

BAB IV

ANALISIS DAN PERANCANGAN

4.1 ANALISIS SISTEM

Pada bagian ini, dilakukan analisis terhadap sistem yang digunakan dalam penelitian ini untuk memahami kinerja, efektivitas, dan keefisienan sistem secara keseluruhan. Analisis ini bertujuan untuk mengidentifikasi kelebihan dan kekurangan sistem, serta memberikan rekomendasi untuk perbaikan guna meningkatkan kinerja dan pengalaman pengguna.

4.1.1 Gambaran Umum Desa Maju Jaya

Kantor Desa Maju Jaya beralamat di Jalan Bahtera Desa Maju Jaya. Kecamatan Pelepat Ilir, Kabupaten Muaro Bungo, merupakan tempat kerja pemerintahan sebagai perangkat daerah yang mempunyai tanggung jawab dibawah pimpinan Kepala Desa. Kantor Desa sendiri memiliki tanggung jawab pemerintahan untuk melakukan tugas pemerintahan sesuai dengan ketentuan perundangan yang ada seperti menyelenggarakan tata praja pemerintahan, penetapan peraturan di desa, pembinaan masalah pertanahan, pembinaan ketentraman dan ketertiban, melakukan upaya perlindungan masyarakat, administrasi kependudukan, dan penataan dan pengelolaan wilayah.

Kantor Desa Maju Jaya menadi salah satu instansi pemerintahan yang mengelola data-data penduduk, yang melayani berbagai kebutuhan masyarakat diantaranya yaitu melakukan pelayanan data kelahiran, kematian, perpindahan, pendatang, data penduduk, data keterangan usaha dan sebagainya. Dengan membantu kebutuhan masyarakat, Desa Maju Jaya mengharapkan terwujudnya

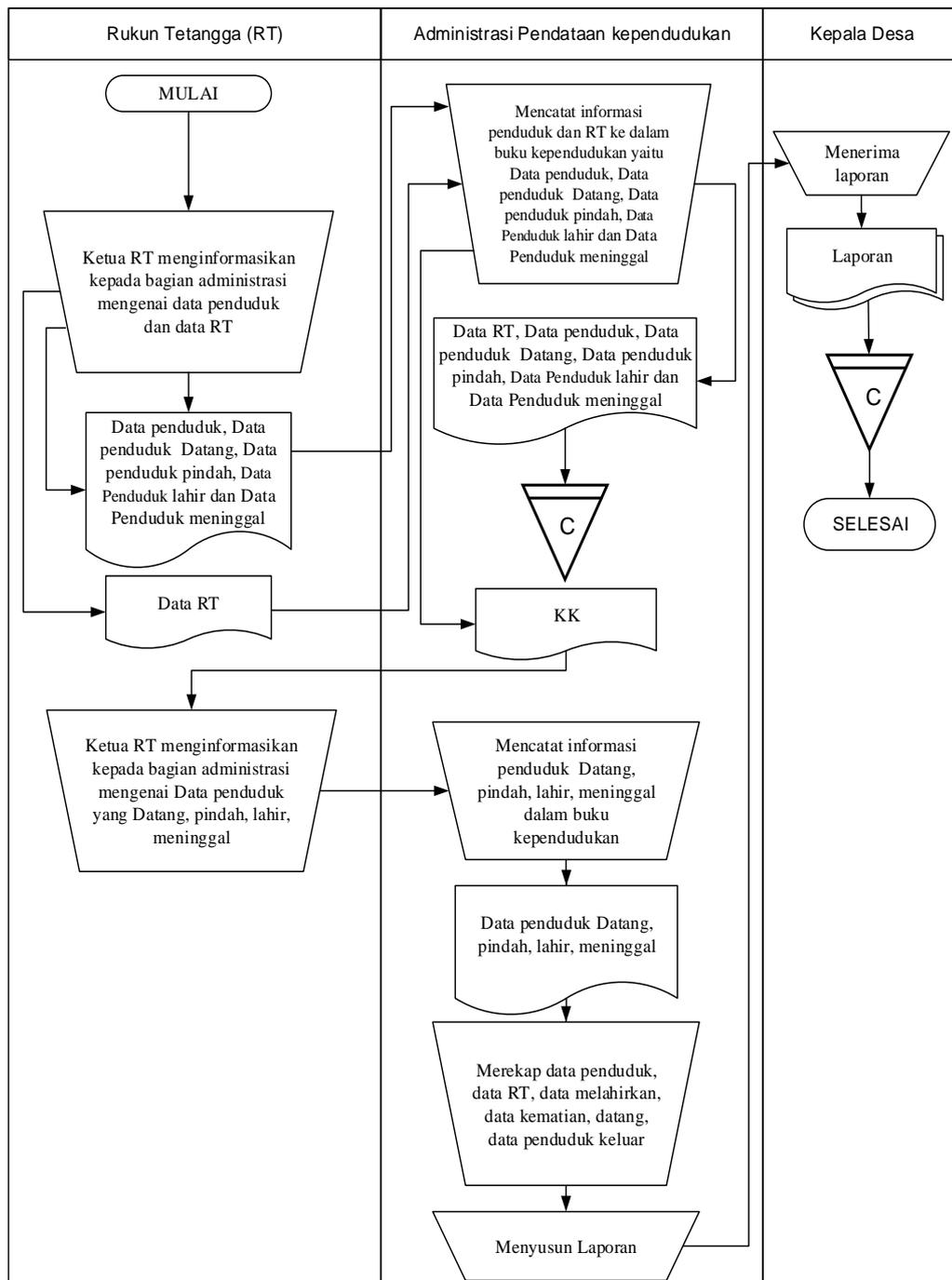
kepercayaan pada antara penduduk desa dengan pihak Desa Maju Jaya dalam mewujudkan Desa yang lebih baik serta mampu menjalin hubungan saling bekerja sama antara penduduk sekitar dan pihak pemerintahan.

4.1.2 Analisis Sistem Yang Sedang Berjalan

Analisis sistem yang sedang berjalan merupakan salah satu cara atau teknik untuk menguraikan masalah dan mencari gambaran dari sistem yang ada atau teknik yang sedang berjalan, apakah tetap dipertahankan atau tidak yang dapat di jelaskan sebagai berikut :

1. Masing-masing RT (rukun tetangga) menginformasikan kepada bagian administrasi khususnya bagian pendataan penduduk mengenai data-data penduduk dan data RT.
2. Bagian administrasi menerima informasi dan menyimpan data penduduk dan data RT dengan cara mencatat ke dalam buku kependudukan.
3. Saat mencatat, data penduduk akan disimpan dan dipisahkan berdasarkan RT masing masing
4. Untuk penduduk yang pindah, lahir, meninggal, akan didata kembali dan dicatat pada buku kependudukan guna mengetahui jumlah penduduk yang ada
5. Kemudian Setiap bulan bagian administrasi akan membuat laporan mengenai kegiatan pengolahan data kependudukan untuk dilaporkan kepada

Berdasarkan sistem yang berjalan diatas maka disusun gambaran *Flowchart* documentnya sebagai berikut ini :



Gambar 4.1 Bagan Alir Dokumen Kependudukan

Berdasarkan hasil pengamatan, maka penulis memperoleh kesimpulan bahwa proses pengolahan data Kependudukan pada Desa Maju Jaya saat ini masih terdapat beberapa kendala yaitu :

1. Dalam proses pengolahan data kependudukannya sudah menggunakan komputer tetapi belum menggunakan sistem informasi kependudukan sehingga kegiatan yang dilakukan belum maksimal karena proses penyimpanan data penduduk, penduduk lahir, penduduk meninggal, penduduk masuk wilayah, penduduk keluar wilayah serta arsip surat menyurat masih menggunakan *file-file* yang terpisah dan ditempatkan pada folder yang cukup banyak.
2. Bagi masyarakat yang membutuhkan informasi berkaitan dengan program pemerintahan seperti pendataan penduduk maka harus datang langsung ke Kantor Desa Tebat Patah untuk mendapatkan informasi yang dibutuhkan hal tersebut di nilai mempersulit masyarakat apabila tidak langsung membawa persyaratan ke kantor, maka masyarakat akan bolak-balik dari kantor ke rumah untuk menyiapkan data yang dibutuhkan.
3. Mempersulit admin dalam mencetak laporan karena mesti memilah kembali riwayat data-data lama serta mesti cek satu persatu tanggal pengolahan data yang dilakukan sebelumnya sehingga membumbutuhkannya waktu lama dalam pembuatan laporan.

Dari permasalahan di atas, maka penulis memperoleh kesimpulan dimana proses manual yang butuh waktu cukup lama dalam mengerjakannya serta ketidakakuratan data yang dihasilkan.

4.1.3 Solusi Pemecahan Masalah

Dari permasalahan tersebut salah satu solusi yang dapat di gunakan oleh Desa Maju Jaya untuk mengatasi permasalahan tersebut adalah dengan merancang suatu sistem informasi Kependudukan pada Desa Maju Jaya. Adapun solusi dari sistem yang akan dirancang tersebut sebagai berikut :

1. Dalam proses pengolahan data kependuduknya sudah menggunakan sistem informasi kependudukan sehingga kegiatan yang dilakukan menjadi maksimal karena proses penyimpanan data penduduk, penduduk lahir, penduduk meninggal, penduduk masuk wilayah menggunakan database MySQL yang datanya sudah saling terintegrasi.
2. Bagi masyarakat yang membutuhkan informasi berkaitan dengan program pemerintahan seperti pendataan penduduk, bisa dilakukan secara *online* karena website yang disediakan menampilkan informasi yang di butuhkan masyarakat tentang kependudukan seperti profil, galeri, berita, formulir maupun kontak kantor desa.
3. Mempermudah admin dalam mencetak laporan karena data kependudukan saling terintegrasi selain itu laporan dapat dicetak berdasarkan tanggal yaitu hari, bulan maupun tahun

Berdasarkan dari solusi pemecahan masalah di atas, maka penulis jadikan sebagai landasan dalam pembangunan sistem informasi Kependudukan pada Desa Maju Jaya, mengenai fungsi apa saja yang harus ada didalam sistem yang akan dibangun.

4.2 ANALISIS KEBUTUHAN SISTEM

Berdasarkan beberapa permasalahan yang ada, penulis tertarik untuk merekomendasi suatu sistem yang dapat dijadikan sebagai *alternative* dalam mengenalkan serta membantu pengolahan data.

4.2.1 Analisis Proses Sistem

Analisis proses sistem bertujuan untuk mengidentifikasi dan mengevaluasi langkah-langkah yang terlibat dalam suatu proses dalam rangka meningkatkan efisiensi dan efektivitasnya. Proses ini melibatkan pemetaan alur kerja, pengumpulan data, serta analisis setiap tahapan untuk menemukan potensi perbaikan.

A. Analisis Kebutuhan Fungsional

Kebutuhan fungsional sistem menggambarkan proses atau fungsi yang harus dikerjakan oleh sistem untuk memenuhi kebutuhan *user* secara rinci mengenai data dan informasi yang berhubungan dengan Admin. Fungsi sistem yang harus dilakukan adalah sebagai berikut :

1. Fungsi Admin

Fungsi-fungsi yang dapat dilakukan Admin antara lain :

a. Fungsi *Login*

Fungsi ini merupakan fungsi yang dilakukan Admin untuk masuk kedalam sistem dengan menginput *Username* dan *Password*.

b. Mengelola data Admin

Fungsi ini merupakan fungsi yang dilakukan Admin untuk melihat, menambah, mengubah dan menghapus data Admin

c. Mengelola data Profil

Fungsi ini merupakan fungsi yang dilakukan Profil untuk melihat, menambah, mengubah dan menghapus data Profil

d. Mengelola data Galeri

Fungsi ini merupakan fungsi yang dilakukan Admin untuk melihat, menambah, mengubah dan menghapus data Galeri

e. Mengelola data Berita

Fungsi ini merupakan fungsi yang dilakukan Admin untuk melihat, menambah, mengubah dan menghapus data Berita

f. Mengelola data Penduduk

Fungsi ini merupakan fungsi yang dilakukan Admin untuk melihat, menambah, mengubah dan menghapus data Penduduk

g. Mengelola data Pendetang baru

Fungsi ini merupakan fungsi yang dilakukan Admin untuk melihat, menambah, mengubah dan menghapus data Pendetang baru

h. Mengelola data Perpindahan

Fungsi ini merupakan fungsi yang dilakukan Admin untuk melihat, menambah, mengubah dan menghapus data Perpindahan

i. Mengelola data Kelahiran

Fungsi ini merupakan fungsi yang dilakukan Admin untuk melihat, menambah, mengubah dan menghapus data Kelahiran

j. Mengelola data Kematian

Fungsi ini merupakan fungsi yang dilakukan Admin untuk melihat,

menambah, mengubah dan menghapus data Kematian

k. Mencetak laporan

Fungsi ini merupakan fungsi yang dilakukan Admin untuk melihat dan mencetak laporan

l. Fungsi *Logout*

Fungsi ini merupakan fungsi yang dilakukan Admin untuk menutup koneksi atau keluar dari sistem

2. Fungsi Penduduk

Fungsi-fungsi yang dapat dilakukan Penduduk antara lain :

a. Fungsi *Login*

Fungsi ini merupakan fungsi yang dilakukan Penduduk untuk masuk kedalam sistem dengan menginput *Username* dan *password*.

b. Melihat informasi data Profil

Fungsi ini merupakan fungsi yang dilakukan untuk melihat informasi data Profil

c. Melihat informasi data Galeri

Fungsi ini merupakan fungsi yang dilakukan untuk melihat informasi data Galeri

d. Melihat informasi data Berita

Fungsi ini merupakan fungsi yang dilakukan untuk melihat informasi data Berita

- e. Melihat informasi data Surat kependudukan

Fungsi ini merupakan fungsi yang dilakukan untuk melihat informasi data Surat kependudukan

- f. Fungsi *Logout*

Fungsi ini merupakan fungsi yang dilakukan Penduduk untuk menutup koneksi atau keluar dari sistem

B. Analisis Kebutuhan *Non Fungsional*

Selain kebutuhan fungsional yang akan dipenuhi, sistem yang dirancang juga diharapkan memenuhi kebutuhan *non fungsional* sebagai berikut :

- a. *Usability*

Sistem memiliki rancangan *interface* yang mudah digunakan.

- b. *Security*

Memiliki system keamanan menggunakan fitur *login* dan *logout* dengan menginput *Username* dan *Password* dengan benar.

- c. *Flexibility*

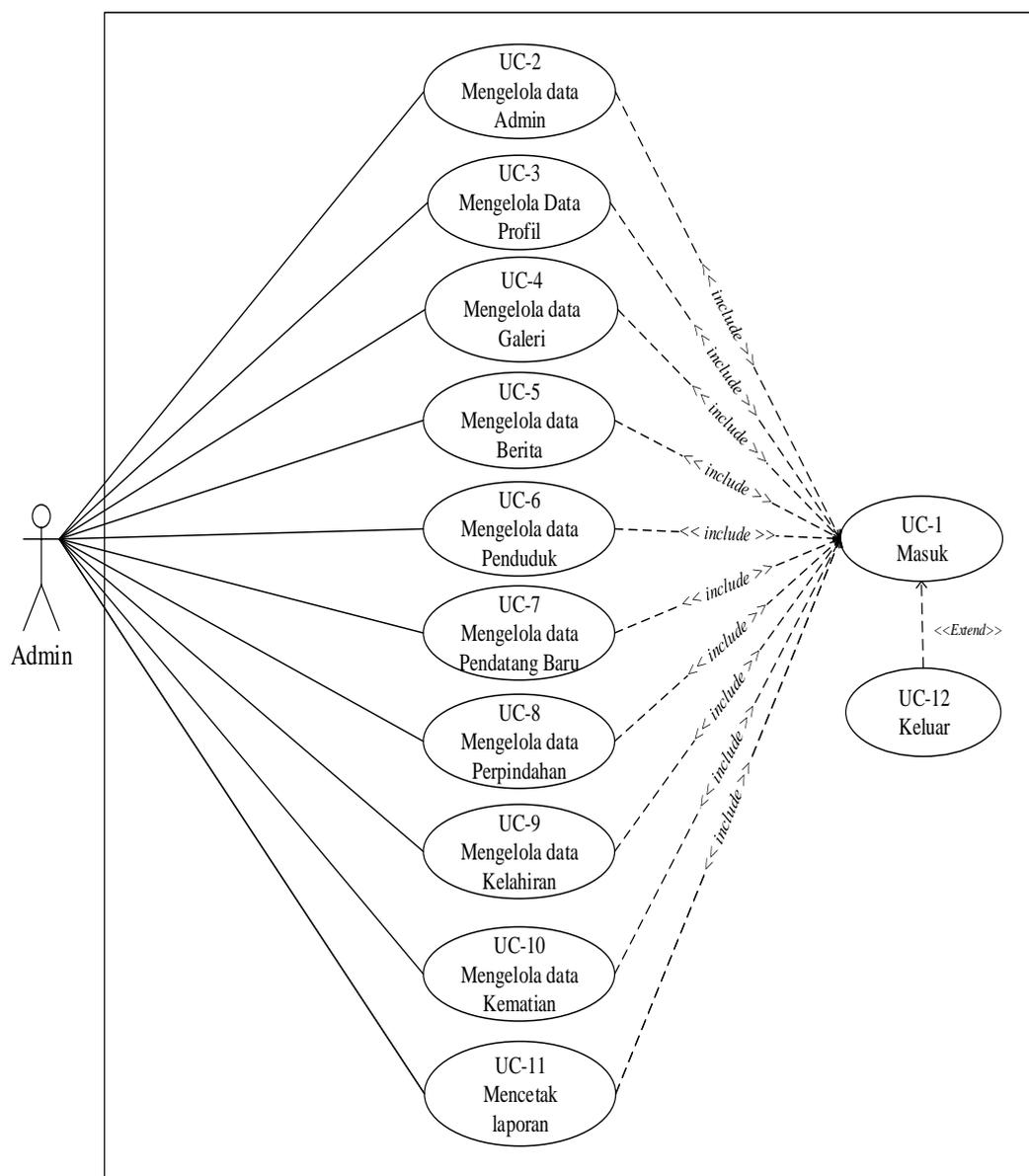
Kemudahan dalam menemukan data yang diperlukan karena sistem memiliki pengorganisasian data yang baik.

4.2.2 Use case diagram

Use case diagram adalah representasi visual dari interaksi antara aktor (pengguna atau sistem lain) dan sistem yang sedang dianalisis. Diagram ini menggambarkan berbagai fungsi atau layanan yang disediakan oleh sistem, serta bagaimana aktor berinteraksi dengan fungsi-fungsi tersebut.

A. Use case diagram Untuk Admin

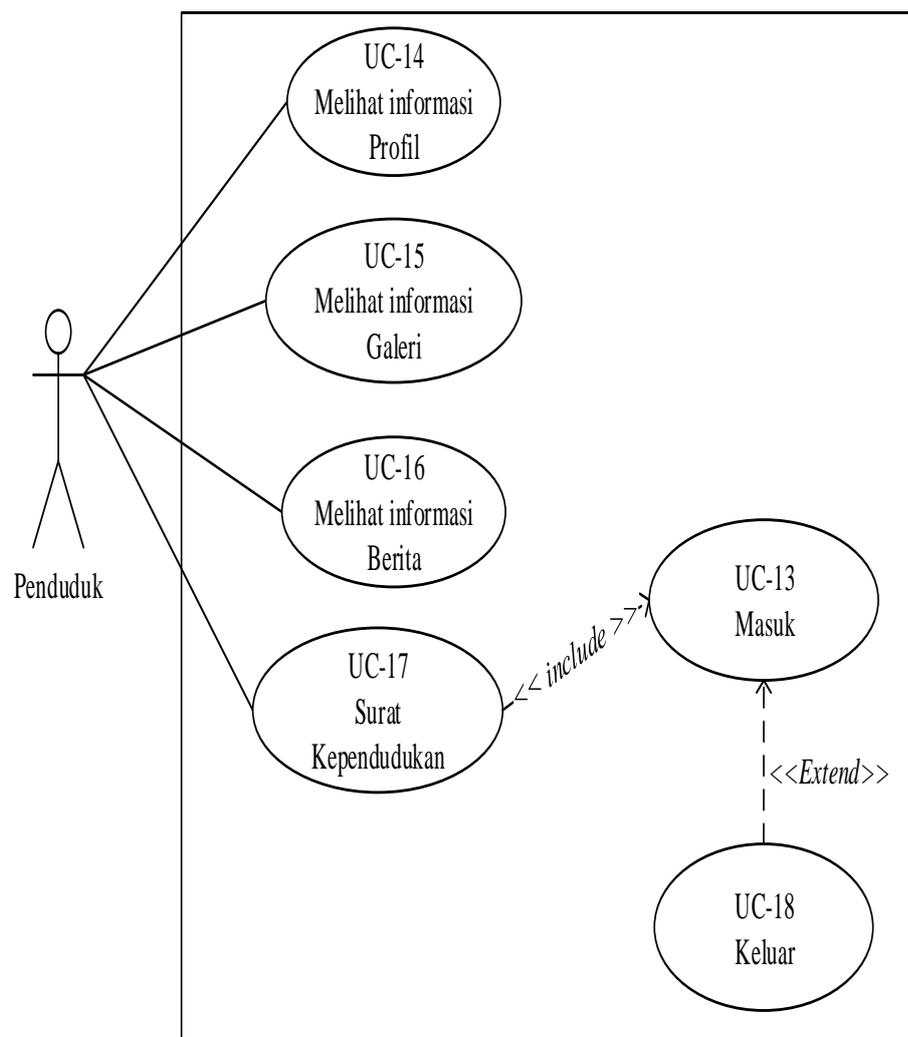
Diagram Use case menyajikan interaksi antara Use case dan Admin di dalam sistem yang akan dikembangkan. Use case diagram berikut ini menggambarkan bagaimana Admin sebagai pengguna yang dapat berinteraksi dengan sistem, dan mengoperasikan sistem seperti terlihat pada gambar berikut :



Gambar 4.2 Use case diagram Admin

B. Use case diagram Untuk Penduduk

Diagram Use case menyajikan interaksi antara Use case dan Penduduk di dalam sistem yang akan dikembangkan. Use case diagram berikut ini menggambarkan bagaimana Penduduk sebagai pengguna yang dapat berinteraksi dengan sistem, dan mengoperasikan sistem seperti terlihat pada gambar berikut :



Gambar 4.3 Use case diagram Penduduk

4.2.3 Deskripsi Use case

Didalam *Use case* terdapat metode berbasis text untuk menggambarkan dan mendokumentasikan proses yang kompleks sebagai berikut ini :

1. Deskripsi Use case Admin

Deskripsi *Use case* Admin merupakan upaya pengolahan data menjadi sesuatu yang dapat diutarakan secara jelas dan tepat dengan tujuan agar dapat dimengerti sebagai berikut ini.

a. Deskripsi Use case Login

Deskripsi *Use case* digunakan untuk menceritakan secara ringkas bagaimana tahapan menggunakan sistem dan apa saja yang bisa dilakukannya. Melalui deskripsi *Use case* dapat diketahui fungsi apa saja yang ada pada sistem, berikut ini merupakan *Use case login* yang berfungsi menceritakan langkah-langkah bagaimana user agar dapat masuk ke dalam sistem :

Tabel 4.1 Deskripsi Use case login

Nama	<i>Login</i>	
ID Use case	UC.1.	
Aktor	Admin	
Deskripsi	Aktor melakukan <i>Login</i> untuk otentikasi dalam pengecekan dan modifikasi data	
Exception	Proses <i>Login</i> gagal, dikarenakan <i>username</i> atau <i>password</i> tidak sesuai, kosong atau belum terdata sebelumnya.	
Pre condition	<i>Username</i> dan <i>Password</i> telah tersimpan dalam <i>database</i>	
	Aktor	Sistem
Skenario normal		
	1. Aktor melakukan <i>Login</i> dengan memasukkan <i>Username</i> dan <i>Password</i>	
	2. Aktor mengklik <i>Login</i>	
		3. Mencocokkan <i>Username</i> dan <i>Username</i> dengan <i>database</i>

	4. Data cocok, sistem menampilkan Halaman menu utama
Skenario alternatif	
2.a	Aktor memilih batal masuk menu utama dengan mengklik <i>Cancel</i>
2.b	Tutup Halaman <i>login</i>
3.a	Mencocokkan <i>username</i> dan <i>Password</i> dengan database, dan data tidak cocok
3.b	Tapilkan pesan Peringatan data yang <i>diinputkan</i> salah, kemudian tampil <i>form login</i> dan Admin kembali <i>inputkan Username</i> dan <i>Password</i> dengan benar
Post condition	Aktor sudah masuk sistem untuk mengolah data

b. Deskripsi *Use case* mengelola data admin

Deskripsi *Use case* digunakan untuk menceritakan secara ringkas bagaimana tahapan menggunakan sistem dan apa saja yang bisa dilakukannya. Melalui deskripsi *Use case* dapat diketahui fungsi-fungsi apa saja yang ada pada sistem, berikut ini merupakan deskripsi *Use case* admin yang berfungsi menceritakan langkah-langkah bagaimana user menggunakan fungsi yang terdapat pada halaman admin.

Tabel 4.2 Deskripsi *Use case* Mengelola Data Admin

Nama	Mengelola data admin
Id Use case	UC.2.
Aktor	Admin
Deskripsi	Aktor dapat menambah, mengubah dan menghapus data admin
Exception	Pengolahan data tidak dapat dilakukan jika belum <i>login</i>
Pre condition	Aktor telah dalam keadaan <i>login</i>
Aktor	Sistem
Skenario normal	
1. Aktor memilih menu data admin	
	2. Menampilkan halaman menu data admin
3. Kondisi : – Jika aktor memilih tombol tambah pada halaman data admin maka s1 berlaku – Jika aktor memilih tombol edit pada	

halaman data admin maka s2 berlaku – Jika aktor memilih tombol hapus pada halaman data admin maka s3 berlaku	
S1 – fungsi tambah	
1. Aktor memilih tombol tambah pada halaman data admin	
	2. Sistem menampilkan halaman tambah data admin
3. Aktor menginputkan data admin pada halaman tambah data admin	
4. Aktor klik tombol simpan	
	5. Validasi <i>input</i> data admin lengkap
	6. Menyimpan data ke dalam <i>database</i>
S2 – fungsi edit	
1. Aktor memilih data admin untuk diedit pada <i>gridview</i>	
2. Aktor klik tombol edit pada halaman data admin	
	3. Sistem menampilkan halaman edit data admin
4. Aktor memodifikasi data admin pada halaman edit data admin	
5. Aktor klik tombol update	
	6. Validasi modifikasi data admin lengkap
	7. Menyimpan data ke dalam <i>database</i>
S3 – fungsi hapus	
1. Aktor memilih data admin untuk dihapus pada <i>gridview</i>	
2. Aktor klik tombol hapus	
	3. Validasi apakah data akan dihapus, dengan menampilkan pesan “apakah data tersebut akan dihapus” pilih : <i>yes</i> atau <i>no</i>
4. Aktor menghapus data admin dengan memilih tombol <i>yes</i>	
	5. Data admin dihapus dari <i>database</i>
Skenario alternatif	

S1-5a	Saat validasi, sistem menampilkan pesan bahwa data yang diinput belum lengkap
S1-5b	Sistem memberikan kesempatan pada aktor untuk mengulang <i>input</i> data admin.
S2-6a	Saat validasi, sistem menampilkan pesan bahwa data yang dimodifikasi belum lengkap
S2-6b	Sistem memberikan kesempatan pada aktor untuk mengulang modifikasi data admin
S3-3a	Sistem menampilkan pesan peringatan hapus data, dan aktor memilih <i>no</i>
S3-3b	Sistem memberikan kesempatan pada aktor untuk mengulang memilih data admin untuk dihapus
Post condition	Aktor berhasil mengelola data admin

c. Deskripsi *Use case* mengelola data Profil

Deskripsi *Use case* digunakan untuk menceritakan secara ringkas bagaimana tahapan menggunakan sistem dan apa saja yang bisa dilakukannya. Melalui deskripsi *Use case* dapat diketahui fungsi-fungsi apa saja yang ada pada sistem, berikut ini merupakan deskripsi *Use case* Profil yang berfungsi menceritakan langkah-langkah bagaimana user menggunakan fungsi yang terdapat pada halaman Profil.

Tabel 4.3 Deskripsi *Use case* Mengelola Data Profil

Nama	Mengelola data Profil
Id Use case	UC.3.
Aktor	Admin
Deskripsi	Aktor dapat menambah, mengubah dan menghapus data Profil
Exception	Pengolahan data tidak dapat dilakukan jika belum <i>login</i>
Pre condition	Aktor telah dalam keadaan <i>login</i>
Aktor	Sistem
Skenario normal	
1. Aktor memilih menu data Profil	
	2. Menampilkan halaman menu data Profil
3. Kondisi :	
– Jika aktor memilih tombol tambah	

<p>pada halaman data Profil maka s1 berlaku</p> <ul style="list-style-type: none"> – Jika aktor memilih tombol edit pada halaman data Profil maka s2 berlaku – Jika aktor memilih tombol hapus pada halaman data Profil maka s3 berlaku 	
S1 – fungsi tambah	
1. Aktor memilih tombol tambah pada halaman data Profil	
	2. Sistem menampilkan halaman tambah data Profil
3. Aktor menginputkan data Profil pada halaman tambah data Profil	
4. Aktor klik tombol simpan	
	5. Validasi <i>input</i> data Profil lengkap
	6. Menyimpan data ke dalam <i>database</i>
S2 – fungsi edit	
1. Aktor memilih data Profil untuk diedit pada <i>gridview</i>	
2. Aktor klik tombol edit pada halaman data Profil	
	3. Sistem menampilkan halaman edit data Profil
4. Aktor memodifikasi data Profil pada halaman edit data Profil	
5. Aktor klik tombol update	
	6. Validasi modifikasi data Profil lengkap
	7. Menyimpan data ke dalam <i>database</i>
S3 – fungsi hapus	
1. Aktor memilih data Profil untuk dihapus pada <i>gridview</i>	
2. Aktor klik tombol hapus	
	3. Validasi apakah data akan dihapus, dengan menampilkan pesan “apakah data tersebut akan dihapus” pilih : <i>yes</i> atau <i>no</i>
4. Aktor menghapus data Profil dengan memilih tombol <i>yes</i>	
	5. Data Profil dihapus dari <i>database</i>
Skenario alternatif	

S1-5a	Saat validasi, sistem menampilkan pesan bahwa data yang diinput belum lengkap
S1-5b	Sistem memberikan kesempatan pada aktor untuk mengulang <i>input</i> data Profil.
S2-6a	Saat validasi, sistem menampilkan pesan bahwa data yang dimodifikasi belum lengkap
S2-6b	Sistem memberikan kesempatan pada aktor untuk mengulang modifikasi data Profil
S3-3a	Sistem menampilkan pesan peringatan hapus data, dan aktor memilih <i>no</i>
S3-3b	Sistem memberikan kesempatan pada aktor untuk mengulang memilih data Profil untuk dihapus
Post condition	Aktor berhasil mengelola data Profil

d. Deskripsi *Use case* mengelola data Galeri

Deskripsi *Use case* digunakan untuk menceritakan secara ringkas bagaimana tahapan menggunakan sistem dan apa saja yang bisa dilakukannya. Melalui deskripsi *Use case* dapat diketahui fungsi-fungsi apa saja yang ada pada sistem, berikut ini merupakan deskripsi *Use case* Galeri yang berfungsi menceritakan langkah-langkah bagaimana user menggunakan fungsi yang terdapat pada halaman Galeri.

Tabel 4.4 Deskripsi *Use case* Mengelola Data Galeri

Nama	Mengelola data Galeri
Id Use case	UC.4.
Aktor	Admin
Deskripsi	Aktor dapat menambah, mengubah dan menghapus data Galeri
Exception	Pengolahan data tidak dapat dilakukan jika belum <i>login</i>
Pre condition	Aktor telah dalam keadaan <i>login</i>
Aktor	Sistem
Skenario normal	
1. Aktor memilih menu data Galeri	
	2. Menampilkan halaman menu data Galeri
3. Kondisi :	
– Jika aktor memilih tombol tambah	

<p>pada halaman data Galeri maka s1 berlaku</p> <ul style="list-style-type: none"> – Jika aktor memilih tombol edit pada halaman data Galeri maka s2 berlaku – Jika aktor memilih tombol hapus pada halaman data Galeri maka s3 berlaku 	
S1 – fungsi tambah	
1. Aktor memilih tombol tambah pada halaman data Galeri	
	2. Sistem menampilkan halaman tambah data Galeri
3. Aktor menginputkan data Galeri pada halaman tambah data Galeri	
4. Aktor klik tombol simpan	
	5. Validasi <i>input</i> data Galeri lengkap
	6. Menyimpan data ke dalam <i>database</i>
S2 – fungsi edit	
1. Aktor memilih data Galeri untuk diedit pada <i>gridview</i>	
2. Aktor klik tombol edit pada halaman data Galeri	
	3. Sistem menampilkan halaman edit data Galeri
4. Aktor memodifikasi data Galeri pada halaman edit data Galeri	
5. Aktor klik tombol update	
	6. Validasi modifikasi data Galeri lengkap
	7. Menyimpan data ke dalam <i>database</i>
S3 – fungsi hapus	
1. Aktor memilih data Galeri untuk dihapus pada <i>gridview</i>	
2. Aktor klik tombol hapus	
	3. Validasi apakah data akan dihapus, dengan menampilkan pesan “apakah data tersebut akan dihapus” pilih : <i>yes</i> atau <i>no</i>
4. Aktor menghapus data Galeri dengan memilih tombol <i>yes</i>	
	5. Data Galeri dihapus dari <i>database</i>
Skenario alternatif	

S1-5a	Saat validasi, sistem menampilkan pesan bahwa data yang diinput belum lengkap
S1-5b	Sistem memberikan kesempatan pada aktor untuk mengulang <i>input</i> data Galeri.
S2-6a	Saat validasi, sistem menampilkan pesan bahwa data yang dimodifikasi belum lengkap
S2-6b	Sistem memberikan kesempatan pada aktor untuk mengulang modifikasi data Galeri
S3-3a	Sistem menampilkan pesan peringatan hapus data, dan aktor memilih <i>no</i>
S3-3b	Sistem memberikan kesempatan pada aktor untuk mengulang memilih data Galeri untuk dihapus
Post condition	Aktor berhasil mengelola data Galeri

e. Deskripsi *Use case* mengelola data Berita

Deskripsi *Use case* digunakan untuk menceritakan secara ringkas bagaimana tahapan menggunakan sistem dan apa saja yang bisa dilakukannya. Melalui deskripsi *Use case* dapat diketahui fungsi-fungsi apa saja yang ada pada sistem, berikut ini merupakan deskripsi *Use case* Berita yang berfungsi menceritakan langkah-langkah bagaimana user menggunakan fungsi yang terdapat pada halaman Berita.

Tabel 4.5 Deskripsi *Use case* Mengelola Data Berita

Nama	Mengelola data Berita
Id Use case	UC.5.
Aktor	Admin
Deskripsi	Aktor dapat menambah, mengubah dan menghapus data Berita
Exception	Pengolahan data tidak dapat dilakukan jika belum <i>login</i>
Pre condition	Aktor telah dalam keadaan <i>login</i>
Aktor	Sistem
Skenario normal	
1. Aktor memilih menu data Berita	
	2. Menampilkan halaman menu data Berita
3. Kondisi :	
– Jika aktor memilih tombol tambah	

<p>pada halaman data Berita maka s1 berlaku</p> <ul style="list-style-type: none"> – Jika aktor memilih tombol edit pada halaman data Berita maka s2 berlaku – Jika aktor memilih tombol hapus pada halaman data Berita maka s3 berlaku 	
S1 – fungsi tambah	
1. Aktor memilih tombol tambah pada halaman data Berita	
	2. Sistem menampilkan halaman tambah data Berita
3. Aktor menginputkan data Berita pada halaman tambah data Berita	
4. Aktor klik tombol simpan	
	5. Validasi <i>input</i> data Berita lengkap
	6. Menyimpan data ke dalam <i>database</i>
S2 – fungsi edit	
1. Aktor memilih data Berita untuk diedit pada <i>gridview</i>	
2. Aktor klik tombol edit pada halaman data Berita	
	3. Sistem menampilkan halaman edit data Berita
4. Aktor memodifikasi data Berita pada halaman edit data Berita	
5. Aktor klik tombol update	
	6. Validasi modifikasi data Berita lengkap
	7. Menyimpan data ke dalam <i>database</i>
S3 – fungsi hapus	
1. Aktor memilih data Berita untuk dihapus pada <i>gridview</i>	
2. Aktor klik tombol hapus	
	3. Validasi apakah data akan dihapus, dengan menampilkan pesan “apakah data tersebut akan dihapus” pilih : <i>yes</i> atau <i>no</i>
4. Aktor menghapus data Berita dengan memilih tombol <i>yes</i>	
	5. Data Berita dihapus dari <i>database</i>
Skenario alternatif	

S1-5a	Saat validasi, sistem menampilkan pesan bahwa data yang <i>diinput</i> belum lengkap
S1-5b	Sistem memberikan kesempatan pada aktor untuk mengulang <i>input</i> data Berita.
S2-6a	Saat validasi, sistem menampilkan pesan bahwa data yang dimodifikasi belum lengkap
S2-6b	Sistem memberikan kesempatan pada aktor untuk mengulang modifikasi data Berita
S3-3a	Sistem menampilkan pesan peringatan hapus data, dan aktor memilih <i>no</i>
S3-3b	Sistem memberikan kesempatan pada aktor untuk mengulang memilih data Berita untuk dihapus
Post condition	Aktor berhasil mengelola data Berita

f. Deskripsi *Use case* mengelola data penduduk

Deskripsi *Use case* digunakan untuk menceritakan secara ringkas bagaimana tahapan menggunakan sistem dan apa saja yang bisa dilakukannya. Melalui deskripsi *Use case* dapat diketahui fungsi-fungsi apa saja yang ada pada sistem, berikut ini merupakan deskripsi *Use case* penduduk yang berfungsi menceritakan langkah-langkah bagaimana user menggunakan fungsi yang terdapat pada halaman penduduk.

Tabel 4.6 Deskripsi *Use case* Mengelola Data Penduduk

Nama	Mengelola data penduduk
Id Use case	UC.6.
Aktor	Admin
Deskripsi	Aktor dapat menambah, mengubah dan menghapus data penduduk
Exception	Pengolahan data tidak dapat dilakukan jika belum <i>login</i>
Pre condition	Aktor telah dalam keadaan <i>login</i>
Aktor	Sistem
Skenario normal	
1. Aktor memilih menu data penduduk	2. Menampilkan halaman menu data penduduk

3. Kondisi : – Jika aktor memilih tombol tambah pada halaman data penduduk maka s1 berlaku – Jika aktor memilih tombol edit pada halaman data penduduk maka s2 berlaku – Jika aktor memilih tombol hapus pada halaman data penduduk maka s3 berlaku	
S1 – fungsi tambah	
1. Aktor memilih tombol tambah pada halaman data penduduk	
	2. Sistem menampilkan halaman tambah data penduduk
3. Aktor menginputkan data penduduk pada halaman tambah data penduduk	
4. Aktor klik tombol simpan	
	5. Validasi <i>input</i> data penduduk lengkap
	6. Menyimpan data ke dalam <i>database</i>
S2 – fungsi edit	
1. Aktor memilih data penduduk untuk diedit pada <i>gridview</i>	
2. Aktor klik tombol edit pada halaman data penduduk	
	3. Sistem menampilkan halaman edit data penduduk
4. Aktor memodifikasi data penduduk pada halaman edit data penduduk	
5. Aktor klik tombol update	
	6. Validasi modifikasi data penduduk lengkap
	7. Menyimpan data ke dalam <i>database</i>
S3 – fungsi hapus	
1. Aktor memilih data penduduk untuk dihapus pada <i>gridview</i>	
2. Aktor klik tombol hapus	
	3. Validasi apakah data akan dihapus, dengan menampilkan pesan “apakah data tersebut akan dihapus” pilih : <i>yes</i> atau <i>no</i>
4. Aktor menghapus data penduduk	

dengan memilih tombol <i>yes</i>		
		5. Data penduduk dihapus dari <i>database</i>
Skenario alternatif		
S1-5a	Saat validasi, sistem menampilkan pesan bahwa data yang <i>diinput</i> belum lengkap	
S1-5b	Sistem memberikan kesempatan pada aktor untuk mengulang <i>input</i> data penduduk.	
S2-6a	Saat validasi, sistem menampilkan pesan bahwa data yang dimodifikasi belum lengkap	
S2-6b	Sistem memberikan kesempatan pada aktor untuk mengulang modifikasi data penduduk	
S3-3a	Sistem menampilkan pesan peringatan hapus data, dan aktor memilih <i>no</i>	
S3-3b	Sistem memberikan kesempatan pada aktor untuk mengulang memilih data penduduk untuk dihapus	
Post condition		Aktor berhasil mengelola data penduduk

g. Deskripsi *Use case* mengelola data pendatang baru

Deskripsi *Use case* digunakan untuk menceritakan secara ringkas bagaimana tahapan menggunakan sistem dan apa saja yang bisa dilakukannya. Melalui deskripsi *Use case* dapat diketahui fungsi-fungsi apa saja yang ada pada sistem, berikut ini merupakan deskripsi *Use case* pendatang baru yang berfungsi menceritakan langkah-langkah bagaimana user menggunakan fungsi yang terdapat pada halaman pendatang baru.

Tabel 4.7 Deskripsi *Use case* Mengelola Data Pendatang baru

Nama	Mengelola data pendatang baru
Id Use case	UC.7.
Aktor	Admin
Deskripsi	Aktor dapat menambah, mengubah dan menghapus data pendatang baru
Exception	Pengolahan data tidak dapat dilakukan jika belum <i>login</i>
Pre condition	Aktor telah dalam keadaan <i>login</i>
Aktor	Sistem

Skenario normal	
1. Aktor memilih menu data pendatang baru	
	2. Menampilkan halaman menu data pendatang baru
3. Kondisi : – Jika aktor memilih tombol tambah pada halaman data pendatang baru maka s1 berlaku – Jika aktor memilih tombol edit pada halaman data pendatang baru maka s2 berlaku – Jika aktor memilih tombol hapus pada halaman data pendatang baru maka s3 berlaku	
S1 – fungsi tambah	
1. Aktor memilih tombol tambah pada halaman data pendatang baru	
	2. Sistem menampilkan halaman tambah data pendatang baru
3. Aktor menginputkan data pendatang baru pada halaman tambah data pendatang baru	
4. Aktor klik tombol simpan	
	5. Validasi <i>input</i> data pendatang baru lengkap
	6. Menyimpan data ke dalam <i>database</i>
S2 – fungsi edit	
1. Aktor memilih data pendatang baru untuk diedit pada <i>gridview</i>	
2. Aktor klik tombol edit pada halaman data pendatang baru	
	3. Sistem menampilkan halaman edit data pendatang baru
4. Aktor memodifikasi data pendatang baru pada halaman edit data pendatang baru	
5. Aktor klik tombol update	
	6. Validasi modifikasi data pendatang baru lengkap
	7. Menyimpan data ke dalam <i>database</i>
S3 – fungsi hapus	
1. Aktor memilih data pendatang baru	

untuk dihapus pada <i>gridview</i>	
2. Aktor klik tombol hapus	
	3. Validasi apakah data akan dihapus, dengan menampilkan pesan “apakah data tersebut akan dihapus” pilih : <i>yes</i> atau <i>no</i>
4. Aktor menghapus data pendatang baru dengan memilih tombol <i>yes</i>	
	5. Data pendatang baru dihapus dari <i>database</i>
Skenario alternatif	
S1-5a	Saat validasi, sistem menampilkan pesan bahwa data yang diinput belum lengkap
S1-5b	Sistem memberikan kesempatan pada aktor untuk mengulang <i>input</i> data pendatang baru.
S2-6a	Saat validasi, sistem menampilkan pesan bahwa data yang dimodifikasi belum lengkap
S2-6b	Sistem memberikan kesempatan pada aktor untuk mengulang modifikasi data pendatang baru
S3-3a	Sistem menampilkan pesan peringatan hapus data, dan aktor memilih <i>no</i>
S3-3b	Sistem memberikan kesempatan pada aktor untuk mengulang memilih data pendatang baru untuk dihapus
Post condition	Aktor berhasil mengelola data pendatang baru

h. Deskripsi *Use case* mengelola data perpindahan

Deskripsi *Use case* digunakan untuk menceritakan secara ringkas bagaimana tahapan menggunakan sistem dan apa saja yang bisa dilakukannya. Melalui deskripsi *Use case* dapat diketahui fungsi-fungsi apa saja yang ada pada sistem, berikut ini merupakan deskripsi *Use case* perpindahan yang berfungsi menceritakan langkah-langkah bagaimana user menggunakan fungsi yang terdapat pada halaman perpindahan.

Tabel 4.8 Deskripsi Use case Mengelola Data Perpindahan

Nama	Mengelola data perpindahan	
Id Use case	UC.8.	
Aktor	Admin	
Deskripsi	Aktor dapat menambah, mengubah dan menghapus data perpindahan	
Exception	Pengolahan data tidak dapat dilakukan jika belum <i>login</i>	
Pre condition	Aktor telah dalam keadaan <i>login</i>	
Aktor		Sistem
Skenario normal		
1. Aktor memilih menu data perpindahan		
		2. Menampilkan halaman menu data perpindahan
3. Kondisi : – Jika aktor memilih tombol tambah pada halaman data perpindahan maka s1 berlaku – Jika aktor memilih tombol edit pada halaman data perpindahan maka s2 berlaku – Jika aktor memilih tombol hapus pada halaman data perpindahan maka s3 berlaku		
S1 – fungsi tambah		
1. Aktor memilih tombol tambah pada halaman data perpindahan		
		2. Sistem menampilkan halaman tambah data perpindahan
3. Aktor menginputkan data perpindahan pada halaman tambah data perpindahan		
4. Aktor klik tombol simpan		
		5. Validasi <i>input</i> data perpindahan lengkap
		6. Menyimpan data ke dalam <i>database</i>
S2 – fungsi edit		
1. Aktor memilih data perpindahan untuk diedit pada <i>gridview</i>		
2. Aktor klik tombol edit pada halaman data perpindahan		
		3. Sistem menampilkan halaman edit data perpindahan

4. Aktor memodifikasi data perpindahan pada halaman edit data perpindahan	
5. Aktor klik tombol update	
	6. Validasi modifikasi data perpindahan lengkap
	7. Menyimpan data ke dalam <i>database</i>
S3 – fungsi hapus	
1. Aktor memilih data perpindahan untuk dihapus pada <i>gridview</i>	
2. Aktor klik tombol hapus	
	3. Validasi apakah data akan dihapus, dengan menampilkan pesan “apakah data tersebut akan dihapus” pilih : <i>yes</i> atau <i>no</i>
4. Aktor menghapus data perpindahan dengan memilih tombol <i>yes</i>	
	5. Data perpindahan dihapus dari <i>database</i>
Skenario alternatif	
S1-5a	Saat validasi, sistem menampilkan pesan bahwa data yang <i>diinput</i> belum lengkap
S1-5b	Sistem memberikan kesempatan pada aktor untuk mengulang <i>input</i> data perpindahan.
S2-6a	Saat validasi, sistem menampilkan pesan bahwa data yang dimodifikasi belum lengkap
S2-6b	Sistem memberikan kesempatan pada aktor untuk mengulang modifikasi data perpindahan
S3-3a	Sistem menampilkan pesan peringatan hapus data, dan aktor memilih <i>no</i>
S3-3b	Sistem memberikan kesempatan pada aktor untuk mengulang memilih data perpindahan untuk dihapus
Post condition	Aktor berhasil mengelola data perpindahan

i. Deskripsi *Use case* mengelola data kelahiran

Deskripsi *Use case* digunakan untuk menceritakan secara ringkas bagaimana tahapan menggunakan sistem dan apa saja yang bisa dilakukannya. Melalui deskripsi *Use case* dapat diketahui fungsi-fungsi

apa saja yang ada pada sistem, berikut ini merupakan deskripsi *Use case* kelahiran yang berfungsi menceritakan langkah-langkah bagaimana user menggunakan fungsi yang terdapat pada halaman kelahiran.

Tabel 4.9 Deskripsi *Use case* Mengelola Data Kelahiran

Nama	Mengelola data kelahiran
<i>Id Use case</i>	UC.9.
Aktor	Admin
Deskripsi	Aktor dapat menambah, mengubah dan menghapus data kelahiran
<i>Exception</i>	Pengolahan data tidak dapat dilakukan jika belum <i>login</i>
<i>Pre condition</i>	Aktor telah dalam keadaan <i>login</i>
Aktor	Sistem
Skenario normal	
1. Aktor memilih menu data kelahiran	
	2. Menampilkan halaman menu data kelahiran
3. Kondisi : – Jika aktor memilih tombol tambah pada halaman data kelahiran maka s1 berlaku – Jika aktor memilih tombol edit pada halaman data kelahiran maka s2 berlaku – Jika aktor memilih tombol hapus pada halaman data kelahiran maka s3 berlaku	
S1 – fungsi tambah	
1. Aktor memilih tombol tambah pada halaman data kelahiran	
	2. Sistem menampilkan halaman tambah data kelahiran
3. Aktor <i>inputkan</i> data kelahiran pada halaman tambah data kelahiran	
4. Aktor klik tombol simpan	
	5. Validasi <i>input</i> data kelahiran lengkap
	6. Menyimpan data ke dalam <i>database</i>
S2 – fungsi edit	
1. Aktor memilih data kelahiran untuk	

	diedit pada <i>gridview</i>	
2.	Aktor klik tombol edit pada halaman data kelahiran	
		3. Sistem menampilkan halaman edit data kelahiran
4.	Aktor memodifikasi data kelahiran pada halaman edit data kelahiran	
5.	Aktor klik tombol update	
		6. Validasi modifikasi data kelahiran lengkap
		7. Menyimpan data ke dalam <i>database</i>
S3 – fungsi hapus		
1.	Aktor memilih data kelahiran untuk dihapus pada <i>gridview</i>	
2.	Aktor klik tombol hapus	
		3. Validasi apakah data akan dihapus, dengan menampilkan pesan “apakah data tersebut akan dihapus” pilih : <i>yes</i> atau <i>no</i>
4.	Aktor menghapus data kelahiran dengan memilih tombol <i>yes</i>	
		5. Data kelahiran dihapus dari <i>database</i>
Skenario alternatif		
S1-5a	Saat validasi, sistem menampilkan pesan bahwa data yang diinput belum lengkap	
S1-5b	Sistem memberikan kesempatan pada aktor untuk mengulang <i>input</i> data kelahiran.	
S2-6a	Saat validasi, sistem menampilkan pesan bahwa data yang dimodifikasi belum lengkap	
S2-6b	Sistem memberikan kesempatan pada aktor untuk mengulang modifikasi data kelahiran	
S3-3a	Sistem menampilkan pesan peringatan hapus data, dan aktor memilih <i>no</i>	
S3-3b	Sistem memberikan kesempatan pada aktor untuk mengulang memilih data kelahiran untuk dihapus	
Post condition	Aktor berhasil mengelola data kelahiran	

j. Deskripsi *Use case* mengelola data kematian

Deskripsi *Use case* digunakan untuk menceritakan secara ringkas bagaimana tahapan menggunakan sistem dan apa saja yang bisa dilakukannya. Melalui deskripsi *Use case* dapat diketahui fungsi-fungsi apa saja yang ada pada sistem, berikut ini merupakan deskripsi *Use case* kematian yang berfungsi menceritakan langkah-langkah bagaimana user menggunakan fungsi yang terdapat pada halaman kematian.

Tabel 4.10 Deskripsi *Use case* Mengelola Data Kematian

Nama	Mengelola data kematian
Id Use case	UC.10.
Aktor	Admin
Deskripsi	Aktor dapat menambah, mengubah dan menghapus data kematian
Exception	Pengolahan data tidak dapat dilakukan jika belum <i>login</i>
Pre condition	Aktor telah dalam keadaan <i>login</i>
Aktor	Sistem
Skenario normal	
1. Aktor memilih menu data kematian	2. Menampilkan halaman menu data kematian
3. Kondisi : – Jika aktor memilih tombol tambah pada halaman data kematian maka s1 berlaku – Jika aktor memilih tombol edit pada halaman data kematian maka s2 berlaku – Jika aktor memilih tombol hapus pada halaman data kematian maka s3 berlaku	
S1 – fungsi tambah	
1. Aktor memilih tombol tambah pada halaman data kematian	2. Sistem menampilkan halaman

	tambah data kematian
3. Aktor menginputkan data kematian pada halaman tambah data kematian	
4. Aktor klik tombol simpan	
	5. Validasi <i>input</i> data kematian lengkap
	6. Menyimpan data ke dalam <i>database</i>
S2 – fungsi edit	
1. Aktor memilih data kematian untuk diedit pada <i>gridview</i>	
2. Aktor klik tombol edit pada halaman data kematian	
	3. Sistem menampilkan halaman edit data kematian
4. Aktor memodifikasi data kematian pada halaman edit data kematian	
5. Aktor klik tombol update	
	6. Validasi modifikasi data kematian lengkap
	7. Menyimpan data ke dalam <i>database</i>
S3 – fungsi hapus	
1. Aktor memilih data kematian untuk dihapus pada <i>gridview</i>	
2. Aktor klik tombol hapus	
	3. Validasi apakah data akan dihapus, dengan menampilkan pesan “apakah data tersebut akan dihapus” pilih : <i>yes</i> atau <i>no</i>
4. Aktor menghapus data kematian dengan memilih tombol <i>yes</i>	
	5. Data kematian dihapus dari <i>database</i>
Skenario alternatif	
S1-5a	Saat validasi, sistem menampilkan pesan bahwa data yang diinput belum lengkap
S1-5b	Sistem memberikan kesempatan pada aktor untuk mengulang <i>input</i> data kematian.
S2-6a	Saat validasi, sistem menampilkan pesan bahwa data yang dimodifikasi belum lengkap
S2-6b	Sistem memberikan kesempatan pada aktor untuk mengulang modifikasi data kematian

1. Deskripsi *Use case Logout*

Berikut ini merupakan Deskripsi *Use case Logout* yang berfungsi menceritakan langkah-langkah bagaimana user untuk keluar dari sistem.

Tabel 4.12 Deskripsi *Use case Logout*

Nama	<i>Logout</i>	
Id Use case	UC.11.	
Aktor	Admin	
Deskripsi	Aktor tidak melakukan pengolahan data dan keluar dari sistem	
Exception	Koneksi gagal ditutup	
Precondition	Aktor sudah <i>Login</i>	
	Aktor	Sistem
Skenario normal		
	1. Aktor klik tombol <i>Logout</i>	
		2. Menutup koneksi ke <i>database</i> aktor
		3. Koneksi berhasil ditutup.
Post condition	Aktor keluar dari sistem	

2. Deskripsi *Use case Penduduk*

Deskripsi *Use case Penduduk* merupakan upaya pengolahan data menjadi sesuatu yang dapat diutarakan secara jelas dan tepat dengan tujuan agar dapat dimengerti sebagai berikut ini.

a. Deskripsi *Use case Login*

Deskripsi *Use case* digunakan untuk menceritakan secara ringkas bagaimana tahapan menggunakan sistem dan apa saja yang bisa dilakukannya. Melalui deskripsi *Use case* dapat diketahui fungsi apa saja yang ada pada sistem, berikut ini merupakan *Use case login* yang berfungsi menceritakan langkah-langkah bagaimana user agar dapat masuk ke dalam sistem

Tabel 4.13 Deskripsi Use case login

Nama	<i>Login</i>	
ID Use case	UC.12.	
Aktor	Penduduk	
Deskripsi	Aktor melakukan <i>Login</i> untuk otentikasi dalam pengecekan dan modifikasi data	
Exception	Proses <i>Login</i> gagal, dikarenakan <i>username</i> atau <i>password</i> tidak sesuai, kosong atau belum terdata sebelumnya.	
Pre condition	<i>Username</i> dan <i>Password</i> telah tersimpan dalam <i>database</i>	
	Aktor	Sistem
Skenario normal		
	1. Aktor melakukan <i>Login</i> dengan menginputkan <i>Username</i> dan <i>Password</i>	
	2. Aktor mengklik <i>Login</i>	
		3. Mencocokkan <i>Username</i> dan <i>Username</i> dengan database
		4. Data cocok, sistem menampilkan Halaman menu utama
Skenario alternatif		
3.a	Aktor memilih batal masuk menu utama dengan mengklik <i>Cancel</i>	
3.b	Tutup Halaman <i>login</i>	
4.a	Mencocokkan <i>username</i> dan <i>Password</i> dengan database, dan data tidak cocok	
4.b	Tampilkan pesan Peringatan data yang diinputkan salah, kemudian tampil <i>form login</i> dan Admin kembali inputkan <i>Username</i> dan <i>Password</i> dengan benar	
Post condition	Aktor sudah masuk sistem untuk mengolah data	

b. Deskripsi Use case melihat informasi Profil

Berikut ini adalah interaksi antar aktor dan sistem pada Use case dengan skenario yang telah di tentukan. Adapun skenario berikut adalah interaksi aktor untuk menampilkan informasi Profil yang akan dijelaskan pada tabel berikut:

Tabel 4.14 Deskripsi Use case Melihat Informasi Profil

Nama	Melihat Informasi Profil
Id Use case	UC.13.
Aktor	Penduduk

Deskripsi	Aktor dapat melihat informasi Profil	
Exception	-	
Pre condition	Data Profil sebelumnya telah diinput Admin	
Aktor		Sistem
Skenario normal		
1. Aktor Klik menu Profil		
		2. Membuka koneksi ke database
		3. Menampilkan halaman menu data Profil
4. Melihat informasi Profil yang ditampilkan		
Skenario alternatif		
-		
Post condition	Aktor berhasil melihat informasi Profil	

c. Deskripsi *Use case* melihat informasi Galeri

Berikut ini adalah interaksi antar aktor dan sistem pada *Use case* dengan skenario yang telah di tentukan. Adapun skenario berikut adalah interaksi aktor untuk menampilkan informasi Galeri yang akan dijelaskan pada tabel berikut:

Tabel 4.15 Deskripsi *Use case* Melihat Informasi Galeri

Nama	Melihat Informasi Galeri	
Id Use case	UC.14.	
Aktor	Penduduk	
Deskripsi	Aktor dapat melihat informasi Galeri	
Exception	-	
Pre condition	Data Galeri sebelumnya telah diinput Admin	
Aktor		Sistem
Skenario normal		
1. Aktor Klik menu Galeri		
		2. Membuka koneksi ke database
		3. Menampilkan halaman menu data Galeri
4. Melihat informasi Galeri yang ditampilkan		
Skenario alternatif		
-		
Post condition	Aktor berhasil melihat informasi Galeri	

d. Deskripsi *Use case* melihat informasi Berita

Berikut ini adalah interaksi antar aktor dan sistem pada *Use case* dengan skenario yang telah di tentukan. Adapun skenario berikut adalah interaksi aktor untuk menampilkan informasi Berita yang akan dijelaskan pada tabel berikut:

Tabel 4.16 Deskripsi *Use case* Melihat Informasi Berita

Nama	Melihat Informasi Berita	
Id Use case	UC.15.	
Aktor	Penduduk	
Deskripsi	Aktor dapat melihat informasi Berita	
Exception	-	
Pre condition	Data Berita sebelumnya telah diinput Admin	
Aktor		Sistem
Skenario normal		
1. Aktor Klik menu Berita		2. Membuka koneksi ke database
		3. Menampilkan halaman menu data Berita
4. Melihat informasi Berita yang ditampilkan		
Skenario alternatif		
-		
Post condition	Aktor berhasil melihat informasi Berita	

e. Deskripsi *Use case* Surat kependudukan

berikut ini adalah interaksi antar aktor dan sistem pada *Use case* dengan skenario yang telah di tentukan. adapun skenario berikut adalah interaksi aktor untuk Surat kependudukan yang akan dijelaskan pada tabel berikut:

Tabel 4.17 Deskripsi *Use case* Surat kependudukan

Nama	Surat Kependudukan
Id use case	UC.16.
Aktor	Penduduk

Deskripsi	Aktor masuk menampilkan halaman surat kependudukan	
Exception	Membuat surat tidak dapat dilakukan jika belum menampilkan halaman daftar	
Pre condition	Aktor telah mengakses menu utama	
Aktor		Sistem
Skenario normal		
1. Aktor memilih menu data		2. Menampilkan Halaman
3. Aktor menginputkan data pada Halaman data		
4. Aktor memilih tombol		
		5. Validasi data lengkap?
		6. <i>Input</i> data lengkap, menyimpan data ke dalam <i>database</i>
Skenario alternatif		
5a	Saat Validasi, sistem menampilkan pesan bahwa data yang diinput belum lengkap	
5b	Sistem memberikan kesempatan pada Aktor untuk mengulang <i>input</i> data Daftar.	
Post condition	Aktor berhasil surat kependudukan	

f. Deskripsi *Use case Logout*

Berikut ini merupakan Deskripsi *Use case Logout* yang berfungsi menceritakan langkah-langkah bagaimana user untuk keluar dari sistem.

Tabel 4.18 Deskripsi *Use case Logout*

Nama	<i>Logout</i>	
Id Use case	UC.17.	
Aktor	Penduduk	
Deskripsi	Aktor tidak melakukan pengolahan data dan keluar dari sistem	
Exception	Koneksi gagal ditutup	
Precondition	Aktor sudah <i>Login</i>	
	Aktor	Sistem
Skenario normal		
1. Aktor klik tombol <i>Logout</i>		2. Menutup koneksi ke <i>database</i> aktor
		3. Koneksi berhasil ditutup.
Post condition	Aktor keluar dari sistem	

4.3 ACTIVITY DIAGRAM

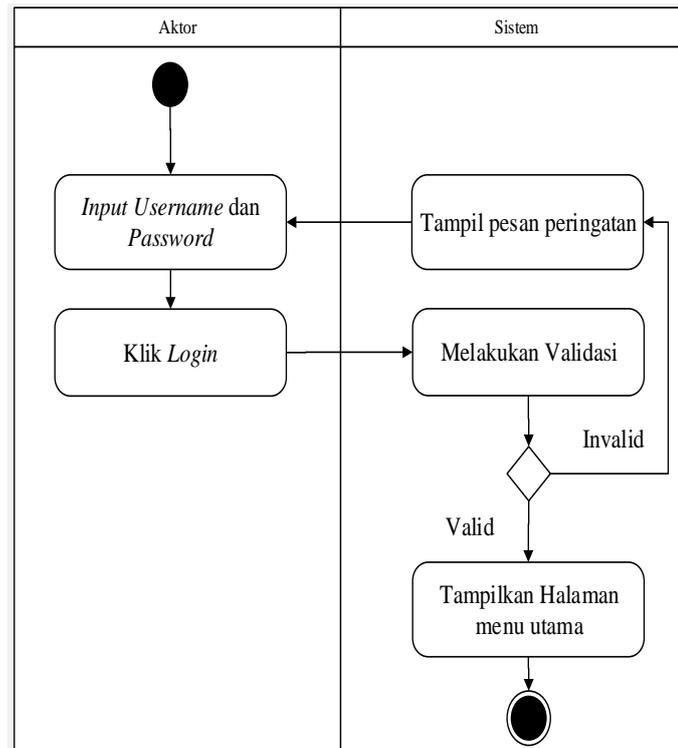
Activity Diagram adalah sebuah cara untuk memodelkan alur kerja (*workflow*) dari deskripsi *Use case* dalam bentuk grafik.

4.3.1 Activity Diagram Admin

Pada *Activity Diagram* berikut ini akan memperlihatkan aliran kendali dari suatu aktivitas ke aktivitas lainnya yang terjadi khususnya pada Admin sebagai aktornya.

1. Activity Diagram Login

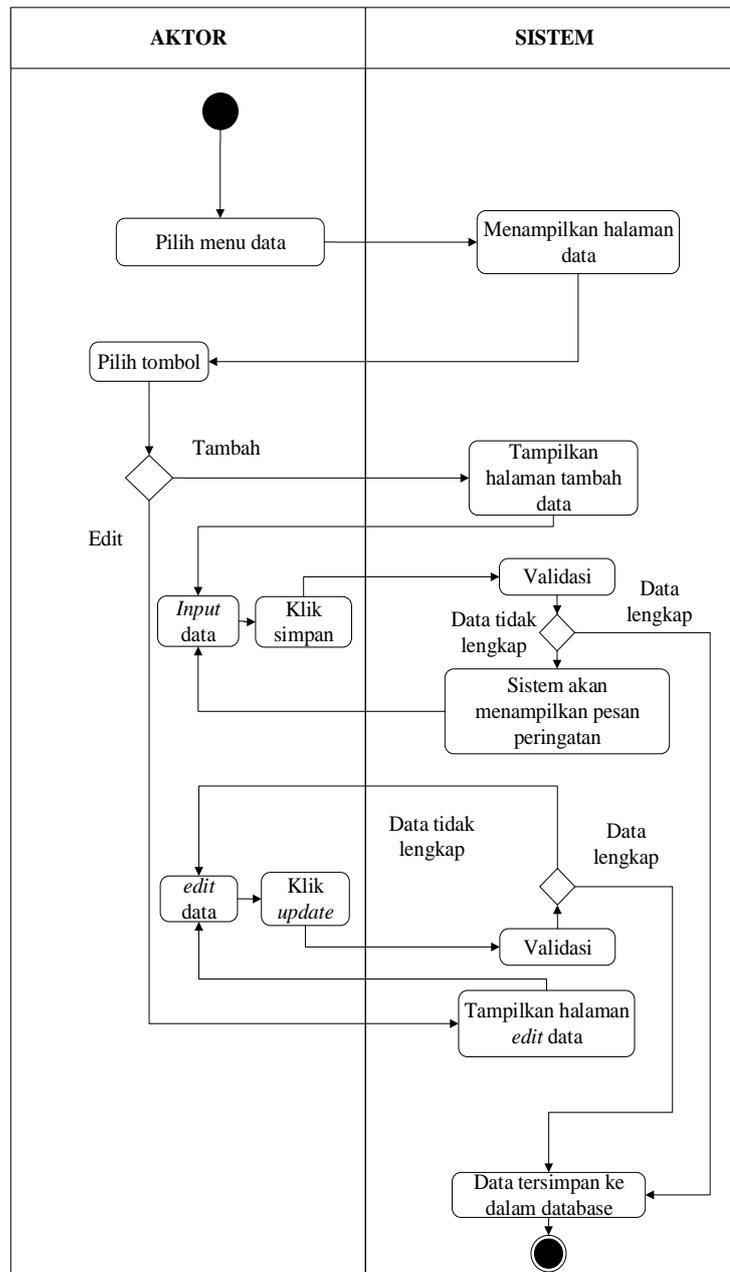
Activity Diagram Login merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari Admin untuk masuk ke sebuah sistem dengan tujuan dapat menampilkan informasi dan melakukan pengolahan data.



Gambar 4.4 Activity Diagram Login

2. *Activity Diagram* Mengolah Data Admin

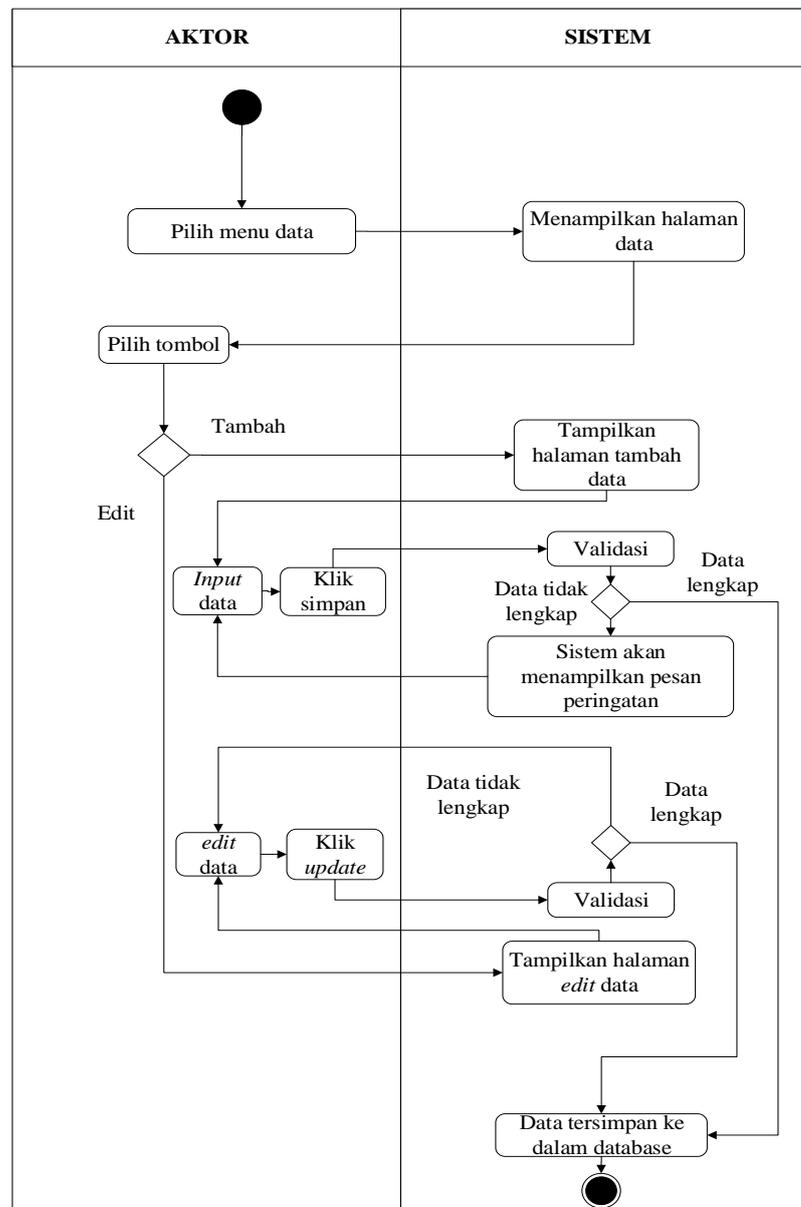
Activity Diagram Admin merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari Admin untuk dapat menampilkan informasi dan melakukan pengolahan data Admin.



Gambar 4.5 Activity Diagram Tambah Data Admin

3. *Activity Diagram* Mengolah Data Profil

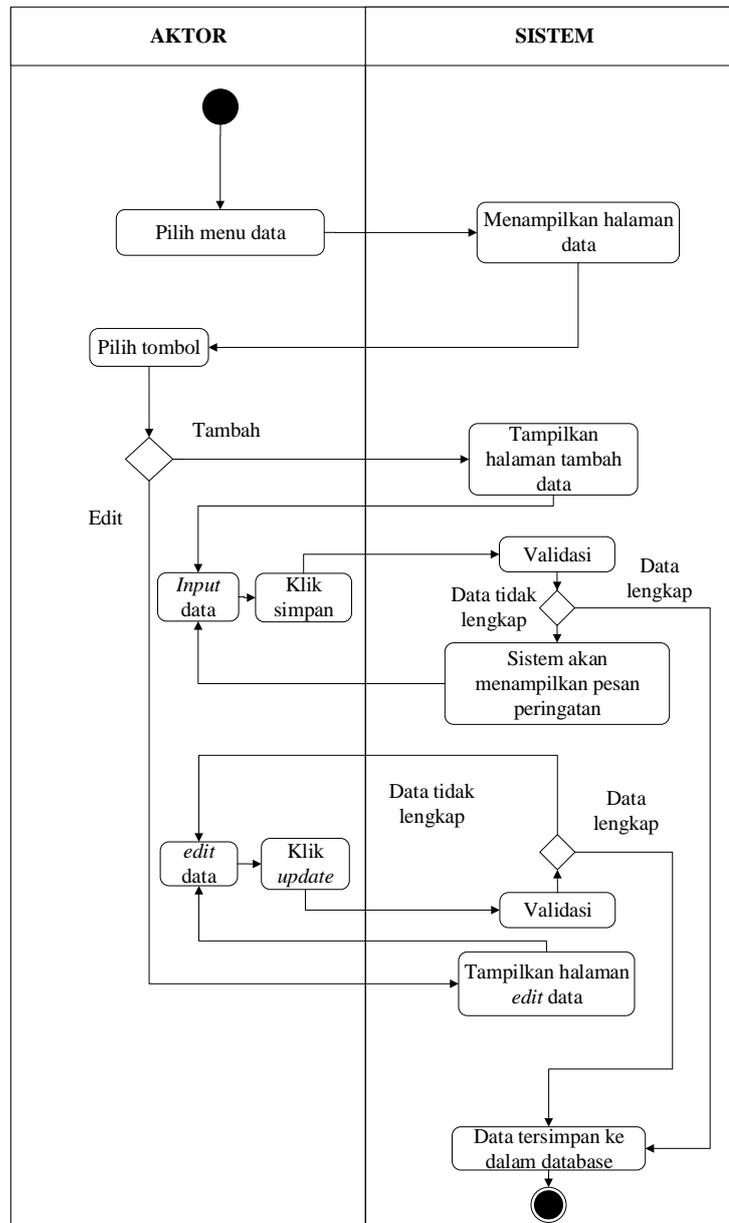
Activity Diagram Profil merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari Admin untuk dapat menampilkan informasi dan melakukan pengolahan data Profil.



Gambar 4.6 Activity Diagram Tambah Data Profil

4. *Activity Diagram* Mengolah Data Galeri

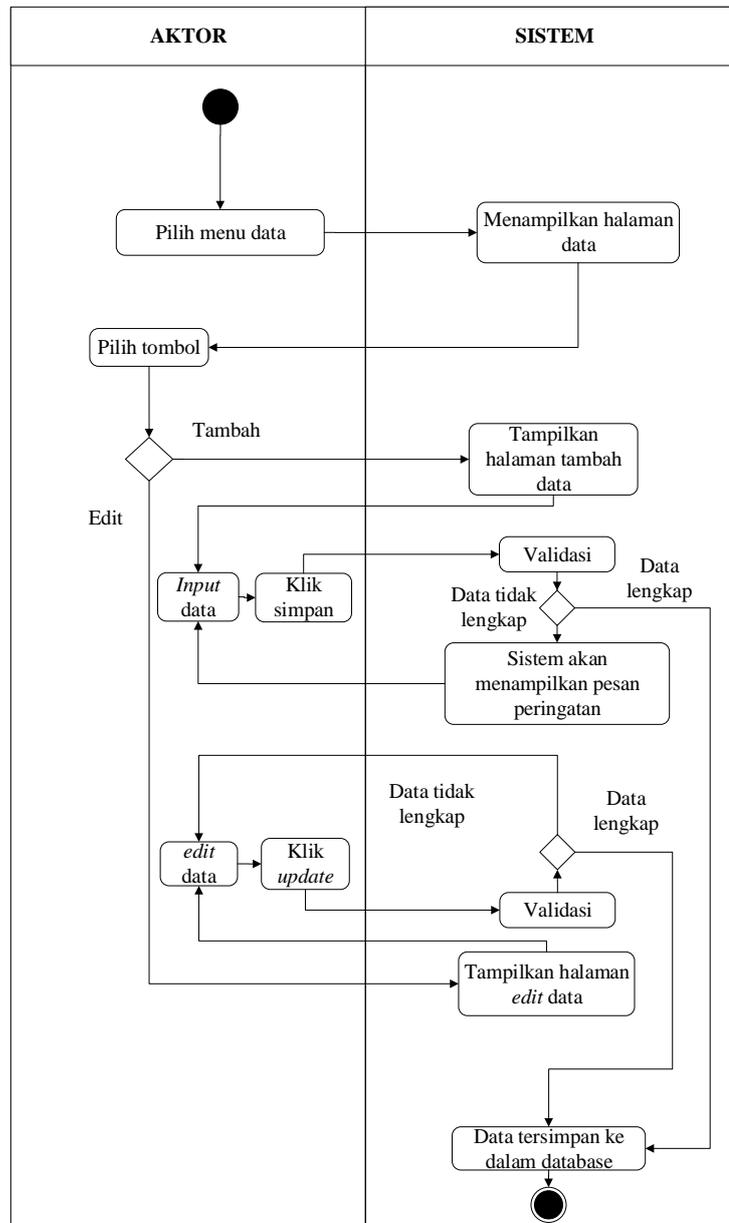
Activity Diagram Galeri merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari Admin untuk dapat menampilkan informasi dan melakukan pengolahan data Galeri.



Gambar 4.7 Activity Diagram Tambah Data Galeri

5. Activity Diagram Mengolah Data Berita

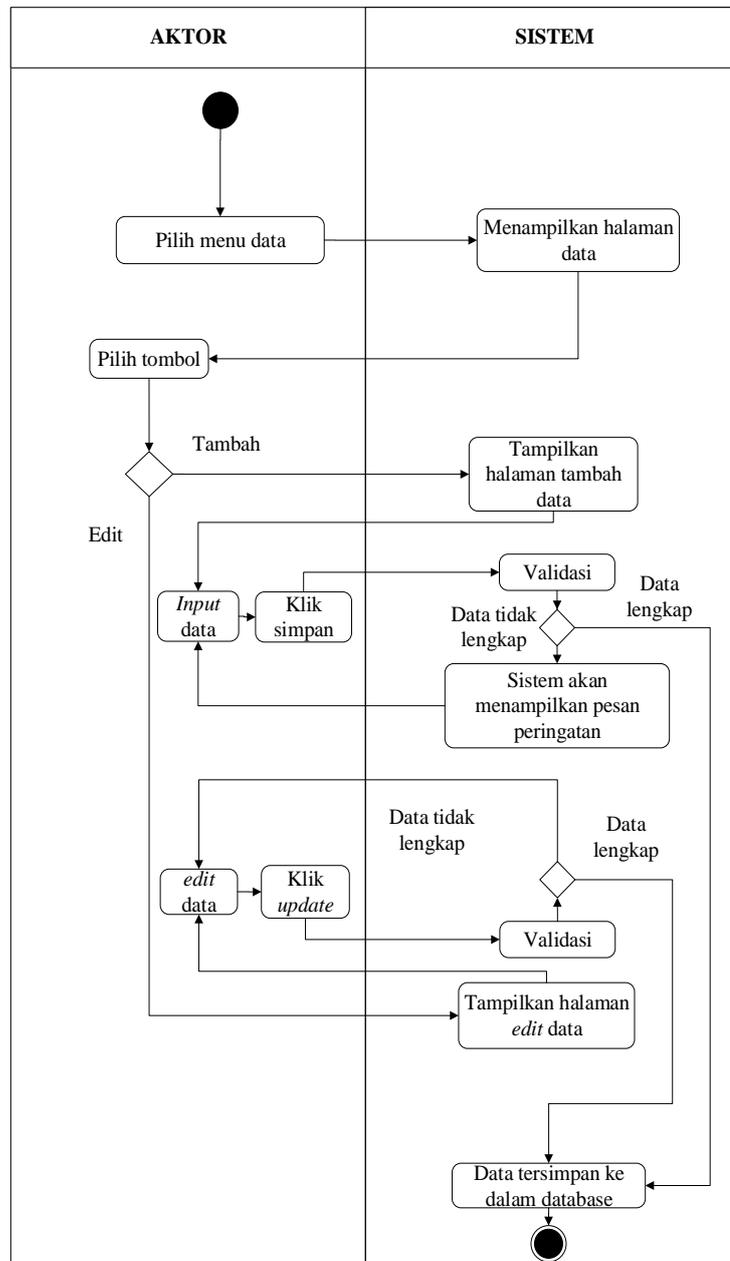
Activity Diagram Berita merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari Admin untuk dapat menampilkan informasi dan melakukan pengolahan data Berita.



Gambar 4.8 Activity Diagram Tambah Data Berita

6. Activity Diagram Mengolah Data Penduduk

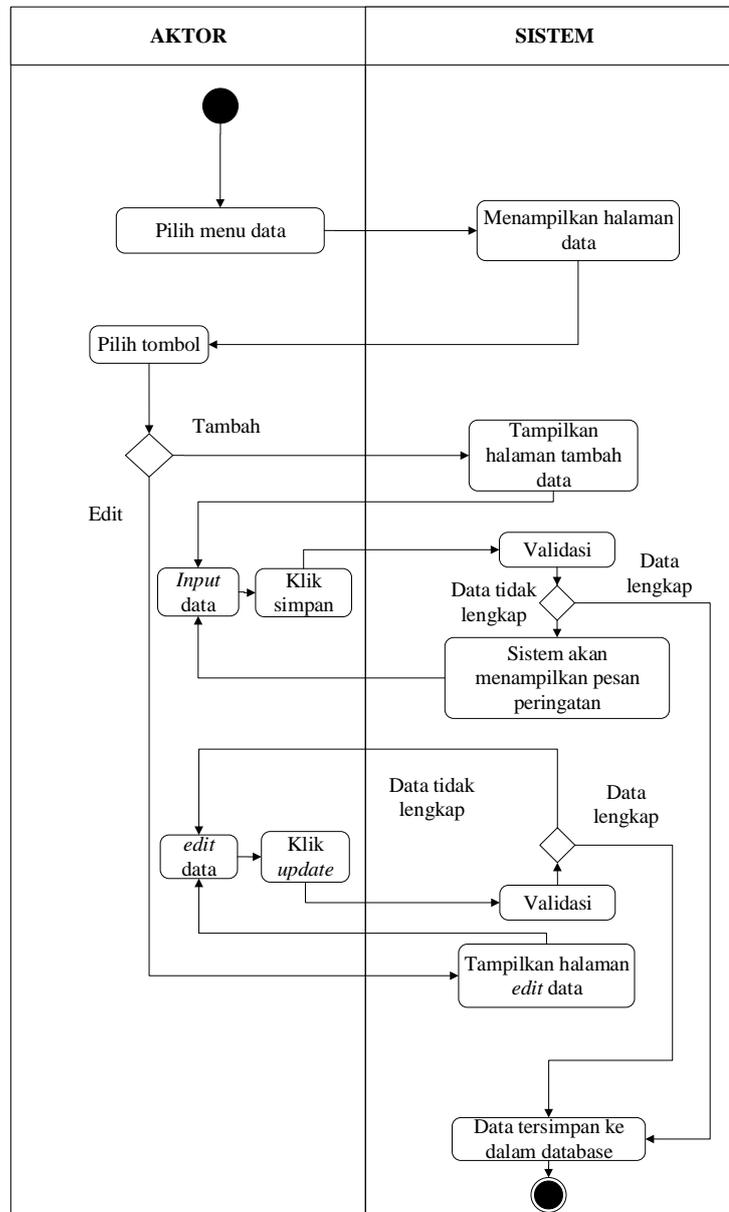
Activity Diagram Penduduk merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari Admin untuk dapat menampilkan informasi dan melakukan pengolahan data Penduduk.



Gambar 4.9 Activity Diagram Tambah Data Penduduk

7. Activity Diagram Mengolah Data Pendetang baru

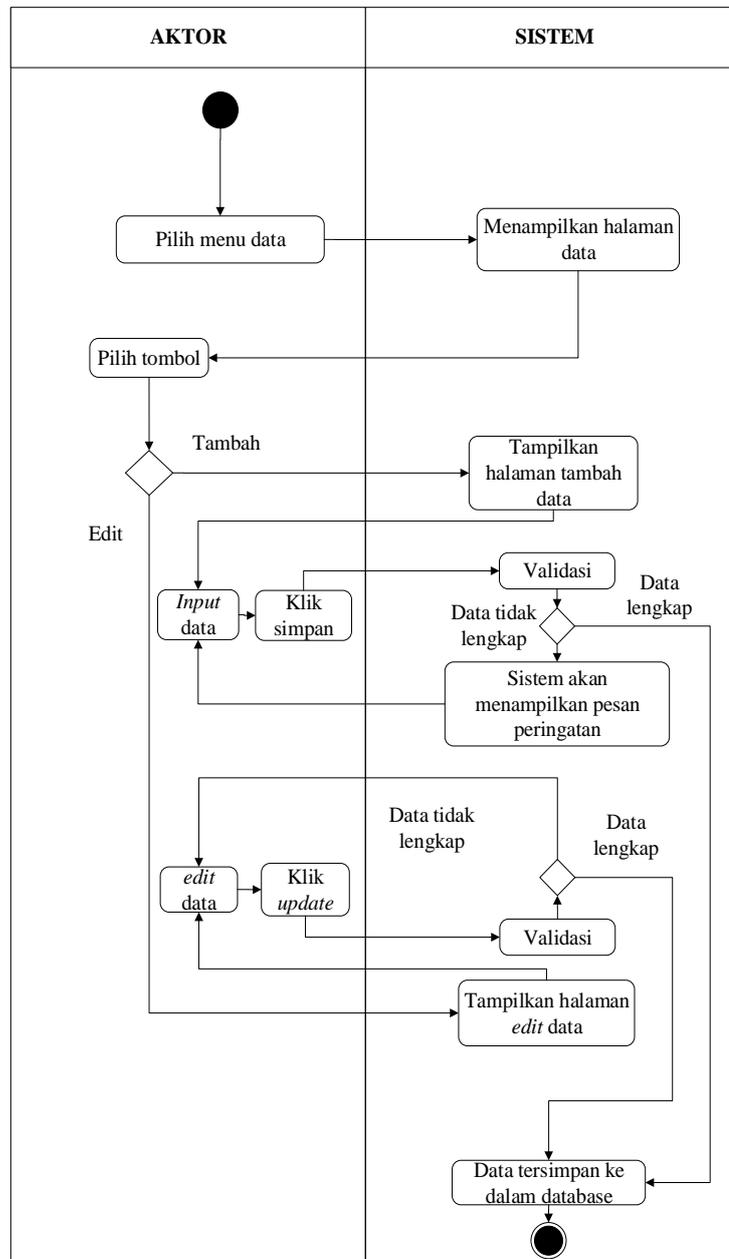
Activity Diagram Pendetang baru merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari Admin untuk dapat menampilkan informasi dan melakukan pengolahan data Pendetang baru.



Gambar 4.10 Activity Diagram Tambah Data Pendetang baru

8. Activity Diagram Mengolah Data Perpindahan

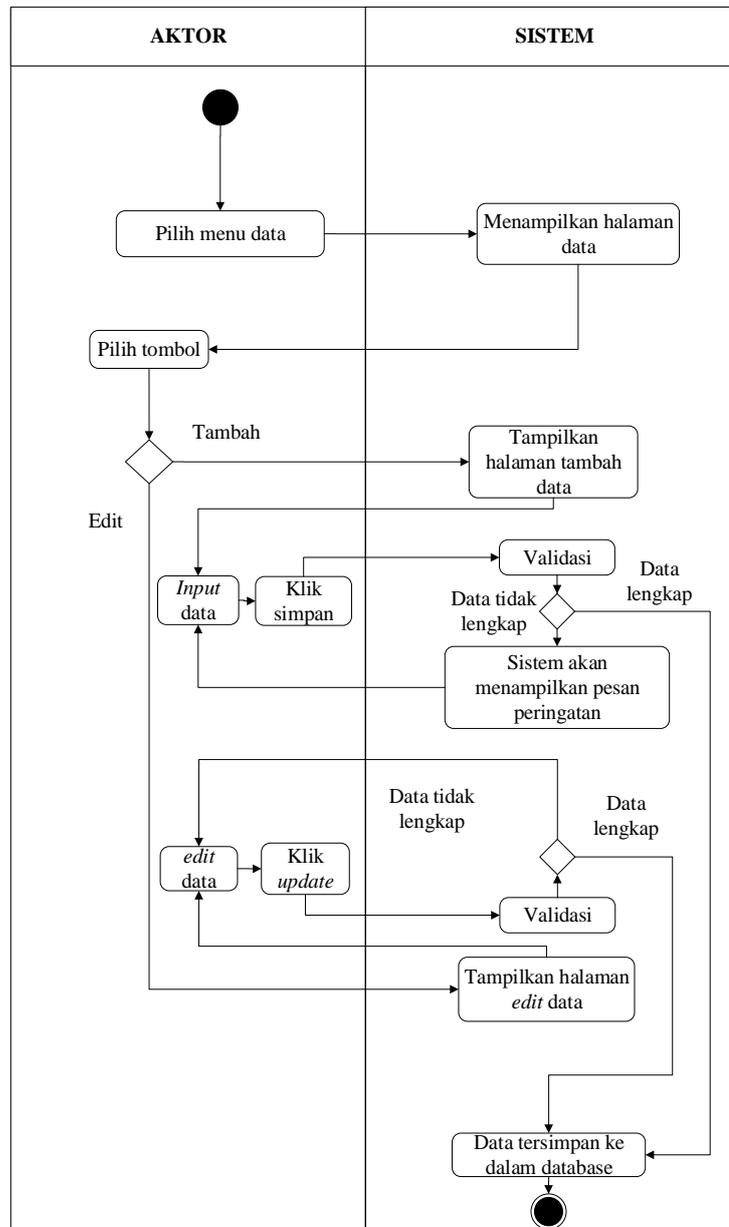
Activity Diagram Perpindahan merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari Admin untuk dapat menampilkan informasi dan melakukan pengolahan data Perpindahan.



Gambar 4.11 Activity Diagram Tambah Data Perpindahan

9. *Activity Diagram* Mengolah Data Kelahiran

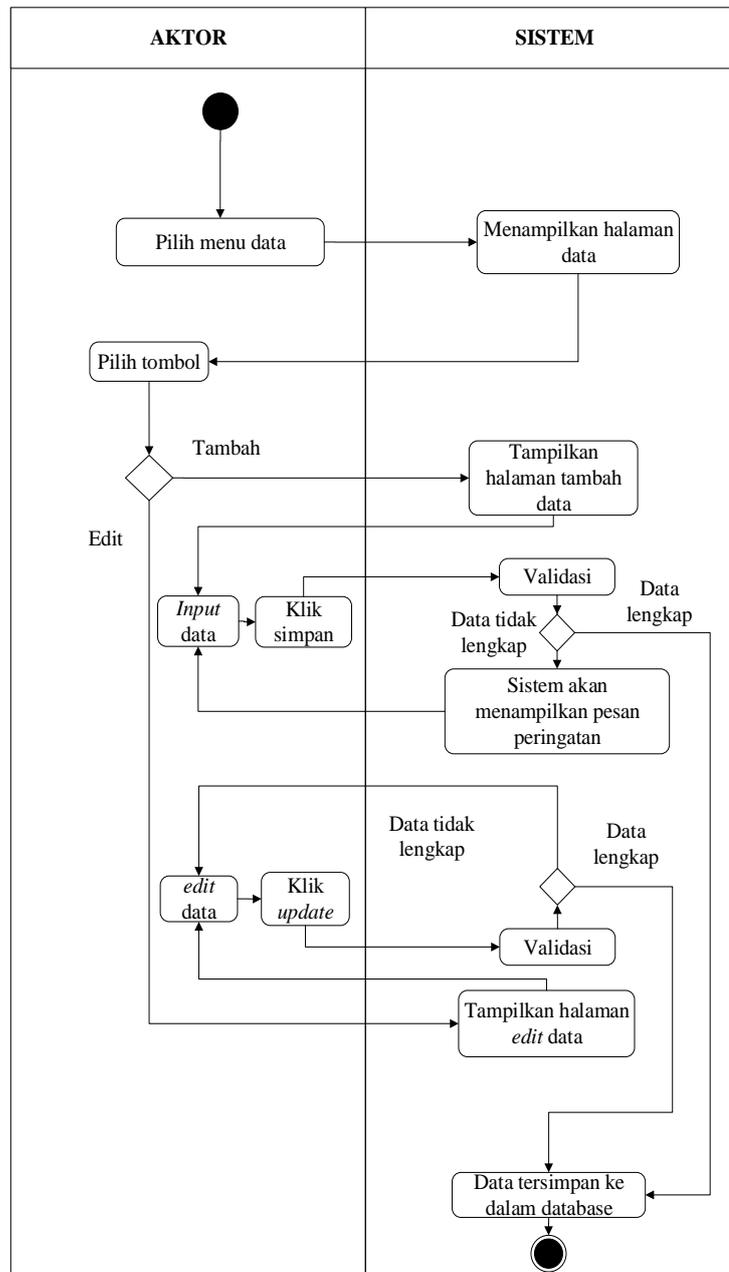
Activity Diagram Kelahiran merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari Admin untuk dapat menampilkan informasi dan melakukan pengolahan data Kelahiran.



Gambar 4.12 Activity Diagram Tambah Data Kelahiran

10. Activity Diagram Mengolah Data Kematian

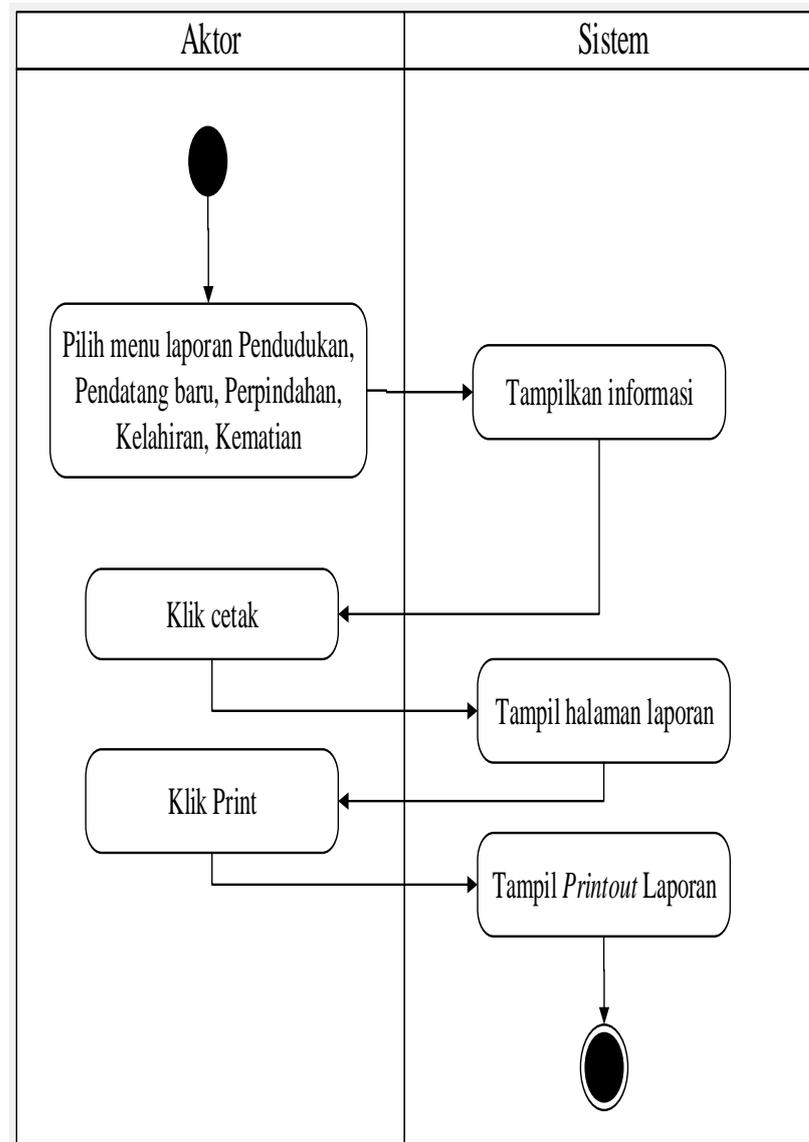
Activity Diagram Kematian merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari Admin untuk dapat menampilkan informasi dan melakukan pengolahan data Kematian.



Gambar 4.13 Activity Diagram Tambah Data Kematian

11. *Activity Diagram* Mencetak Laporan

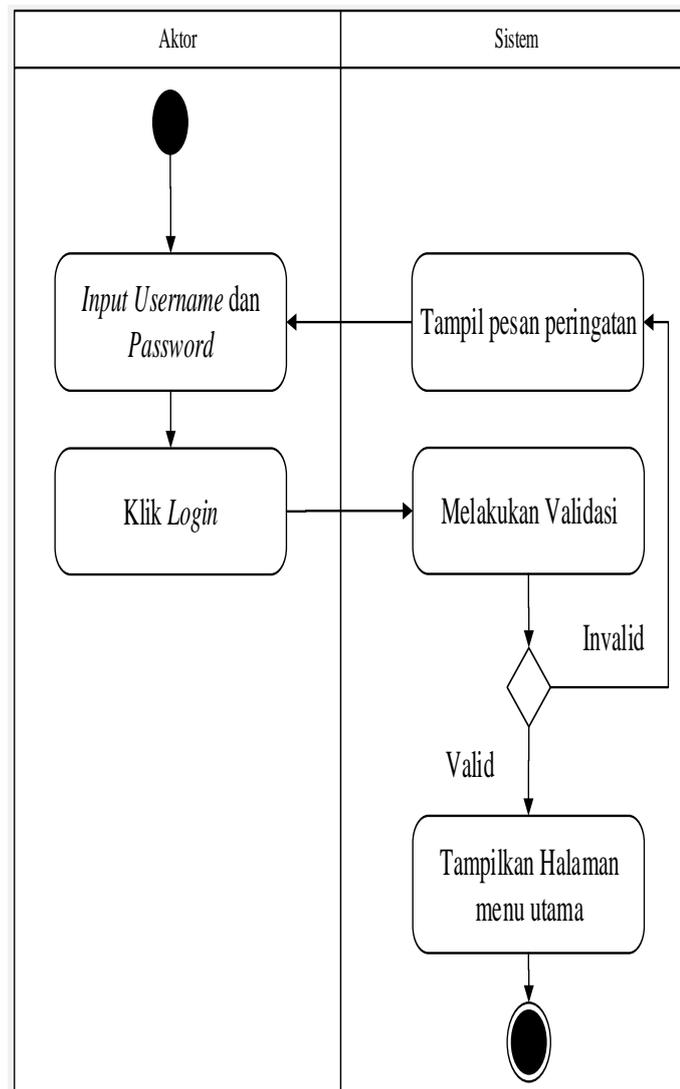
Activity Diagram Mencetak Laporan merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari Admin untuk dapat menampilkan informasi dan melakukan pengolahan data Mencetak Laporan.



Gambar 4.14 Mencetak Laporan

12. *Activity Diagram Logout*

Activity Diagram logout merupakan gambaran berbagai alir aktivitas dalam sistem yang sedang dirancang, pada *Activity Diagram logout* menggambarkan aktor keluar dari sistem.



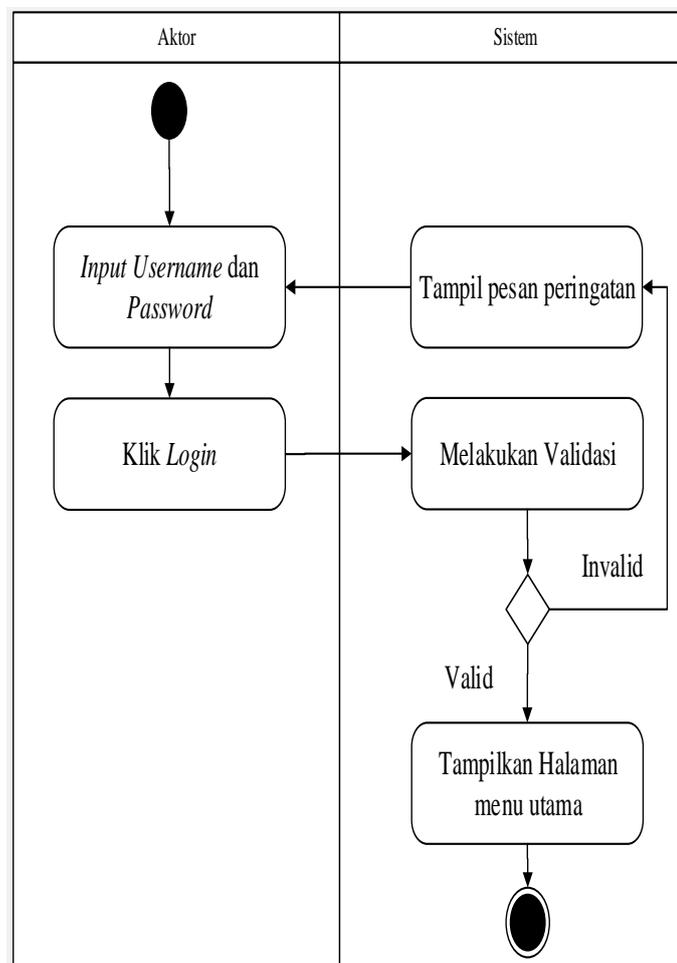
Gambar 4.15 Activity Diagram Logout

4.3.2 Activity Diagram Penduduk

Pada *Activity Diagram* berikut ini akan memperlihatkan aliran kendali dari suatu aktivitas ke aktivitas lainnya yang terjadi khususnya pada Penduduk sebagai aktornya.

13. Activity Diagram Login

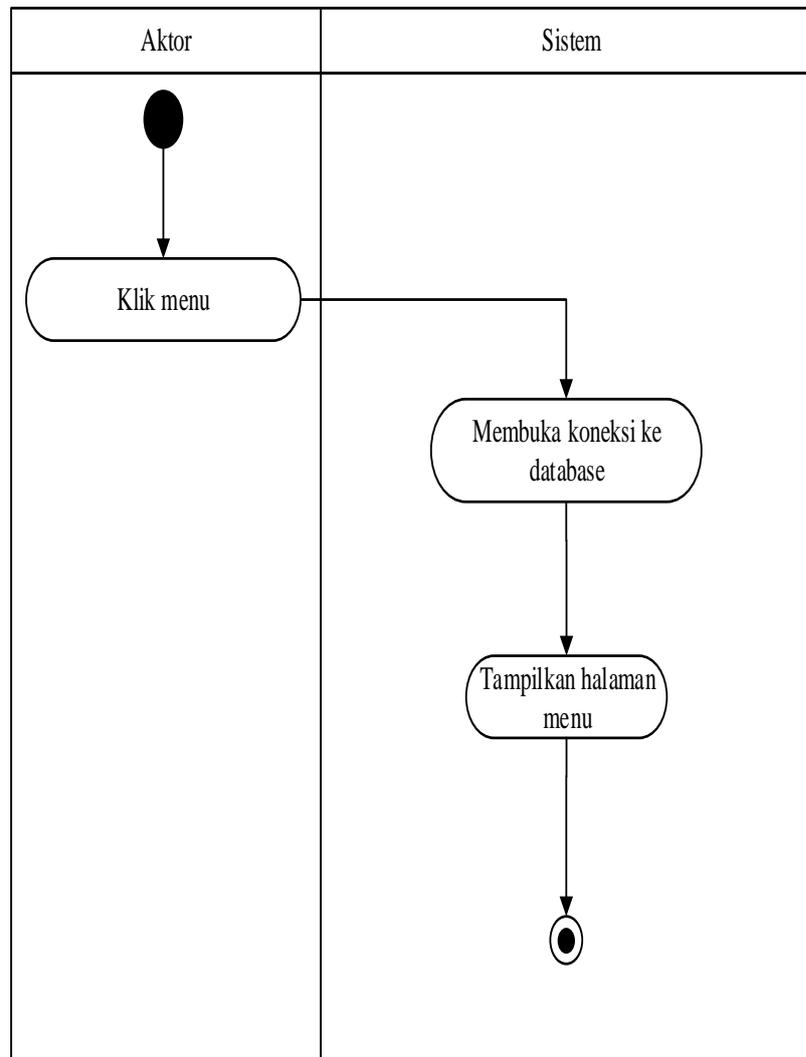
Activity Diagram Login merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari Admin untuk masuk ke sebuah sistem dengan tujuan dapat menampilkan informasi dan melakukan pengolahan data.



Gambar 4.16 Activity Diagram Login

14. *Activity Diagram* Melihat Informasi Profil

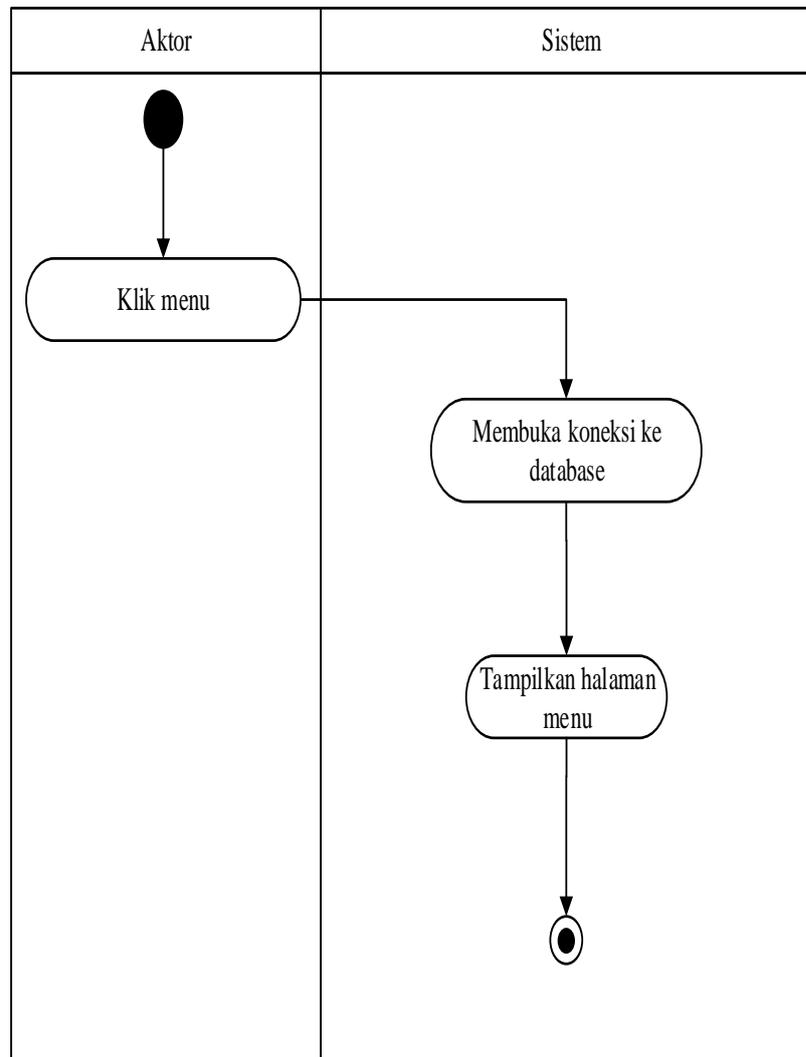
Activity Diagram Melihat informasi Profil merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari aktor untuk dapat melihat informasi Profil



Gambar 4.17 *Activity Diagram* Melihat Informasi Profil

15. *Activity Diagram* Melihat Informasi Galeri

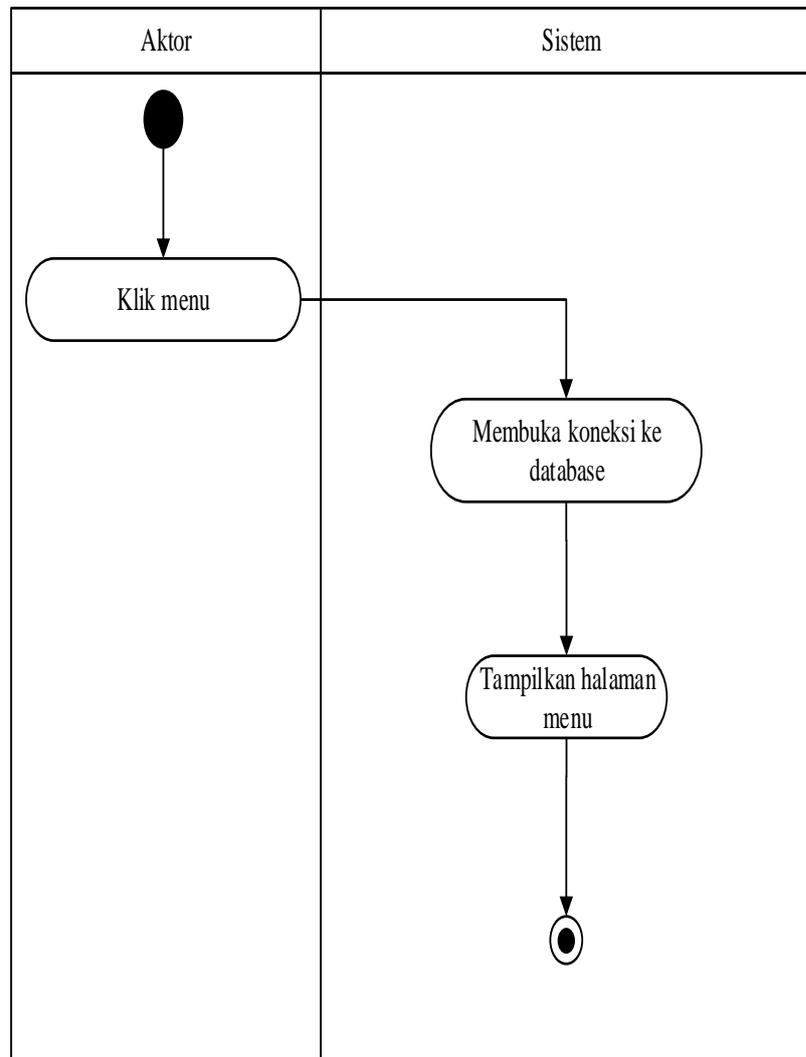
Activity Diagram Melihat informasi Galeri merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari aktor untuk dapat melihat informasi Galeri



Gambar 4.18 *Activity Diagram* Melihat Informasi Galeri

16. *Activity Diagram* Melihat Informasi Berita

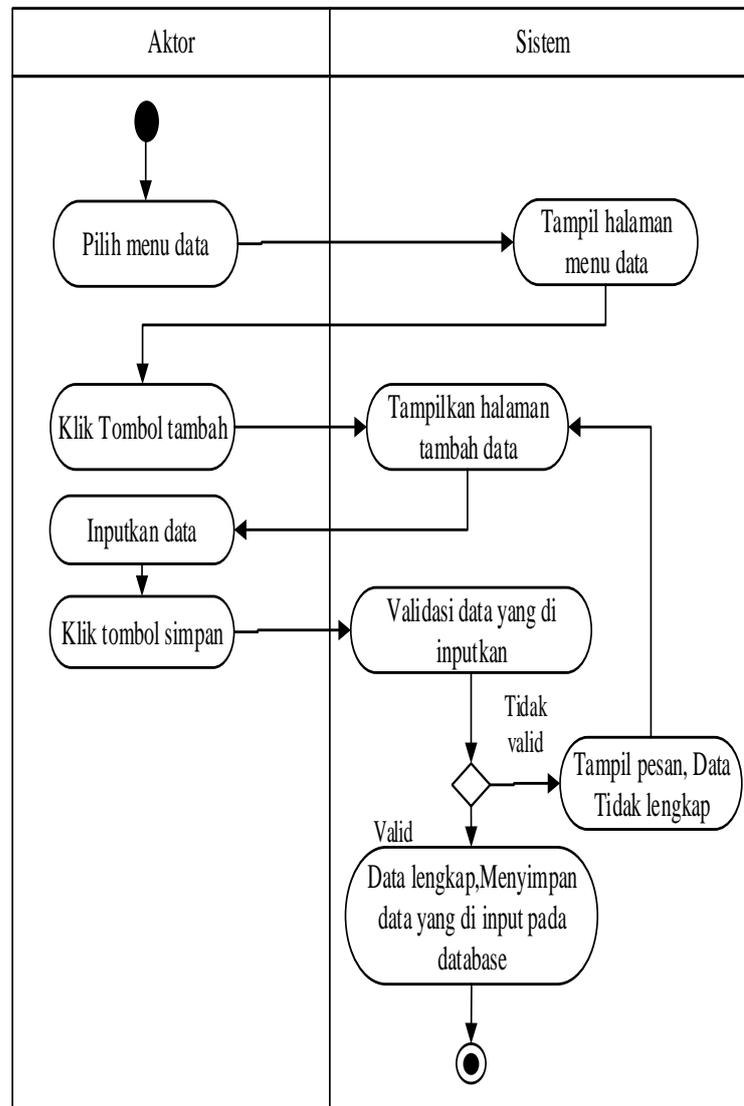
Activity Diagram Melihat informasi Berita merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari aktor untuk dapat melihat informasi Berita



Gambar 4.19 *Activity Diagram* Melihat Informasi Berita

17. *Activity Diagram* Surat kependudukan

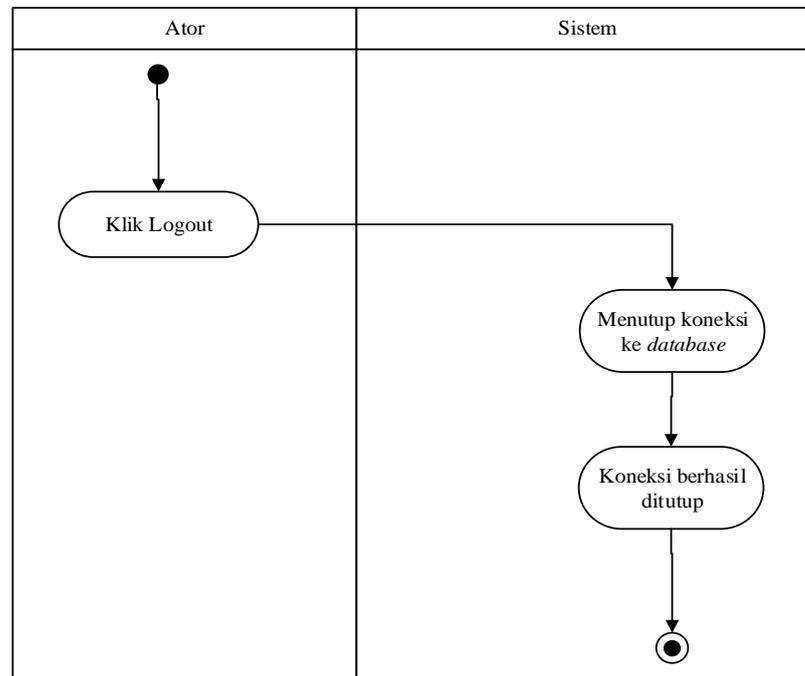
Activity Diagram Surat kependudukan merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari aktor untuk dapat melihat informasi Surat kependudukan



Gambar 4.20 *Activity Diagram* Surat kependudukan

18. *Activity Diagram Logout*

Activity Diagram logout merupakan gambaran berbagai alir aktivitas dalam sistem yang sedang dirancang, pada *Activity Diagram logout* menggambarkan aktor keluar dari sistem.



Gambar 4.21 Activity Diagram Logout

4.4 ANALISIS MASUKAN *OUTPUT* DAN *INPUT*

Berikut ini merupakan tahapan menganalisis data yang didapatkan pada tempat penelitian dimana data tersebut terbagi menjadi analisis *input*, analisis *output* dan analisis kebutuhan data.

4.4.1 Analisis Keluaran (*Output*)

Analisis *output* pengolahan data yang dihasilkan oleh sebuah simulasi dan analisis output ini berguna untuk memprediksi informasi sebuah sistem yang dibutuhkan dalam pelaksanaan kegiatan pada Desa Maju Jaya. Adapun informasi

itu diantaranya sebagai berikut :

Data keadaan penduduk

Data keadaan penduduk merupakan data-data Laporan penduduk yang ada pada Desa Maju Jaya, yang digunakan sebagai landasan dalam perancangan sistem yang dapat dilihat pada gambar berikut ini :

No	DUSUN	Jumlah Penduduk Akhir Bulan Lalu			Tambahkan Bulan Ini						Pengurangan Bulan Ini						L	P	L+P	JLH KK Akhir	
		Jlh KK Awal	L	P	L+P	Penambahan KK	Lahir		Datang		Pengurangan KK	Meninggal		Pindah							
							L	P	L	P		L	P	L	P						
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
1	Lopak Sepang	150	230	262	492	1	2	2	-	-	-	-	-	-	-	1	1	263	243	506	150
2	Semetung	151	262	241	503	-	-	-	2	3	1	-	-	1	1	1	1	495	507	1,002	301

Gambar 4.22 Analisis Input Data keadaan penduduk

Sumber : Desa Maju Jaya

Keterangan Gambar :

Nama Masukan : Data Laporan penduduk

Fungsi : Menyimpan informasi Data Laporan penduduk

Media : Arsip

Frekuensi : Setiap pendataan Laporan penduduk

Struktur Data : Data penduduk yang pindah keluar wilayah Desa Maju Jaya

Hasil Analisis : Informasi yang dihasilkan cukup jelas, namun untuk mempermudah menghasilkan *output* sama dengan data yang *dinputkan* sebelumnya maka akan diintegrasikan data-data yang ada dengan laporan yang dihasilkan.

4.4.2 Analisis Masukan (*Input*)

Analisis *input* yang dibutuhkan pada Desa Maju Jaya adalah sebagai berikut :

1. Data Penduduk

Data Penduduk merupakan data-data Penduduk yang ada pada Desa Maju Jaya, yang digunakan sebagai landasan dalam perancangan sistem yang dapat dilihat pada gambar berikut ini :

DATA PENDUDUK TAHUN 2023

1. JUMLAH PENDUDUK: 1.002
 LAKI - LAKI 495
 PEREMPUAN 507
2. PENDUDUK MENINGGAL 5 ORANG L = 4 P = 1
 1. NAWAWI
 2. ARI SAPUTRA
 3. LANGGA
 4. M. SYAHI ABANDI
 5. SALMA
3. PENDUDUK LAHIR 19 ORANG L = 9 P = 8
 1. KARA OKTAVIOLA, JAMBI 23-10-2023
 2. ARHAM NOVIC, SARANG BURUNG 14-09-2023
 3. AISWA NAHLA, JAMBI 13-09-2023
 4. RADA KURNIA ALVARENDRA, JAMBI 01-06-2023
 5. AZWA HILYATUNNISA, JAMBI 03-08-2023
 6. GUCEN SYAH ALNATRA, NUARO JAMBI 09-01-2023
 7. RIVALDO PUTRA PANDHAN, NUARO JAMBI 01-03-2023
 8. AQILA ARUHI, TEBAL PALLY, 19-01-2023
 9. MUHAMMAD AZAH SAPUTRA, NUARO JAMBI 21-06-2023
 10. FARHAN PANDHAN, JAMBI 01-04-2023
 11. BAYI MASTURA LAKI-LAKI
 12. BAYI MUSTIKA PEREMPUAN
 13. BAYI AGUSTIN ANELIA PEREMPUAN
 14. BAYI ARINI LAKI-LAKI
 15. BAYI NAWAH LAKI-LAKI
 16. BAYI NAILIA LAKI-LAKI
 17. BAYI NURUL BADI'AH PEREMPUAN
- PENDUDUK KELUAR = 2 ORANG

Gambar 4.23 Analisis *Input* Data Penduduk

Sumber : Desa Maju Jaya

Keterangan Gambar :

Nama Masukan	: Data Penduduk
Fungsi	: Menyimpan informasi data Penduduk Desa Maju Jaya
Media	: Arsip
Frekuensi	: Setiap ada pendataan Penduduk
Struktur Data	: nomor, nama, nik, jenis kelamin
Hasil Analisis	: Informasi yang dihasilkan cukup jelas, namun dalam proses pendataan Penduduk beberapa record yang ada pada data Penduduk tidak terisi atau dibiarkan kosong. Sehingga menjadi tidak bergunanya record tersebut disertakan kedalam struktur data <i>form</i> yang akan dirancang struktur data yang baru untuk mempermudah integrasi antar data yang terkait struktur data akan disusun kembali.

4.4.3 Analisis Kebutuhan Data

Kebutuhan data dari sistem yang akan dibangun adalah sebagai berikut :

1. Data Profil

Nama Data	: Profil
Deskripsi	: Kumpulan data yang berisikan informasi-informasi mengenai data Profil.
Media	: Pengarsipan dalam <i>database</i> .

Kaitan dengan data masukan : Digunakan untuk menambah data Profil.

Kaitan dengan data keluaran : Digunakan untuk menampilkan data Profil.

2. Data Galeri

Nama Data : Galeri

Deskripsi : Kumpulan data yang berisikan informasi-informasi mengenai data Galeri.

Media : Pengarsipan dalam *database*.

Kaitan dengan data masukan : Digunakan untuk menambah data Galeri.

Kaitan dengan data keluaran : Digunakan untuk menampilkan data Galeri.

3. Data Berita

Nama Data : Berita

Deskripsi : Kumpulan data yang berisikan informasi-informasi mengenai data Berita.

Media : Pengarsipan dalam *database*.

Kaitan dengan data masukan : Digunakan untuk menambah data Berita.

Kaitan dengan data keluaran : Digunakan untuk menampilkan data Berita.

4. Data Penduduk

Nama Data : Penduduk

Deskripsi : Kumpulan data yang berisikan informasi-informasi mengenai data Penduduk.

Media : Pengarsipan dalam *database*.

Kaitan dengan data masukan : Digunakan untuk menambah data Penduduk.

Kaitan dengan data keluaran : Digunakan untuk menampilkan data Penduduk.

5. Data Pendetang baru

Nama Data : Pendetang baru

Deskripsi : Kumpulan data yang berisikan informasi-informasi mengenai data Pendetang baru.

Media : Pengarsipan dalam *database*.

Kaitan dengan data masukan : Digunakan untuk menambah data Pendetang baru.

Kaitan dengan data keluaran : Digunakan untuk menampilkan data Pendetang baru.

6. Data Perpindahan

Nama Data : Perpindahan

Deskripsi : Kumpulan yang berisikan informasi-informasi mengenai data Perpindahan.

Media : Pengarsipan dalam *database*.

Kaitan dengan data masukan : Digunakan menambah data Perpindahan.

Kaitan dengan data keluaran : Digunakan menampilkan Perpindahan.

7. Data Kelahiran

Nama Data : Kelahiran

Deskripsi : Kumpulan yang berisikan informasi-informasi mengenai data Kelahiran.

Media : Pengarsipan dalam *database*.

Kaitan dengan data masukan : Digunakan menambah data Kelahiran.

Kaitan dengan data keluaran : Digunakan menampilkan data Kelahiran.

8. Data Kematian

Nama Data : Kematian

Deskripsi : Kumpulan yang berisikan informasi-informasi mengenai data Kematian.

Media : Pengarsipan dalam *database*.

Kaitan dengan data masukan : Digunakan menambah data Kematian.

Kaitan dengan data keluaran : Digunakan menampilkan data Kematian.

9. Data Admin

Nama Data : Admin

Deskripsi : Kumpulan yang berisikan informasi-informasi mengenai data Admin.

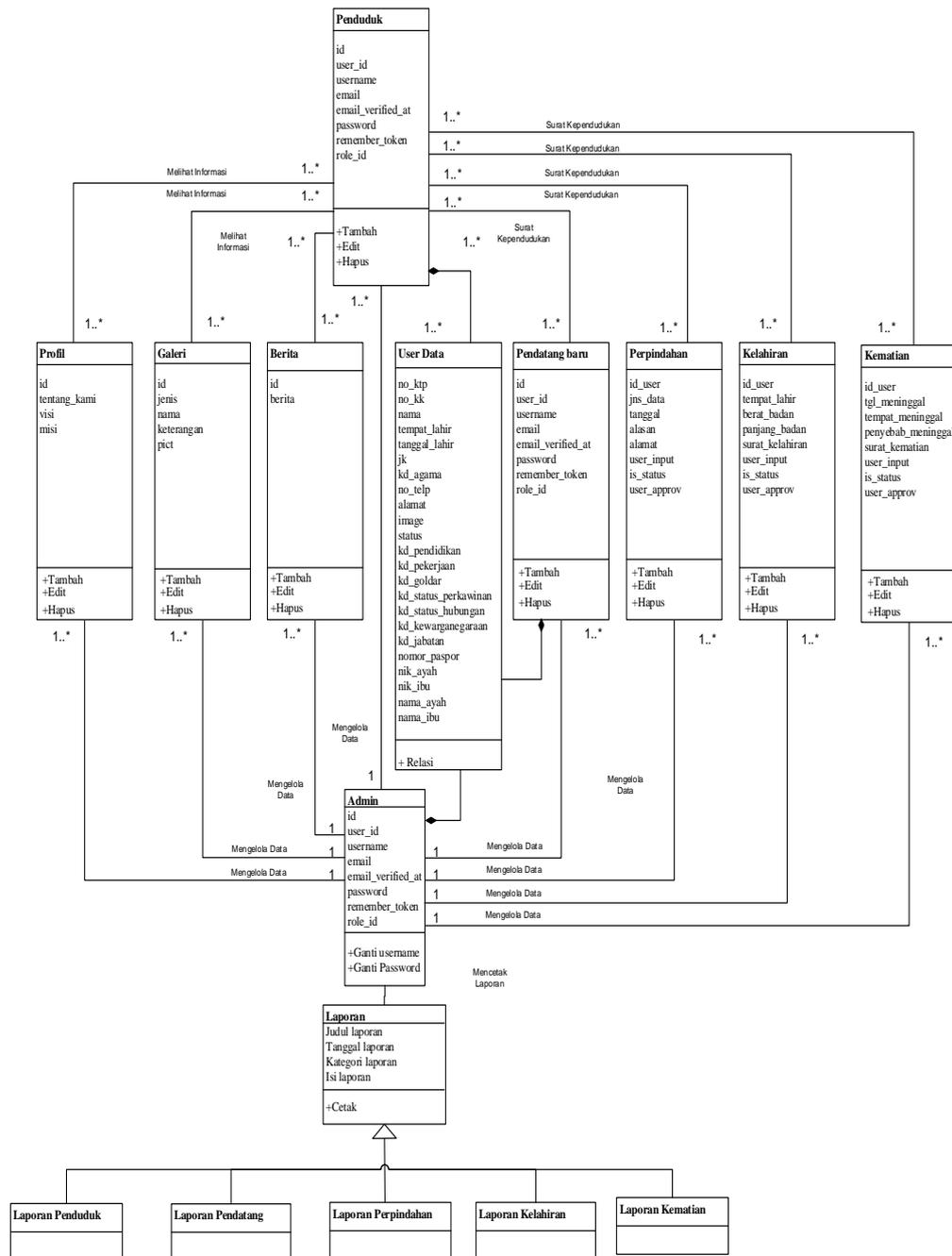
Media : Pengarsipan dalam *database*.

Kaitan dengan data masukan : Digunakan menambah data Admin.

Kaitan dengan data keluaran : Digunakan menampilkan data Admin.

Analisis kebutuhan data diatas digambarkan dengan menggunakan Class

Diagram seperti terlihat pada Gambar Berikut ini:



Gambar 4.24 Class Diagram

4.5 RANCANGAN SISTEM *OUTPUT* DAN *INPUT*

Berikut ini merupakan tahapan mendesain atau merancang sistem yang akan dibangun dimana pada tahap ini menentukan fungsi-fungsi apa saja yang nantinya akan dimiliki system dan menentukan bagaimana tampilan system yang seharusnya.

4.5.1 RANCANGAN *OUTPUT*

Rancangan-rancangan tampilan *output* dari sistem yang akan dibangun adalah sebagai berikut :

1. Rancangan Laporan Data Penduduk

Rancangan *output* laporan data Penduduk ini menampilkan informasi mengenai data-data Penduduk pada Desa Maju Jaya. Di bawah ini merupakan rancangan dari *output* data Penduduk:

No	id_penduduk	nik	nama_lengkap	tempat_lahir	tanggal_lahir	jenis_kelamin	alamat	rt	rw	desa	kecamatan	kabupaten	provinsi	agama	status	golongan_darah	pekerjaan	kewarganegaraan
999	xxx	xxx	xxx	xxx	Dd/ mm/ yyyy	xxx	xxx	99	99	xx	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx

Jambi/tgl/bln

TTD

ADMIN

Gambar 4.25 Rancangan Laporan Data Penduduk

2. Rancangan Laporan Data Pendatang baru

Rancangan *output* laporan data Pendatang baru ini menampilkan informasi mengenai data-data Pendatang baru pada Desa Maju Jaya. Di bawah ini merupakan rancangan dari *output* data Pendatang baru:

LOGO		Administrasi Kependudukan Pada Desa Maju Jaya												LOGO
LAPORAN PENDATANG BARU														
Jalan Bahtera Desa Maju Jaya, Kecamatan Pelepat Ilir, Kabupaten Muaro Bungo														
No	id_data_kedatangan	tanggal	nomor_kartu_keluarga	nik_pemohon	alasan	alamat_asal	rt_asal	rw_asal	desa_asal	kode_pos_asal	no_telepon_asal	kecamatan_asal	kabupaten_asal	provinsi_asal
999	xxx	xxx	999	999	xxx	xxx	xxx	xxx	xxx	999	999	xxx	xxx	xxx

Jambi/ig/Thu
TID

ADMIN

Gambar 4.26 Rancangan Laporan Data Pendatang baru

3. Rancangan Laporan Data Perpindahan

Rancangan *output* laporan data Perpindahan ini menampilkan informasi mengenai data-data Perpindahan pada Desa Maju Jaya. Di bawah ini merupakan rancangan dari *output* data Perpindahan:

LOGO		Administrasi Kependudukan Pada Desa Maju Jaya												LOGO
LAPORAN PERPINDAHAN														
Jalan Bahtera Desa Maju Jaya, Kecamatan Pelepat Ilir, Kabupaten Muaro Bungo														
No	id_data_perpindahan	tanggal	nomor_kartu_keluarga	nik_pemohon	alasan_pindah	alamat_tujuan_pindah	rt_tujuan_pindah	rw_tujuan_pindah	desa_tujuan_pindah	kode_pos_tujuan_pindah	no_telepon_tujuan_pindah	kecamatan_tujuan_pindah	kabupaten_tujuan_pindah	provinsi_tujuan_pindah
999	xxx	DD mm YY	999	999	xxx	xxx	999	999	xxx	999	999	xxx	xxx	xxx

Jambi/ig/Thu
TID

ADMIN

Gambar 4.27 Rancangan Laporan Data Perpindahan

4. Rancangan Laporan Data Kelahiran

Rancangan *output* laporan data Kelahiran ini menampilkan informasi mengenai data-data Kelahiran pada Desa Maju Jaya. Di bawah ini merupakan rancangan dari *output* data Kelahiran:

No.	id_data_kelahiran	tanggal	nama	jenis_kelamin	tempat_kelahiran	hari_kelahiran	tanggal_kelahiran	jam_kelahiran	jenis_kelahiran	anak_ke	berat_bayi	panjang_bayi	penolong_kelahiran	nik_bu	nik_ayah
999	xxx	Dd/ mm/ yy	xxx	xxx	xxx	xxx	Dd/ mm/ yy	999	xxx	999	999	999	xxx	999	999

Jumlah/ghIn
TID
ADMIN

Gambar 4.28 Rancangan Laporan Data Kelahiran

5. Rancangan Laporan Data Kematian

Rancangan *output* laporan data Kematian ini menampilkan informasi mengenai data-data Kematian pada Desa Maju Jaya. Di bawah ini merupakan rancangan dari *output* data Kematian:

No.	id_data_kematian	tanggal	nik	hari_meninggal_dunia	jam_meninggal_dunia	tanggal_meninggal_dunia	tempat_meninggal_dunia	sebab_meninggal_dunia
999	xxx	Dd/ mm/ yy	999	xxx	999	Dd/ mm/ yy	xxx	xxx

Jumlah/ghIn
TID
ADMIN

Gambar 4.29 Rancangan Laporan Data Kematian

4.5.2 RANCANGAN *INPUT*

Rancangan-rancangan tampilan *input* dari sistem yang akan dibangun adalah sebagai berikut :

1. Rancangan *Form Login*

Rancangan *form input login* digunakan untuk membuat *form input login*.

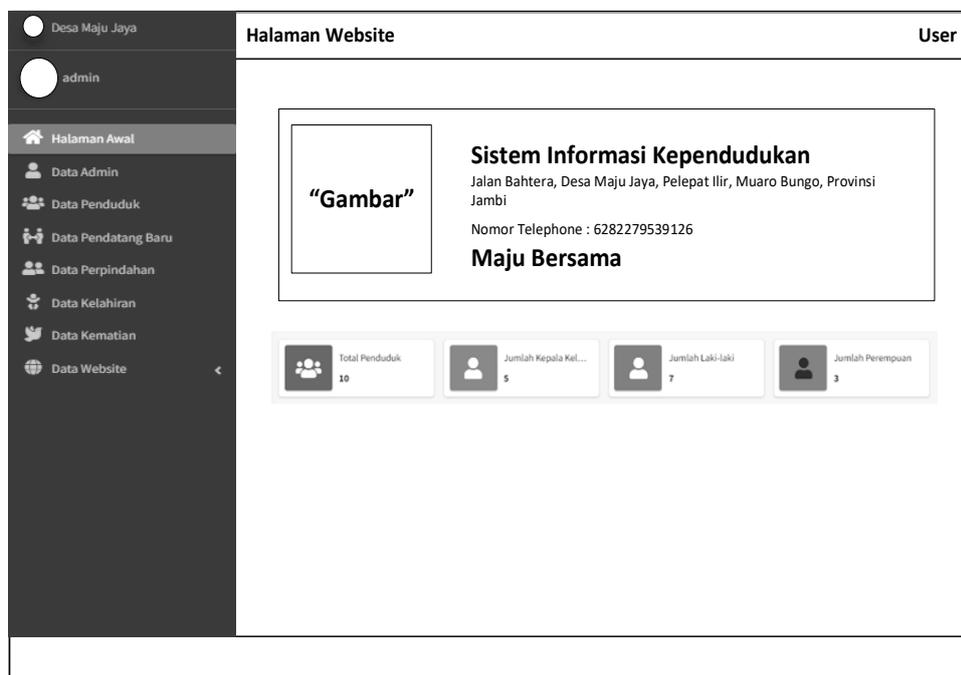
Yang dapat digunakan oleh pemakai yang memiliki hak akses yang mana pemakai memiliki hak sepenuhnya untuk mengoperasikan aplikasi yang dibuat. Adapun rancangan tampilan *form login* adalah sebagai berikut :

Sistem Informasi Kependudukan	
Masuk untuk memulai sesi	
Pilih Jenis Masuk	<input type="text"/>
Masukkan Nama	<input type="text"/>
Masukkan Kata Sandi	<input type="text"/>
<input type="checkbox"/> Ingat Saya	<input type="button" value="Masuk"/>

Gambar 4.30 Rancangan *Input Login*

2. Rancangan Menu Utama

Rancangan Menu Utama merupakan rencana atau kerangka dasar yang digunakan penulis sebelum masuk ke dalam tahap perograman, dengan tujuan agar rencana desain awal sesuai dengan program yang akan dibuat.



Gambar 4.31 Rancangan *Input Form* Menu Utama

3. Rancangan Menu Admin

Rancangan Menu Admin merupakan rencana atau kerangka dasar yang digunakan penulis sebelum masuk ke dalam tahap perograman, dengan tujuan agar rencana desain awal sesuai dengan program yang akan dibuat.

a. Rancangan *Input Form* Menu Admin

Rancangan *input form* menu Admin digunakan untuk mengintegrasikan semua fungsi-fungsi yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Admin. Berikut merupakan rancangan Admin :

Gambar 4.32 Rancangan *Input Form* Menu Admin

b. Rancangan *Input Form* Tambah Admin

Rancangan *input form* Tambah Admin digunakan untuk menambah data-data yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Admin. Berikut merupakan rancangan tampilan Admin :

Gambar 4.33 Rancangan *Input Form* Tambah Admin

4. Rancangan Menu Profil

Rancangan Menu Profil merupakan rencana atau kerangka dasar yang digunakan penulis sebelum masuk ke dalam tahap perograman, dengan tujuan agar rencana desain awal sesuai dengan program yang akan dibuat.

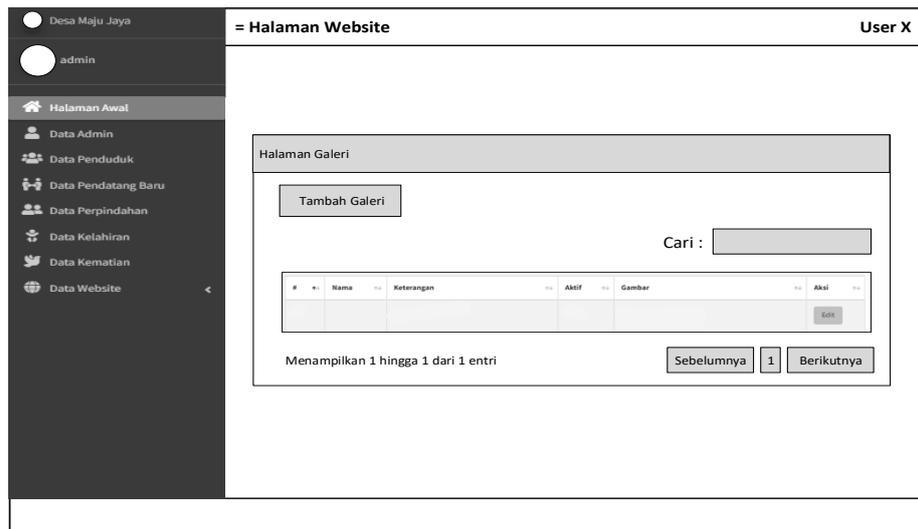
Gambar 4.34 Rancangan *Input Form* Menu Profil

5. Rancangan Menu Galeri

Rancangan Menu Galeri merupakan rencana atau kerangka dasar yang digunakan penulis sebelum masuk ke dalam tahap perograman, dengan tujuan agar rencana desain awal sesuai dengan program yang akan dibuat.

a. Rancangan *Input Form* Menu Galeri

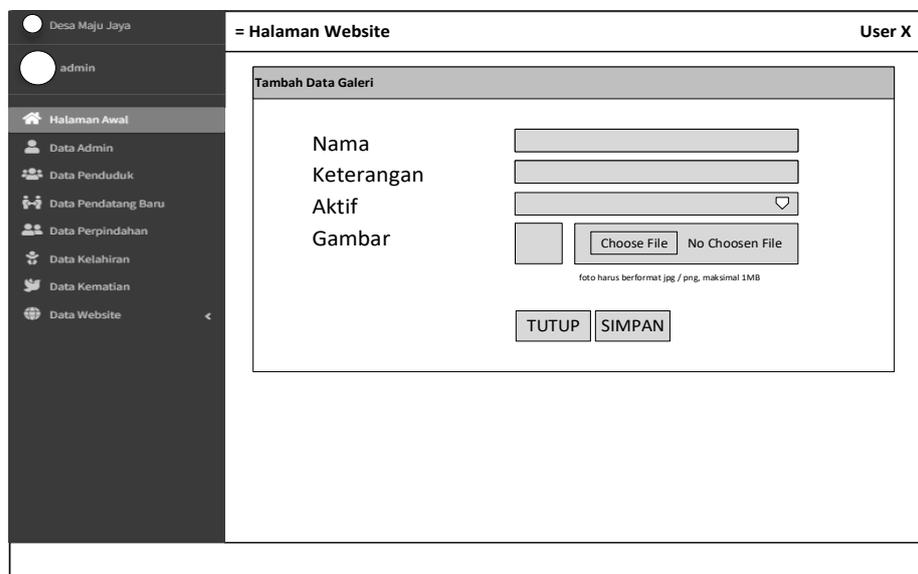
Rancangan *input form* menu Galeri digunakan untuk mengintegrasikan semua fungsi-fungsi yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Galeri. Berikut merupakan rancangan tampilan Galeri :



Gambar 4.35 Rancangan *Input Form* Menu Galeri

b. Rancangan *Input Form* Tambah Galeri

Rancangan *input form* Tambah Galeri digunakan untuk menambah data-data yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Galeri. Berikut merupakan rancangan tampilan Galeri :



Gambar 4.36 Rancangan *Input Form* Tambah Galeri

6. Rancangan Menu Berita

Rancangan Menu Berita merupakan rencana atau kerangka dasar yang digunakan penulis sebelum masuk ke dalam tahap perograman, dengan tujuan agar rencana desain awal sesuai dengan program yang akan dibuat.

a. Rancangan *Input Form* Menu Berita

Rancangan *input form* menu Berita digunakan untuk mengintegrasikan semua fungsi-fungsi yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Berita. Berikut merupakan rancangan tampilan Berita

The screenshot shows a web application interface for 'Desa Maju Jaya'. The top left corner displays the user 'admin'. The main content area is titled 'Halaman Berita' and contains a 'Tambah Berita' button, a search bar labeled 'Cari:', and a table with columns for 'Kode Berita', 'Judul', 'Tanggal', 'Berita', 'Status', and 'Aksi'. Below the table, it indicates 'Menampilkan 1 hingga 1 dari 1 entri' and includes navigation buttons 'Sebelumnya', '1', and 'Berikutnya'.

Gambar 4.37 Rancangan *Input Form* Menu Berita

b. Rancangan *Input Form* Tambah Berita

Rancangan *input form* Tambah Berita digunakan untuk menambah data-data yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Berita. Berikut merupakan rancangan tampilan Berita :

The image shows a web application interface for adding news data. The interface is divided into a sidebar menu on the left and a main content area on the right. The sidebar menu includes the following items: 'Desa Maju Jaya', 'admin', 'Halaman Awal', 'Data Admin', 'Data Penduduk', 'Data Pendetang Baru', 'Data Perpindahan', 'Data Kelahiran', 'Data Kematian', and 'Data Website'. The main content area is titled 'Tambah Data Berita' and contains the following input fields: 'Judul Berita' (text input), 'Tanggal Berita' (date input with a calendar icon), 'Isi Berita' (text area), and 'Aktif' (checkbox input). Below the input fields are two buttons: 'TUTUP' and 'SIMPAN'. The top of the interface shows the page title '= Halaman Website' and the user name 'User X'.

Gambar 4.38 Rancangan *Input Form* Tambah Berita

7. Rancangan Menu Penduduk

Rancangan Menu Penduduk merupakan rencana atau kerangka dasar yang digunakan penulis sebelum masuk ke dalam tahap perograman, dengan tujuan agar rencana desain awal sesuai dengan program yang akan dibuat.

a. Rancangan *Input Form* Menu Penduduk

Rancangan *input form* menu Penduduk digunakan untuk mengintegrasikan semua fungsi-fungsi yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Penduduk. Berikut merupakan rancangan tampilan Penduduk :

The screenshot shows a web application interface for 'Desa Maju Jaya'. The user is logged in as 'admin'. The sidebar menu includes 'Halaman Awal', 'Data Admin', 'Data Penduduk', 'Data Pendaatang Baru', 'Data Perpindahan', 'Data Kelahiran', 'Data Kematian', and 'Data Website'. The main content area is titled 'Halaman Admin' and contains a 'Tambah Admin' button, a search bar labeled 'Cari:', and a table with columns for '#', 'No KTP', 'Nama', 'Tempat Lahir', 'Tanggal Lahir', 'JK', 'Jabatan', and 'Aksi'. Below the table, it indicates 'Menampilkan 1 hingga 1 dari 1 entri' and navigation buttons for 'Sebelumnya', '1', and 'Berikutnya'.

Gambar 4.39 Rancangan *Input Form* Menu Penduduk

b. Rancangan *Input Form* Tambah Penduduk

Rancangan *input form* Tambah Penduduk digunakan untuk menambah data-data yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Penduduk. Berikut merupakan rancangan tampilan Penduduk :

The screenshot shows a web application interface for 'Desa Maju Jaya'. The user is logged in as 'admin'. The sidebar menu includes 'Halaman Awal', 'Data Admin', 'Data Penduduk', 'Data Pendaatang Baru', 'Data Perpindahan', 'Data Kelahiran', 'Data Kematian', and 'Data Website'. The main content area is titled 'Tambah Data Penduduk' and contains a form with the following fields: No Nik, No Kk, Nama, Tempat Lahir, Tanggal Lahir, Jk, Kd Agama, No Telp, Alamat, Image, Status, Kd Pendidikan, and Kd Pekerjaan.

Gambar 4.40 Rancangan *Input Form* Tambah Penduduk

8. Rancangan Menu Pdatang baru

Rancangan Menu Pdatang baru merupakan rencana atau kerangka dasar yang digunakan penulis sebelum masuk ke dalam tahap perograman, dengan tujuan agar rencana desain awal sesuai dengan program yang akan dibuat.

a. Rancangan *Input Form* Menu Pdatang baru

Rancangan *input form* menu Pdatang baru digunakan untuk mengintegrasikan semua fungsi-fungsi yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Pdatang baru. Berikut merupakan rancangan tampilan Pdatang baru :

Gambar 4.41 Rancangan *Input Form* Menu Pdatang baru

b. Rancangan *Input Form* Tambah Pdatang baru

Rancangan *input form* Tambah Pdatang baru digunakan untuk menambah data-data yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Pdatang baru. Berikut merupakan rancangan tampilan Pdatang baru :

The image shows a web application interface for adding new resident data. The interface is divided into a sidebar menu on the left and a main content area on the right. The sidebar menu includes the following items: 'Desa Maju Jaya', 'admin', 'Halaman Awal', 'Data Admin', 'Data Penduduk', 'Data Pendetang Baru', 'Data Perpindahan', 'Data Kelahiran', 'Data Kematian', and 'Data Website'. The main content area is titled 'TAMBAH DATA PENDATANG BARU' and contains the following input fields: 'Nama Penduduk', 'Tanggal Datang', 'Alasan Datang', and 'Alamat Sebelumnya'. There are two buttons at the bottom: 'TUTUP' and 'SIMPAN'.

Gambar 4.42 Rancangan *Input Form* Tambah Pendatang baru

9. Rancangan Menu Perpindahan

Rancangan Menu Perpindahan merupakan rencana atau kerangka dasar yang digunakan penulis sebelum masuk ke dalam tahap perograman, dengan tujuan agar rencana desain awal sesuai dengan program yang akan dibuat.

a. Rancangan *Input Form* Menu Perpindahan

Rancangan *input form* menu Perpindahan digunakan untuk mengintegrasikan semua fungsi-fungsi yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Perpindahan. Berikut merupakan rancangan Perpindahan :

Desa Maju Jaya = Halaman Website User X

admin

Halaman Awal

- Data Admin
- Data Penduduk
- Data Pendetang Baru
- Data Perpindahan
- Data Kelahiran
- Data Kematian
- Data Website

Halaman Perpindahan

Tambah Perpindahan

Cari :

#	No KTP	Nama	Tempat / Tanggal Lahir	JK	Tanggal Pindah	Alasan Pindah	Alamat Pindah	Aksi
								Edit

Menampilkan 1 hingga 1 dari 1 entri

Sebelumnya 1 Berikutnya

Gambar 4.43 Rancangan *Input Form* Menu Perpindahan

b. Rancangan *Input Form* Tambah Perpindahan

Rancangan *input form* Tambah Perpindahan digunakan untuk menambah data-data yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Perpindahan. Berikut merupakan rancangan tampilan Perpindahan :

Desa Maju Jaya = Halaman Website User X

admin

Halaman Awal

- Data Admin
- Data Penduduk
- Data Pendetang Baru
- Data Perpindahan
- Data Kelahiran
- Data Kematian
- Data Website

TAMBAH DATA PERPINDAHAN

Nama Penduduk

Tanggal Pindah

Alasan Pindah

Alamat Sebelumnya

TUTUP SIMPAN

Gambar 4.44 Rancangan *Input Form* Tambah Perpindahan

10. Rancangan Menu Kelahiran

Rancangan Menu Kelahiran merupakan rencana atau kerangka dasar yang digunakan penulis sebelum masuk ke dalam tahap perograman, dengan tujuan agar rencana desain awal sesuai dengan program yang akan dibuat.

a. Rancangan *Input Form* Menu Kelahiran

Rancangan *input form* menu Kelahiran digunakan untuk mengintegrasikan semua fungsi-fungsi yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Kelahiran. Berikut merupakan rancangan Kelahiran :

Gambar 4.45 Rancangan *Input Form* Menu Kelahiran

b. Rancangan *Input Form* Tambah Kelahiran

Rancangan *input form* Tambah Kelahiran digunakan untuk menambah data-data yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Kelahiran. Berikut merupakan rancangan tampilan Kelahiran :

The image shows a web application interface for adding birth data. The interface is divided into a sidebar menu on the left and a main content area on the right. The sidebar menu includes the following items: 'Desa Maju Jaya', 'admin', 'Halaman Awal', 'Data Admin', 'Data Penduduk', 'Data Pendatang Baru', 'Data Perpindahan', 'Data Kelahiran', 'Data Kematian', and 'Data Website'. The main content area is titled 'Tambah Data Kelahiran' and contains the following input fields: 'Nama Penduduk' (a text input field with a dropdown arrow), 'Tempat Lahir' (a large text area), 'Panjang Badan' (a text input field), and 'Berat Badan' (a text input field). At the bottom of the form, there are two buttons: 'TUTUP' and 'SIMPAN'. The top of the interface shows the page title '= Halaman Website' and the user name 'User X'.

Gambar 4.46 Rancangan *Input Form* Tambah Kelahiran

11. Rancangan Menu Kematian

Rancangan Menu Kematian merupakan rencana atau kerangka dasar yang digunakan penulis sebelum masuk ke dalam tahap perograman, dengan tujuan agar rencana desain awal sesuai dengan program yang akan dibuat.

a. Rancangan *Input Form* Menu Kematian

Rancangan *input form* menu Kematian digunakan untuk mengintegrasikan semua fungsi-fungsi yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Kematian. Berikut merupakan rancangan Kematian :

Desa Maju Jaya = Halaman Website User X

admin

Halaman Awal

- Data Admin
- Data Penduduk
- Data Pendetang Baru
- Data Perpindahan
- Data Kelahiran
- Data Kematian
- Data Website

Halaman Kematian

Tambah Kematian

Cari :

#	No KTP	Nama	Tempat / Tanggal Lahir	JK	Tanggal Meninggal	Tempat Meninggal	Penyebab Meninggal	Aksi
								Edit

Menampilkan 1 hingga 1 dari 1 entri

Sebelumnya 1 Berikutnya

Gambar 4.47 Rancangan *Input Form* Menu Kematian

b. Rancangan *Input Form* Tambah Kematian

Rancangan *input form* Tambah Kematian digunakan untuk menambah data-data yang dibutuhkan untuk melakukan pengolahan atau menampilkan data Kematian. Berikut merupakan rancangan tampilan Kematian :

Desa Maju Jaya = Halaman Website User X

admin

Halaman Awal

- Data Admin
- Data Penduduk
- Data Pendetang Baru
- Data Perpindahan
- Data Kelahiran
- Data Kematian
- Data Website

Tambah Data Meninggal

Nama Penduduk

Tanggal Meninggal

Tempat Meninggal

Penyebab

Meninggal

TUTUP SIMPAN

Gambar 4.48 Rancangan *Input Form* Tambah Kematian

4.5.2.1 Rancangan *Input* Khusus Penduduk

Rancangan *Input* Khusus Penduduk merupakan rancangan yang direncanakan untuk Penduduk sebagai aktor yang melakukan kunjungan pada sistem.

1. Rancangan *Input* menu utama

Rancangan *input form* menu utama digunakan untuk menampilkan data menu Utama. Berikut merupakan rancangan tampilan menu Utama:



Gambar 4.49 Rancangan *Input Form* Menu utama

2. Rancangan *Input Form* Profil

Rancangan *input form* Profil digunakan untuk menampilkan data Profil.

Berikut merupakan rancangan tampilan Profil:



Gambar 4.50 Rancangan *Input Form* Profil

3. Rancangan *Input Form* Galeri

Rancangan *input form* Galeri digunakan untuk menampilkan data Galeri.

Berikut merupakan rancangan tampilan Galeri:

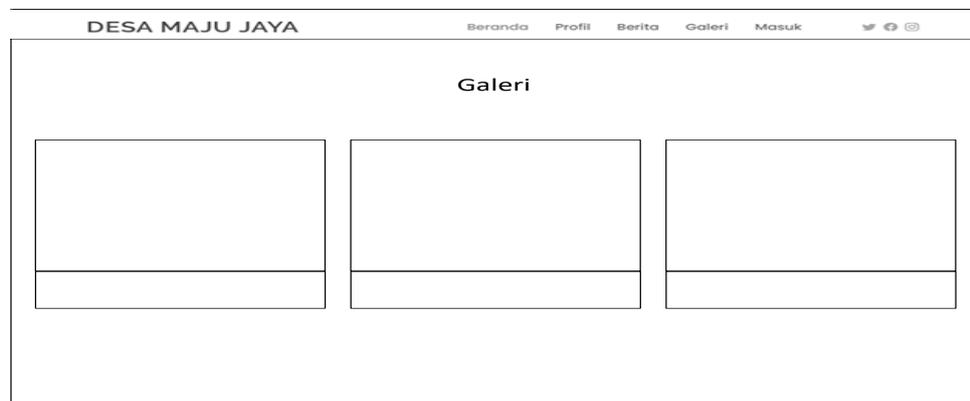


Gambar 4.51 Rancangan *Input Form* Galeri

4. Rancangan *Input Form* Berita

Rancangan *input form* Berita digunakan untuk menampilkan data Berita.

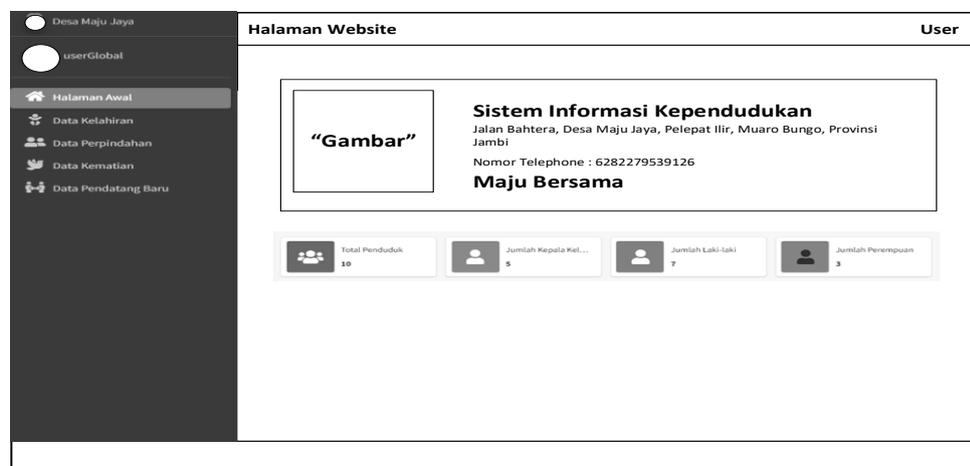
Berikut merupakan rancangan tampilan Berita:



Gambar 4.52 Rancangan *Input Form* Berita

5. Rancangan *Input Form* Surat kependudukan

Rancangan *input form* Surat kependudukan digunakan untuk menampilkan data Surat kependudukan. Berikut merupakan rancangan tampilan Surat kependudukan:



Gambar 4.53 Rancangan *Input Form* Surat kependudukan

4.6 RANCANGAN STRUKTUR DATA

Struktur data yang digunakan dalam rancangan sistem yang akan dibangun dapat dilihat dari tabel-tabel yang tersedia berikut ini.

1. Tabel Admin

Tabel Admin digunakan untuk menyimpan data-data Admin. Nama tabel *login* dalam *database* adalah Admin. Rancangan tabel *login* dapat di lihat pada tabel

Tabel 4.19 Rancangan Tabel Admin

Nama Field	Tipe	Panjang	Keterangan
<i>id</i>	bigint	20	<i>Primary Key</i>
user_id	int	11	user_id
username	varchar	15	username
email	varchar	255	email
email_verified_at	timestamp	-	email_verified_at
password	varchar	255	password
remember_token	varchar	100	remember_token
role_id	tinyint	4	role_id

2. Tabel Profil

Tabel Profil digunakan untuk menyimpan data-data Profil. Nama tabel *login* dalam *database* adalah Profil. Rancangan tabel *login* dapat di lihat pada tabel

Tabel 4.20 Rancangan Tabel Profil

Nama Field	Tipe	Panjang	Keterangan
id	bigint	20	<i>Primary Key</i>
tentang_kami	varchar	255	tentang_kami
visi	varchar	100	visi
misi	varchar	255	misi

3. Tabel Galeri

Tabel Galeri digunakan untuk menyimpan data-data Galeri. Nama tabel *login* dalam *database* adalah Galeri. Rancangan tabel *login* dapat di lihat pada tabel

Tabel 4.21 Rancangan Tabel Galeri

Nama Field	Tipe	Panjang	Keterangan
<i>id</i>	int	11	<i>Primary Key</i>
jenis	varchar	7	jenis
nama	varchar	100	nama
keterangan	varchar	100	keterangan
pict	longtext	-	pict

4. Tabel Berita

Tabel Berita digunakan untuk menyimpan data-data Berita. Nama tabel *login* dalam *database* adalah Berita. Rancangan tabel *login* dapat di lihat pada tabel

Tabel 4.22 Rancangan Tabel Berita

Nama Field	Tipe	Panjang	Keterangan
id	bigint	20	<i>Primary Key</i>
Judul_berita	varchar	255	Judul_berita
Isi_berita	varchar	255	Isi_berita

5. Tabel Penduduk

Tabel Penduduk digunakan untuk menyimpan data-data Penduduk. Nama tabel *login* dalam *database* adalah Penduduk. Rancangan tabel *login* dapat di lihat pada tabel

Tabel 4.23 Rancangan Tabel Penduduk

Nama Field	Tipe	Panjang	Keterangan
<i>id</i>	bigint	20	<i>Primary Key</i>
user_id	int	11	user_id
username	varchar	15	username
email	varchar	255	email
email_verified_at	Timestamp	-	email_verified_at
password	varchar	255	password
remember_token	varchar	100	remember_token
role_id	tinyint	4	role_id

6. Tabel Pendatang baru

Tabel Pendatang baru digunakan untuk menyimpan data-data Pendatang baru. Nama tabel *login* dalam *database* adalah Pendatang baru. Rancangan tabel *login* dapat di lihat pada tabel

Tabel 4.24 Rancangan Tabel Pendatang baru

Nama Field	Tipe	Panjang	Keterangan
<i>id</i>	bigint	20	<i>Primary Key</i>
user_id	int	11	user_id
username	varchar	15	username
email	varchar	255	email
email_verified_at	timestamp	-	email_verified_at
password	varchar	255	password
remember_token	varchar	100	remember_token
role_id	tinyint	4	role_id

7. Tabel Perpindahan

Tabel Perpindahan digunakan untuk menyimpan data-data Perpindahan. Nama tabel *login* dalam *database* adalah Perpindahan. Rancangan tabel *login* dapat di lihat pada tabel

Tabel 4.25 Rancangan Tabel Perpindahan

Nama Field	Tipe	Panjang	Keterangan
id_user	bigint	20	<i>Primary Key</i>

jns_data	varchar	7	jns_data
tanggal	date	-	tanggal
alasan	text	-	alasan
alamat	text	-	alamat
user_input	varchar	25	user_input
is_status	smallint	6	is_status

8. Tabel Kelahiran

Tabel Kelahiran digunakan untuk menyimpan data-data Kelahiran. Nama tabel *login* dalam *database* adalah Kelahiran. Rancangan tabel *login* dapat di lihat pada tabel

Tabel 4.26 Rancangan Tabel Kelahiran

Nama Field	Tipe	Panjang	Keterangan
id_user	bigint	20	<i>Primary Key</i>
tempat_lahir	varchar	100	tempat_lahir
berat_badan	Float	-	berat_badan
panjang_badan	Float	-	panjang_badan
surat_kelahiran	Text	-	surat_kelahiran
user_input	varchar	25	user_input
is_status	smallint	6	is_status

9. Tabel Kematian

Tabel Kematian digunakan untuk menyimpan data-data Kematian. Nama tabel *login* dalam *database* adalah Kematian. Rancangan tabel *login* dapat di lihat pada tabel

Tabel 4.27 Rancangan Tabel Kematian

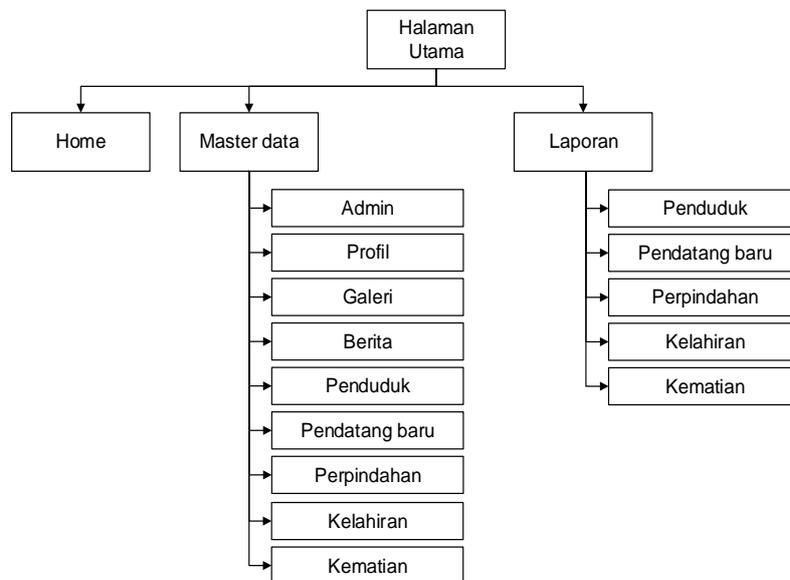
Nama Field	Tipe	Panjang	Keterangan
id_user	bigint	20	<i>Primary Key</i>
tgl_meninggal	date	-	tgl_meninggal
tempat_meninggal	text	-	tempat_meninggal
penyebab_meninggal	text	-	penyebab_meninggal
surat_kematian	text	-	surat_kematian
user_input	varchar	25	user_input

is_status	smallint	6	is_status
user_approv	varchar	25	user_approv

4.7 RANCANGAN STRUKTUR PROGRAM

4.7.1. Rancangan Struktur Program Khusus Admin

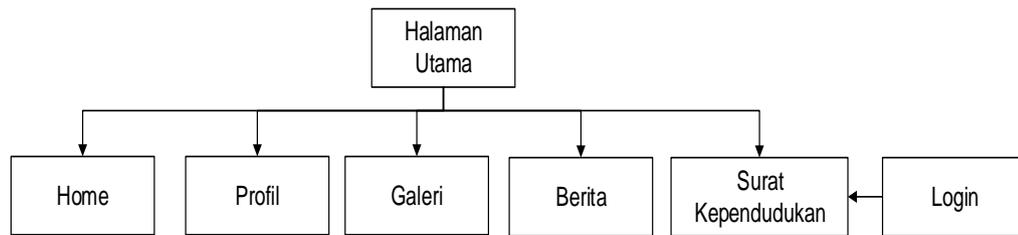
Rancangan struktur program merupakan gambaran mengenai hubungan antara menu utama dengan modul/sub program yang ada. Adapun struktur program dari Halaman utama sistem yang akan di bangun adalah sebagai berikut:



Gambar 4.54 Struktur Program Halaman Utama

4.7.2. Rancangan Struktur Program Khusus Penduduk

Rancangan struktur program merupakan gambaran mengenai hubungan antara menu utama dengan modul/sub program yang ada. Adapun struktur program dari Halaman utama sistem yang akan di bangun adalah sebagai berikut:



Gambar 4.55 Struktur Program Halaman Utama