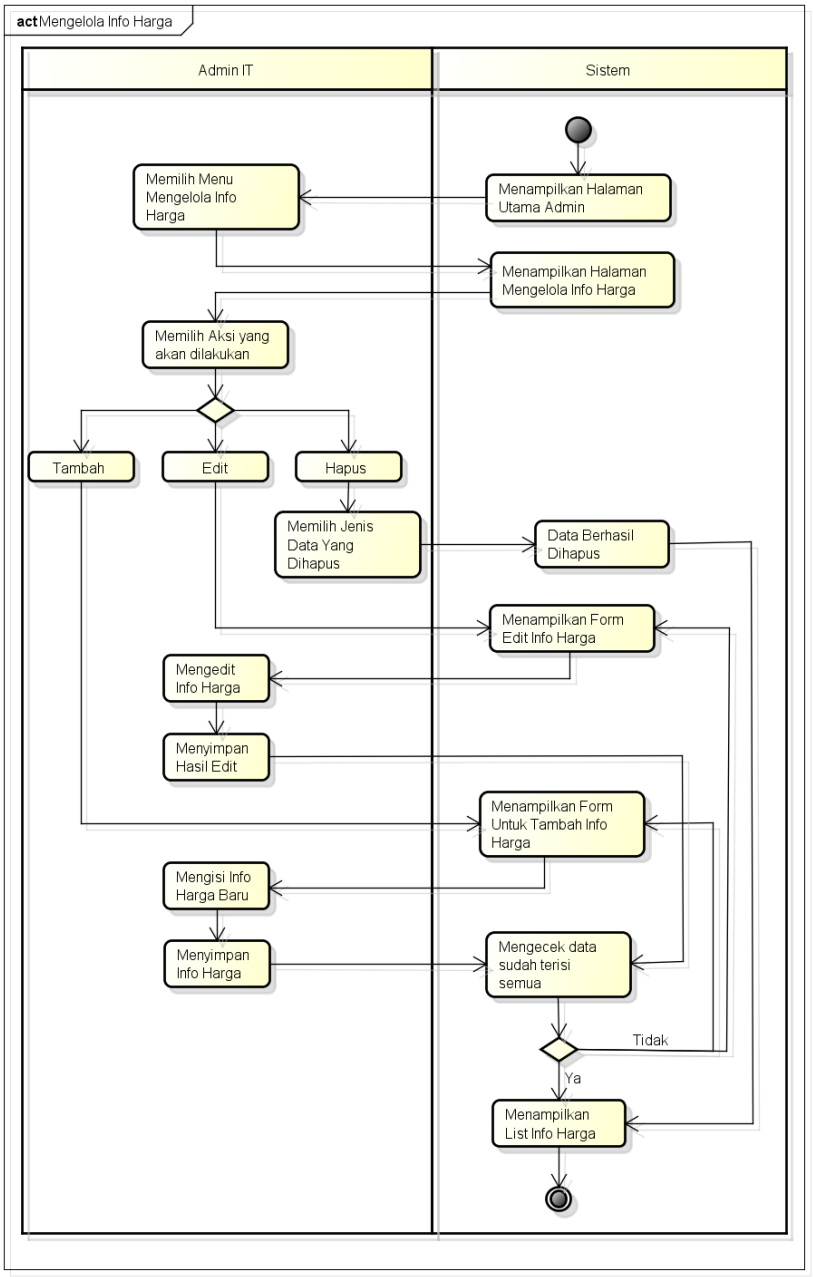
Activity diagram mengelola pengumuman memodelkan aliran kerja atau *workflow* dari urutan proses mengelola pengumuman.



Gambar 3.20 *Activity* Diagram Mengelola Pengumuman

12. *Activity* Diagram Mengelola Info Harga

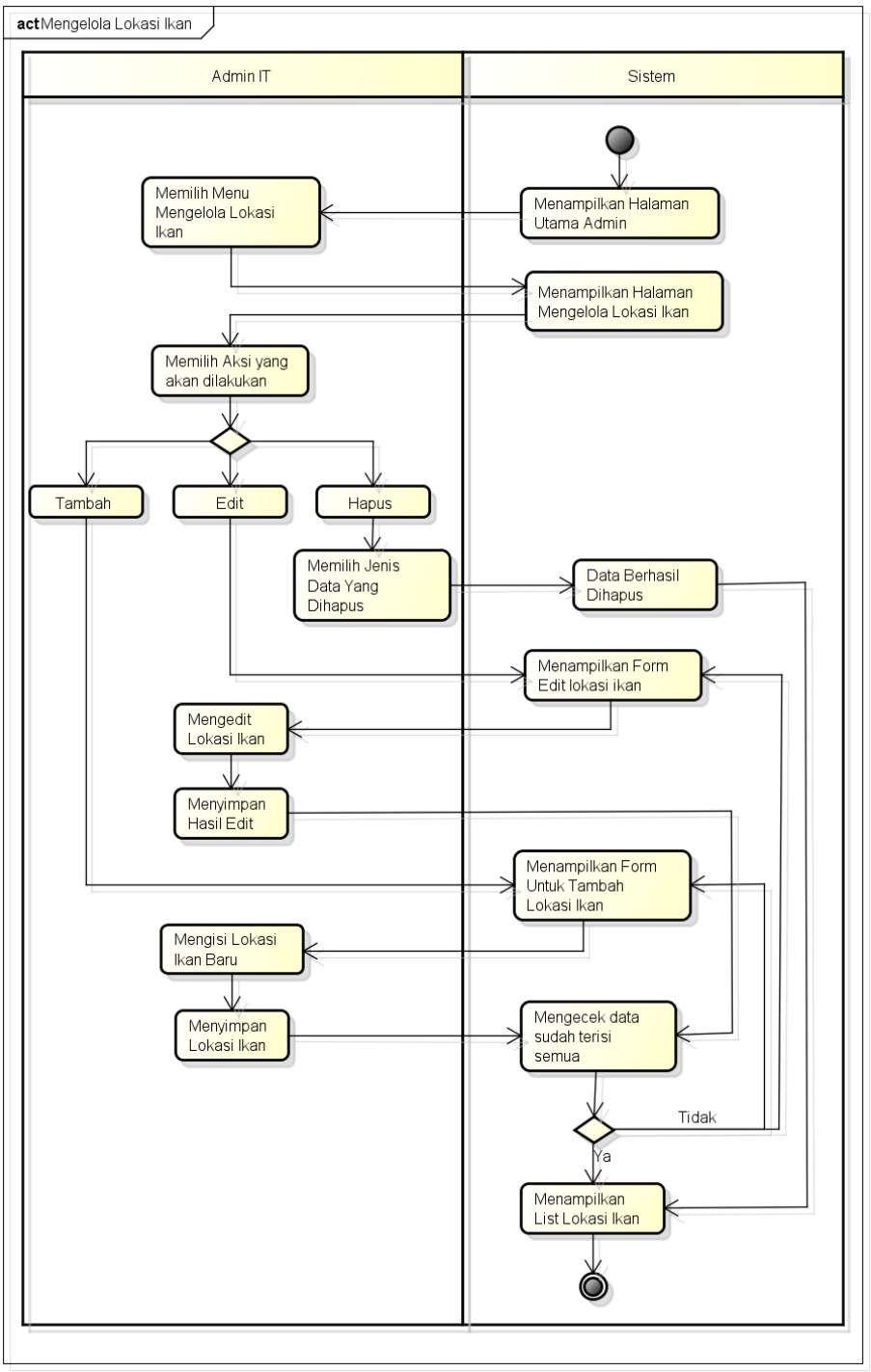
Activity diagram mengelola info harga memodelkan aliran kerja atau *workflow* dari urutan proses mengelola info harga.



Gambar 3.21 *Activity* Diagram Mengelola Info Harga

13. *Activity* Diagram Mengelola Lokasi Ikan

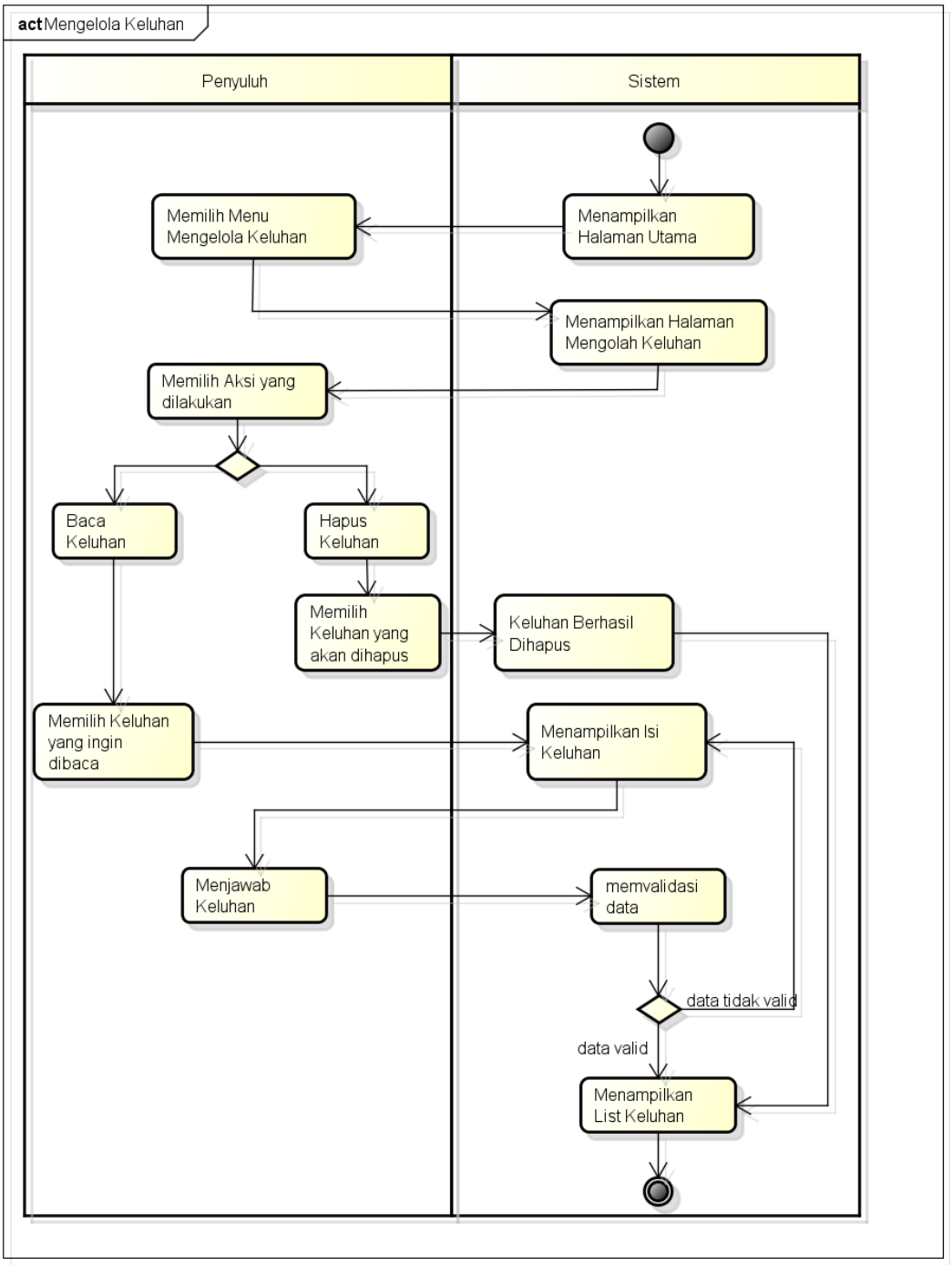
Activity diagram mengelola lokasi ikan memodelkan aliran kerja atau *workflow* dari urutan proses mengelola lokasi ikan.



Gambar 3.22 *Activity* Diagram Mengelola Lokasi Ikan

14. *Activity* Diagram Mengelola Keluhan

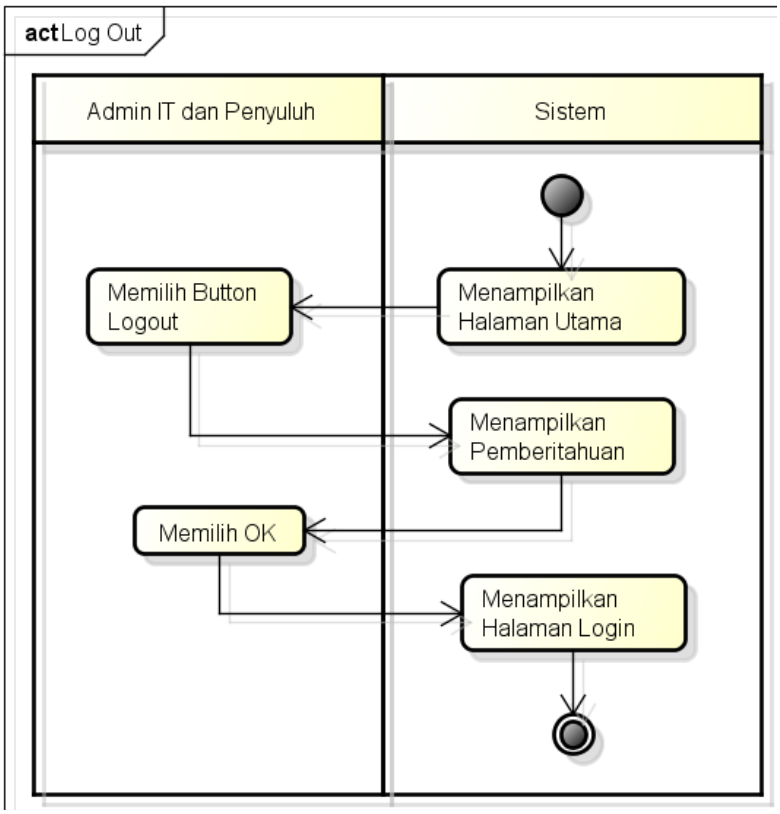
Activity diagram mengelola keluhan memodelkan aliran kerja atau *workflow* dari urutan proses mengelola keluhan.



Gambar 3.23 *Activity* Diagram Mengelola Keluhan

15. *Activity* Diagram Logout *Back End*

Activity diagram *logout back end* memodelkan aliran kerja atau *workflow* dari urutan proses *logout back end.*

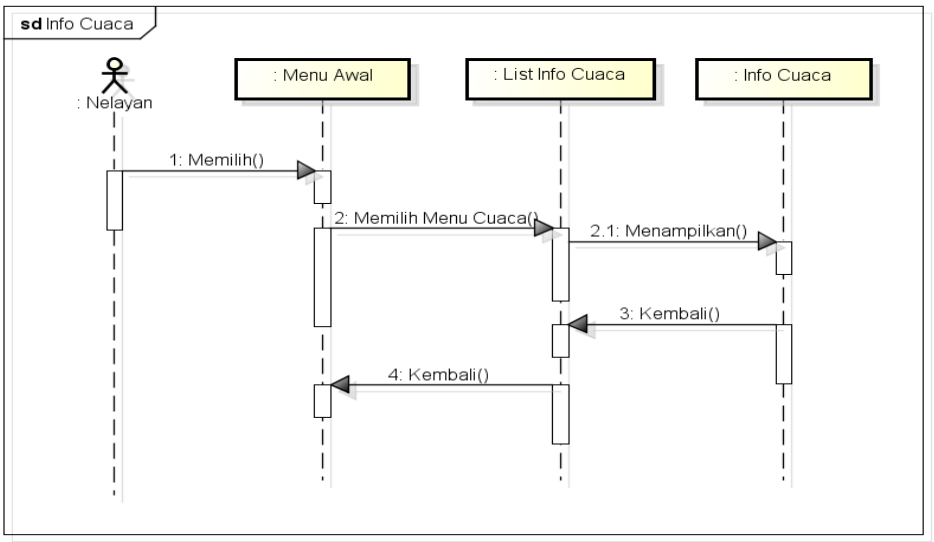


Gambar 3.24 *Activity* Diagram *Logout Back End*

3.1.5.5 *Sequence* Diagram

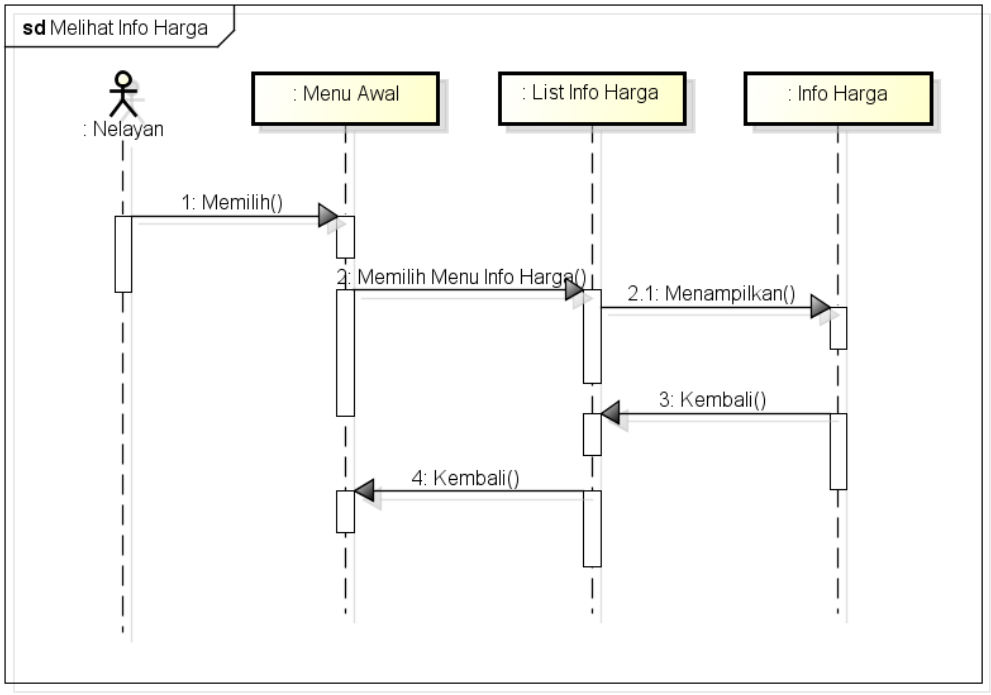
Sequence Diagram menjelaskan secara detail urutan proses yang dilakukan sistem juga menjelaskan interaksi yang terjadi antar objek *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

1. Sequence Diagram Melihat Cuaca



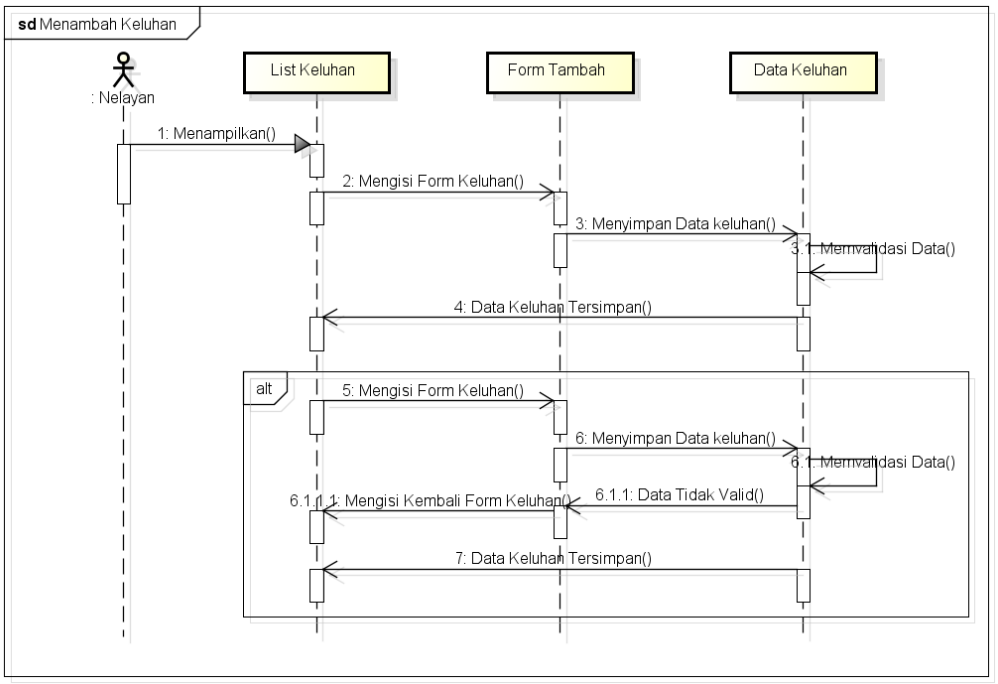
Gambar 3.25 *Sequence* Diagram Melihat Cuaca

2. Sequence Diagram Melihat Info Harga



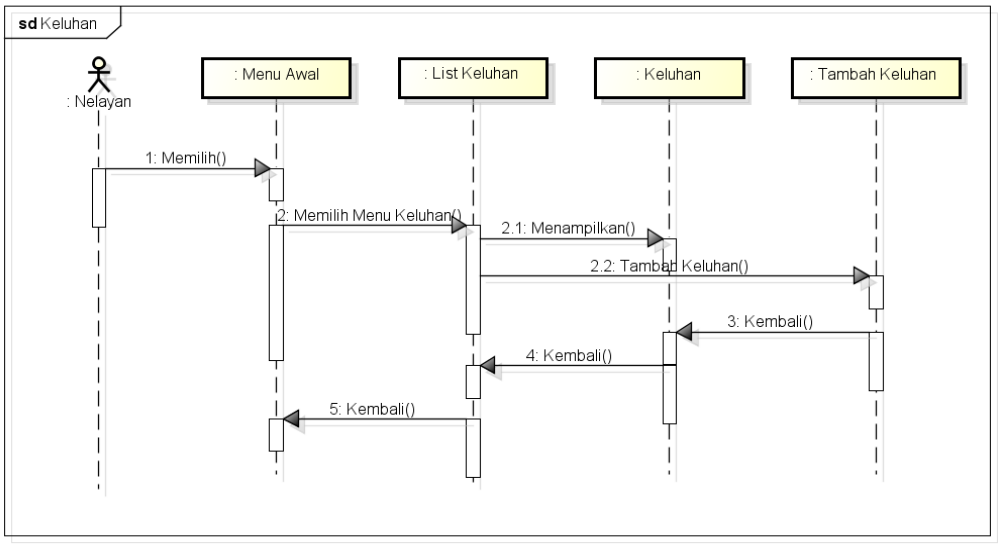
Gambar 3.26 *Sequence* Diagram Melihat Info Harga

3. Sequence Diagram Menambah Keluhan



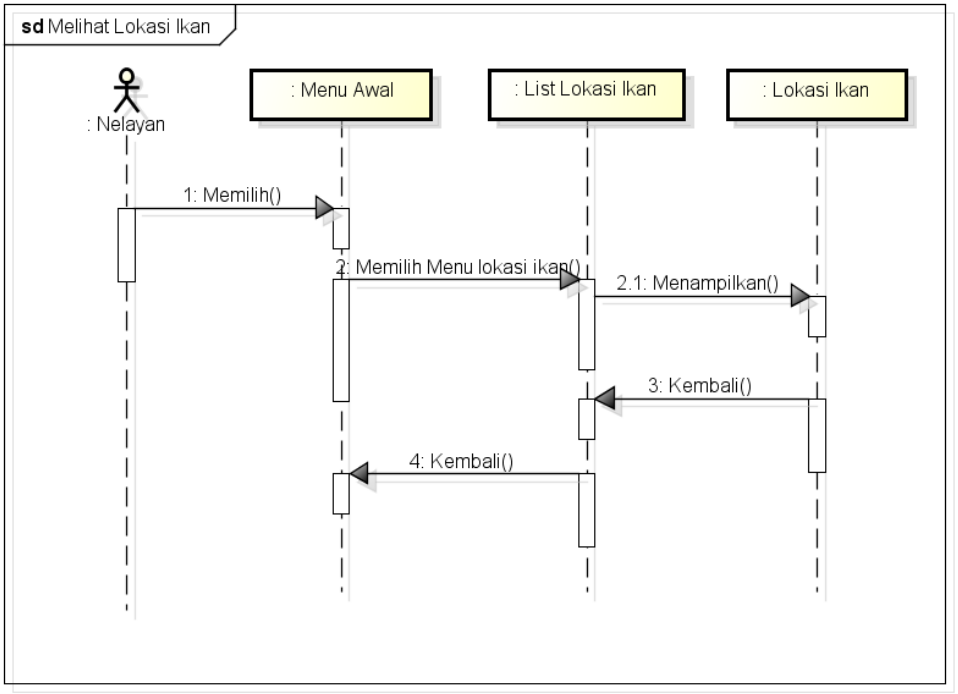
Gambar 3.27 *Sequence* Diagram Menambah Keluhan

4. Sequence Diagram Melihat Keluhan



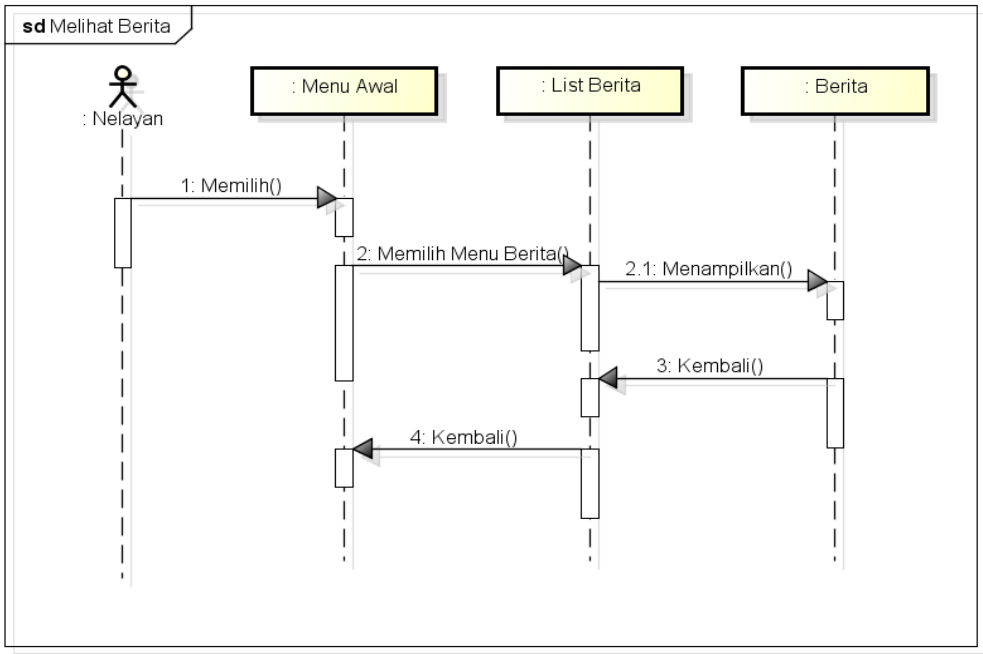
Gambar 3.28 *Sequence* Diagram Melihat Keluhan

5. Sequence Diagram Melihat Lokasi Ikan



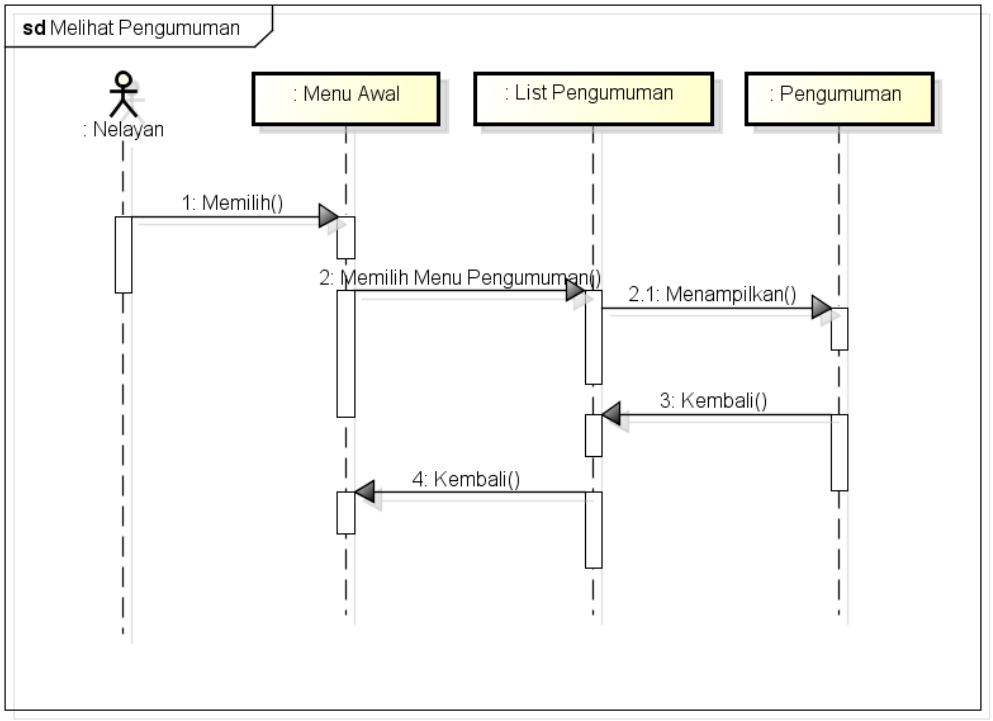
Gambar 3.29 *Sequence* Diagram Melihat Lokasi Ikan

6. Sequence Diagram Melihat Berita



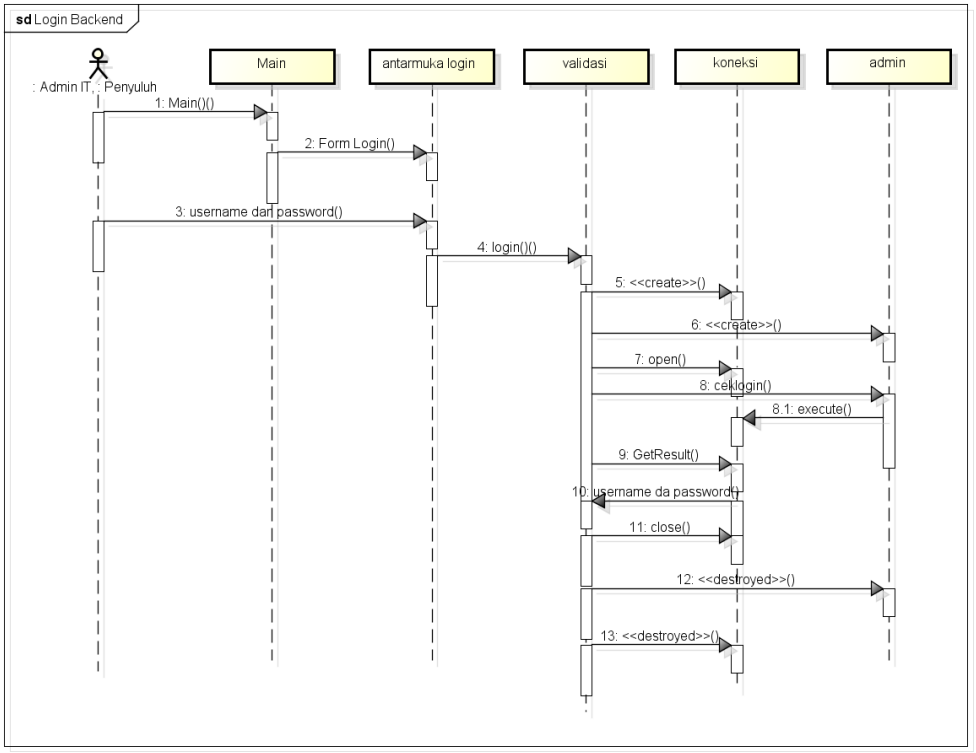
Gambar 3.30 *Sequence* Diagram Melihat Berita

7. Sequence Diagram Melihat Pengumuman



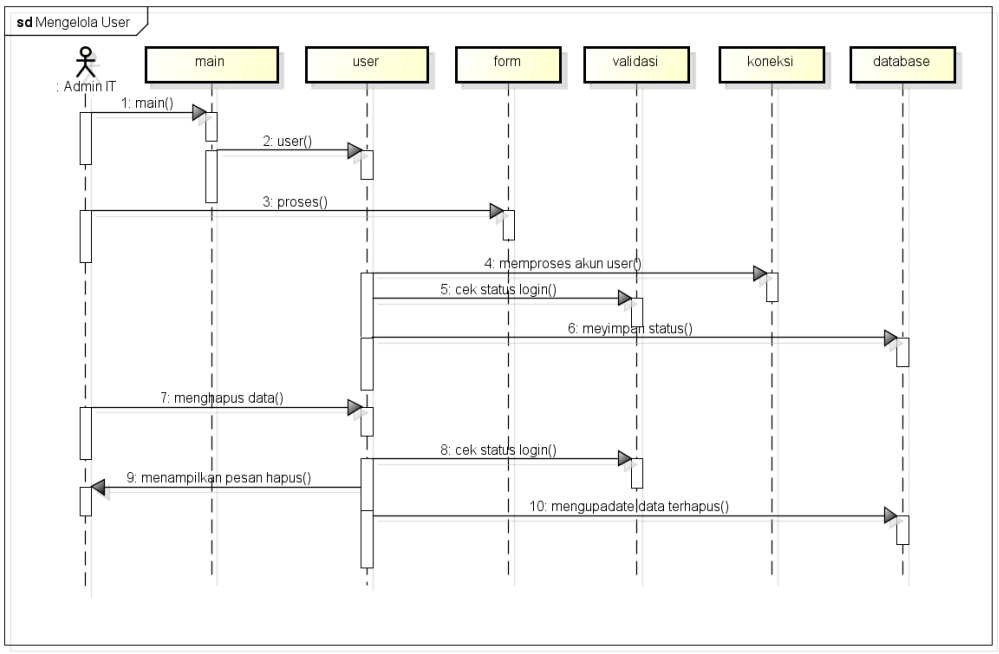
Gambar 3.31 *Sequence* Diagram Melihat Pengumuman

8. Sequence Diagram Login Back End



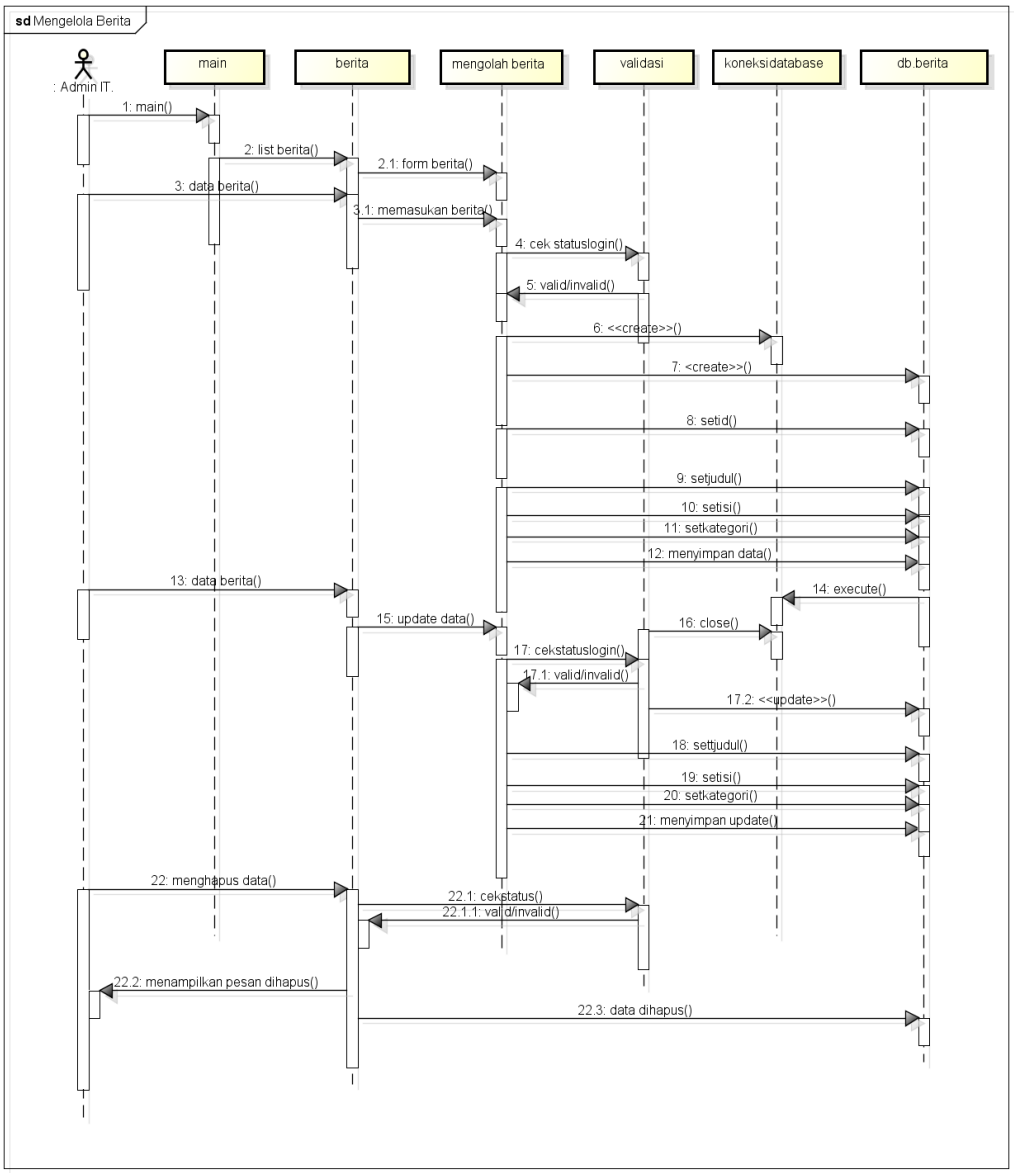
Gambar 3.32 *Sequence* Diagram *Login Back End*

9. Sequence Diagram Mengelola User



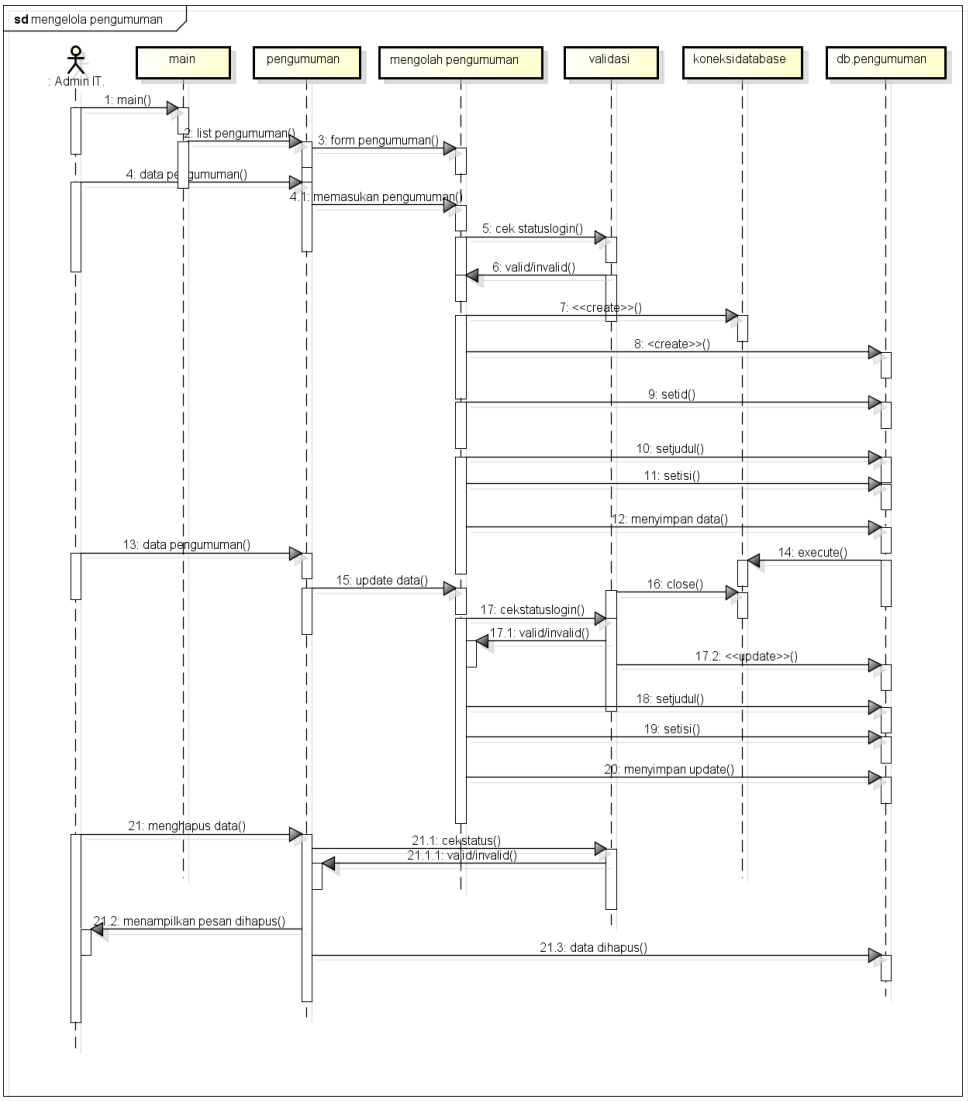
Gambar 3.33 *Sequence* Diagram Mengelola User

10. Sequence Diagram Mengelola Berita



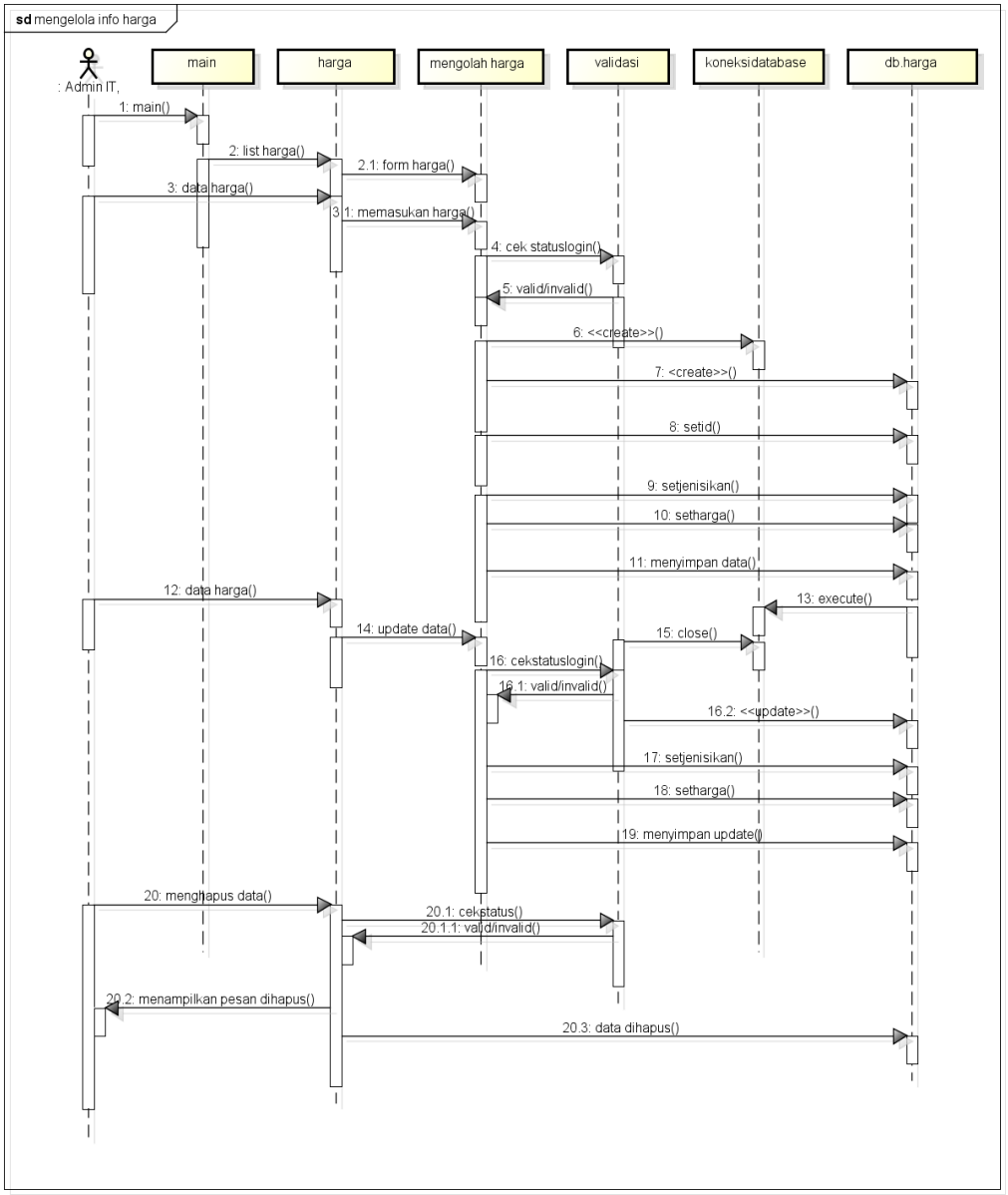
Gambar 3.34 *Sequence* Diagram Mengelola Berita

11. Sequence Diagram Mengelola Pengumuman



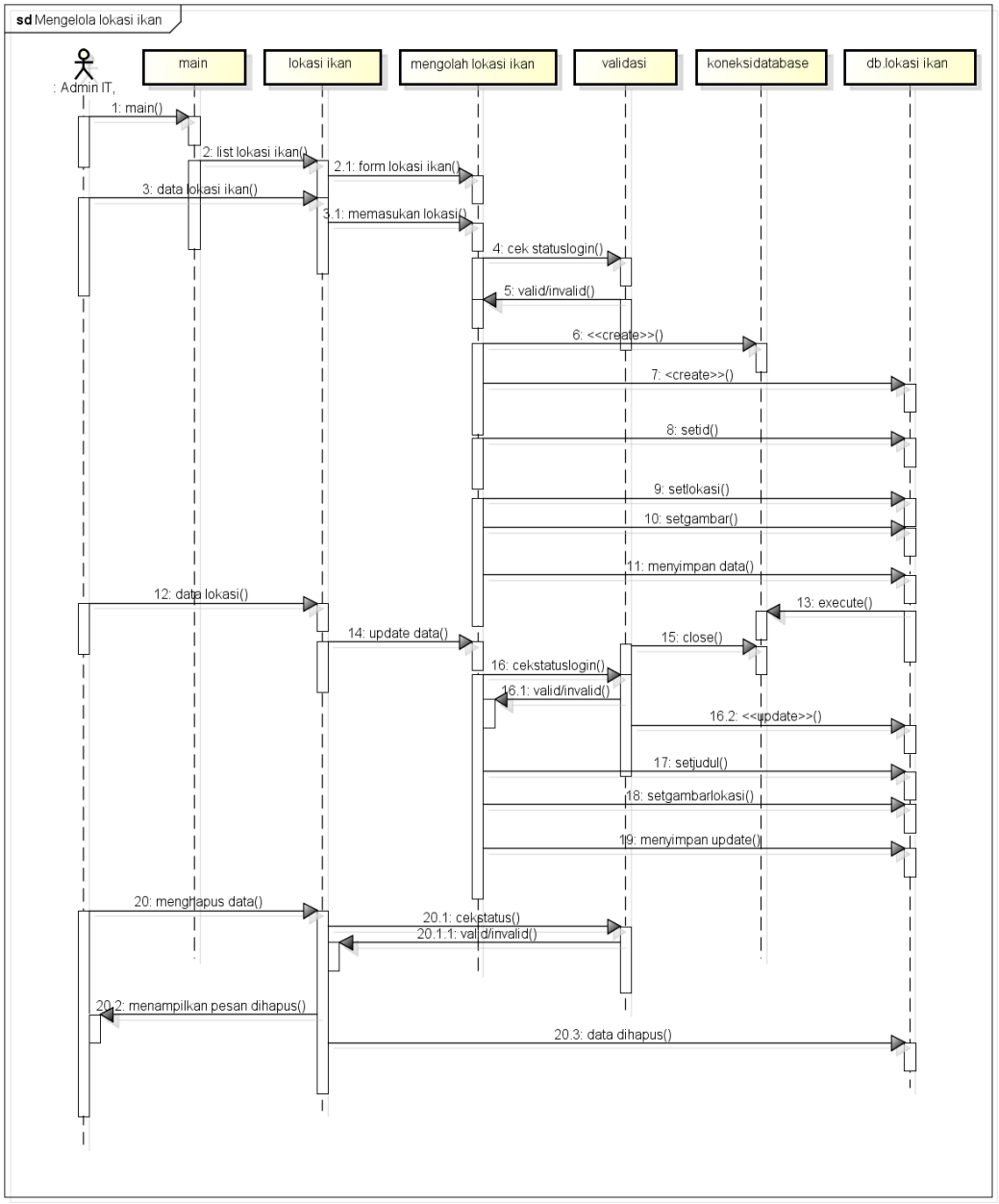
Gambar 3.35 *Sequence* Diagram Mengelola Pengumuman

12. Sequence Diagram Mengelola Info Harga



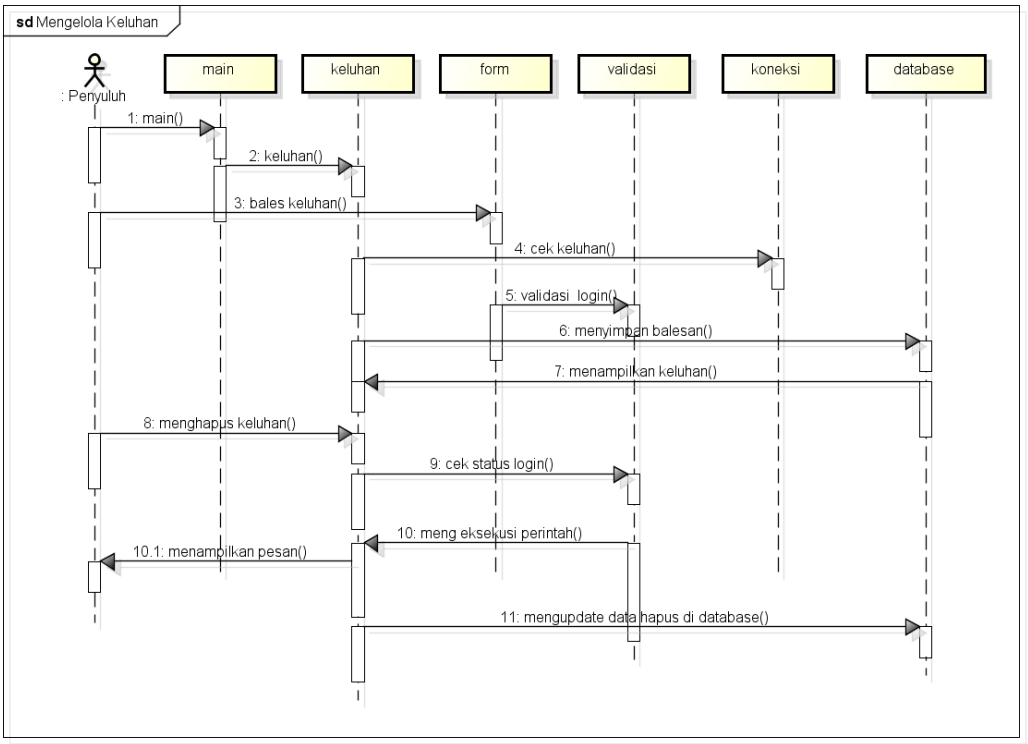
Gambar 3.36 *Sequence* Diagram Mengelola Info Harga

13. Sequence Diagram Mengelola Lokasi Ikan



Gambar 3.37 *Sequence* Diagram Mengelola Lokasi Ikan

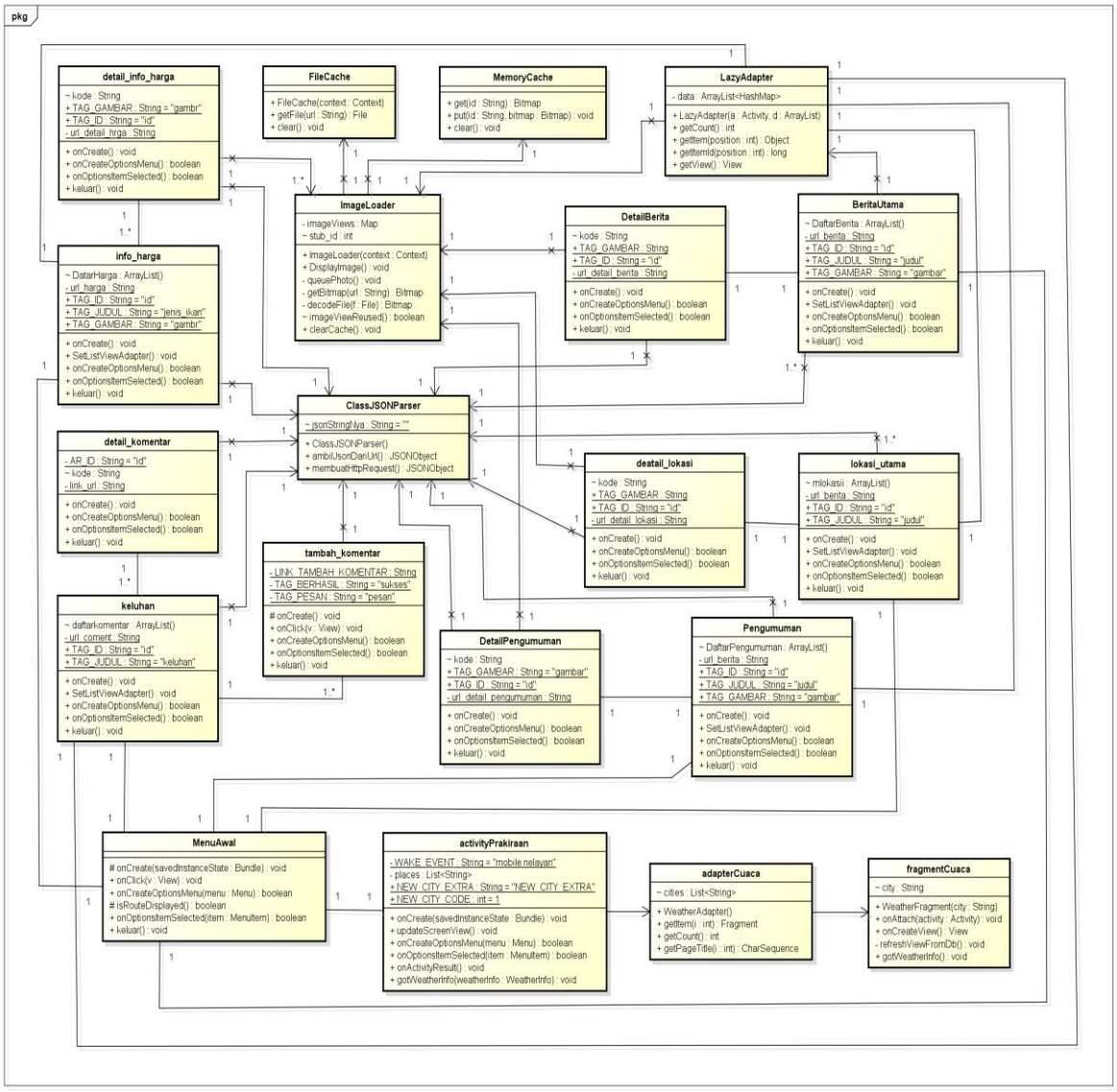
14. Sequence Diagram Mengelola Keluhan



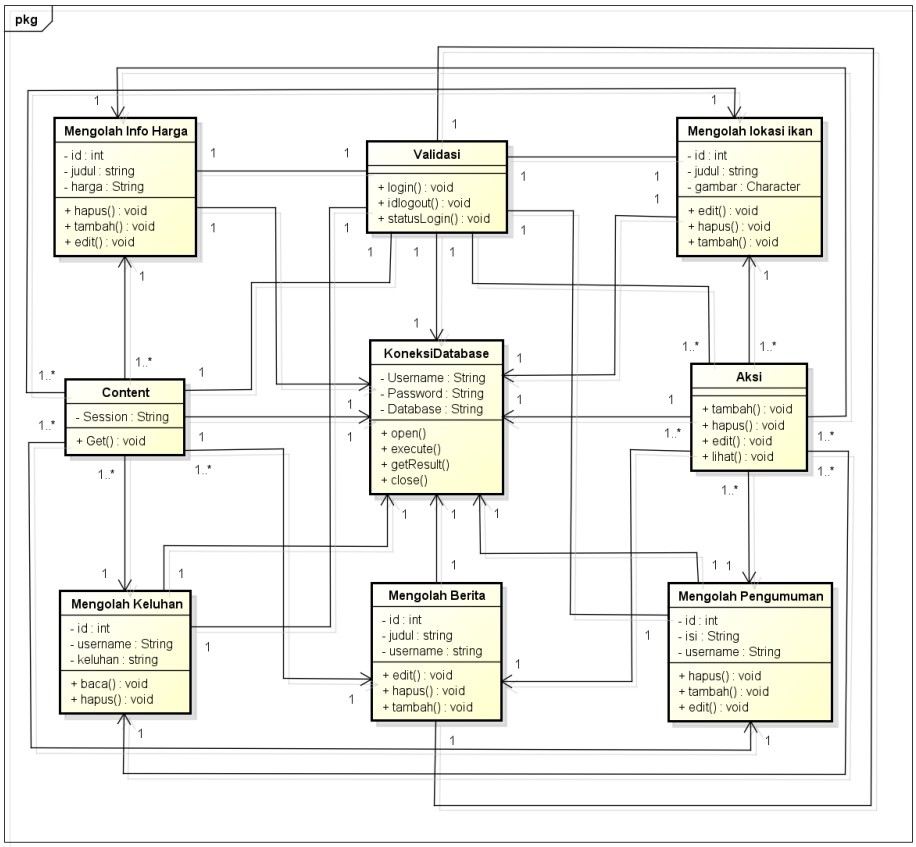
Gambar 3.38 *Sequence* Diagram Mengelola Keluhan

3.1.5.6 *Class Diagram*

Class diagram adalah kumpulan objek-objek dan mempunyai struktur umum, behavior umum, relasi dan semantik yang umum. Class-class ditentukan dengan cara memeriksa objek-objek dan kolaborasi diagram. *Class* diagram digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam sistem.



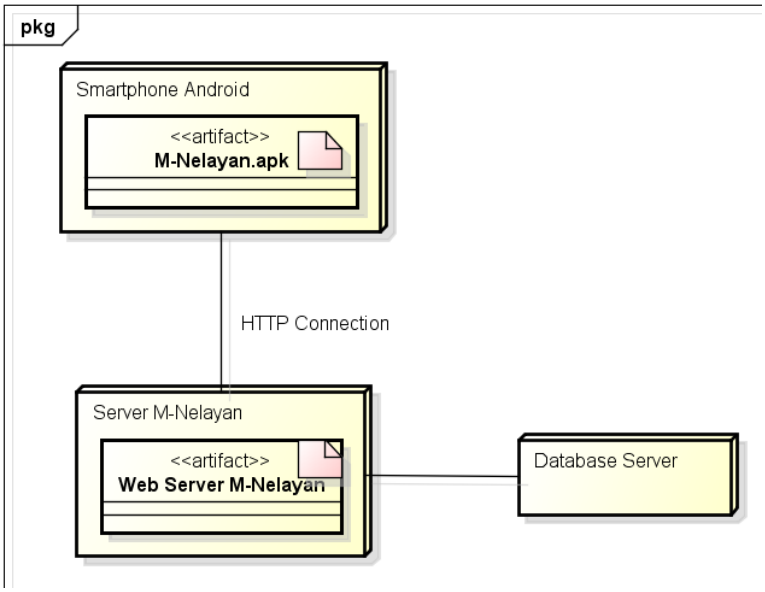
Gambar 3.39 *Class* Diagram M-Nelayan bagian *Front End*



Gambar 3.40 *Class* Diagram M-Nelayan bagian *Back End*

3.1.5.7 *Deployment Diagram*

Deployment diagram menggambarkan bagaimana infrastruktur sistem dari aplikasi M-Nelayan yang terdiri dari *Smartphone* Android, *server*, dan basis data dapat terhubung. *Deployment* diagram menggambarkan secara rinci bagaimana komponen di-*deploy* dalam infrastruktur sistem. Berikut ini *deployment* diagram dari sistem aplikasi M-Nelayan.



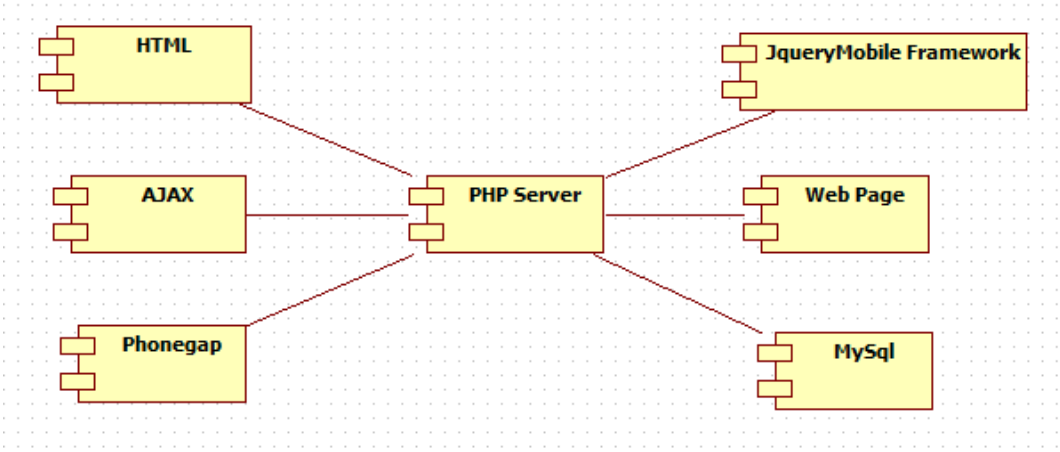
Gambar 3.41 Deployment Diagram

Deployment diagram sistem aplikasi M-Nelayan terdiri dari tiga tipe *Node*, yaitu *client* yang diwakili dengan *smartphone* Android, *sever* M-Nelayan, dan basis data. Berikut ini penjelasan masing-masing *node*:

1. *Node smartphone* Android, merupakan perangkat (*handphone)* tempat aplikasi M-Nelayan berada. *Node smartphone* Android berkomunikasi dengan server menggunakan jalur internet (HTTP Connections).
2. Node server, merupakan tempat web server yang melayani segala bentuk permintaan *client* seperti *request* data berita kelautan dan data komunikasi.
3. *Node* basis data, merupakan tempat penyimpanan data.

3.1.5.8 *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti perangkat lunak. Komponen piranti lunak yang dimaksud adalah modul yang berisi *source code* atau *binary code*, yang memiliki *library* atau *executable* yang muncul pada *compile time*, *link time*, ataupun *runtime.*

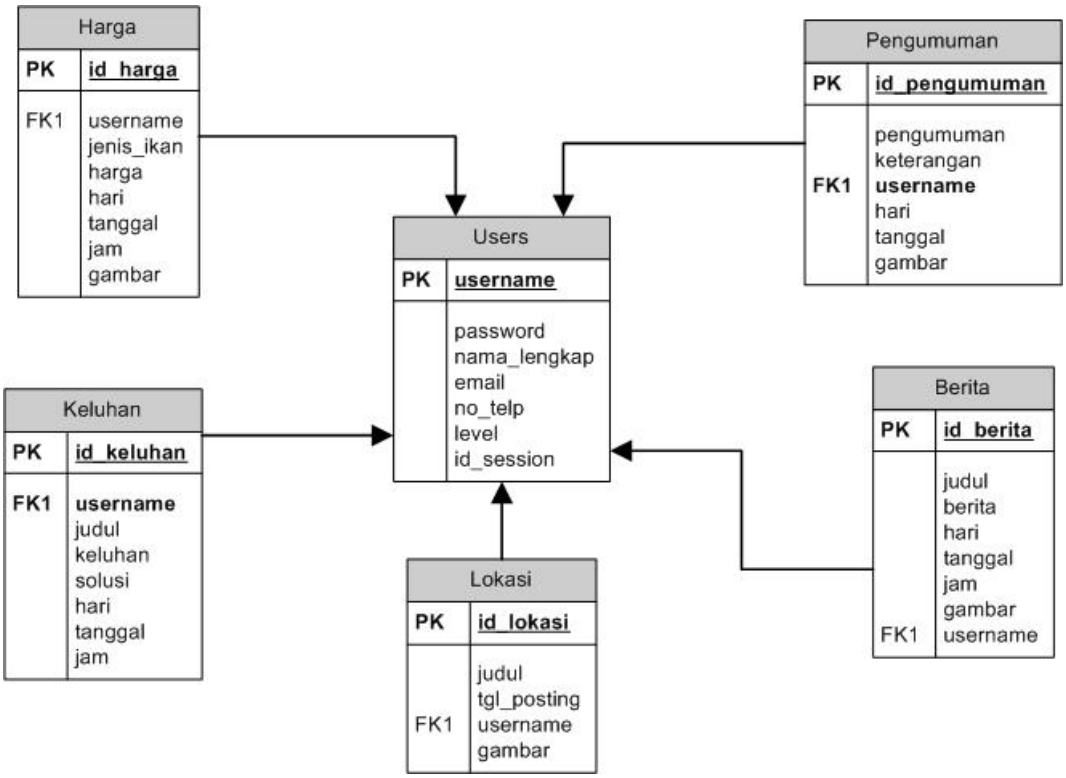


Gambar 3.42 *Component* Diagram

3.2 Perancangan

3.2.1 Perancangan Data

3.2.1.1 Skema Relasi



Gambar 3.43 Skema Relasi

3.2.1.2 Struktur Tabel

Perancangan struktur tabel adalah perancangan tabel-tabel yang akan digunakan pada database. Tebel-tabel yang terdapat dalam database yang digunakan dalam sistem aplikasi M-Nelayan adalah:

Tabel 3.26 Perancangan Struktur Tabel Harga

Nama *Field*	Tipe	Panjang	Kunci
Id_harga	*int*	5	*Primary key*
Username	*varchar*	30	-
jenis_ikan	*varchar*	30	-
Harga	*varchar*	30	-
Hari	*varchar*	25	-
Tanggal	*date*	-	-
Jam	*time*	-	-
Gambar	*varchar*	100	-

Tabel 3.27 Perancangan Struktur Tabel User

Nama *Field*	Tipe	Panjang	Kunci
Username	*Varchar*	50	*Primary key*
Password	*Varchar*	50	-
Nama_lengkap	*Varchar*	50	-
Email	*Varchar*	40	-
No_telp	*Varchar*	20	-
Level	*varchar*	20	-

Tabel 3.28 Perancangan Struktur Tabel Pengumuman

Nama *Field*	Tipe	Panjang	Kunci
Id_pengumuman	*int*	5	*Primary key*
Pengumuman	*Varchar*	40	-
Keterangan	*Text*	-	-
Username	*Varchar*	30	-
Hari	*Varchar*	30	-
Tanggal	*Date*	-	-

Jam	*Time*	-	-
Gambar	*varchar*	100	-

Tabel 3.29 Perancangan Struktur Tabel Keluhan

Nama *Field*	Tipe	Panjang	Kunci
Id_keluhan	*Int*	5	*Primary key*
Username	*Varchar*	30	-
Judul	*Varchar*	100	-
Keluhan	*Text*	-	-
Solusi	*Text*	-	-
Hari	*Varchar*	20	-
Tanggal	*Date*	-	-
Jam	*time*	-	-

Tabel 3.30 Perancangan Struktur Tabel Lokasi

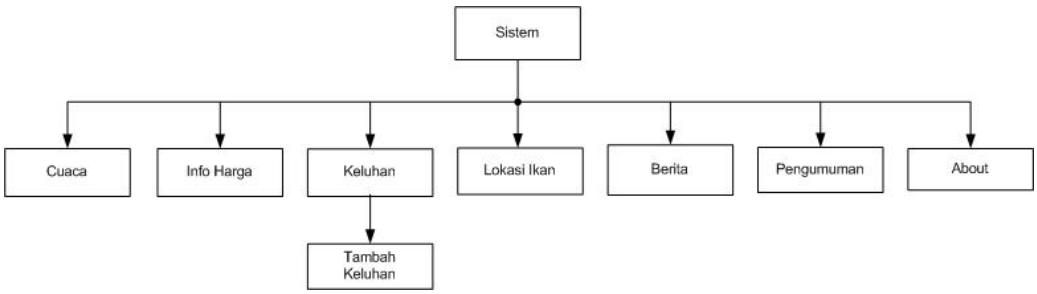
Nama *Field*	Tipe	Panjang	Kunci
Id_lokasi	*int*	10	*Primary key*
Judul	*varchar*	100	-
Tgl_posting	*date*	-	-
Gambar	*varchar*	100	-
Username	*varchar*	35	-

Tabel 3.31 Perancangan Struktur Tabel Berita

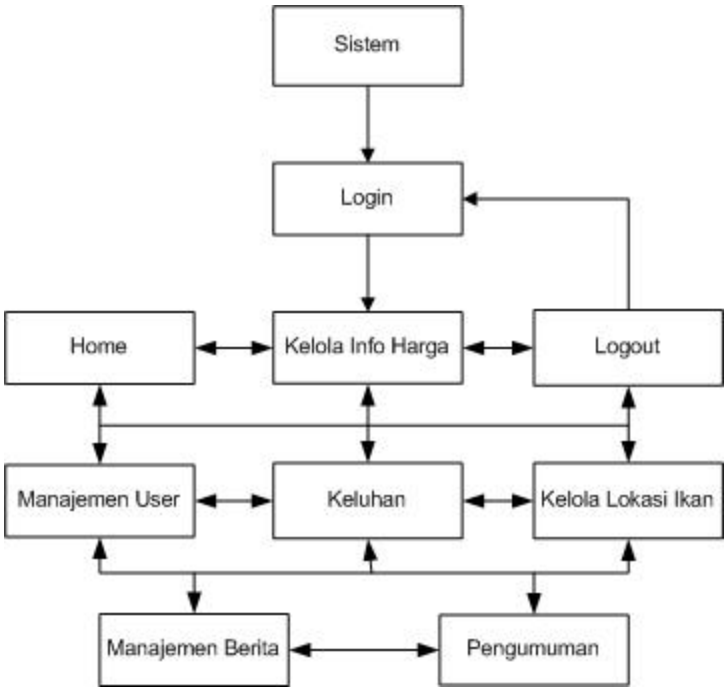
Nama *Field*	Tipe	Panjang	Kunci
Id_berita	*int*	10	*Primary key*
Username	*varchar*	30	-
Judul	*varchar*	100	-
Berita	*text*	-	-
Hari	*Varchar*	20	-
Tanggal	*Date*	-	-
Jam	*Time*	-	-
Gambar	*varchar*	100	-

3.2.2 Perancangan Menu

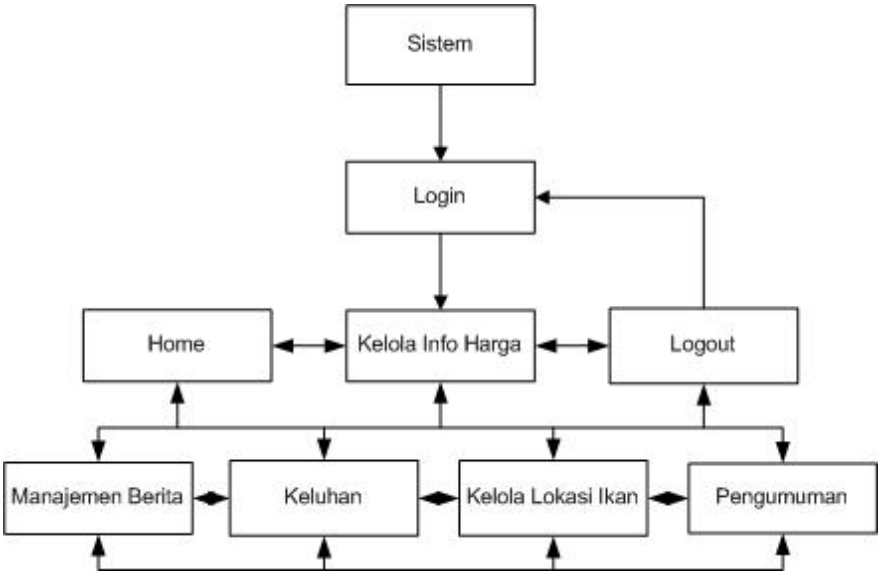
Perancangan menu menjelaskan struktur menu yang akan dibuat pada suatu sistem.



Gambar 3.44 Perancangan Menu untuk Nelayan



Gambar 3.45 Perancangan Menu Untuk Admin Web

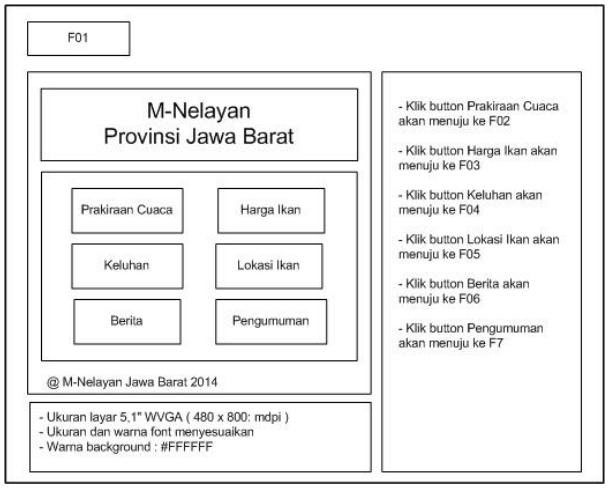


Gambar 3.46 Perancangan Menu Untuk Penyuluh

3.2.3 Perancangan Antarmuka

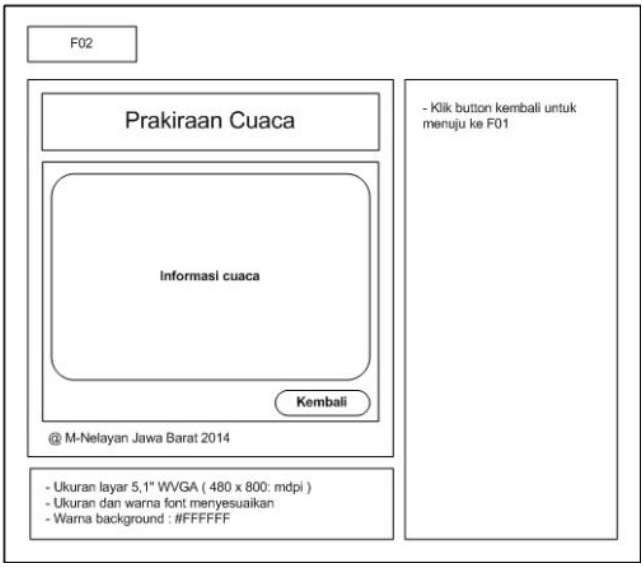
3.2.3.1 Perancangan Antarmuka Aplikasi *Front End*

1. Perancangan Antarmuka Halaman Menu Utama



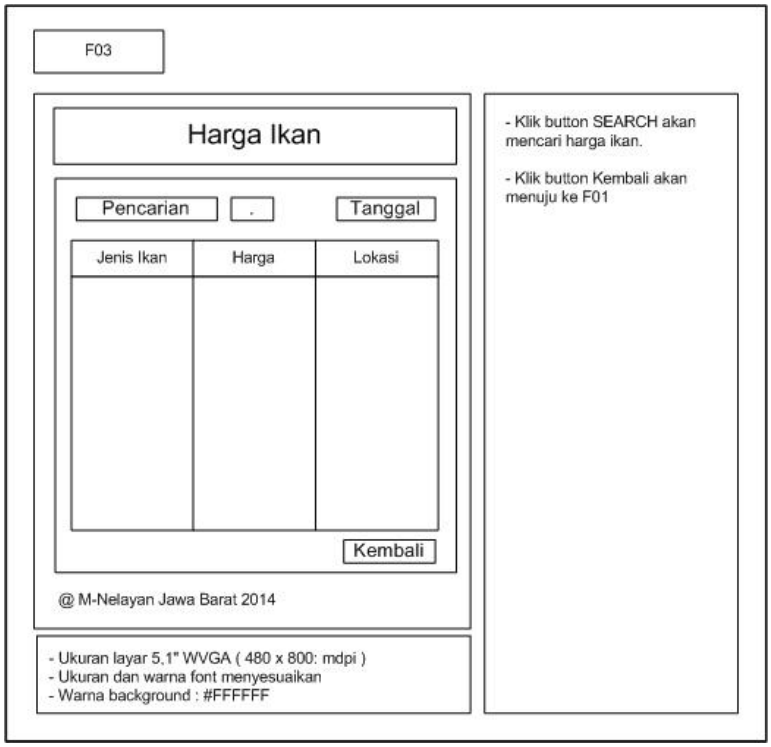
Gambar 3.47 Perancangan Antarmuka Halaman Menu Utama

2. Perancangan Antarmuka Halaman Prakiraan Cuaca



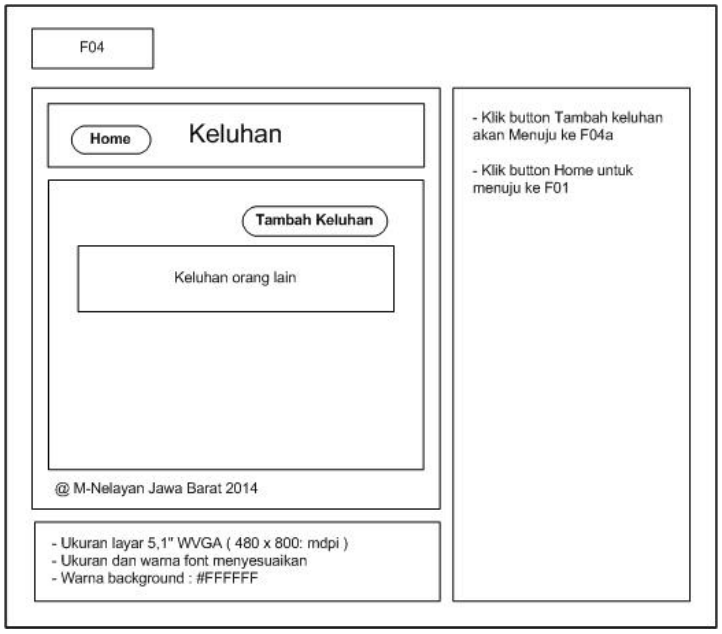
Gambar 3.48 Perancangan Antarmuka Halaman Prakiraan Cuaca

3. Perancangan Antarmuka Halaman Info Harga Ikan



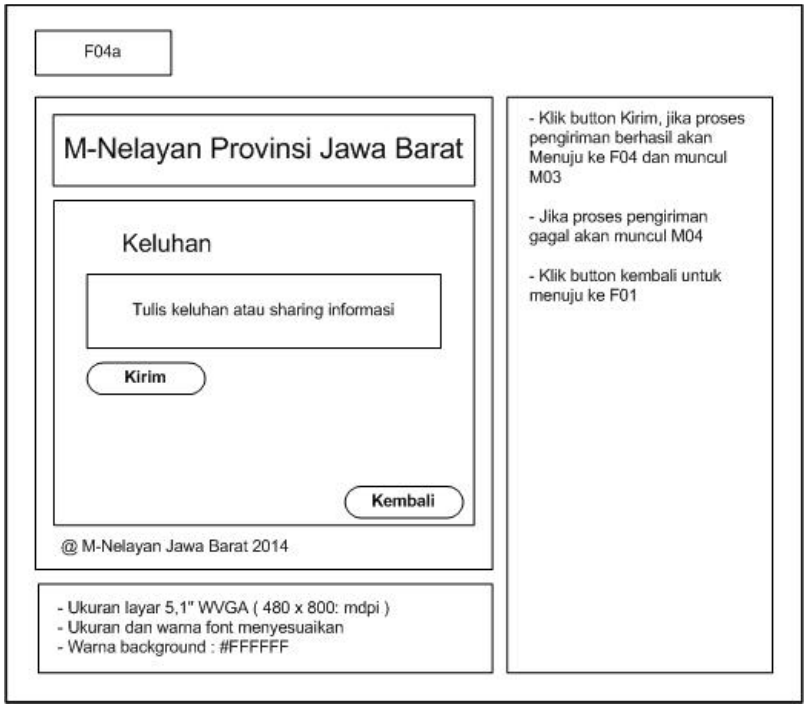
Gambar 3.49 Perancangan Antarmuka Halaman Info Harga Ikan

4. Perancangan Antarmuka Halaman Keluhan



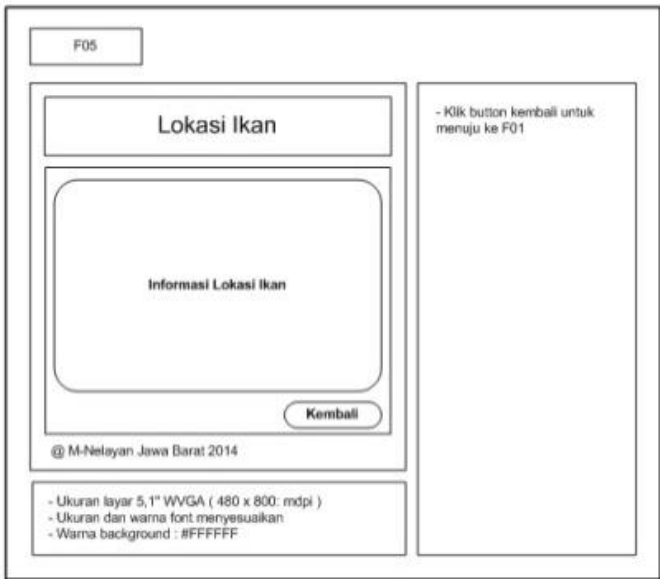
Gambar 3.50 Perancangan Antarmuka Halaman Keluhan

5. Perancangan Antarmuka Halaman Tambah Keluhan



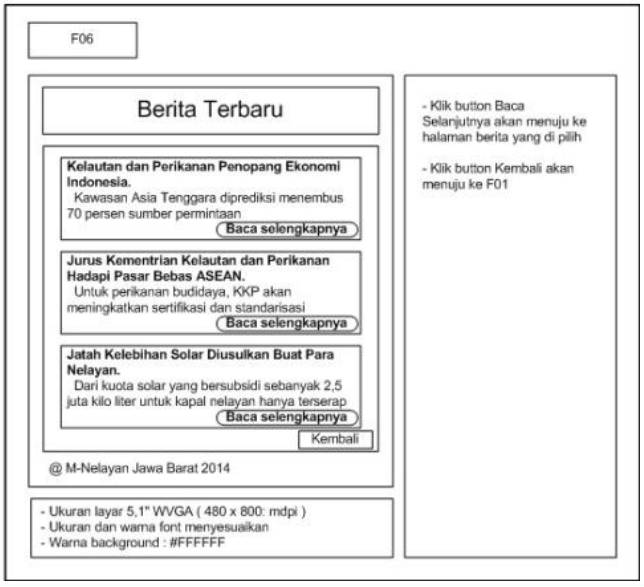
Gambar 3.51 Perancangan Antarmuka Halaman Tambah Keluhan

6. Perancangan Antarmuka Halaman Lokasi Ikan



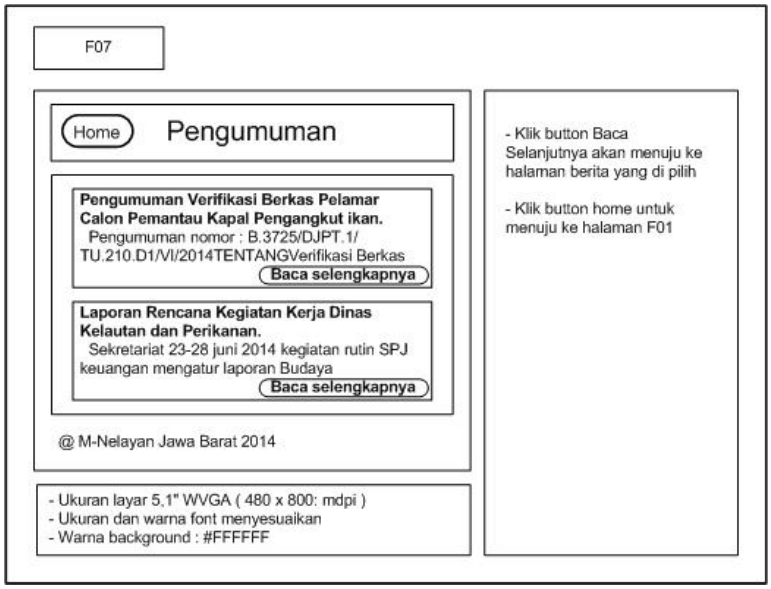
Gambar 3.52 Perancangan Antarmuka Halaman Lokasi Ikan

7. Perancangan Antarmuka Halaman Berita Terbaru



Gambar 3.53 Perancangan Antarmuka Halaman Berita Terbaru

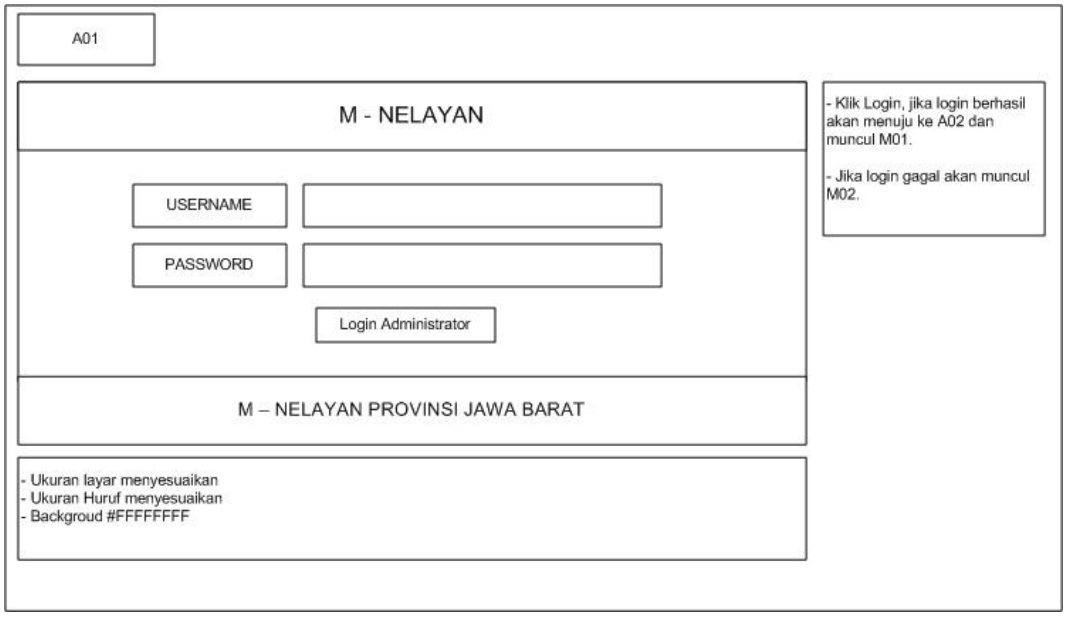
8. Perancangan Antarmuka Halaman Pengumuman



Gambar 3.54 Perancangan Antarmuka Halaman Pengumuman

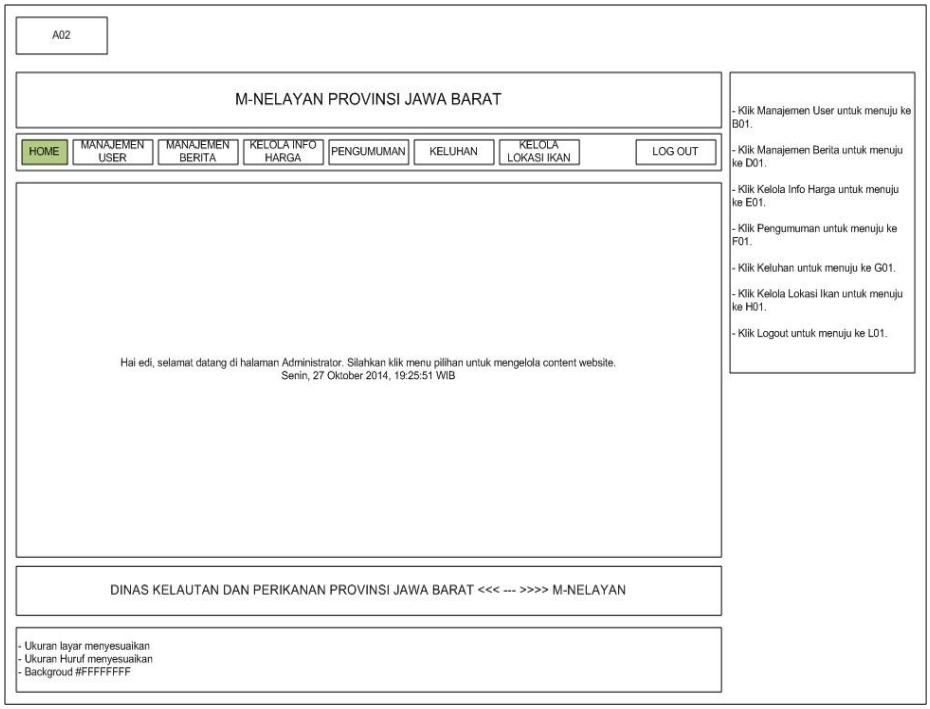
3.2.3.2 *Perancangan Antarmuka Aplikasi Back End*

1. Perancangan Antarmuka Halaman Login Admin



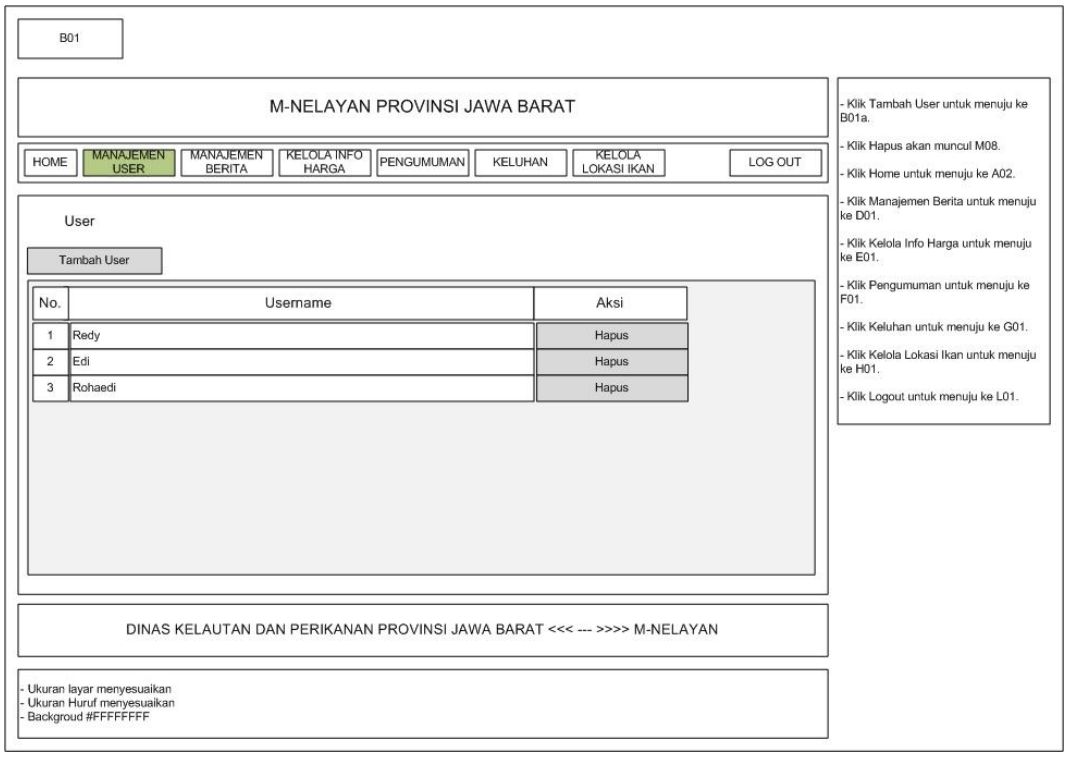
Gambar 3.55 Perancangan Antarmuka Halaman Login Admin

2. Perancangan Antarmuka Halaman Menu Utama Back End



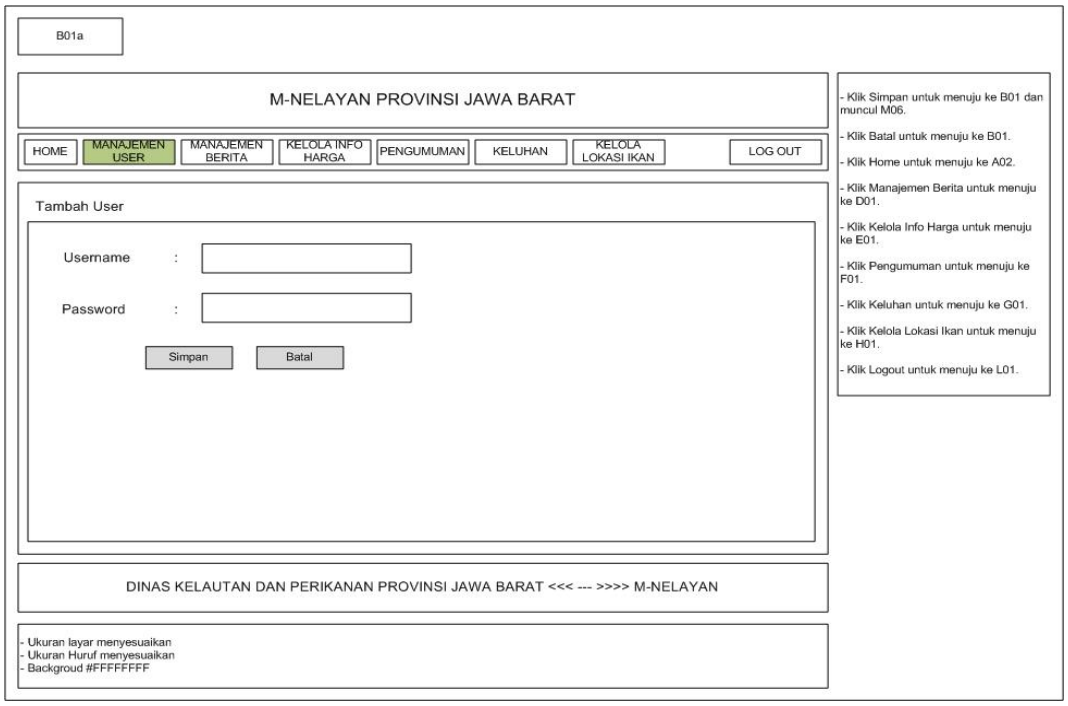
Gambar 3.56 Perancangan Antarmuka Halaman Menu Utama Back End

3. Perancangan Antarmuka Halaman Manajemen User



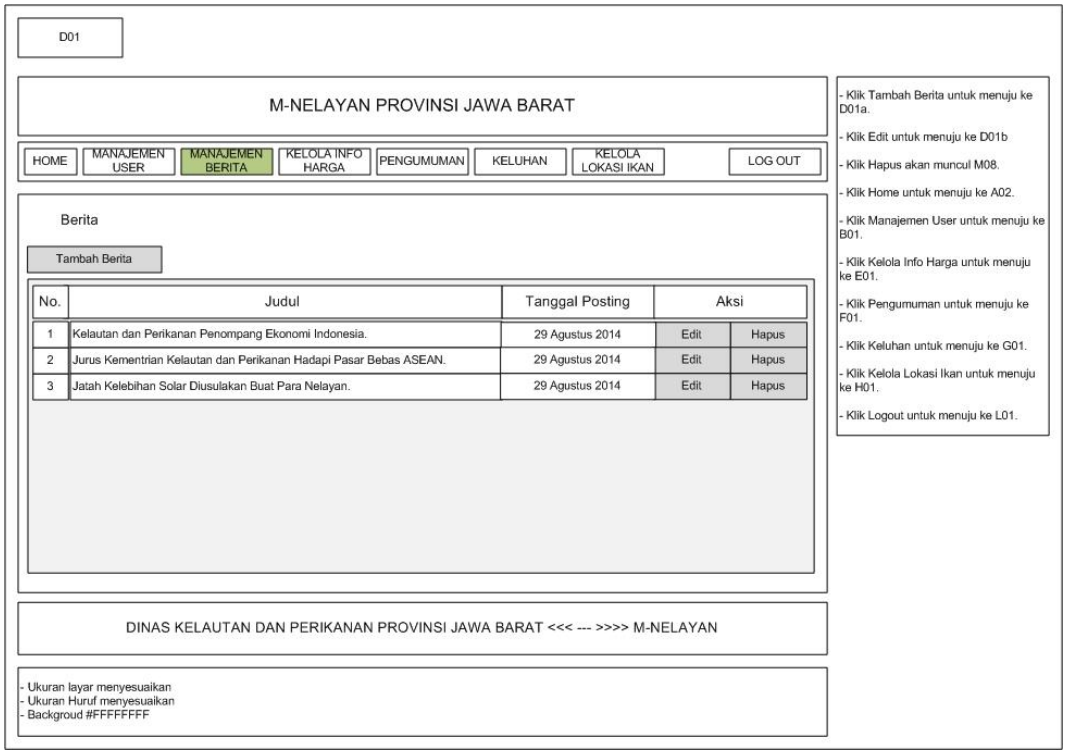
Gambar 3.57 Perancangan Antarmuka Halaman Manajemen User

4. Perancangan Antarmuka Halaman Tambah User



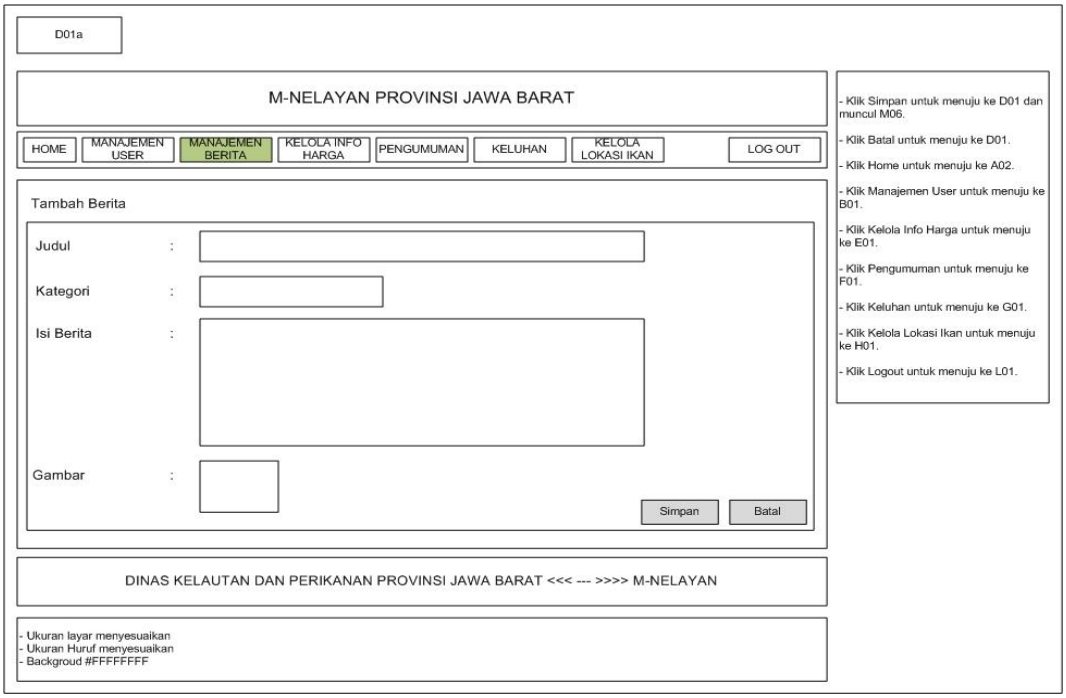
Gambar 3.58 Perancangan Antarmuka Halaman Tambah User

5. Perancangan Antarmuka Halaman Manajemen Berita



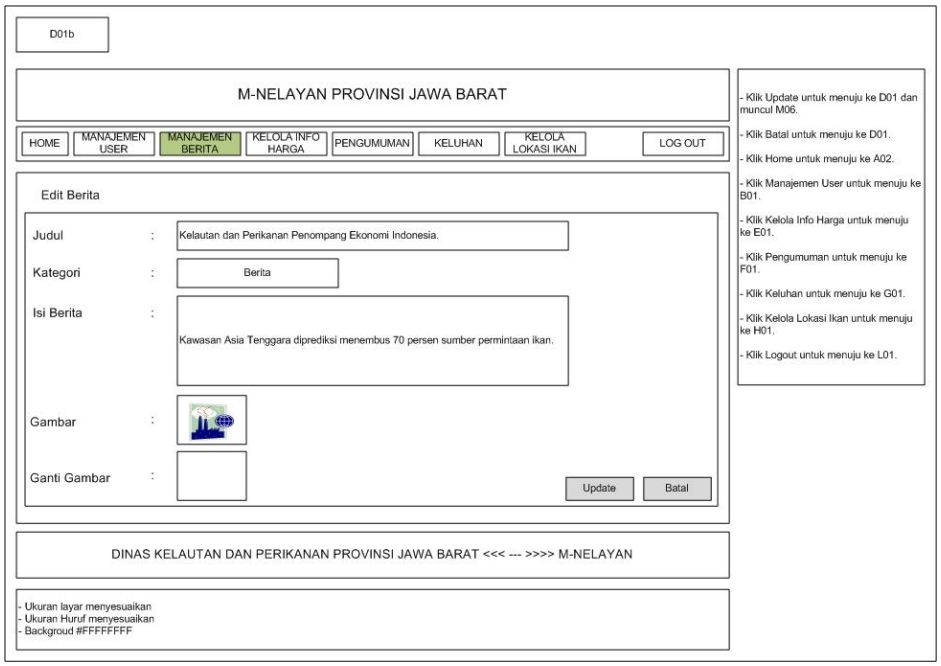
Gambar 3.59 Perancangan Antarmuka Halaman Manajemen Berita

6. Perancangan Antarmuka Halaman Tambah Berita



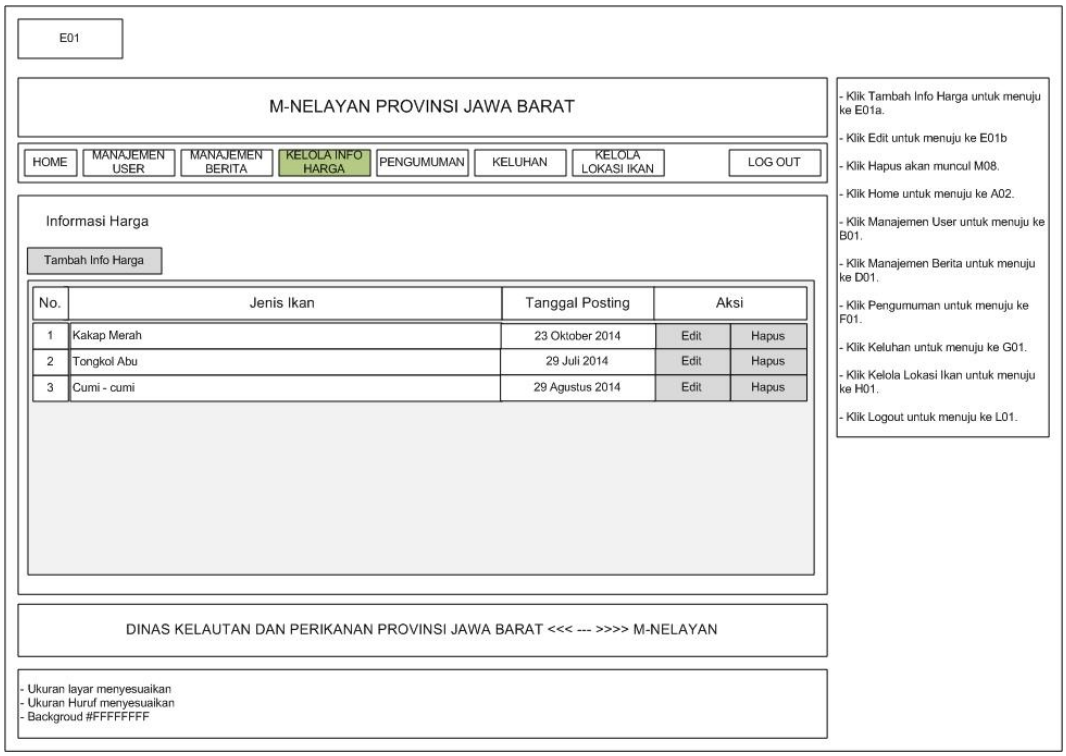
Gambar 3.60 Perancangan Antarmuka Halaman Tambah Berita

7. Perancangan Antarmuka Halaman Edit Berita



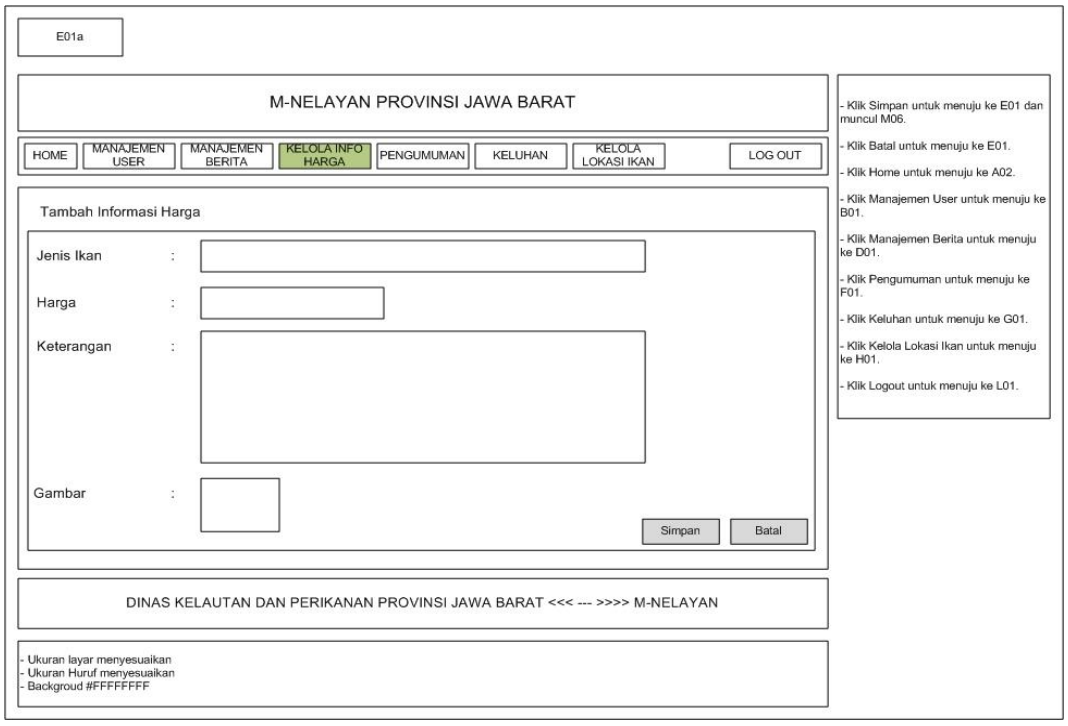
Gambar 3.61 Perancangan Antarmuka Halaman Edit Berita

8. Perancangan Antarmuka Halaman Kelola Info Harga



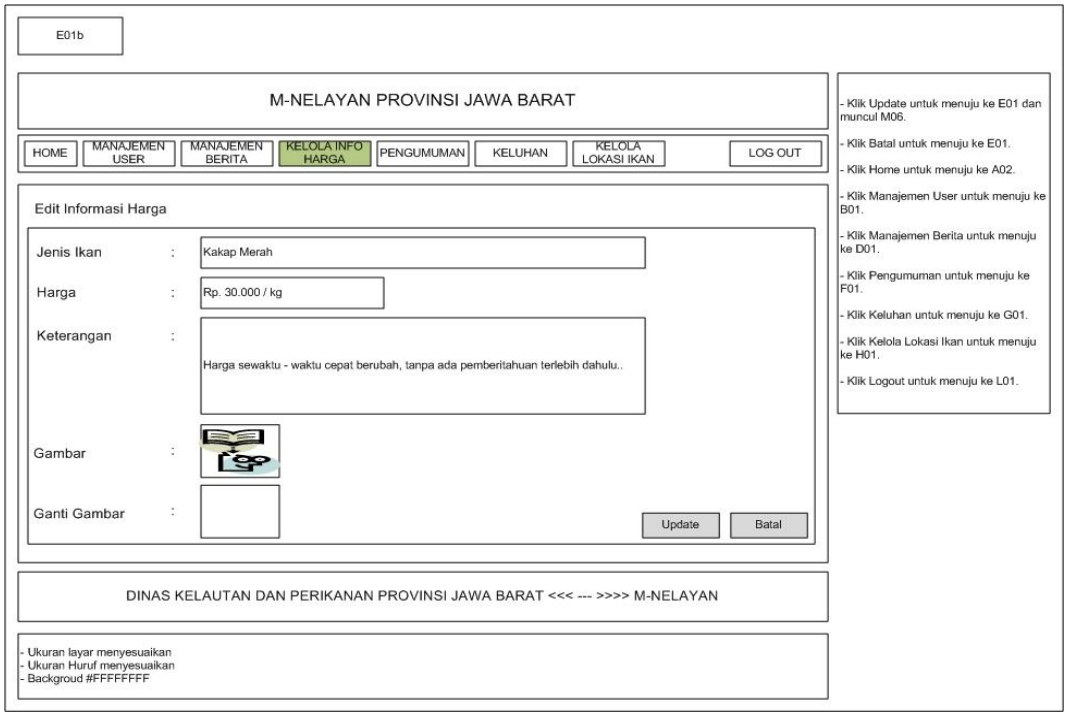
Gambar 3.62 Perancangan Antarmuka Halaman Kelola Info Harga

9. Perancangan Antarmuka Halaman Tambah Info Harga



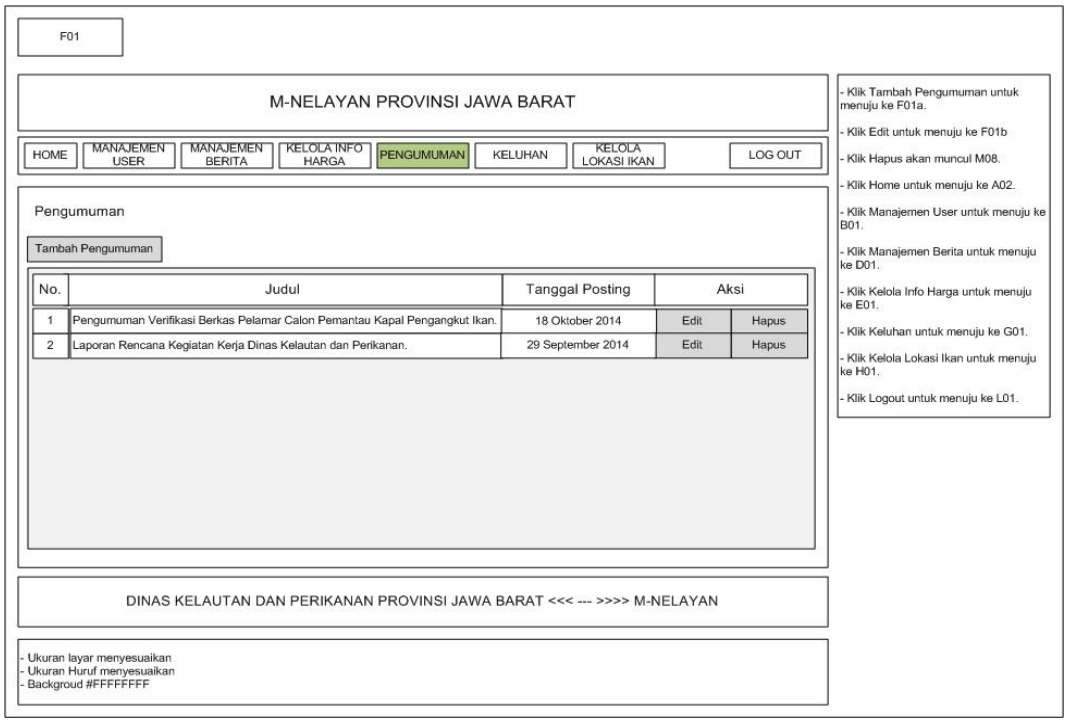
Gambar 3.63 Perancangan Antarmuka Halaman Tambah Info Harga

10. Perancangan Antarmuka Halaman Edit Info Harga



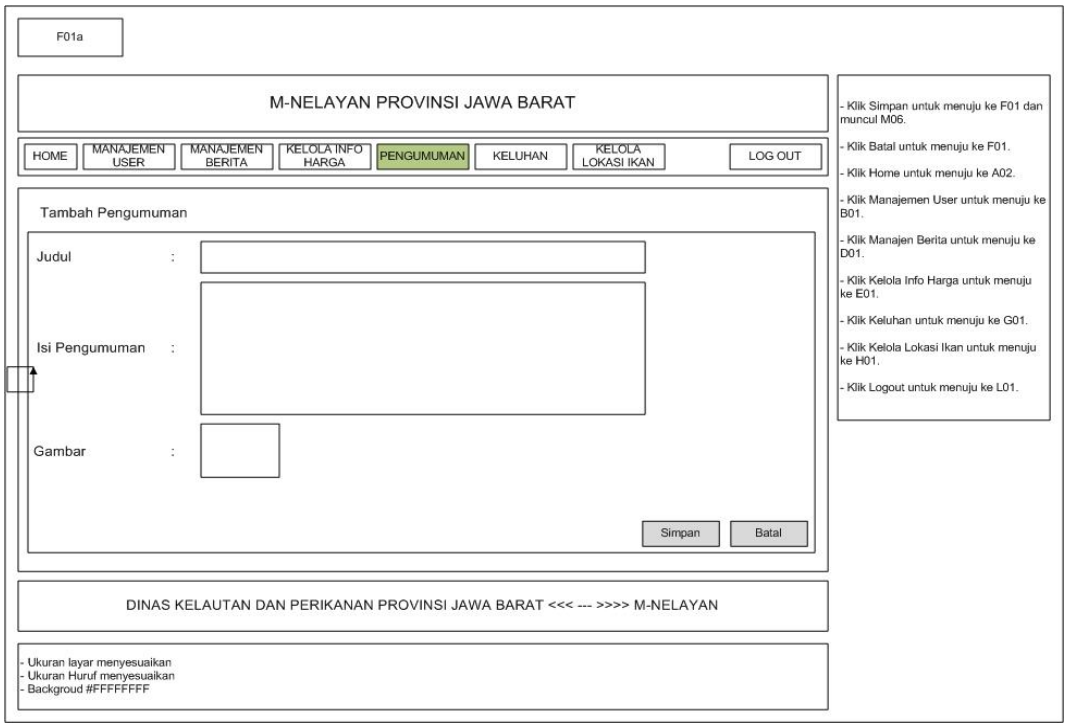
Gambar 3.64 Perancangan Antarmuka Halaman Edit Info Harga

11. Perancangan Antarmuka Halaman Pengumuman



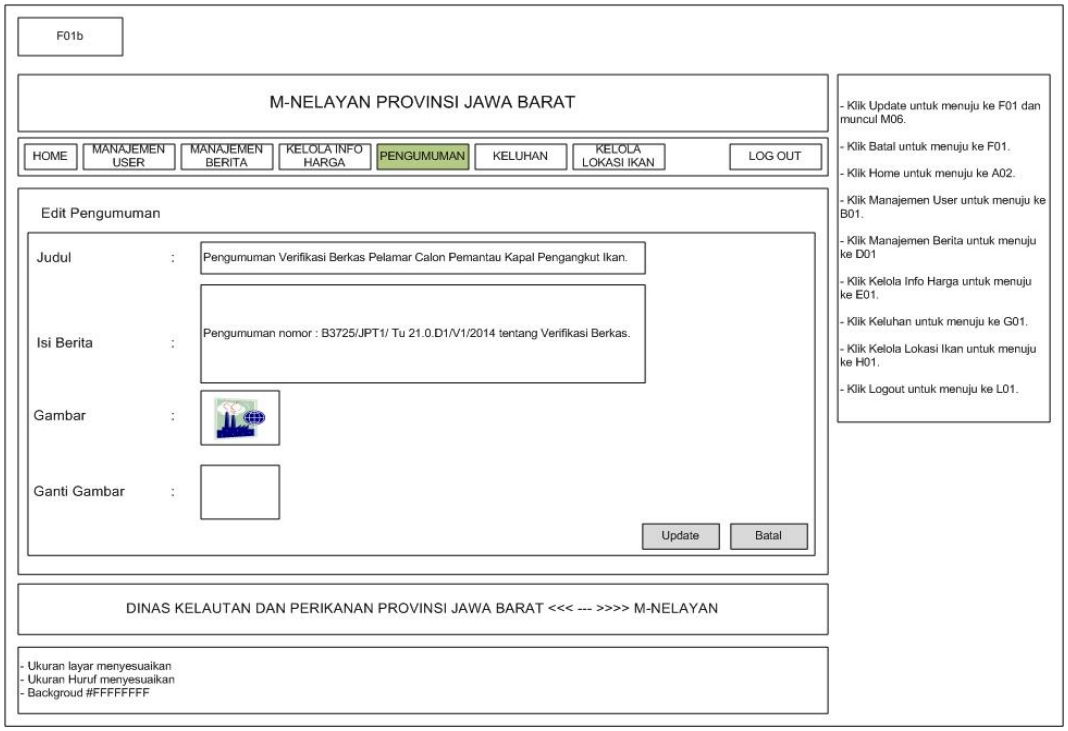
Gambar 3.65 Perancangan Antarmuka Halaman Pengumuman

12. Perancangan Antarmuka Halaman Tambah Pengumuman



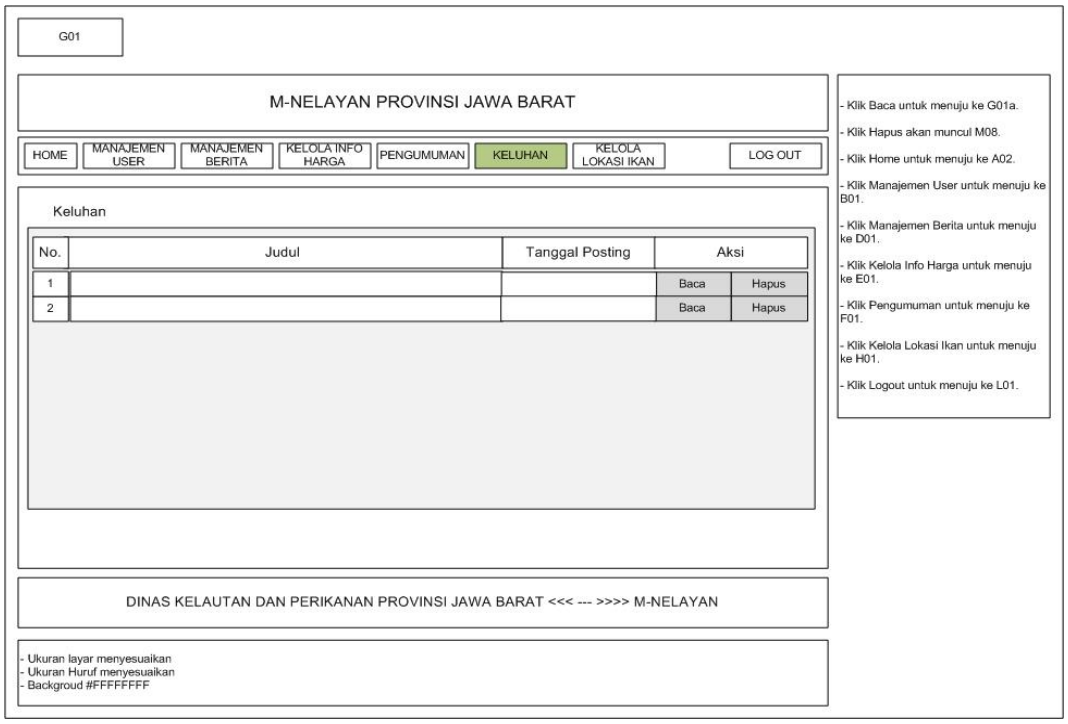
Gambar 3.66 Perancangan Antarmuka Halaman Tambah Pengumuman

13. Perancangan Antarmuka Halaman Edit Pengumuman



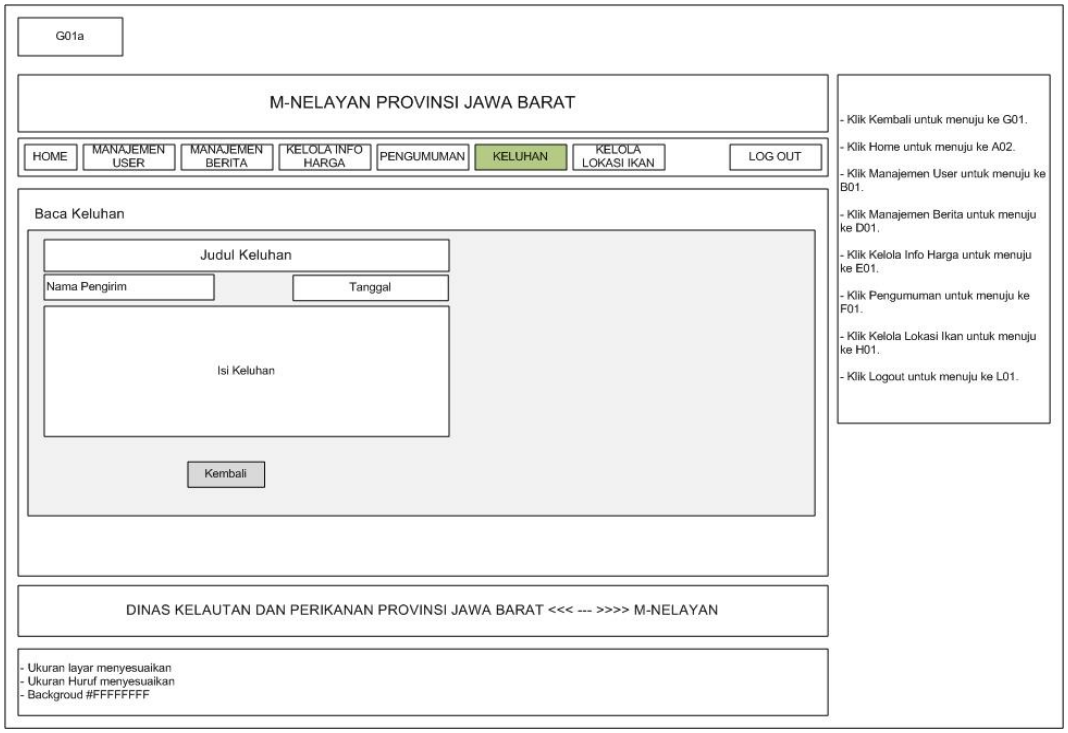
Gambar 3.67 Perancangan Antarmuka Halaman Edit Pengumuman

14. Perancangan Antarmuka Halaman Keluhan



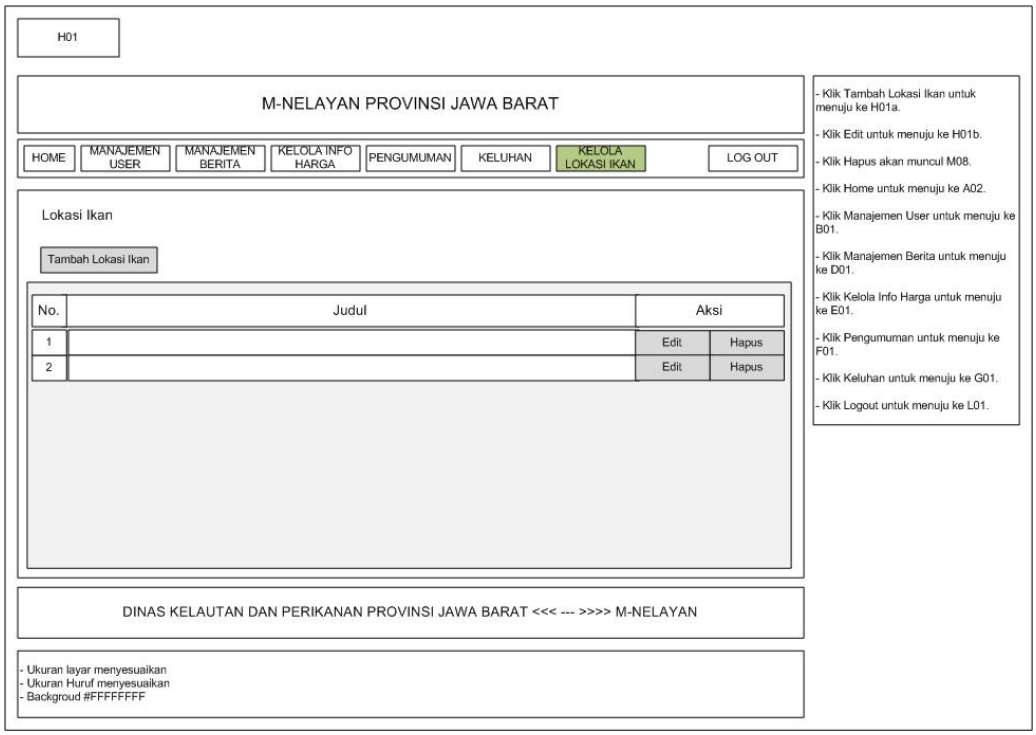
Gambar 3.68 Perancangan Antarmuka Halaman Keluhan

15. Perancangan Antarmuka Halaman Baca Keluhan



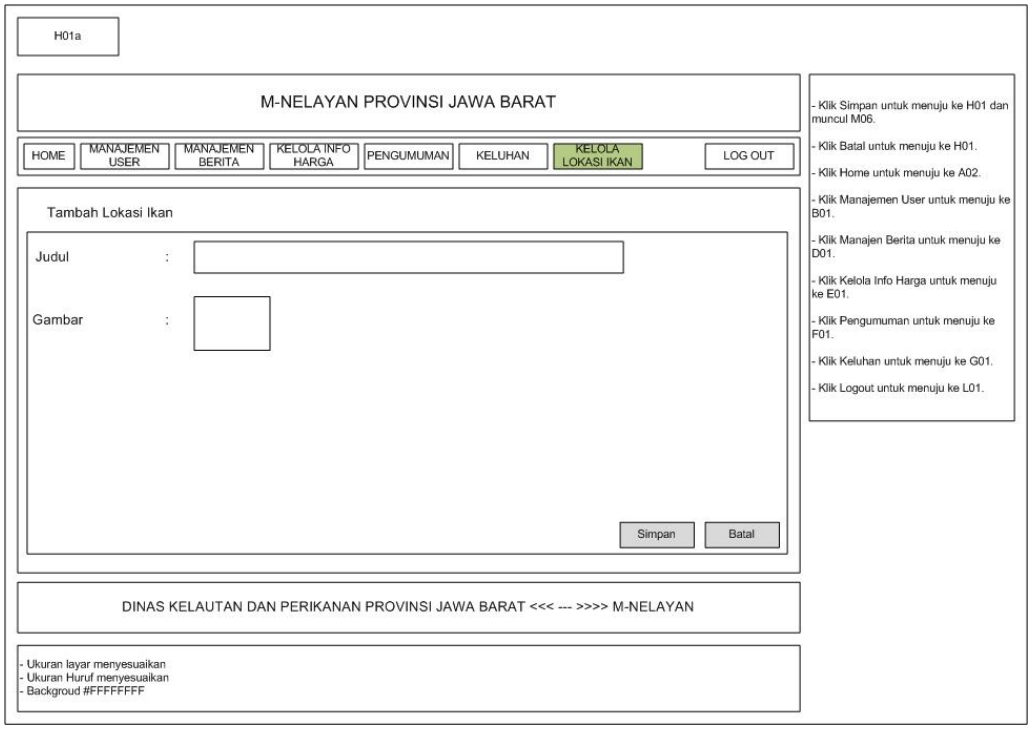
Gambar 3.69 Perancangan Antarmuka Halaman Baca Keluhan

16. Perancangan Antarmuka Halaman Kelola Lokasi Ikan



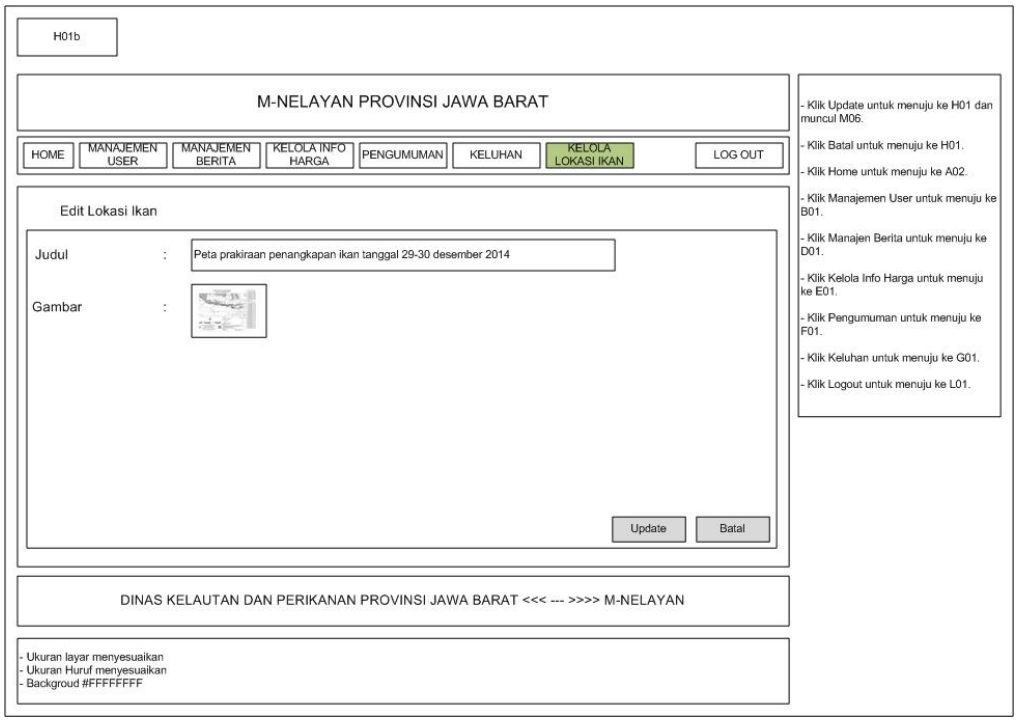
Gambar 3.70 Perancangan Antarmuka Halaman Kelola Lokasi Ikan

17. Perancangan Antarmuka Halaman Tambah Lokasi Ikan



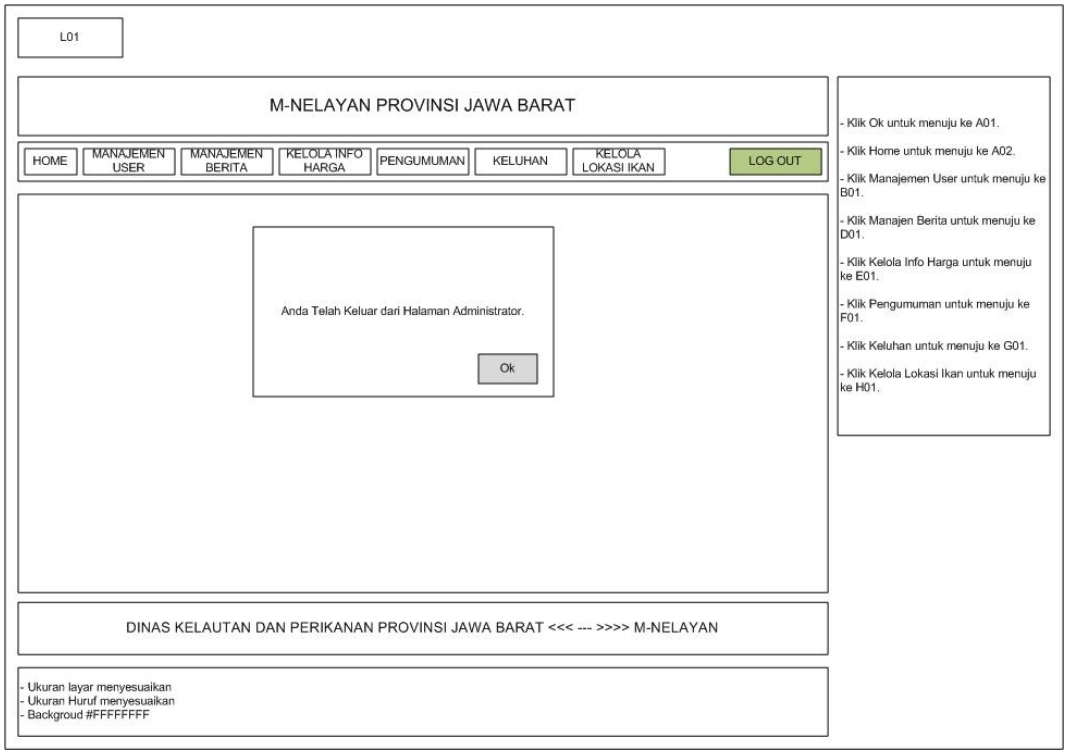
Gambar 3.71 Perancangan Antarmuka Halaman Tambah Lokasi Ikan

18. Perancangan Antarmuka Halaman Edit Lokasi Ikan



Gambar 3.72 Perancangan Antarmuka Halaman Edit Lokasi Ikan

19. Perancangan Antarmuka Halaman *Logout Back End*

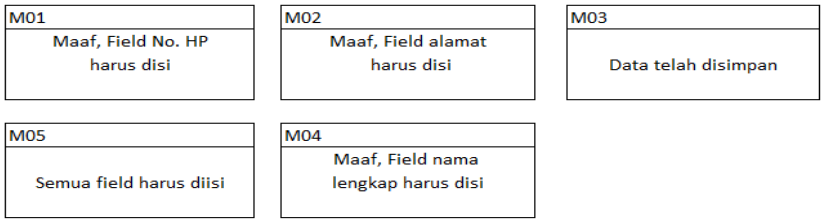


Gambar 3.73 Perancangan Antarmuka Halaman *Logout Back End*

3.2.4 Perancangan Pesan

Perancangan pesan di buat sebagai bentuk pemberitahuan atau respon sistem terhadap pengguna, baik itu ketika pengguna melakukan kesalahan terhadap sistem, maupun hanya sekedar pemberitahuan informasi.

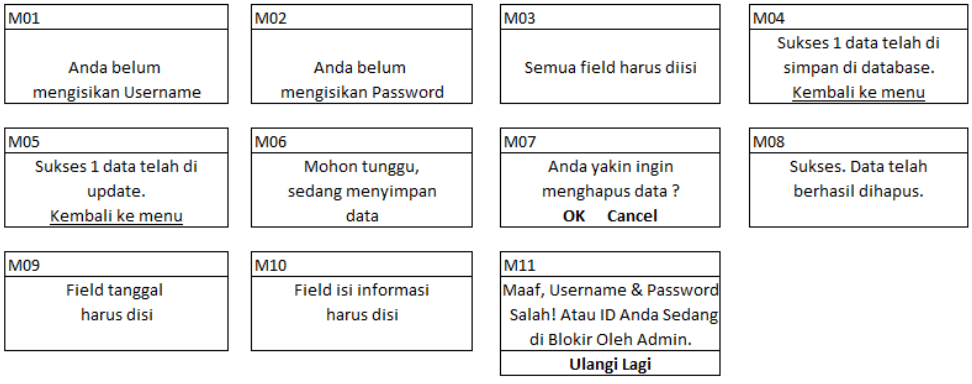
3.2.4.1 Perancangan Pesan Aplikasi *Front End*



Gambar 3.74 Perancangan Pesan Aplikasi *Front End*

No Pesan	Isi Pesan	Jenis Pesan
M01	Maaf, field no. HP harus diisi!	Informasi
M02	Maaf, field alamat harus diisi!	Informasi
M03	Data telah disimpan	Informasi
M04	Semua field harus diisi	Informasi
M05	Maaf, field nama lengkap harus diisi!	Informasi

3.2.4.2 Perancangan Pesan Aplikasi *Back End*



Gambar 3.75 Perancangan Pesan Aplikasi *Back End*

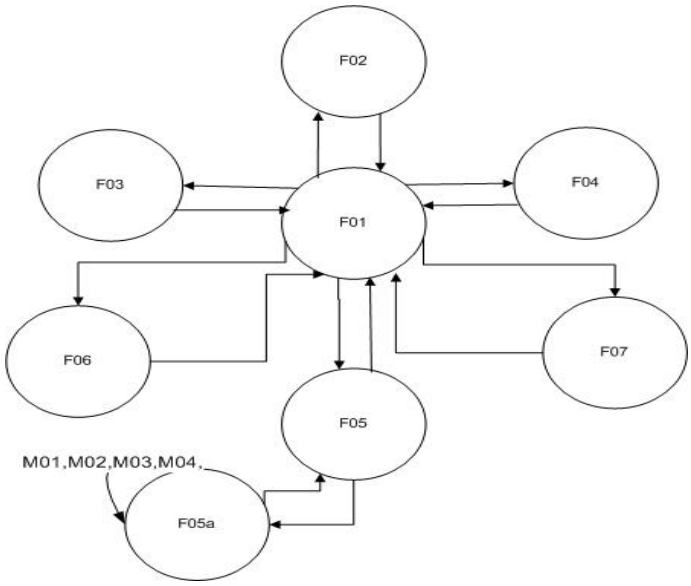
No Pesan	Isi Pesan	Jenis Pesan
M01	Anda belum mengisi *username*	Informasi

M02	Anda belum mengisi *password*	Informasi
M03	Semua *field* harus diisi	Informasi
M04	Sukses. Data telah disimpan di database. Kembali ke menu	Informasi
M05	Sukses. Data telah di*update*. Kembali ke menu.	Informasi
M06	Mohon tunggu, sedang menyimpan data.	Informasi
M07	Anda yakin ingin menghapus data?	Informasi
M08	Sukses. Data telah berhasil dihapus.	Informasi
M09	*Field* tanggal harus diisi	Informasi
M10	*Field* isi informasi harus diisi	Informasi
M11	Maaf, *username* dan *password* salah! **Ulangi lagi**	Informasi

3.2.5 Jaringan Semantik

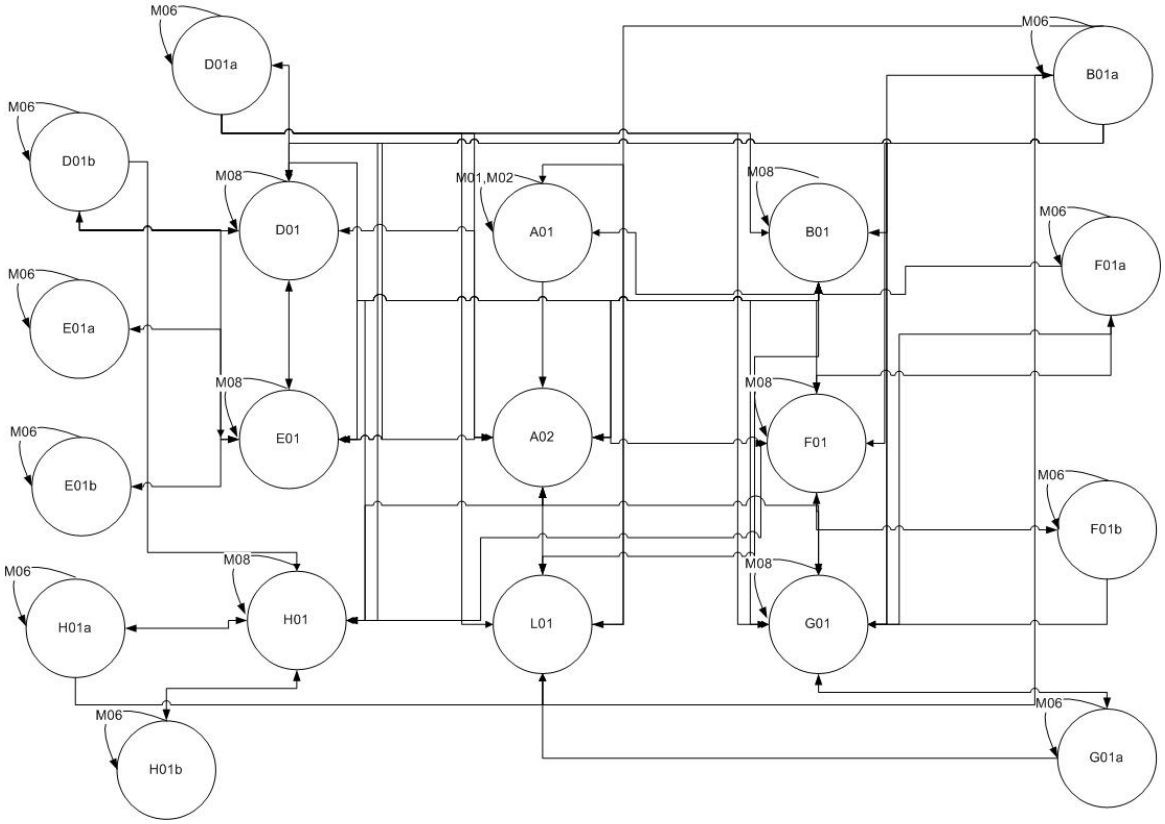
Jaringan semantik menjelaskan keterhubungan antar *form* yang ada di dalam aplikasi, begitu juga dengan pesan yang terdapat di aplikasi.

3.2.5.1 Jaringan Semantik Aplikasi *Front End* untuk Nelayan



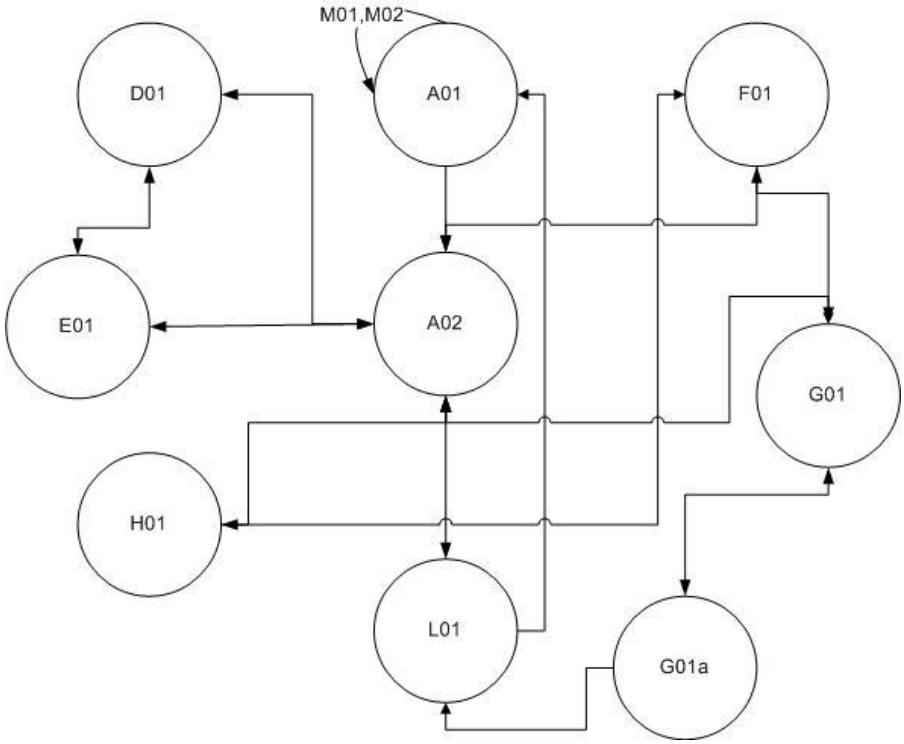
Gambar 3.76 Jaringan Semantik Aplikasi *Front End* Untuk Nelayan

3.2.5.2 Jaringan Semantik Aplikasi *Back End* untuk Admin Web



Gambar 3.77 Jaringan Semantik Aplikasi *Back End* untuk Admin Web

3.2.5.3 Jaringan Semantik Aplikasi *Back End* untuk Tenaga Penyuluh



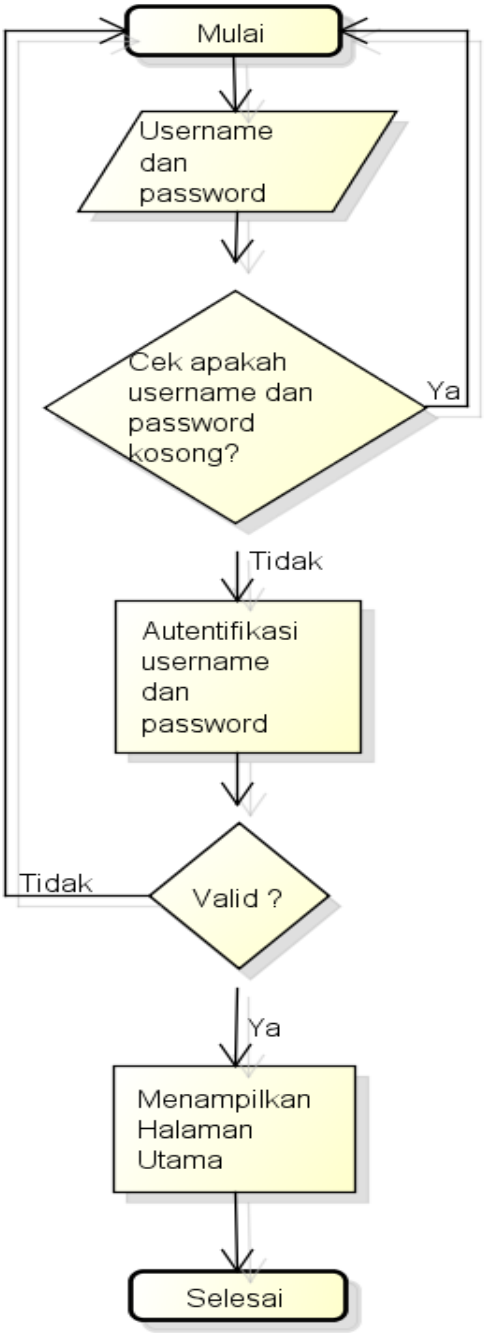
Gambar 3.78 Jaringan Semantik Aplikasi *Back End* Untuk Bagian Penyuluh

3.2.6 Perancangan Prosedural

Perancangan prosedural merupakan representasi algoritma dari aplikasi M-Nelayan yang dibangun. Perancangan prosedural pada aplikasi M-Nelayan digambarkan dengan menggunakan *flowchart.* Adapun penjelasan beserta gambaran dari masing-masing *flowchart* yang ada adalah sebagai berikut:

3.2.6.1 *Flowchart* Proses Login

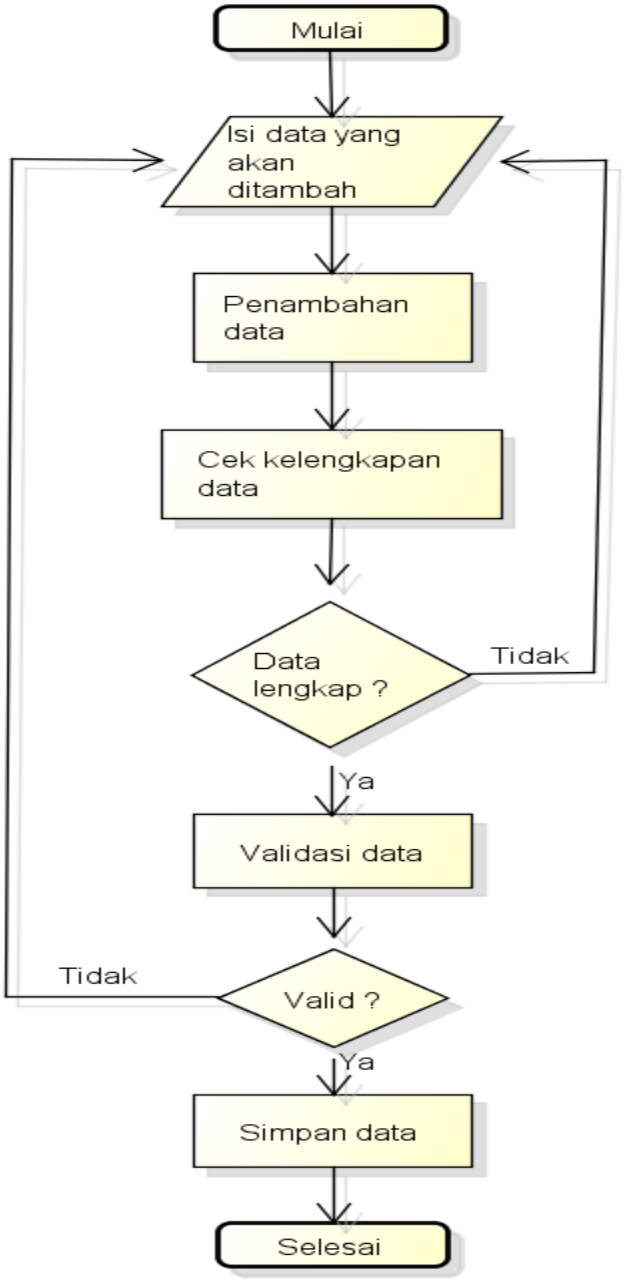
Flowchart proses login adalah ketika user pertama kali akan mengakses sistem dengan menginputkan *username* dan *password*.



Gambar 3.79 *Flowchart* Proses Login

3.2.6.2 *Flowchart* Proses Tambah Data

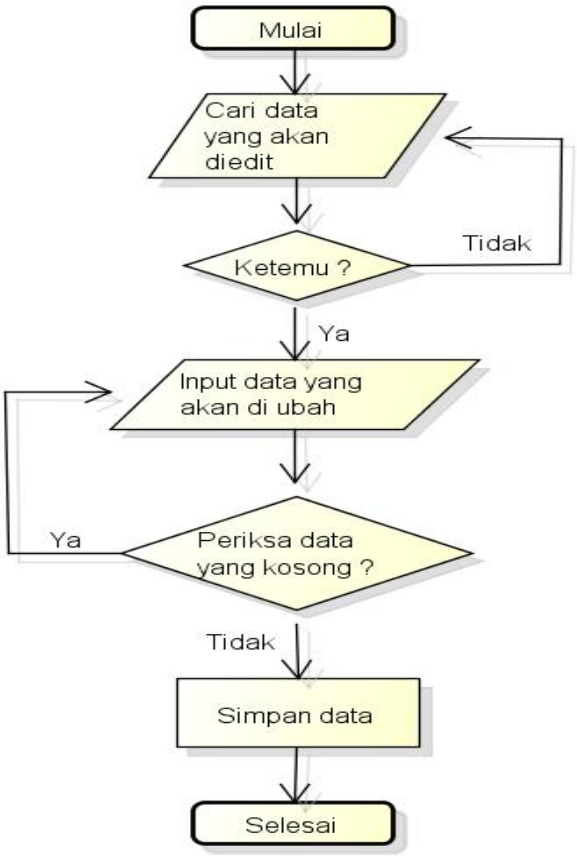
Flowchart proses tambah data adalah ketika user akan melakukan penambahan terhadap suatu data.



Gambar 3.80 *Flowchart* Proses Tambah Data

3.2.6.3 *Flowchart* Proses Edit Data

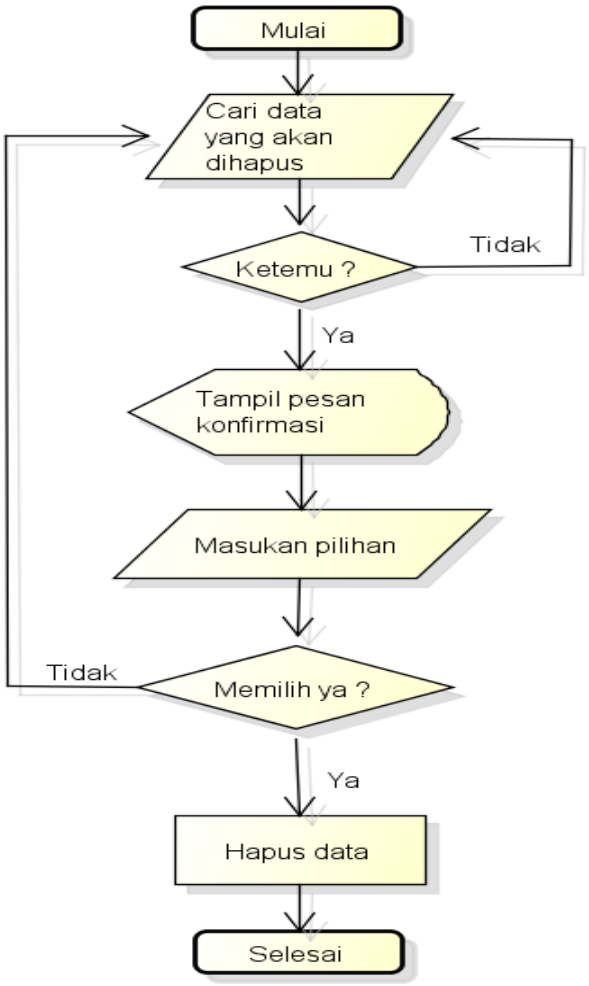
Flowchart proses edit data adalah ketika user akan melakukan suatu perubahan data atau edit data.



Gambar 3.81 *Flowchart* Proses Edit Data

3.2.6.4 *Flowchart* Proses Hapus Data

Flowchart proses hapus data adalah ketika user akan melakukan penghapusan terhadap suatu data.



Gambar 3.82 *Flowchart* Proses Hapus Data