BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem

Pada tahap perancangan ini akan dijelaskan mengenai perancangan aplikasi pada objek yang digunakan, perancangan arsitektur program yang akan dibuat, perancangan tampilan dan perancangan menu. Analisis dan perancangan adalah serangkaian kegiatan yang selalu beriringan dalam setiap pengembangan *software*, sebagai sebuah hubungan sebab dan akibat yang memunculkan sebuah siklus hidup sistem. Walaupun dalam kenyataannya pengembangan sistem yang sederhana, aktivitas ini tidak tampak.

4.1.1 Tujuan Perancangan Sistem

Tujuan dari perancangan ini merupakan suatu kegiatan pengembangan prosedur dan proses yang sedang berjalan untuk menghasilkan sesuatu yang baru atau memperbaharui yang ada untuk meningkatkan kinerja sistem itu sendiri sehingga dapat memenuhi hasil yang diinginkan. Rancangan aplikasi yang baru, akan diterapkan suatu kegiatan untuk menemukan dan mengembangkan metode, prosedur dan proses suatu data agar tujuan dari suatu organisasi dapat tercapai.

Adapun tujuan dari tahap perancangan ini adalah untuk menghasilkan perancangan aplikasi berupa pemodelan menggunakan pendekatan berorientasi objek sehingga dapat memperbaiki atau meningkatkan efisiensi kerja sistem yang sedang berjalan.

4.1.2 Gambaran Umum Sistem Yang Diusulkan

Gambaran umum tentang aplikasi yang diusulkan pada proses perancangan ini adalah suatu aplikasi yang digunakan untuk membuat data kependudukan dengan harapan mampu menangani permasalahan yang ada pada pendataan Penduduk yang sedang berjalan di kecamatan majalaya.

Aplikasi ini sendiri dibangun atas dasar kebutuhan akan data kependudukan yang lebih signifikan. Pengguna dari aplikasi ini dikhususkan pada petugas pendataan sesuai dengan tugas yang telah di berikan langsung oleh camat

4.1.3 Perancangan Prosedur Yang Diusulkan

Pada tahap ini, perancangan prosedur yang diusulkan dilakukan untuk memperbaiki atau meningkatkan efisiensi kerja. Tahap perancangan yang digambarkan merupakan tahap perancangan untuk membangun suatu sistem dan mengkonfigurasikan komponen-komponen perangkat lunak dan perangkat keras sehingga menghasilkan sistem yang baik. Tahapan perancangan prosedur ini akan dijelaskan dengan menggunakan pemodelan sistem informasi berorientasi objek dengan *UML*.Untuk menunjang hal tersebut setiap penduduk yang akan melakukan proses pendataan harus mengikuti beberapa prosedur di antara lain :

Penduduk yang akan mengurus data Kelahiran

1. Membawa surat keterangan melahirkan dari rumah sakit atau bidan
2. Membuat surat Kelahiran dari desa

Penduduk yang akan mengurus data perpindahan (pidah dari Kecamatan atau Masuk ke kecamatan)

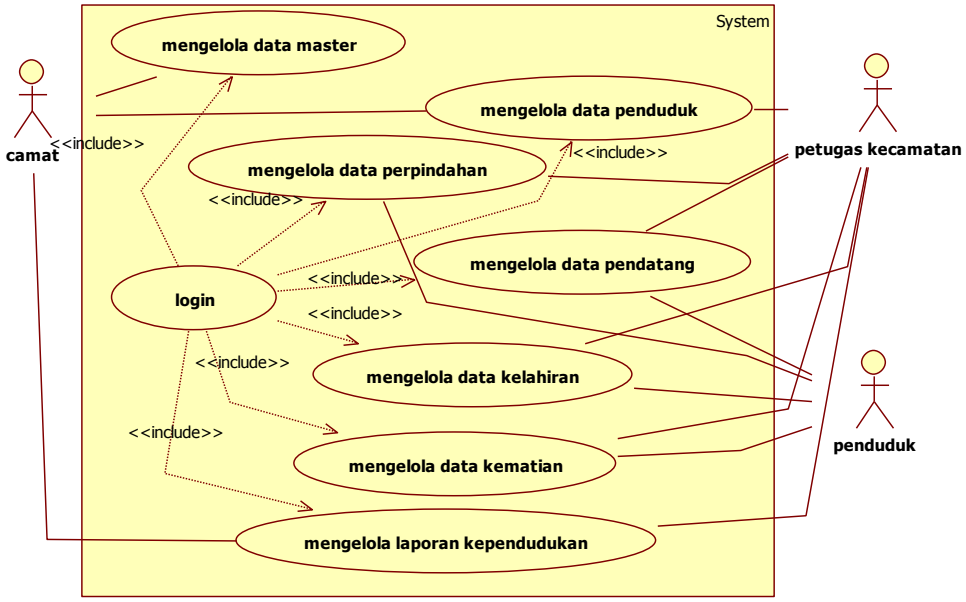
1. Membawa surat keterangan pindah dari tempat tinggal asal jika si penduduk ingin menetap di Kecamatan Majalaya
2. Membuat surat keterangan pindah dari kecamatan jika penduduk tersebut ingin pindah dari tempat tinggal yang dia tempati saat ini.

Penduduk yang akan mengurus surat Keterangan Kematian

1. Membawa surat keterangan meninggal dari rumah sakit (bila meninggal di rumah sakit)
2. Membawa surat keterangan meninggal dari RW dimana dia bertempat.
3. Membawa Surat keterangan meninggal dari Desa dimana dia bertempat.

4.1.3.1 *Use Case Diagram* Yang Diusulkan

Usecase diagram yang diusulkan merupakan gambaran diagram yang menggambarkan semua kasus yang akan ditangani oleh perangkat lunak beserta actor atau pelakunya. Berikut *use case* diagram yang diusulkan pada perancangan Sistem Informasi Kependudukan :



Gambar 4.1 *Use Case* Yang Diusulkan

4.1.3.2 Identifikasi Use Case dan Deskripsi Yang Diusulkan

Identifikasi *use case* yang bekerja pada *system* adalah sebagai berikut:

Tabel 4.1 Daftar Deskripsi *Use Case* Usulan

No	*Use Case*	Deskripsi
UC – 01	*Login*	Menentukan hak akses *user* pada saa tmasuk kesistem
UC – 02	Mengelola Data Master	Fungsional untuk menambah, menghapus atau mengubah data master seperti data *user*, data penduduk yang melahirkan, meninggal, pindah
UC – 03	Mengelola Data Penduduk	Fungsional untuk melakukan pencatatan penduduk yang ada di kecamatan

UC – 04	Mengelola perpindahan	Fungsional untuk mencatat perpindahan warga, yang akan pindah keluar kecamatan atau berpindah rumah
UC – 05	Mengelola pendatang	Fungsional untuk mencatat penduduk pendatang baru. Agar tercatat kedalam data kependudukan
UC – 06	Mengelola kelahiran	Fungsional untuk melakukan pencatatan kelahiran yang terjadi di lingkungan.
UC – 07	Mengelola kematian	Fungsional untuk melakukan pencatatan warga yang meninggal dunia.
UC – 08	Mengelola Laporan	Fungsional untuk mengakses laporan penduduk

4.1.3.3 Identifikasi Aktor dan Deskripsi Yang Diusulkan

Tabel 4.2 Identifikasi Aktor dan Deskripsi Yang Diusulkan

No	**Aktor**	**Deskripsi**
P-01	*Admin* (camat)	Pengguna sistem yang mendapat hak akses untuk mengelola data master. Dan menerima laporan kependudukan
P-02	Petugas	Pengguna sistem yang mendapatkan sesuatu untuk melakukan pendataan penduduk.

4.1.3.4 Skenario Use Case yang diusulkan

Adapun skenario use case yang diusulkan yang diusulkan sebagai langkah dari awal hingga proses *output* sebagai berikut :

1. Skenario *Use Case Login*

Tabel 4.3 Skenario *Use Case* Login

Identifikasi	
Judul	*Login*
No. *Use case*	UC-01
Deskripsi *use case*	Menentukan hak akses user pada saat masuk ke system
Aktor	Admin (camat), petugas
Trigger	Aktor menekan tombol *Login*
Kondisi awal	Aplikasi menampilkan sebuah *form* untuk *login*
***Scenario* utama**	
Aksi-aktor	**Reaksi system**
1.user mengisi *username & password* dan memilih hak akses.	
2.user menekan tombol *login.*	
	3.Sistem memeriksa inputan berdasarkan

	username, password dan hak akses pada *database.*
Kondisi akhir *scenario* utama	4.Sistem menampilkan halaman utama berdasarkan hak akses dari *user.*
Skenario Alternatif – 1	
Aksi-aktor	**Reaksi system**
1. *User* menekan tombol *login* tanpa mengisi *field username* dan *field password* dan hanya memilih hak akses	
	2.Form login menampilkan pesan kesalahan yang menyatakan ada kesalah pada *username* atau *password.*
	3.Form utama tidak ditampilkan
Kondisi akhir *scenario* alternative -1	Aplikasi menampilkan *form login.*
Skenario Alternatif – 2	
Aksi-aktor	**Reaksi system**
1. *User* memilih hak akses yang tidak sesuai	
	3.Form login menampilkan pesan kesalahan yang menyatakan ada

	kesalahan pada hak akses
	4.Form utama tidak ditampilkan
Kondisi akhir *scenario* alternative -2	Aplikasi menampilkan *form login.*

2. Skenario *Use Case* Mengelola Data Master

Tabel 4.4 Skenario *Use Case* Mengelola Data Master

Identifikasi	
Judul	Mengelola Data Master
No. *Use case*	UC-02
Deskripsi *use case*	Melakukan pencatatan data master. Menambah, menghapus, atau mengubah data yang ada pada system
Aktor	Admin(camat)
Trigger	Aktor menekan tombol Tambah/Simpan/Edit/Hapus/Batal
Kondisi awal	Aplikasi menampilkan sebuah *form*
***Scenario* utama**	
Aksi-aktor	**Reaksi system**
1. Admin memilih menu data master yang akan dikelola	

	2.Menampilkan form data master yang dipilih
3.Mengklik tombol tambah pada form	
4. Mengisi *field* pada *form* data master yang dipilih dan menekan tombol Tambah/Simpan/Hapus/Edit/Batal	
Kondisi akhir *scenario* utama	5.Menyimpan data master *database*
Skenario Alternatif – 1	
Aksi-aktor	**Reaksi system**
1.Untuk melakukan perubahan. Menghapus data master. Klik terlebih dahulu data yang akan dikelola	
	2.Menampilkan data yang dipilih pada *form*
3.Klik tombol hapus / edit	
Kondisi akhir *scenario* alternative -1	4.Menyimpan perubahan data master

3. Skenario *Use Case* Mengelola Data Penduduk

Tabel 4.5 Skenario *Use Case* Mengelola Data Penduduk

Identifikasi	
Judul	Mengelola Penduduk
No. *Use case*	UC-03
Deskripsi *use case*	Melakukan pencatatan penduduk
Aktor	Admin (camat) , petugas
Trigger	Aktor menekan tombol tambah / simpan / rubah / hapus / reset
Kondisi awal	Aplikasi menampilkan sebuah *form*
***Scenario* utama**	
Aksi-aktor	**Reaksi system**
1.Menekan menu data penduduk	
	2.Menampilkan *form* dengan data penduduk pada table
3.Menekan tombol tambah	
	4.Menampilkan form tambah data penduduk
5.Mengisi seluruh field penting dan menekan tombol simpan	
6.Menutup form tambah data	
7.Menekan tombol refresh pda	

form data penduduk	
Kondisi akhir *scenario* utama	8.Menyimpan data penduduk ke *database* menampilkan data penduduk pada table
Skenario Alternatif – 1	
Aksi-aktor	**Reaksi system**
1.Klik tombol Reset	
	2.Data penduduk yang telah di input akan hilang.
Kondisi akhir *scenario* alternative -1	3.Menampilkan *form* kosong

4. Skenario *Use Case* Mengelola Data Perpindahan

Tabel 4.6 Skenario *Use Case* Mengelola Data Perpindahan

Identifikasi	
Judul	Mengelola Perpindahan
No. *Use case*	UC-04
Deskripsi *use case*	Melakukan pencatatan penduduk yang melakukan perpindahan kependudukan
Aktor	Petugas kecamatan
Trigger	Aktor menekan tombol Simpan/ Edit/ Rubah/ Hapus/ Batal

Kondisi awal	Aplikasi menampilkan sebuah *form*
***Scenario* utama**	
Aksi-aktor	**Reaksi system**
1.Menekan menu Perpindahan penduduk	
	2.Menampilkan *form* perpindahan penduduk
3.Menekan tombol cari pada *field* kartu tanda penduduk	
	4.Menampilkan data penduduk
5.Memilih no kartu tanda penduduk dan mengisi *field* yang ada	
6.Menekan tombol simpan	
Kondisi akhir *scenario* utama	7.Menyimpan perubahan data penduduk ke *database*
Skenario Alternatif – 1	

5. Skenario *Use Case* Mengelola Data Pendatang

Tabel 4.7 Skenario *Use Case* Mengelola Data Pendatang

Identifikasi	
Judul	Mengelola Pendatang
No. *Use case*	UC-05

Deskripsi *use case*	Melakukan pencatatn penduduk pendatang baru
Aktor	Petugas kecamatan
Trigger	Aktor menekan tombol Simpan / Hapus/ Batal/ Edit
Kondisi awal	Aplikasi menampilkan sebuah *form*
***Scenario* utama**	
Aksi-aktor	**Reaksi system**
1.Mengklik menu Penduduk Pendatang	
	2.Menampilkan *Form* Penduduk pendatang
3.Menginput data penduduk yang melakukan perpindahan ke kecamatan majalaya menekan tombol simpan.	
Kondisi akhir *scenario* utama	4.Menyimpan data penduduk ke *database.*
Skenario Alternatif – 1	

6. Skenario *Use Case* Mengelola Data Kelahiran

Tabel 4.8 Skenario *Use Case* Mengelola Data Kelahiran

Identifikasi	
Judul	Mengelola Kelahiran
No. *Use case*	UC-06
Deskripsi *use case*	Melakukan pencatatan kelahiran penduduk
Aktor	Petugas kecamatan
Trigger	Aktor menekan tombol Simpan/ Edit/ Hapus/ Batal
Kondisi awal	Aplikasi menampilkan sebuah *form*
***Scenario* utama**	
Aksi-aktor	**Reaksi system**
1.User memilih menu kelahiran	
	2.Menampilkan *form* kelahiran
3.Mengisi *field* yang ada di *form* kelahiran dan menekan tombol simpan.	
Kondisi akhir *scenario* utama	4.Menyimpan data kelahiran ke *database*

7. Skenario *Use Case* Mengelola Data Kematian

Tabel 4.9 Skenario *Use Case* Mengelola Data Kematian

Identifikasi	
Judul	Mengelola kematian
No. *Use case*	UC-07
Deskripsi *use case*	Melakukan pencatatan kematian penduduk
Aktor	Petugas kecamatan
Trigger	Aktor menekan tombol Simpan/ Edit/ Hapus/ Batal
Kondisi awal	Aplikasi menampilkan sebuah *form*
***Scenario* utama**	
Aksi-aktor	**Reaksi system**
1.Mengklik menu Kematian penduduk	
	2.Menampilkan *form* kematian penduduk
3.Mengisi *field* yang ada di *form* dan menekan tombol simpan	
Kondisi akhir *scenario* utama	4.Menyimpan data kematian penduduk

8. Skenario *Use Case* Mengelola Data Laporan

Tabel 4.10 Skenario *Use Case* Mengelola Data Laporan

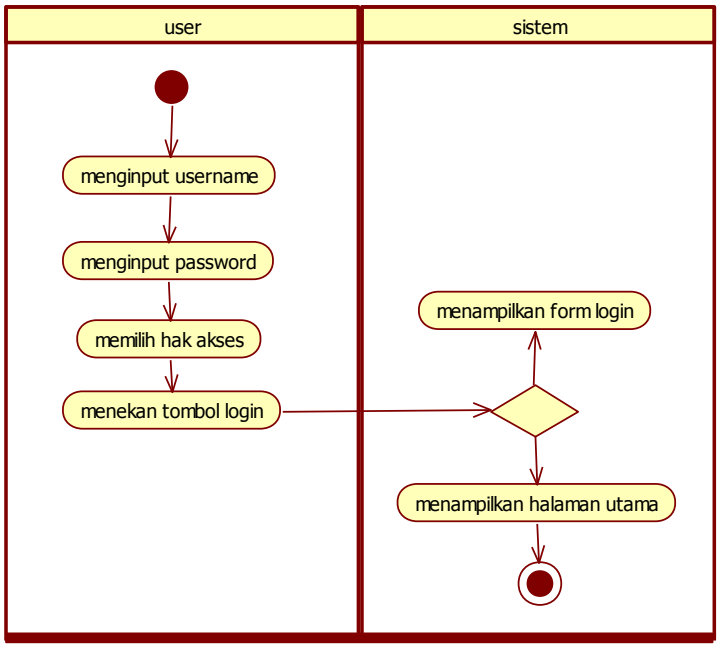
Identifikasi	
Judul	Mengelola Laporan
No. *Use case*	UC-08
Deskripsi *use case*	Melakukann pencetakan laporan
Aktor	Petugas kecamatan, admin (camat)
Trigger	Aktor menekan tombol Tampilkan/cetak
Kondisi awal	Aplikasi menampilkan sebuah *form*
***Scenario* utama**	
Aksi-aktor	**Reaksi system**
1.Mengklik menu Laporan data penduduk	
	2.Menampilkan *form* Laporan
3.Mengisi periode laporan yang akan ditampilkan lalu klik tombol tampilkan	
	4.Menampilkan data sesuai periode
5.Memilih data yang akan dicetak lalu tekan tombol cetak	
Kondisi akhir *scenario* utama	6.Menampilkan laporan penduduk yang

	dipilih

4.1.3.5 Activity Diagram Yang Diusulkan

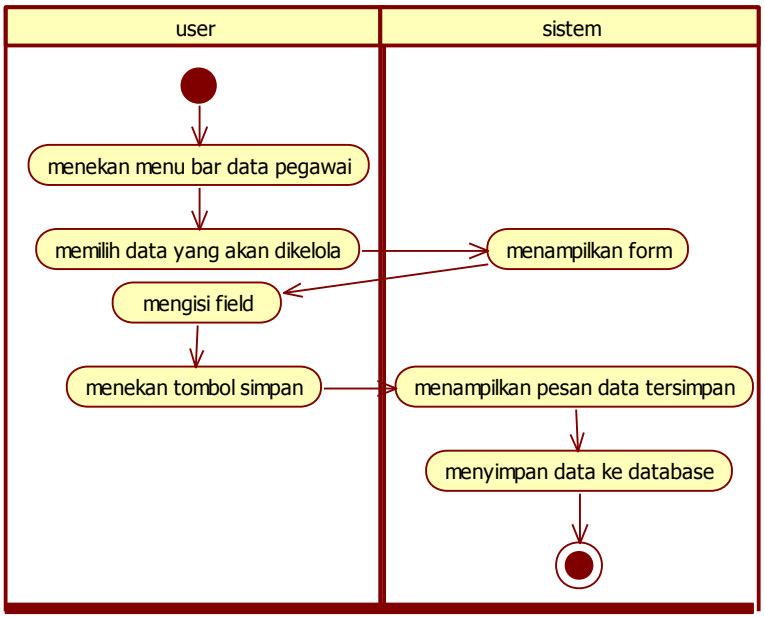
Activity diagram yang diusulkan ini menggambarkan ilustrasi aliran sebuah sistem yang akan berlangsung pada Kantor kecamatan:

1. *Activity Diagram Login*



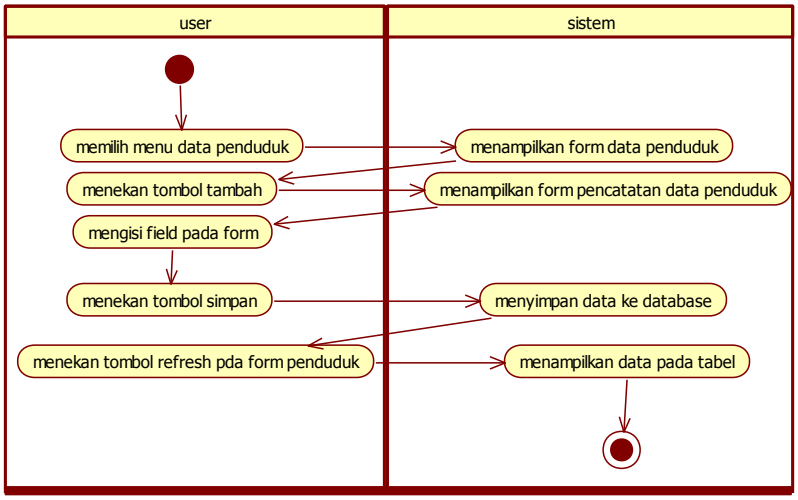
Gambar 4.2 Activity Diargam Login

2. *Activity Diagram* Mengelola Data Maste



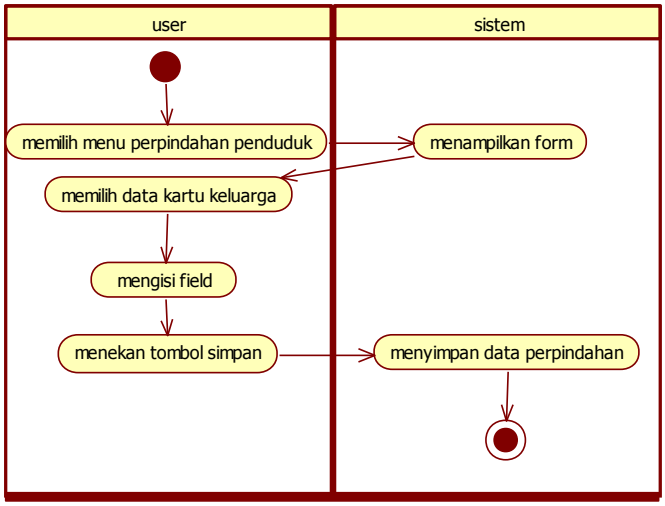
Gambar 4.3 *Activity Diagram* Mengelola Data Master

3. *Activity Diagram* Mengelola Data Penduduk



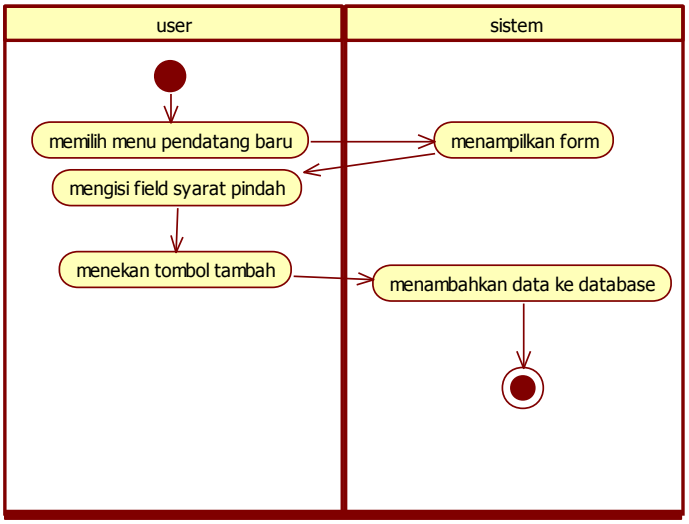
Gambar 4.4 *Activity Diagram* Mengelola Data Penduduk

4. *Activity Diagram* Mengelola Perpindahan



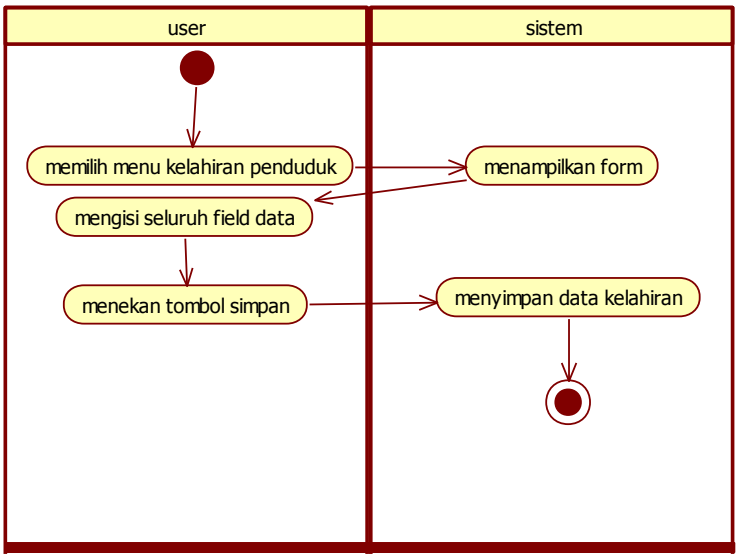
Gambar 4.5 *Activity Diagram* Mengelola Perpindahan

5. *Activity Diagram* Mengelola Pendatang



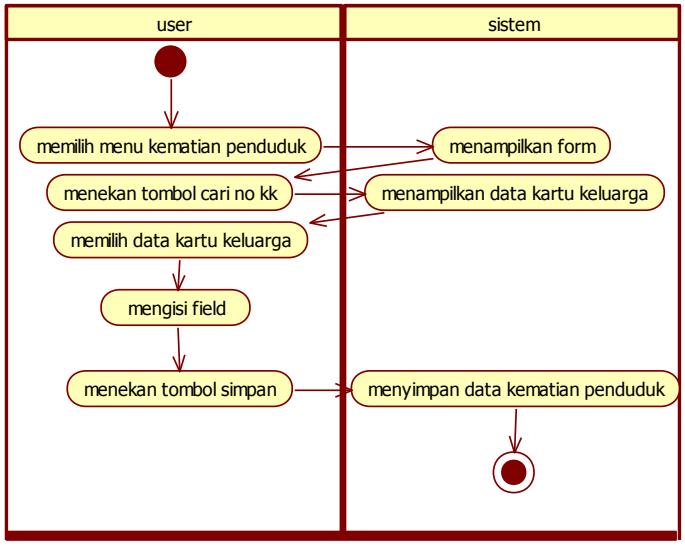
Gambar 4.6 *Activity Diagram* Mengelola Pendatang

6. *Activity Diagram* Mengelola Kelahiran



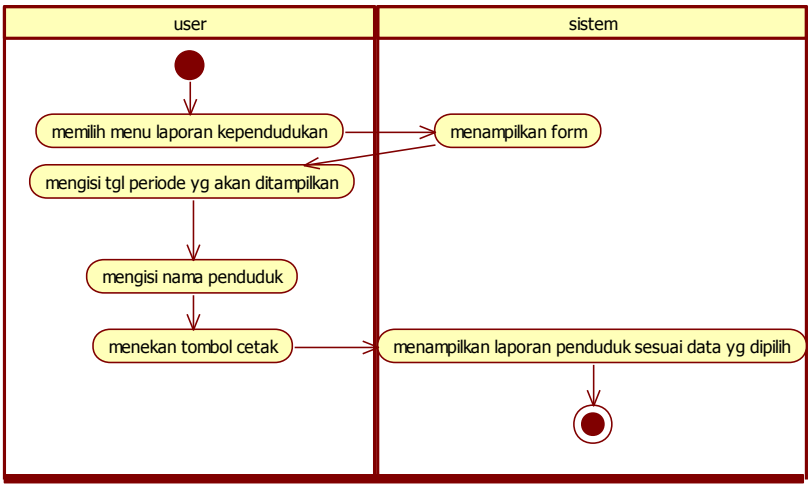
Gambar 4.7 *Activity Diagram* Mengelola Kelahiran

7. *Activity Diagram* Mengelola Kematian



Gambar 4.8 *Activity Diagram* Mengelola Kematian

8. *Activity Diagram* Mengelola Laporan

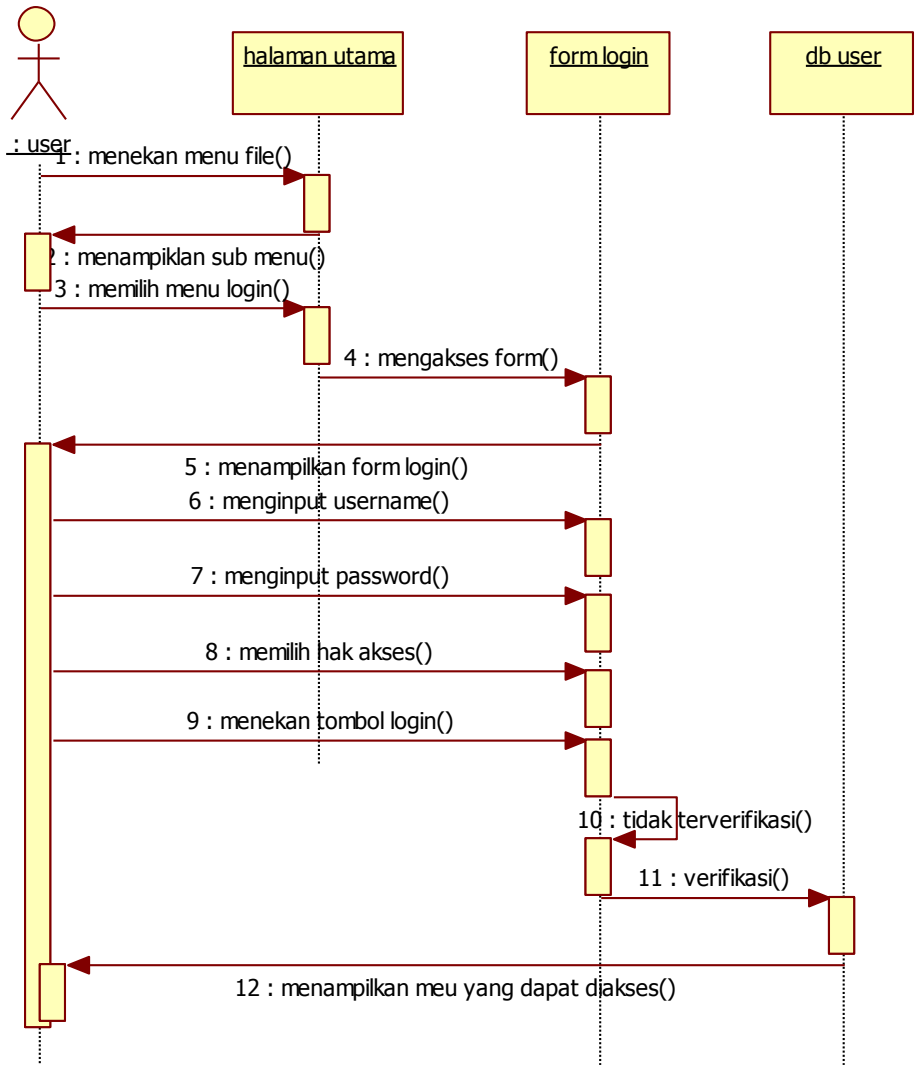


Gambar 4.9 *Activity Diagram* Mengelola Laporan

4.1.3.6 Diagram Sequence

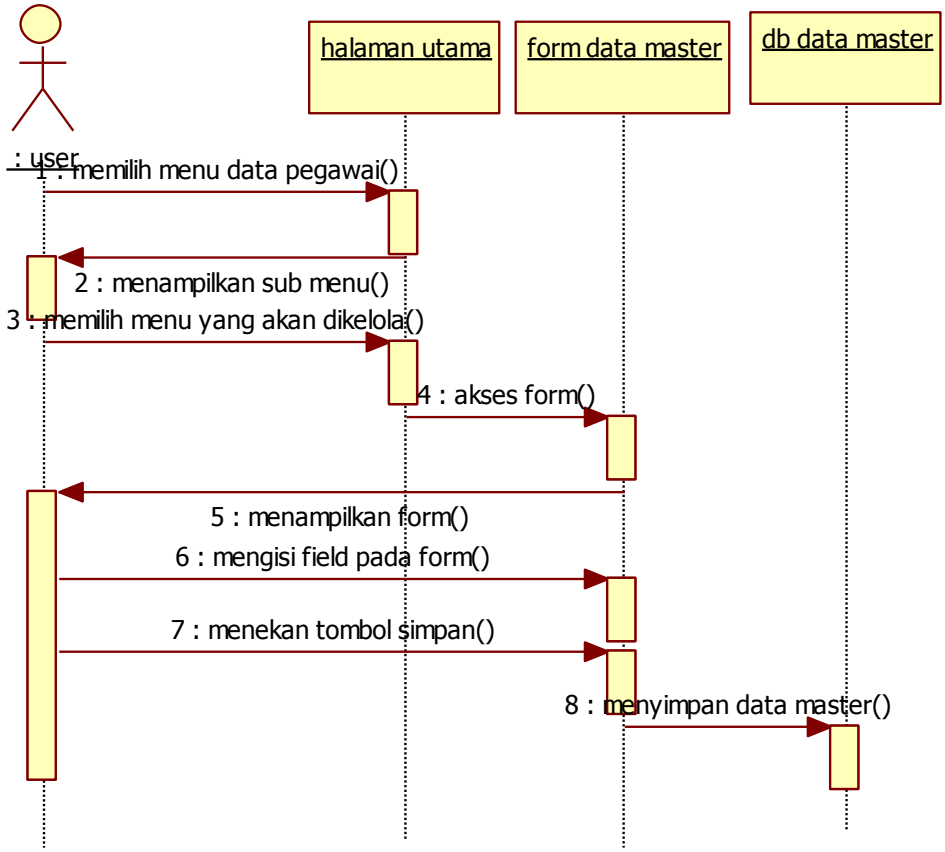
Diagram sequence merupakan diagram untuk menggambarkan perilaku system terhadap suatu interaksi yang dilakukan pada system tersebut berupa pesan yang digambarkan terhadap waktu. Diagram *sequence* yang diusulkan untuk masing-masing bagian dari sistem yang diusulkan akan digambarkan seperti berikut ini :

1. *Diagram Sequence Login*



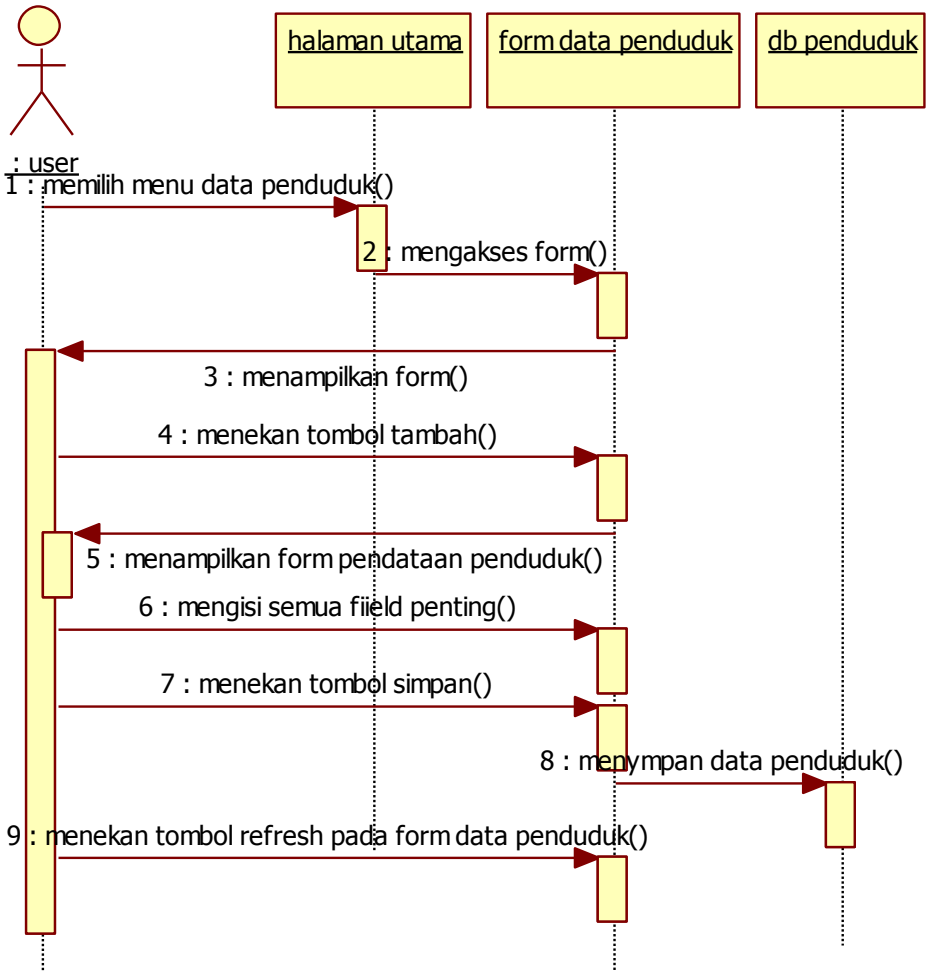
Gambar 4.10 Diagram Sequence Login

2. *Diagram Sequence* Mengelola Data Master



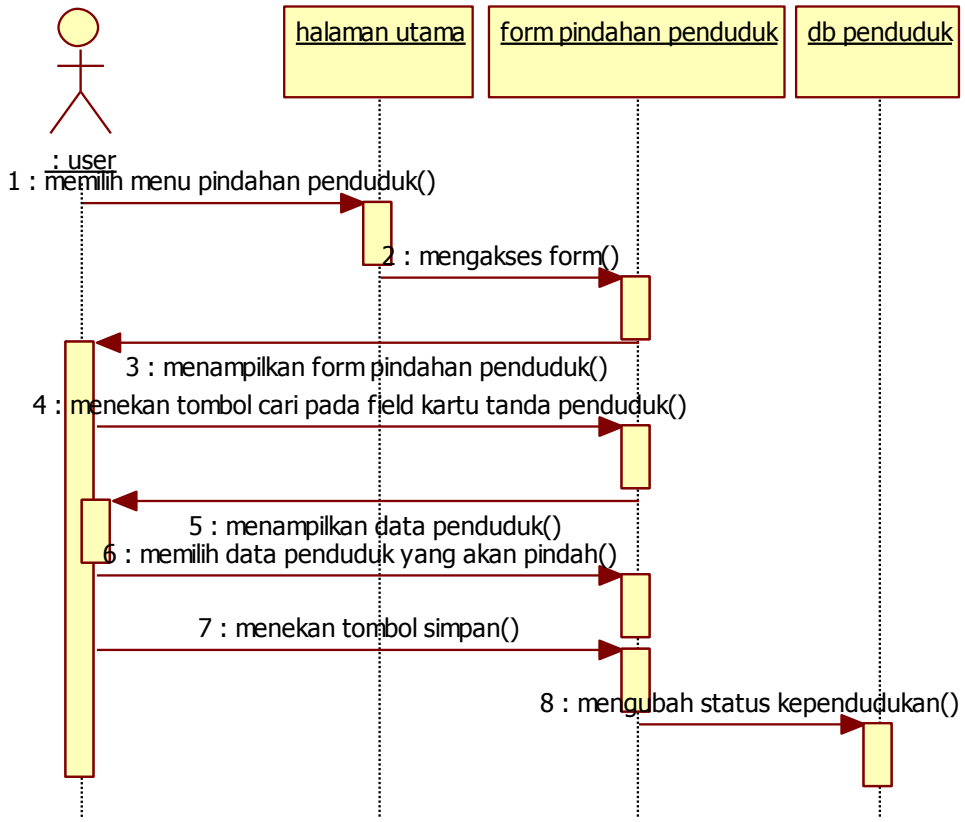
Gambar 4.11 *Diagram Sequence* Data Master

3. *Diagram Sequence* Mengelola Data Penduduk



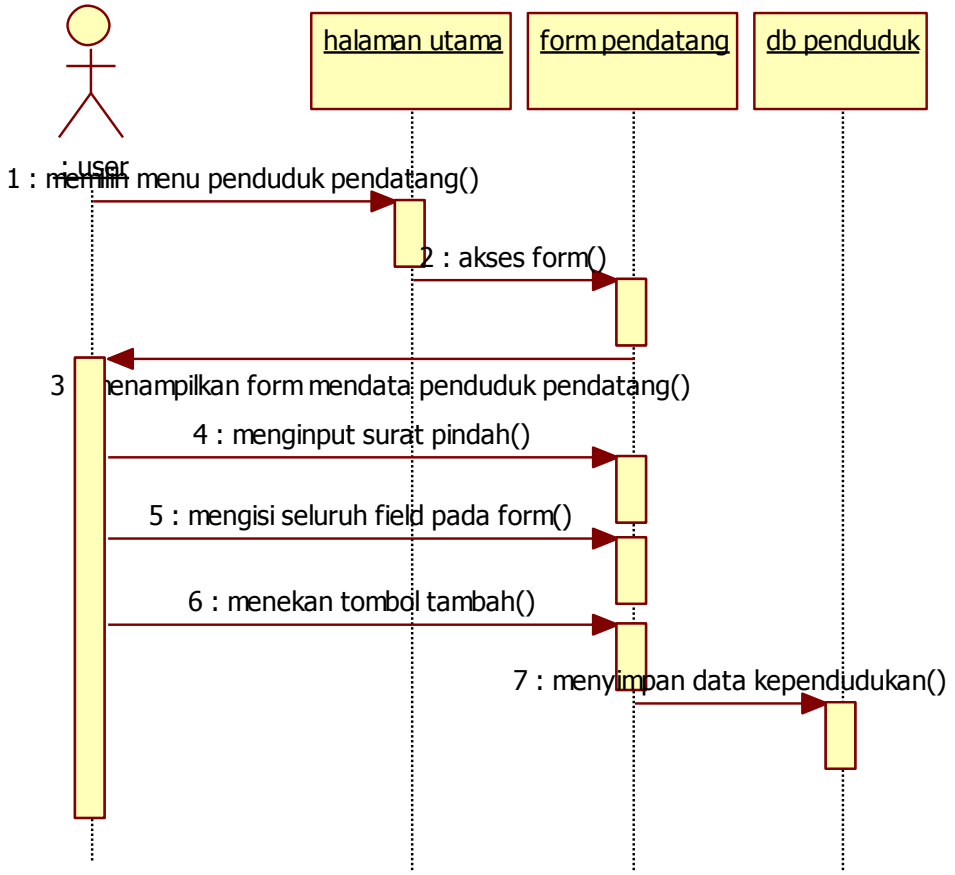
Gambar 4.12 *Diagram Sequence* MengelolaData Penduduk

4. *Diagram Sequence* Mengelola Perpindahan



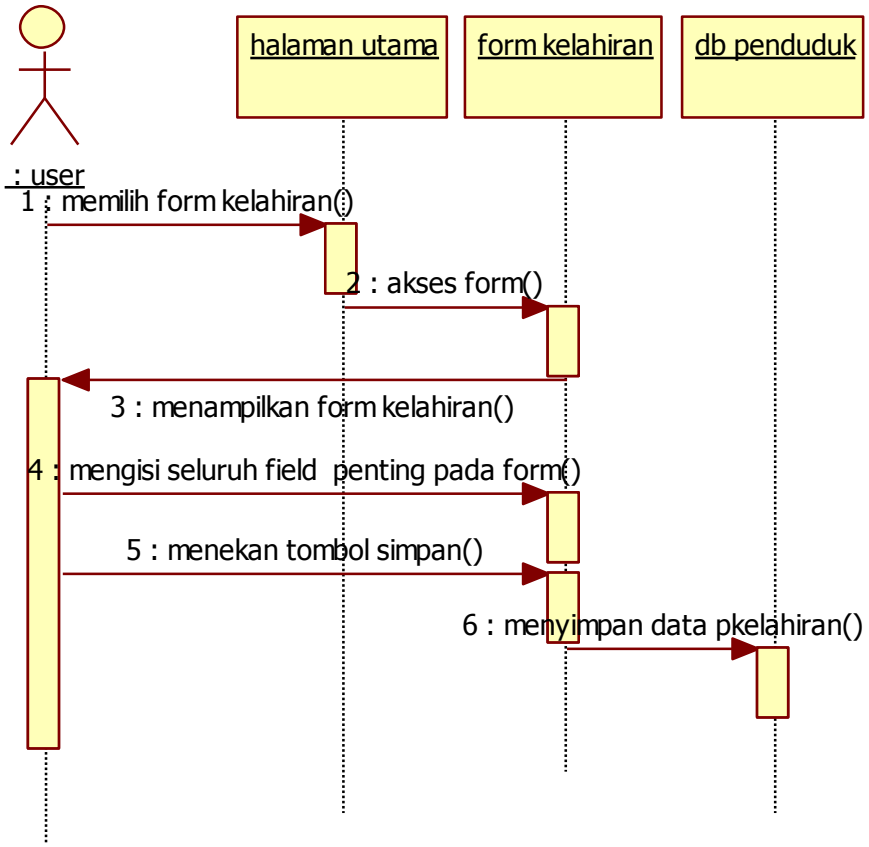
Gambar 4.13 *Diagram Sequence* Mengelola Perpindahan

5. *Diagram Sequence* Mengelola Pendatang



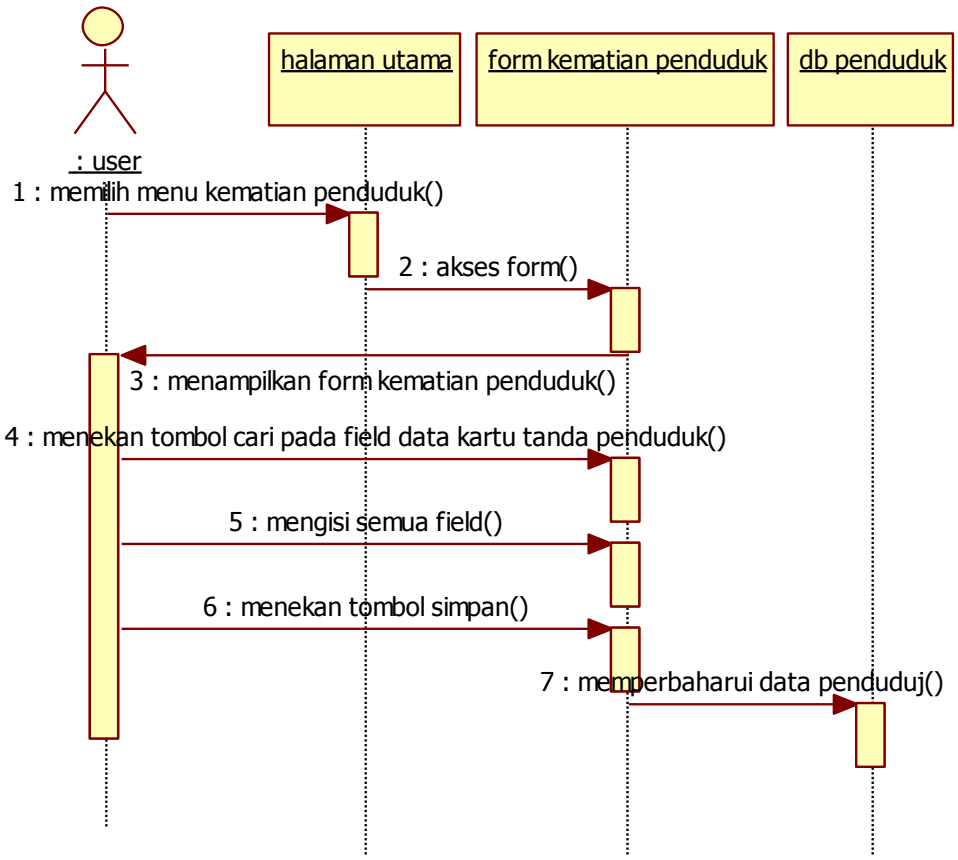
Gambar 4.14 *Diagram Sequence* Mengelola Perpindahan

6. *Diagram Sequence* Mengelola Kelahiran



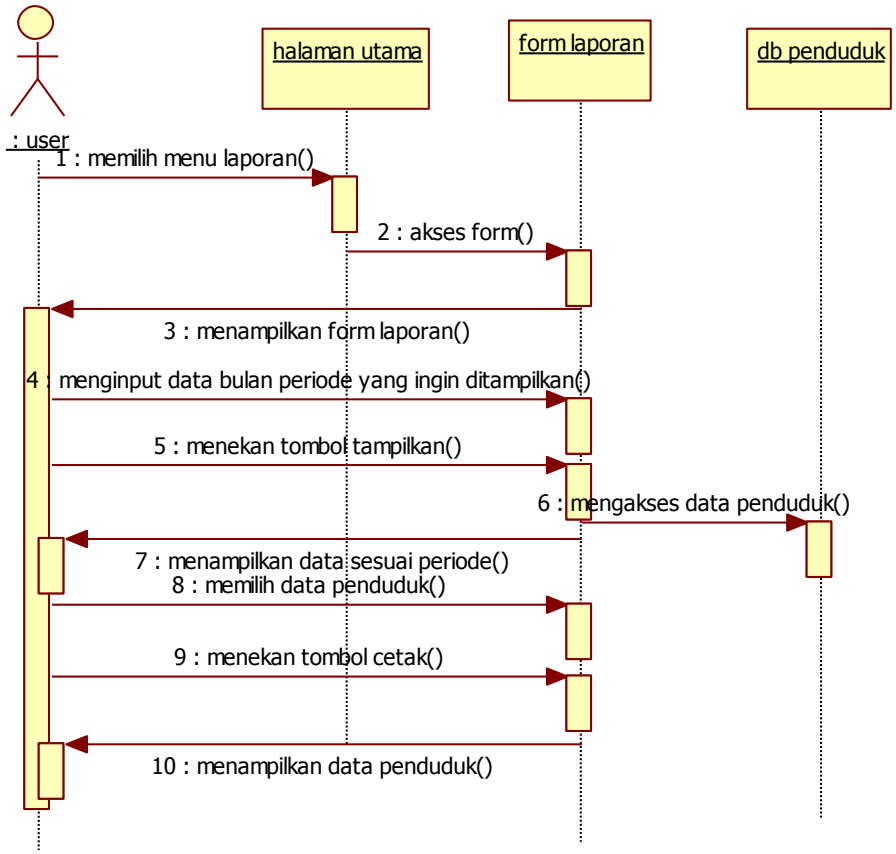
Gambar 4.15 *Diagram Sequence* Mengelola Kelahiran

7. *Diagram Sequence* Mengelola Kematian



Gambar 4.16 *Diagram Sequence* Mengelola Kematian

8. *Diagram Sequence* Mengelola Laporan

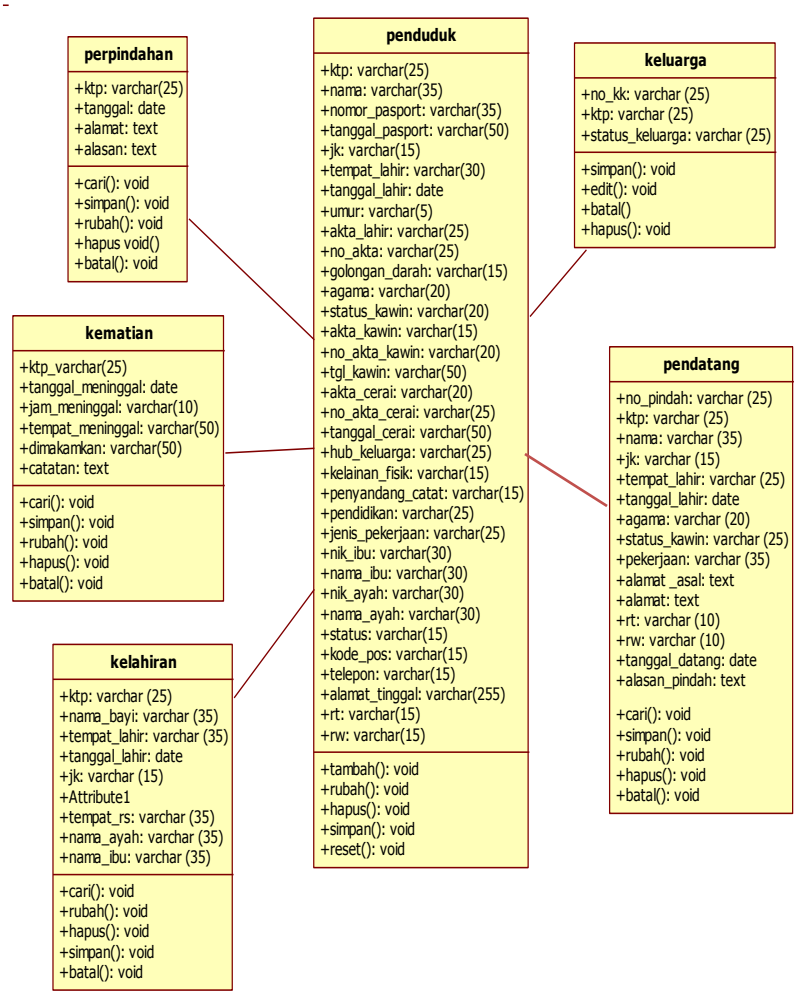


Gambar 4. 17 *Diagram Sequence* Mengelola Laporan

4.1.3.7 Diagram Class

Class Diagram adalah sebuah spesifikasi yang jika di instansiakan menghasil kan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawar kan layanan untuk memanipulasi keadaan tersebut (fungsi).*Class diagram* digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam system. *Class*

diagram memberikan gambaran system secara statis dan relasi antar mereka.Dengan melihat karakteristik system beserta proses-proses yang terjadi,maka dapat dibuat *class* diagram sebagai berikut :

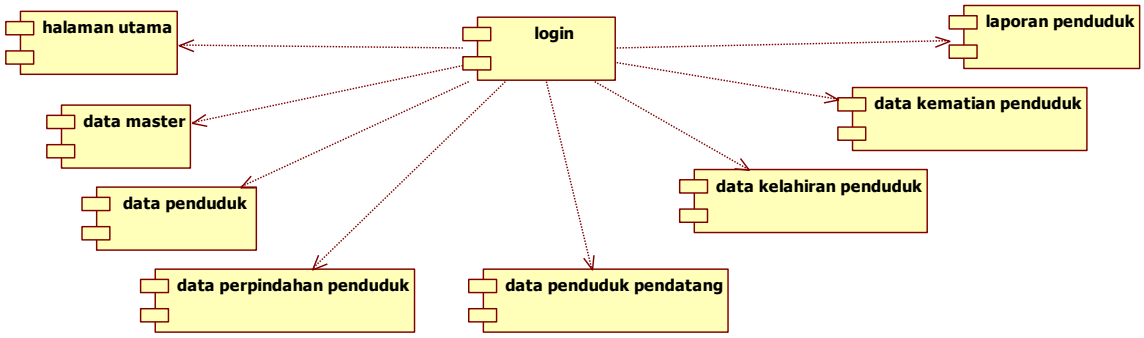


Gambar 4. 18 Diagram Class

4.1.3.8 Component Diagram

Component Diagram adalah diagram yang menampilkan komponen dalam istem dan hubungan antara komponen piranti lunak dalam hal ini mewakili

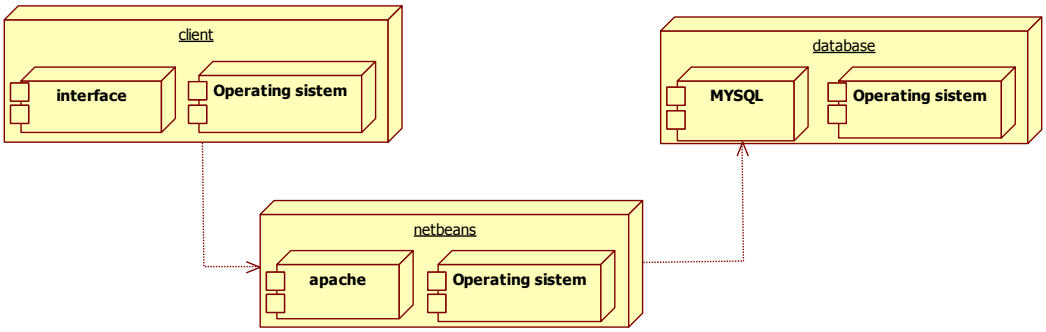
potongan-potongan yang independen yang bisa diperbaharui atau ditambahkan sewaktu-waktu oleh karena itu komponen lebih banyak didorong oleh kepentingan teknis tergantung sistem yang berjalan di dalam suatu organisasi.



Gambar 4.19 *Component Diagram* Yang Diusulkan

4.1.3.9 Deployment Diagram

Deployment / physical diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal. Sebuah node adalah *server*, *workstation*, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.



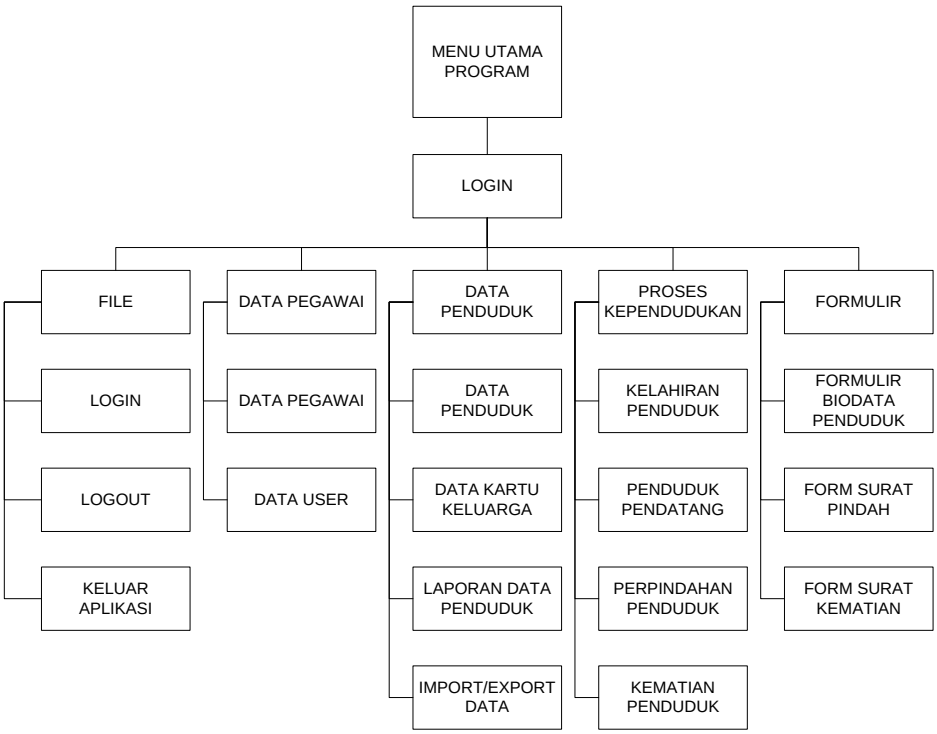
Gambar 4.20 *Deployment Diagram* Yang Diusulkan

4.2 Perancangan Antar Muka

Perancangan antar muka merupakan tahapan untuk membuat tampilan atau design dari sistem yang akan dibuat. Rancangan tampilan yang dibuat meliputi rancangan input dan rancangan output dari sistem yang akan dibuat. Perancangan perancangan antar muka pada Aplikasi data kependudukandi Kecamatan Majalaya adalah sebagai berikut.

4.2.1 Struktur Menu

Perancangan menu dibuat sebagai alat antar muka dengan pengguna untuk memudahkan pengoperasian aplikasi. Berikut rancangan menu aplikasi ini:



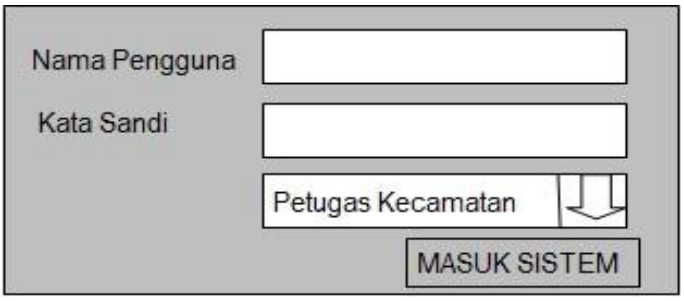
Gambar 4.21 Struktur Menu

4.2.2 Perancangan *Input*

Perancangan *input* diperlukan untuk menentukan tampilan program yang berfungsi sebagai tempat memasukan data. Perancangan input merupakan perancangan tampilan yang akan digunakan guna memasukkan data pada sistem untuk kemudian diproses. Dalam perancangan input ini, data yang dimasukkan akan mempengaruhi hasil yang ditampilkan. Adapun perancangan-perancangan input yang ada dalam perancangan ini adalah :

1. *Form Login*

   Rancangan *form* ini berfungsi untuk menginputkan *username* dan *password* ketika kita akan *login*.



Gambar 4.22 Perancangan *Form Login*

2. *Form* Menu Utama

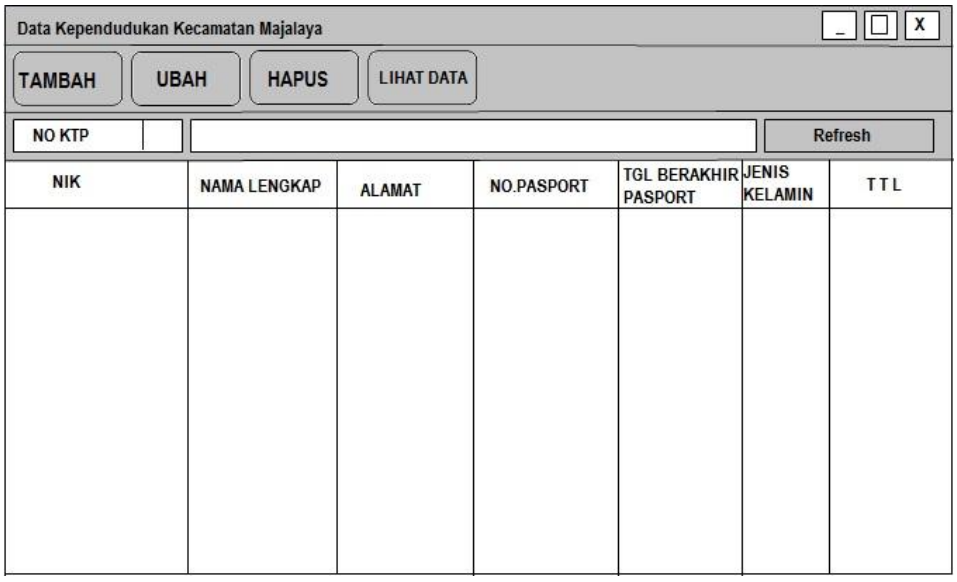
Rancangan *form* ini berfungsi sebagai halaman utama.



Gambar 4.23 Perancangan *Form* Utama

3. *Form* Menu Data Penduduk

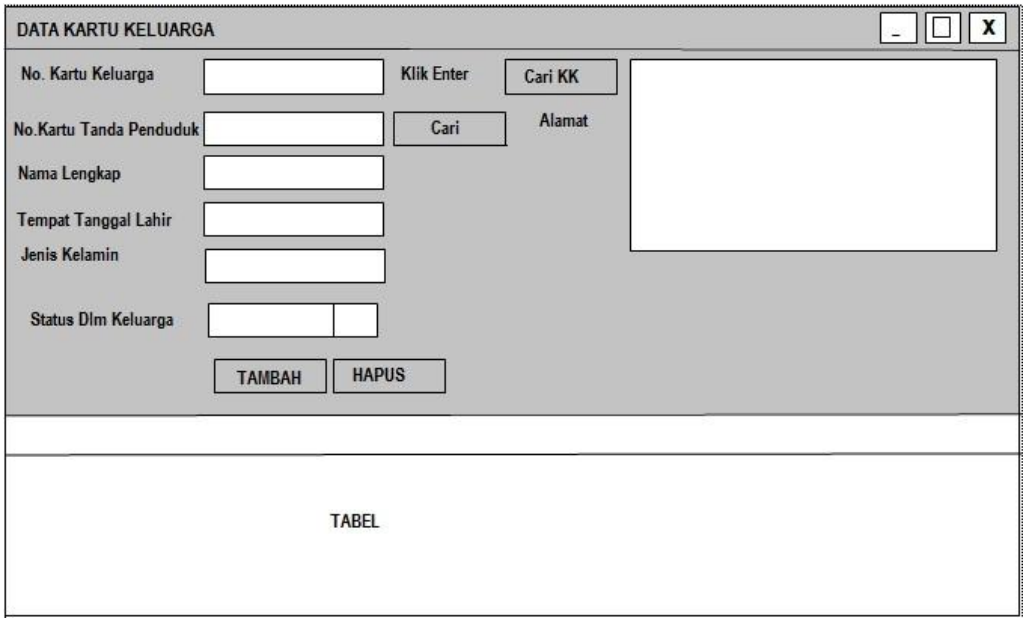
Rancangan *form* ini berfungsi sebagai menampilkan, menambah, merubah dan menghapus data penduduk.



Gambar 4.24 Perancangan *Form* Menu Data Penduduk

4. *Form* Data Kartu Keluarga

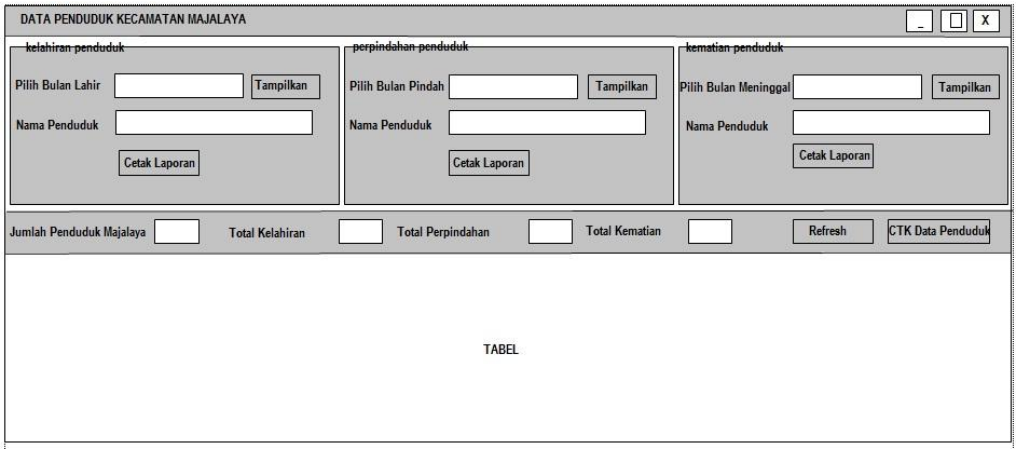
Rancangan *form* ini berfungsi untuk menginputkan data penduduk dan mengelompokkannya ke dalam data kartu keluarga. Dimana setiap penduduk akan memiliki keluarga masing-masing yang sesuai dengan Kartu Keluarga.



Gambar 4.25 Perancangan *Form* Data Kartu Keluarga

5. Perancangan Form Laporan Penduduk

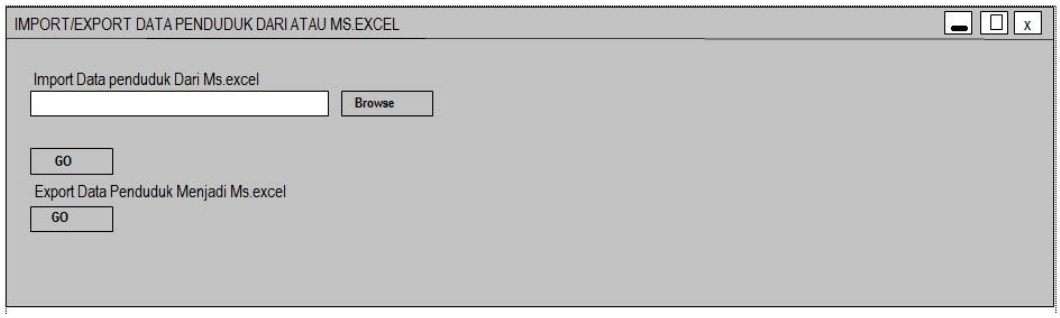
Form ini berfungsi untuk menampilkan atau mencetak laporan data penduduk yang melahirkan, pindah dan meninggal.



Gambar 4.26 Perancangan Form Laporan Penduduk

6. Perancangan Form Import/Export Data Penduduk ke Ms. Excel

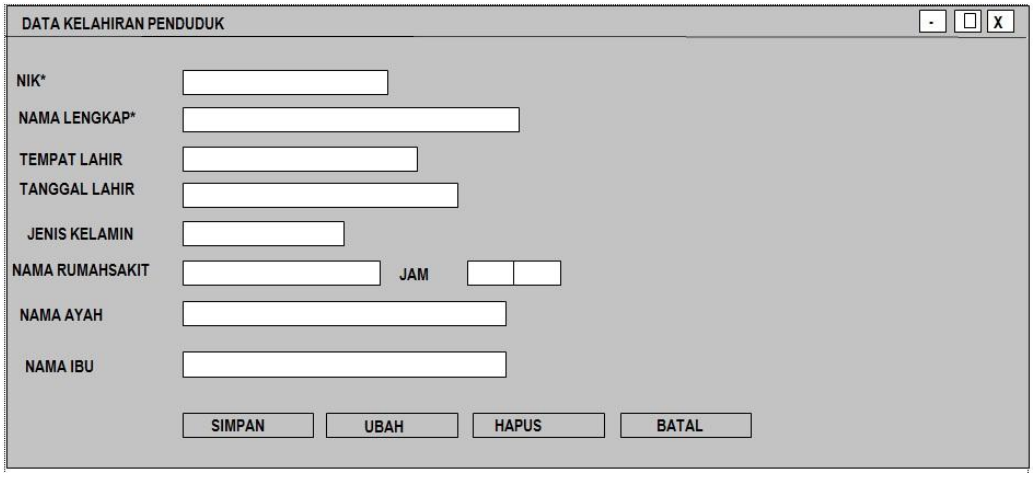
Form ini berfungsi untuk mengimport data dari file Excel ke database penduduk atau mengexport data dari database penduduk ke file Excel.



Gambar 4.27 Perancangan Form Import/Export Data Penduduk

7. *Form* Kelahiran

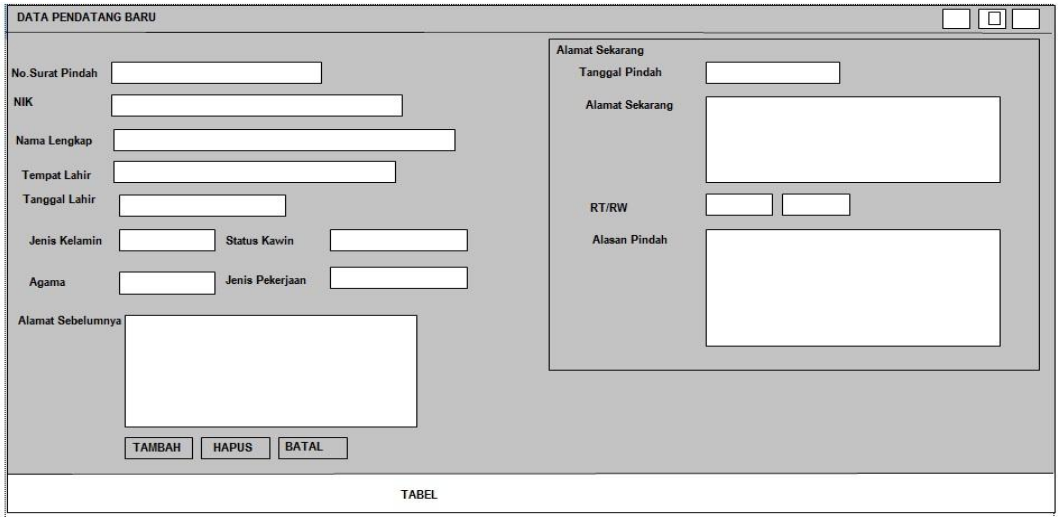
Rancangan *form* ini berfungsi untuk menginputkan data kelahiran penduduk di kecamatan majalaya.



Gambar 4.28 Perancangan *Form* Kelahiran

8. Form Penduduk Pendatang

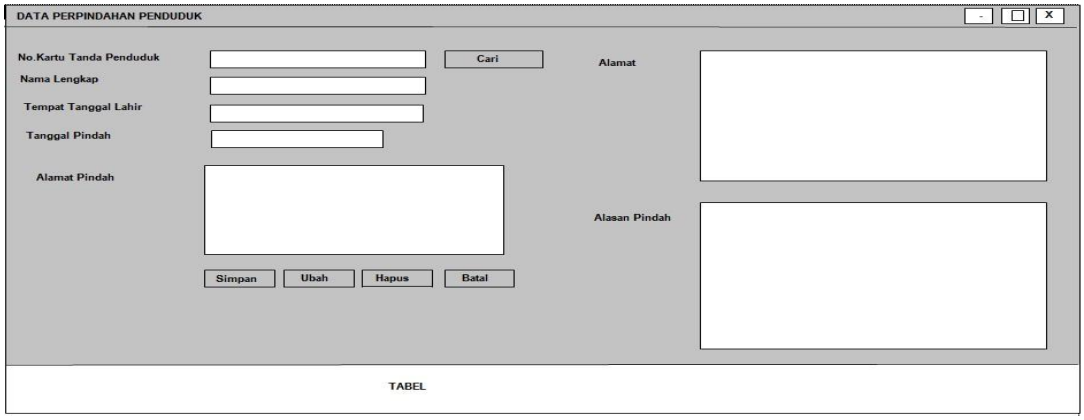
Form ini bergfungsi untuk mencatat data penduduk yang pindah dari kecamatan atau kota lain ke kecamatan majalaya.



Gambar 4.29 Perancangan Form Penduduk Pendatang

9. Form Penduduk Pindah

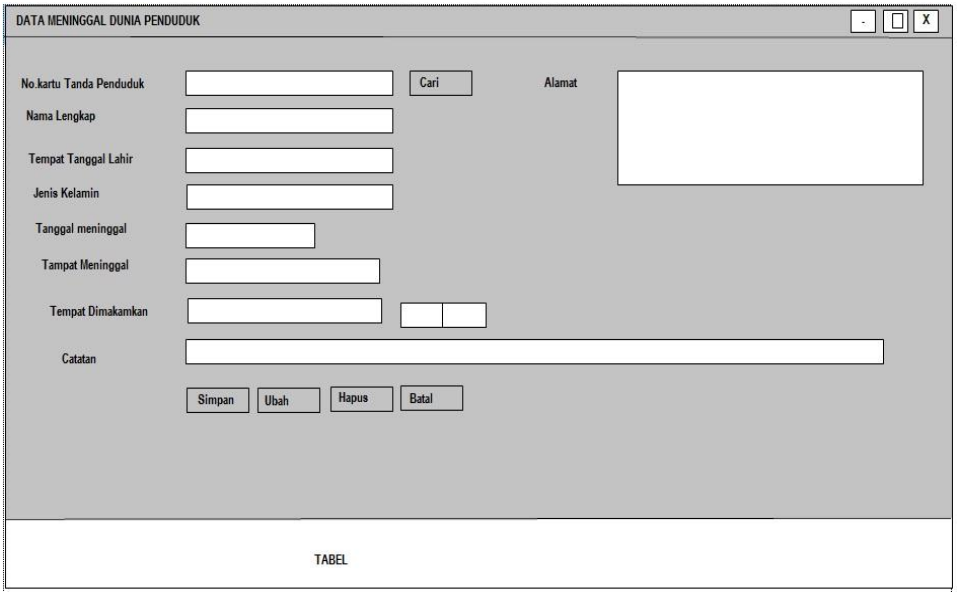
Form perpindahan penduduk ini berfungsi untuk mencatat data penduduk yang akan pindah tempat tinggal atau meninggalkan kecamatan majalaya.



Gambar 4.30 Perancangan Form Penduduk Pindah

10. *Form* Data Penduduk Meninggal

Rancangan *form* ini berfungsi untuk mencatat data penduduk yang meninggal dunia di kecamatan majalaya.



Gambar 4.31 Perancangan *Form* Data Penduduk Meninggal

4.2.3 Perancangan Output

Perancangan *output* diperlukan untuk menentukan tampilan program yang berfungsi sebagai tempat menampilkan informasi dan data yang telah di *input*kan. Dari aplikasi ini, output yang dihasilkan sebagai berikut :

1. Perancangan Surat Pindah



PEMERINTAH DAERAH TINGKAT II BANDUNG

"RT."+$F{penduduk_rt}+" - RW."+$F

DESA. MAJALAYA KEC. MAJALAYA KAB.BANDUNG

Surat Pengantar Keterangan Pindah :

Nomor : 4 75/ /RT. / /20

Nama	: $F{penduduk_nama}
Jenis Kelamin	: $F{penduduk_jk}
Tempat / Tanggal Lahir	: $F{penduduk_tempat_lahir}+","+$F{penduduk_tanggal_lahir}
Kewarganegaraan	: Indonesia
Agama	: $F{penduduk_agama}
Status Perkawinan	: $F{penduduk_status_kawin}
Pekerjaan	: $F{penduduk_jenis_pekerjaan}
Alamat	: Kp. $F{penduduk_alamat_tinggal} $F /Rt. $F
	: Desa Majalaya
	: Kec. Majalaya Kab. Bandung
No KTP / KK	: $F{perpindahan_ktp}
Pindah Ke	: Kp. $F{perpindahan_alamat} /Rt.
	: Desa. Majalaya
	: Kecamatan. Majalaya
	: Kabupaten. Bandung
	: Provinsi.Jawa Barat
	: Pada Tanggal. $F{perpindahan_tanggal}
Alasan Pindah	: $F{perpindahan_alasan}

Mengetahui,

Ketua RW $F.....

( )

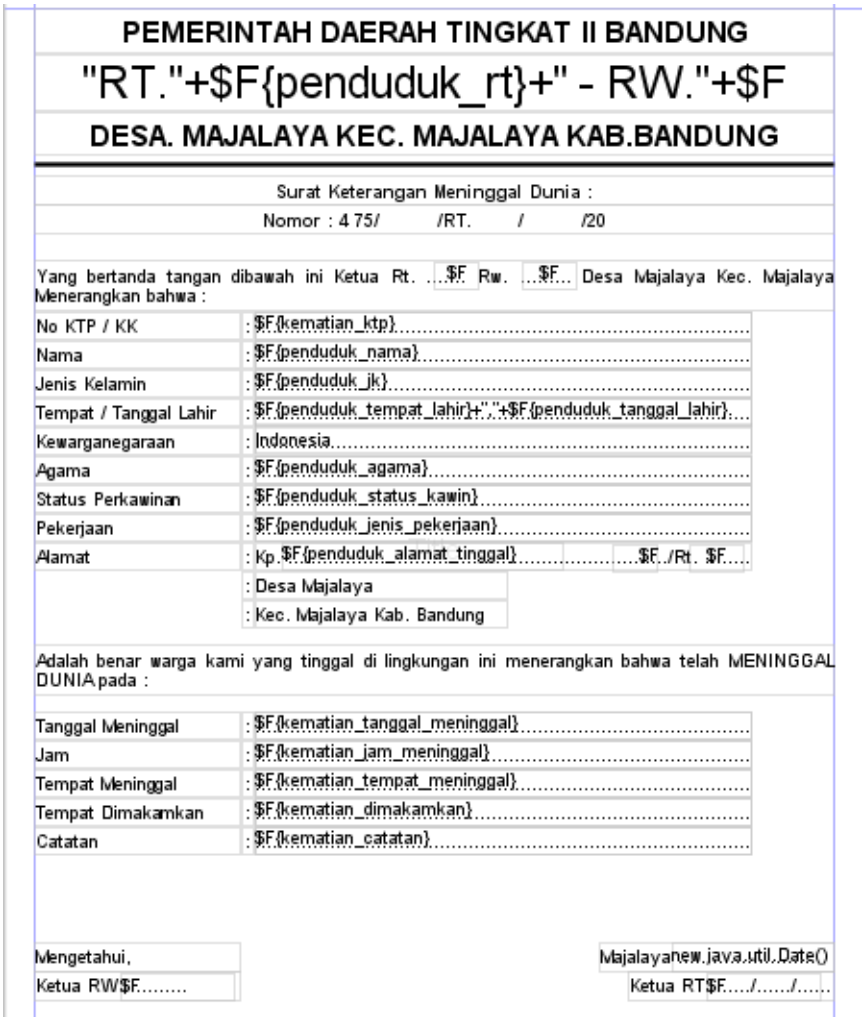
Majalaya

Ketua RT $F...../...../.....

( )

Gambar 4.32 Surat Keterangan Pindah

2. Perancangan Surat Keterangan Meninggal



PEMERINTAH DAERAH TINGKAT II BANDUNG

"RT."+$F{penduduk_rt}+" - RW."+$F

DESA. MAJALAYA KEC. MAJALAYA KAB.BANDUNG

Surat Keterangan Meninggal Dunia :

Nomor : 4 75/ /RT. / /20

Yang bertanda tangan dibawah ini Ketua Rt. ...$F Rw. ...$F... Desa Majalaya Kec. Majalaya Menerangkan bahwa :

No KTP / KK	: $F{kematian_ktp}
Nama	: $F{penduduk_nama}
Jenis Kelamin	: $F{penduduk_jk}
Tempat / Tanggal Lahir	: $F{penduduk_tempat_lahir}+","+$F{penduduk_tanggal_lahir}
Kewarganegaraan	: Indonesia
Agama	: $F{penduduk_agama}
Status Perkawinan	: $F{penduduk_status_kawin}
Pekerjaan	: $F{penduduk_jenis_pekerjaan}
Alamat	: Kp. $F{penduduk_alamat_tinggal} ....$F. /Rt. $F....
	: Desa Majalaya
	: Kec. Majalaya Kab. Bandung

Adalah benar warga kami yang tinggal di lingkungan ini menerangkan bahwa telah MENINGGAL DUNIA pada :

Tanggal Meninggal	: $F{kematian_tanggal_meninggal}
Jam	: $F{kematian_jam_meninggal}
Tempat Meninggal	: $F{kematian_tempat_meninggal}
Tempat Dimakamkan	: $F{kematian_dimakamkan}
Catatan	: $F{kematian_catatan}

Mengetahui,

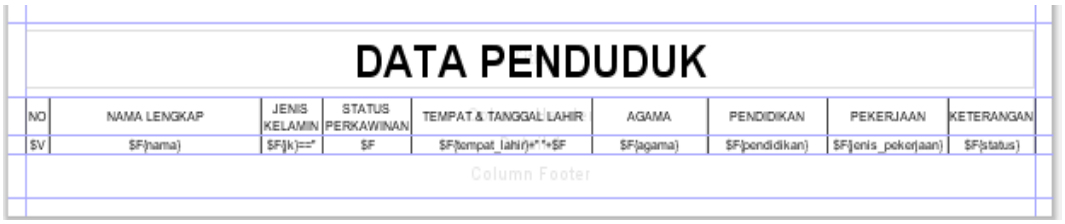
Ketua RW$F.....

Majalaya new.java.util.Date()

Ketua RT$F...../...../.....

Gambar 4.33 Surat Keterangan Meninggal

3. Laporan Data Penduduk



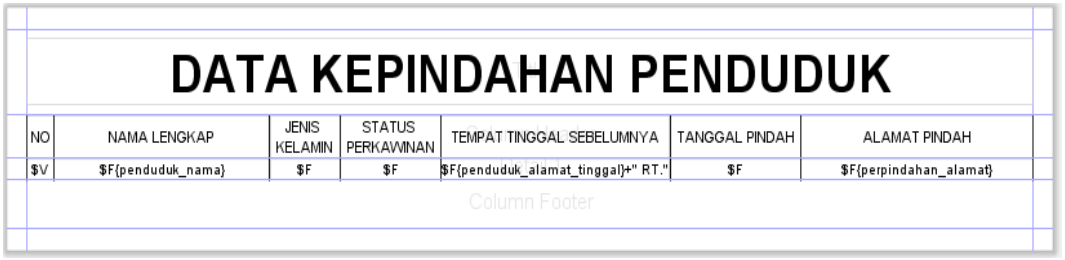
DATA PENDUDUK

NO	NAMA LENGKAP	JENIS KELAMIN	STATUS PERKAWINAN	TEMPAT & TANGGAL LAHIR	AGAMA	PENDIDIKAN	PEKERJAAN	KETERANGAN
$V	$F{nama}	$F{jk}==""	$F	$F{tempat_lahir}+" "+$F	$F{agama}	$F{pendidikan}	$F{jenis_pekerjaan}	$F{status}

Column Footer

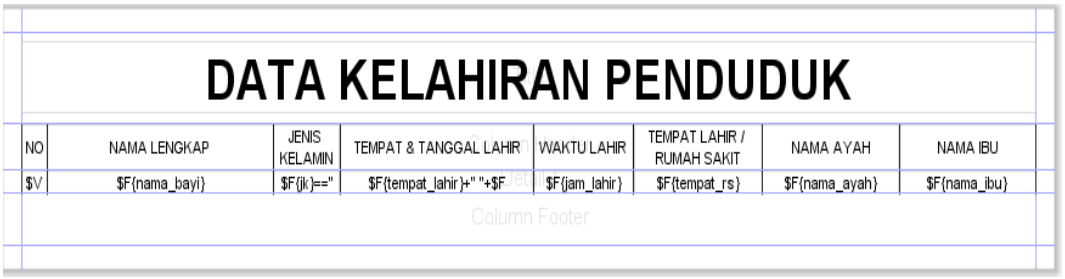
Gambar 4.34 Laporan Data Penduduk

4. Laporan Data Perpindahan Penduduk



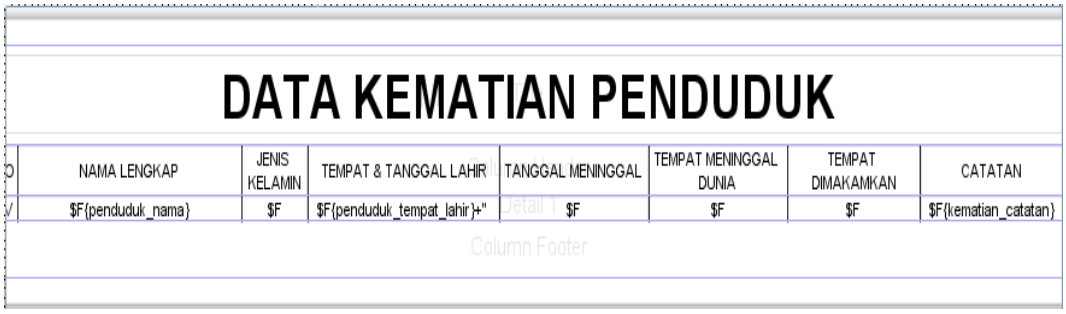
Gambar 4.35 Laporan Data Perpindahan Penduduk

5. Laporan Data Kelahiran Penduduk



Gambar 4.36 Laporan Data Kelahiran Penduduk

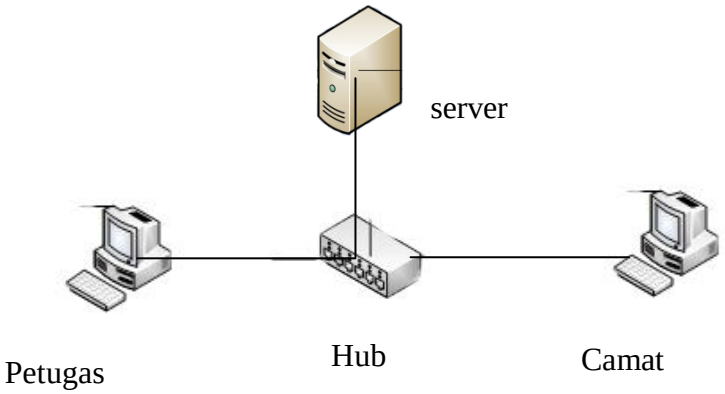
6. Laporan Data Penduduk Meninggal



Gambar 4.37 Laporan Data Penduduk Meninggal

4.3 Perancangan Arsitektur Jaringan

Dalam pengimplementasian aplikasi pendataan.. Jaringan ini berfungsi untuk sebagai penghubung antara komputer satu dan yang lainnya. Berikut adalah skema dari arsitektur jaringan yang digunakan.



Gambar 4.38 Perancangan Arsitektur Jaringan

4.4 Implementasi

Implementasi merupakan pengaplikasian perangkat lunak, perangkat keras serta *database* yang digunakan. Pada tahapan ini akan dilakukan implementasi meliputi lingkungan implementasi, batasan implementasi, dan konfigurasi dari kebutuhan sistem untuk dapat menjalankan aplikasi perangkat lunak yang telah dibangun.

4.4.1 Batasan Implementasi

Berikut adalah hal-hal yang menjadi batasan dalam implementasi dari aplikasi ini :

1. Aplikasi ini dapat diakses oleh Petugas dan Camat.

2. Penginputan data hanya dengan tipe data standar yaitu *real*, integer dan string.
3. Konfigurasi basis data harus dilakukan secara manual dengan memanfaatkan *file Dump* SQL .
4. Aplikasi yang dibuat bertujuan untuk membuat data kependudukan.

4.4.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan program aplikasi penjadwalan ini adalah sebagai berikut :

1. Sistem operasi yang digunakan dalam pembangunan dan pengujian aplikasi ini adalah *Microsoft Windows 7* dan *Microsoft Windows 7.*
2. Bahasa pemrograman yang digunakan untuk membangun applikasi ini adalah bahasa pemrograman java dengan *Java Development Kit* 6.9 (JDK6.9).
3. Perangkat Pemograman Netbeans 7.1.2.
4. Basis data dengan MySQL 5.0.67.
5. Laporan menggunakan iReport-4.0.2 *for* Windows.
6. Java *Libraries.*

4.4.3 Implementasi Perangkat Keras

Berikut adalah perangkat keras yang digunakan dalam implementasi aplikasi penjadwalan ini dengan minimal spesifikasi adalah :

1. CPU Intel Dual Core 2.0 GHz.
2. Memory DDR RAM 512 MB.

3. Harddisk 40 GB.
4. Perangkat keluaran berupa *Monitor* 14".
5. Perangkat masukan berupa *mouse* dan *keyboard*

4.4.4 Implementasi Basis Data

Pada pembangunan aplikasi ini dibangun *database* agar tercipta sistem informasi yang terintegrasi dan dinamis. Selain itu pembuatan dan implementasi *database* merupakan langkah yang memperlihatkan kesesuaian perancangan pada tahapan analisis dan perancangan terhadap pembangunan aplikasi. Berikut adalah basis data yang dibuat menggunakan bahasa SQL (MySQL) dalam Aplikasi Data Kependudukan:

```
CREATE TABLE IF NOT EXISTS `kelahiran` (
  `ktp` varchar(25) NOT NULL,
  `nama_bayi` varchar(35) NOT NULL,
  `tempat_lahir` varchar(35) NOT NULL,
  `tanggal_lahir` date NOT NULL,
  `jam_lahir` varchar(10) NOT NULL,
  `jk` varchar(15) NOT NULL,
  `tempat_rs` varchar(35) NOT NULL,
  `nama_ayah` varchar(35) NOT NULL,
  `nama_ibu` varchar(35) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
CREATE TABLE IF NOT EXISTS `keluarga` (
```

```
  `no_kk` varchar(25) NOT NULL,
  `ktp` varchar(25) NOT NULL,
  `status_keluarga` varchar(25) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
CREATE TABLE IF NOT EXISTS `kematian` (
  `ktp` varchar(25) NOT NULL,
  `tanggal_meninggal` date NOT NULL,
  `jam_meninggal` varchar(10) NOT NULL,
  `tempat_meninggal` varchar(50) NOT NULL,
  `dimakamkan` varchar(50) NOT NULL,
  `catatan` text NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
CREATE TABLE IF NOT EXISTS `petugas` (
  `nip_pegawai` varchar(15) NOT NULL,
  `nama_pegawai` varchar(25) NOT NULL,
  `alamat` text NOT NULL,
  `no_telepon` varchar(15) NOT NULL,
  `jabatan` varchar(20) NOT NULL,
  PRIMARY KEY (`nip_pegawai`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
CREATE TABLE IF NOT EXISTS `penduduk` (
  `ktp` varchar(25) NOT NULL,
```

```
`nama` varchar(35) DEFAULT NULL,
`alamat` varchar(255) DEFAULT NULL,
`nomor_pasport` varchar(35) DEFAULT NULL,
`tanggal_passport` varchar(50) DEFAULT NULL,
`jk` varchar(15) DEFAULT NULL,
`tempat_lahir` varchar(30) DEFAULT NULL,
`tanggal_lahir` date DEFAULT NULL,
`umur` varchar(5) DEFAULT NULL,
`akta_lahir` varchar(25) DEFAULT NULL,
`no_akta` varchar(25) DEFAULT NULL,
`golongan_darah` varchar(15) DEFAULT NULL,
`agama` varchar(20) DEFAULT NULL,
`status_kawin` varchar(20) DEFAULT NULL,
`akta_kawin` varchar(15) DEFAULT NULL,
`no_akta_kawin` varchar(20) DEFAULT NULL,
`tgl_kawin` varchar(50) DEFAULT NULL,
`akta_cerai` varchar(20) DEFAULT NULL,
`no_akta_cerai` varchar(25) DEFAULT NULL,
`tanggal_cerai` varchar(50) DEFAULT NULL,
`hub_keluarga` varchar(25) DEFAULT NULL,
`kelainan_fisik` varchar(15) DEFAULT NULL,
`penyandang_cacat` varchar(25) DEFAULT NULL,
```

```
  `pendidikan` varchar(25) DEFAULT NULL,
  `jenis_pekerjaan` varchar(25) DEFAULT NULL,
  `nik_ibu` varchar(30) DEFAULT NULL,
  `nama_ibu` varchar(30) DEFAULT NULL,
  `nik_ayah` varchar(30) DEFAULT NULL,
  `nama_ayah` varchar(30) DEFAULT NULL,
  `status` varchar(15) DEFAULT NULL,
  `kode_pos` varchar(15) DEFAULT NULL,
  `telepon` varchar(15) DEFAULT NULL,
  `alamat_tinggal` varchar(255) DEFAULT NULL,
  `rt` varchar(15) DEFAULT NULL,
  `rw` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`ktp`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
CREATE TABLE IF NOT EXISTS `perpindahan` (
  `ktp` varchar(25) NOT NULL,
  `tanggal` date NOT NULL,
  `alamat` text NOT NULL,
  `alasan` text NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
CREATE TABLE IF NOT EXISTS `user` (
  `nip_pegawai` varchar(15) NOT NULL,
```

```
  `username` varchar(25) NOT NULL,
  `password` varchar(25) NOT NULL,
  `hak_akses` varchar(20) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

4.4.5 Implementasi Antar Muka

Antarmuka dalam sebuah aplikasi adalah hal yang sangat penting, aplikasi *desktop* yang menggunakan bahasa JAVA ini, implementasi antarmuka dilakukan dengan sebuah halaman yang berekstensi .java. Implementasi antarmuka dari halaman yang dibuat dapat dilihat pada tabel di bawah ini.

1. Implementasi Antarmuka

Tabel 4.11 Implementasi Antarmuka

Menu	Deskripsi	Tipe
File	Menu yang berisi form login dan logout untuk masuk ke system	*File.java*
Data Pegawai	Menu yang berisi form data pegawai dan user yang mengakses system	*Datapegawai.java*
Data Penduduk	Menu yang berisi data kependudukan	*Datapenduduk.java*
Proses Kependudukan	Menu yang berisi menu untuk memproses data kependudukan	*Prosespenduduk.java*

2. Impementasi Menu File

Tabel 4.12 Implementasi Menu File

Menu	Deskripsi	Tipe
Login	Menu yang berfungsi menampilkan form login user	*Login.java*
Logout	Menu untuk keluar dari menu aplikasi	*Logout.java*
Keluar aplikasi	Menu untuk keluar dari aplikasi	*Exit.java*

3. Implementasi Menu Data Pegawai

Tabel 4.13 Implementasi Menu Data Pegawai

Menu	Deskripsi	Tipe
Data pegawai	Menu yang berfungsi untuk mengelola data pegawai	*Pegawai.java*
Data user	Menu yang berfungsi untuk mengelola data user	*User.java*

4. Implementasi Menu Data Penduduk

Tabel 4.14 Implementasi Menu Data Penduduk

Menu	Deskripsi	Tipe
Data penduduk	Menu yang berfungsi untuk	*Dpenduduk.java*

	mengelola data penduduk majalaya	
Data Kartu keluarga	Menu yang berfungsi untuk mengelola data keluarga	*DkartuKeluarga.java*
Laporan Data Penduduk	Menu yang berfungsi untuk membuat laporan penduduk	*Laporan.java*
Import/Export data penduduk	Menu yang berfungsi untuk mengexport dan import data penduduk	*Import.java*

5. Implementasi Menu Proses Kependudukan

Tabel 4.15 Implementasi Menu Proses Kependudukan

Menu	Deskripsi	Tipe
Kelahiran Penduduk	Menu yang berfungsi untuk mengelola data kelahiran penduduk	*Kelahiran.java*
Penduduk Pendatang	Menu yang berfungsi untuk mengelola data penduduk pendatang	*Pendatang.java*
Perpindahan Penduduk	Menu yang berfungsi untuk mengelola data perpindahan penduduk	*Perpindahan.java*
Penduduk Meninggal	Menu yang berfungsi untuk mengelola data penduduk meninggal	*Penduduk.java*

4.4.6 Implementasi Instalasi Program

Dalam mengimplementasikan program ke dalam komputer, maka sebelumnya harus dipersiapkan segala kebutuhan yang mendukung agar perangkat lunak dapat berjalan dengan baik pada *platform* teknologi yang kita pakai. Berikut ini adalah beberapa hal yang harus dipersiapkan terlebih dahulu, yaitu :

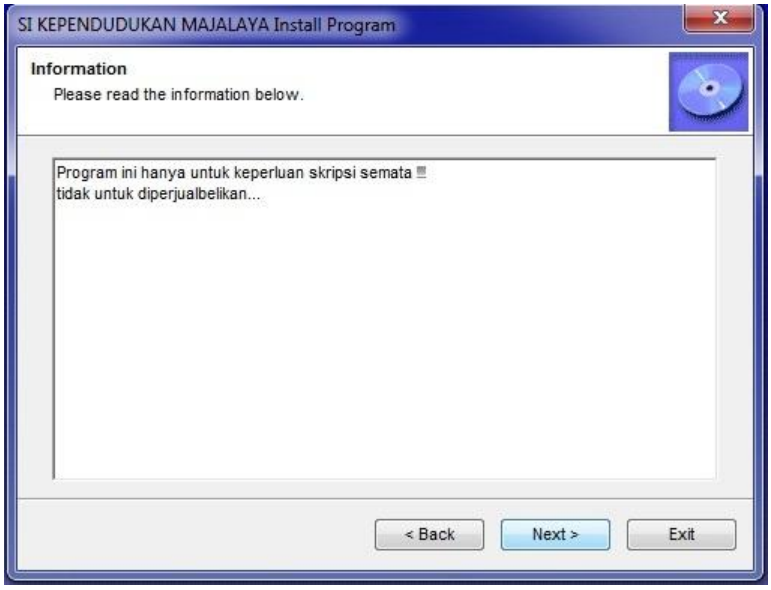
1. Instalasi XAMPP.
2. Lakukan *Dump* SQL untuk mengimport *database* ke dalam XAMPP.
3. Lalu lakukan instalasi SIKependudukan, sebagai berikut :
   a. Klik ganda File Setup SI Kependudukan Majalaya, maka akan tampak halaman “Welcome”, seperti gambar dibawah ini.



Gambar 4.39 Tampilan *Welcome*

b. Kemudian klik *Next*.

c. Maka akan muncul tampilan information yang menjelaskan keterangan program tersebut.Kemudian klik *Next*



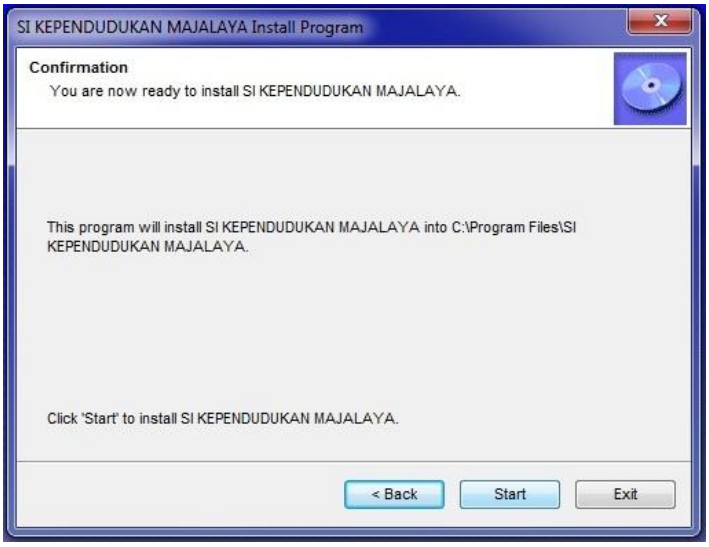
Gambar 4.40 Lokasi Informasi

d. Kemudian muncul tampilan pemilihan lokasi instalasi program, klik "*Browse*" untuk menentukan lokasi. Kali ini kita biarkan saja pada kondisi *default*. Kemudian klik *Next*.



Gambar 4.41 Ready to Install

e. Kemudian muncul tampilan confirmation lalu klik *Next* kembali



Gambar 4.42 *Setup* SI Kependudukan Majalaya

f. Tunggulah proses instalasi sampai selesai



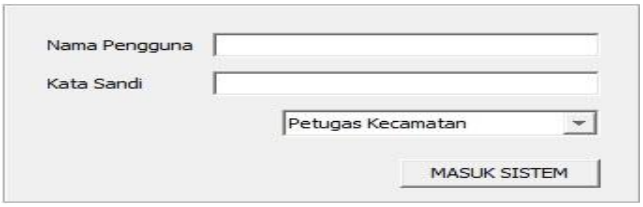
Gambar 4.43 *Finish*

4.4.7 Penggunaan Program

Kegiatan ini bertujuan untuk menerangkan secara singkat penggunaan program pembuatan data kependudukan. Adapun cara pengoperasiannya adalah sebagai berikut :

1. *Form Login*

   Ketika pertama kali program aplikasi ini dijalankan maka yang pertama kali muncul adalah *form login* seperti di bawah ini.



Gambar 4.44 Tampilan *Login*

2. Menu Utama

   Halaman ini merupakan halaman yang berisikan menu-menu yang dapat dilakukan oleh admin pada aplikasi SI Kependudukan Majalaya.



Gambar 4.45 Tampilan Menu Utama

3. Menu data penduduk merupakan menu pada halaman utama. Dimana didalamnya terdapat 4 sub menu, yaitu Data Penduduk, Data Kartu Keluarga, Laporan Data Pnduduk, import/export Data Penduduk



Gambar 4.46 Tampilan Menu Data Penduduk

4. Menu Proses Kependudukan merupakan menu pada halaman utama. Dimana didalamnya terdapat 4 sub menu, yaitu. Kelahiran Penduduk, Penduduk Pendatang, Perpindahan Penduduk, Kematian Penduduk



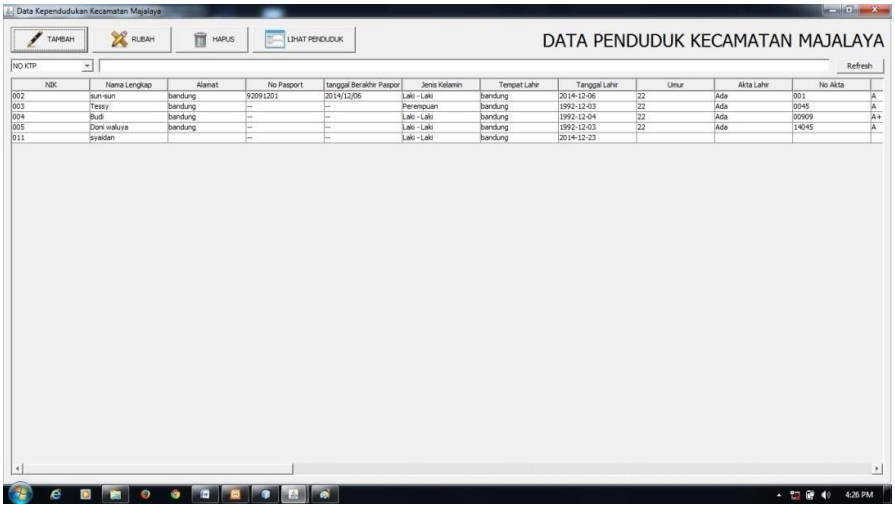
Gambar 4.47 Tampilan Menu Proses Kependudukan

5. Menu formulir merupakan menu pada halaman admin. Dimana didalamnya terdapat 3 sub menu, yaitu formulir Biodata Penduduk, Surat Pengantar Keterangan Pindah, Surat Pengantar Kematian.



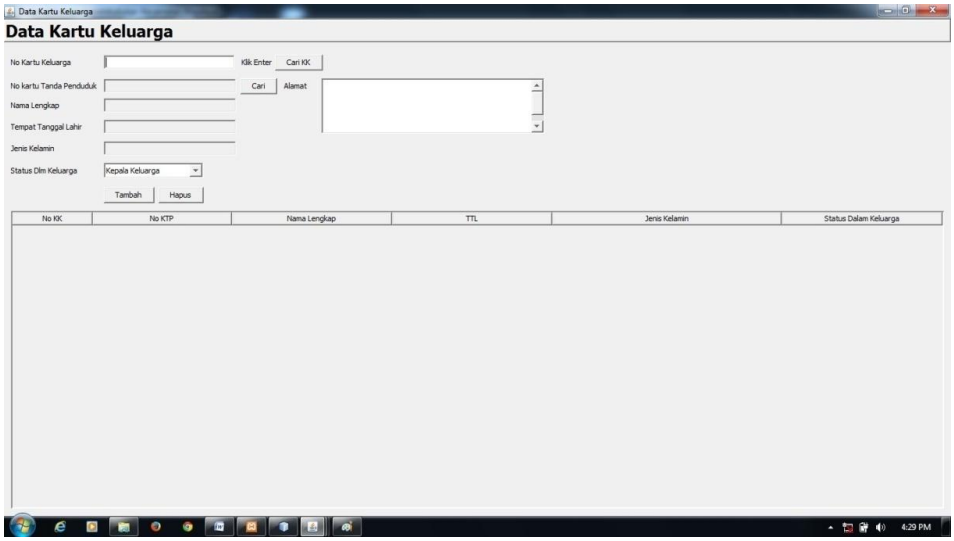
Gambar 4.48 Tampilan Menu Formulir

6. Sub menu Data Kependudukan merupakan *form* pada halaman admin untuk menambah, menghapus, mengubah, menyimpan, serta melihat data.



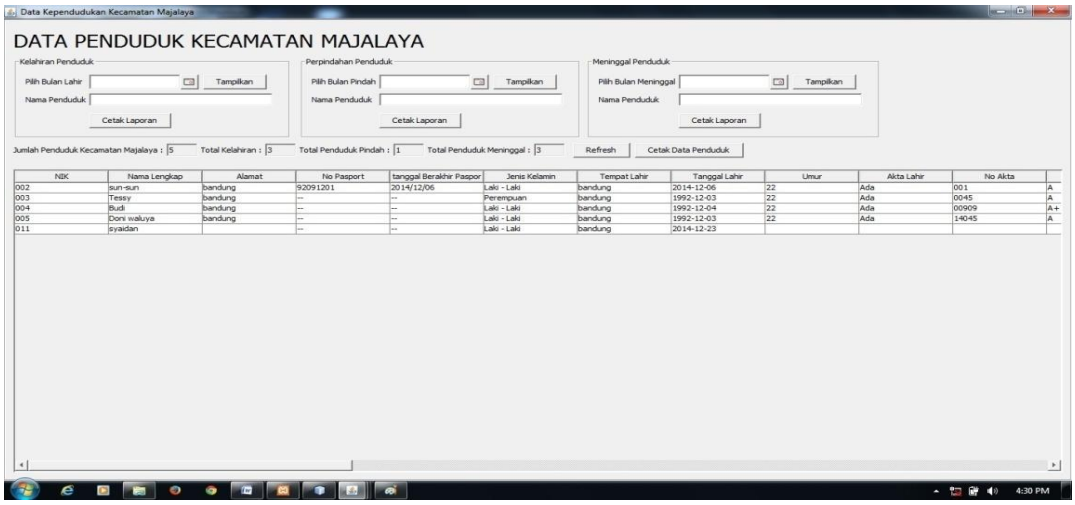
Gambar 4.49 *Form* Data Penduduk

7. Sub menu Data Kartu Keluarga merupakan *form* pada halaman admin untuk mengelompokan data data penduduk sesuai dengan NIK KK.



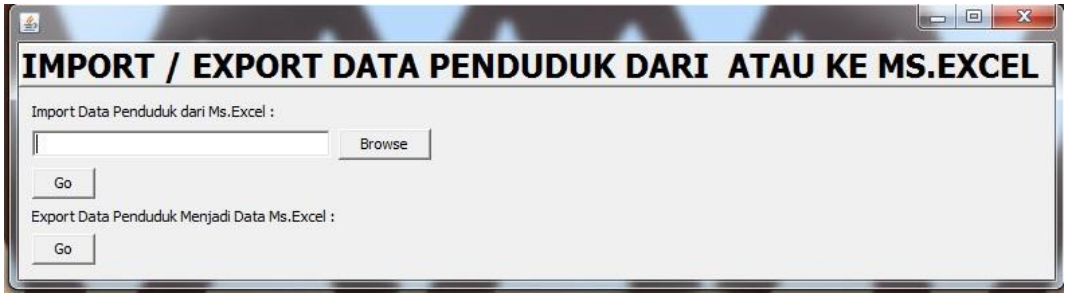
Gambar 4.50 *Form* Data Kartu Keluarga

8. Sub menu Data Kependudukan merupakan *form* pada halaman admin untuk mengetahui jumlah jumlah data penduduk .



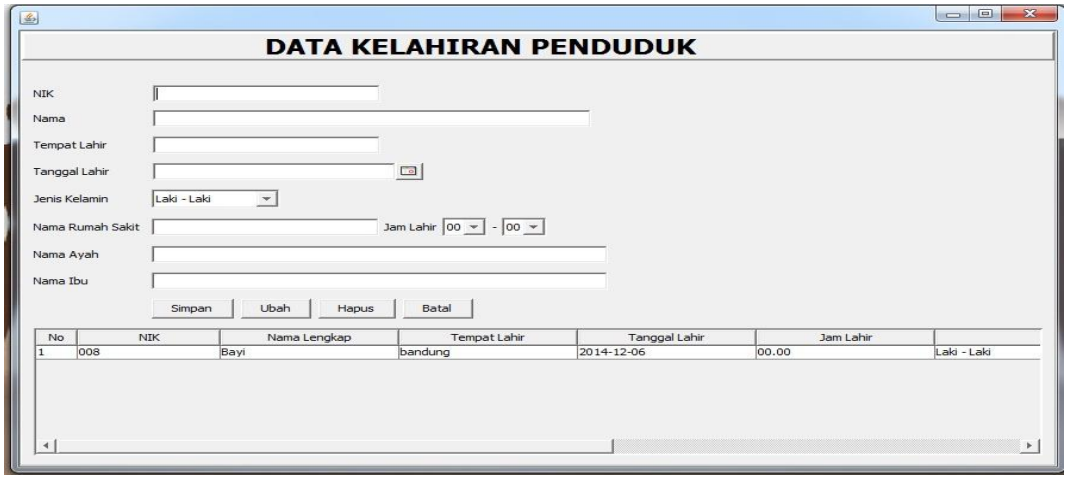
Gambar 4.51 *Form* Laporan Data Penduduk

9. Sub menu import/export data penduduk merupakan *form* pada halaman admin untuk mengimport atau export data penduduk dari atau ke ms.excel.



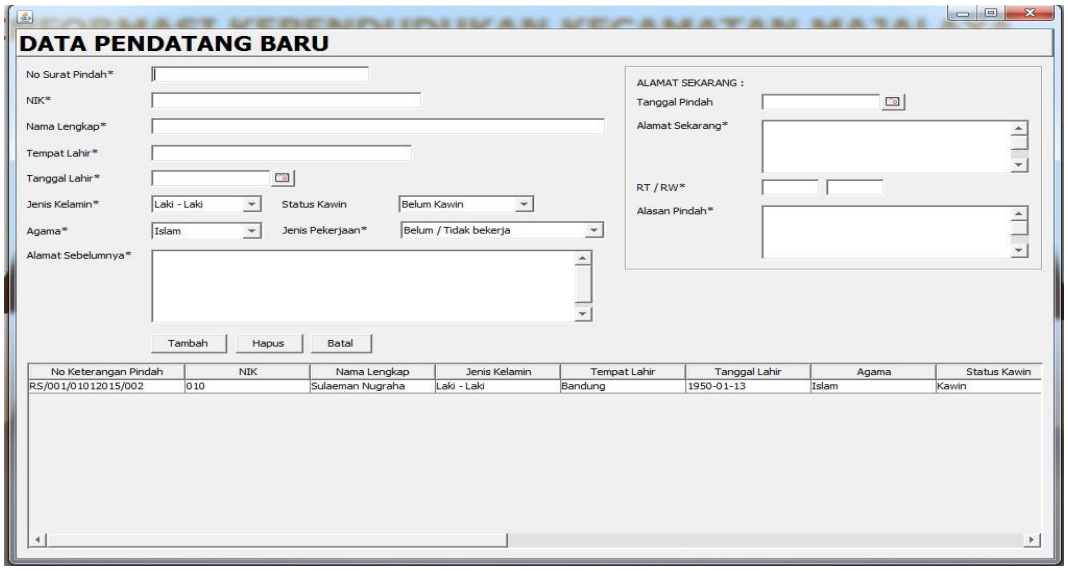
Gambar 4.52 *Form* import/export data penduduk

10. Sub Kelahiran Penduduk merupakan *form* pada halaman admin untuk menginput data penduduk yang melahirkan.



Gambar 4.53 *Form* Kelahiran Penduduk

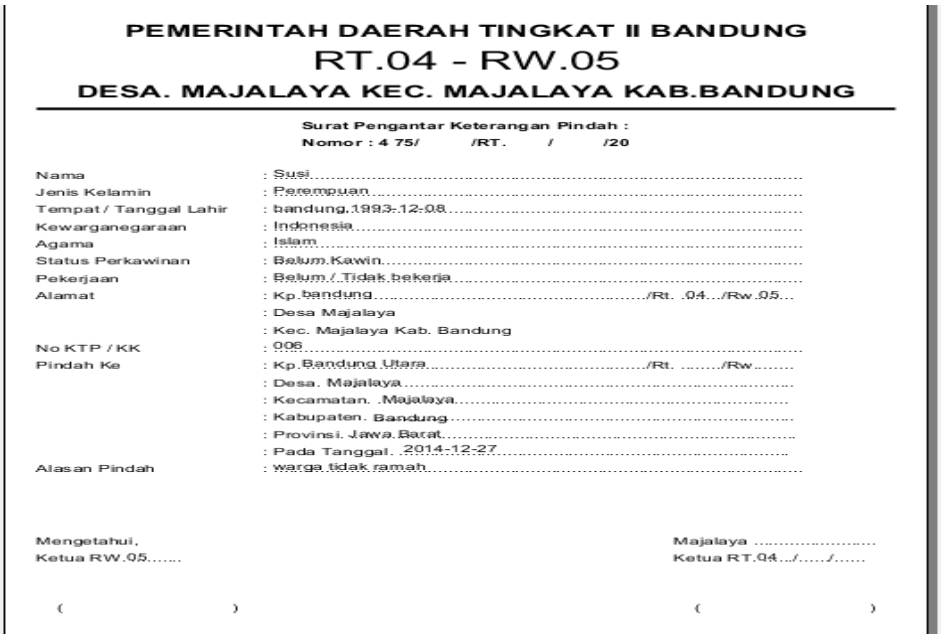
11. Sub Penduduk Pendatang pada halaman admin untuk menginput/mendata penduduk pendatang.



Gambar 4.54 *Form* Penduduk Pendatang

12. Sub menu Data Kepindahan Penduduk merupakan *form* pada halaman admin menginput/mendata penduduk yang pindah. Sesudah menginput data penduduk yang pindah maka sistem akan mencetak surat pindah penduduk.





PEMERINTAH DAERAH TINGKAT II BANDUNG
RT.04 - RW.05
DESA. MAJALAYA KEC. MAJALAYA KAB.BANDUNG

Surat Pengantar Keterangan Pindah :
Nomor : 4 75/ /RT. / /20

Nama	: Susi
Jenis Kelamin	: Perempuan
Tempat / Tanggal Lahir	: bandung,1993-12-08
Kewarganegaraan	: Indonesia
Agama	: Islam
Status Perkawinan	: Belum Kawin
Pekerjaan	: Belum / Tidak bekerja
Alamat	: Kp.bandung /Rt. .04 /Rw.05
	: Desa Majalaya
	: Kec. Majalaya Kab. Bandung
No KTP / KK	: 006
Pindah Ke	: Kp.Bandung Utara /Rt. ...../Rw.....
	: Desa. Majalaya
	: Kecamatan. .Majalaya
	: Kabupaten. Bandung
	: Provinsi. Jawa Barat
	: Pada Tanggal. .2014-12-27
Alasan Pindah	: warga tidak ramah

Mengetahui,
Ketua RW.05.....

Majalaya .....
Ketua RT.04.../...../.....

( ) ( )

Gambar 4.55 *Form* Data Perpindahan Penduduk

13. Sub menu Kematian Penduduk merupakan *form* pada halaman admin untuk menginput/mendata penduduk yang meninggal





PEMERINTAH DAERAH TINGKAT II BANDUNG

RT.05 - RW.04

DESA. MAJALAYA KEC. MAJALAYA KAB.BANDUNG

Surat Keterangan Meninggal Dunia :
Nomor : 4 75/ /RT. / /20

Yang bertanda tangan dibawah ini Ketua Rt. ..05. Rw. ...04... Desa Majalaya Kec. Majalaya Menerangkan bahwa :

No KTP / KK	: 009
Nama	: Bagus
Jenis Kelamin	: Laki - Laki
Tempat / Tanggal Lahir	: Bandung,1980-12-27
Kewarganegaraan	: Indonesia
Agama	: Islam
Status Perkawinan	: Belum Kawin
Pekerjaan	: PNS
Alamat	: Kp.bandung ../Rt. .05.../Rw.04...
	: Desa Majalaya
	: Kec. Majalaya Kab. Bandung

Adalah benar warga kami yang tinggal di lingkungan ini menerangkan bahwa telah MENINGGAL DUNIA pada :

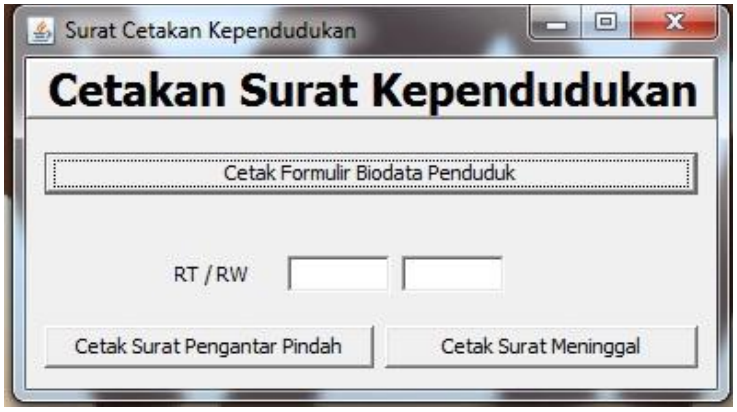
Tanggal Meninggal	: 2014-12-27
Jam	: 05.00
Tempat Meninggal	: Rumah
Tempat Dimakamkan	: Kondang
Catatan	: mati dengan bahagia

Mengetahui,
Ketua RW.04.....

Majalaya January 09, 2015
Ketua RT.05.../...../.....

Gambar 4.56 *Form* Kematian Penduduk

14. Sub menu Formulir data Kependudukan merupakan *form* pada halaman admin untuk Untuk mencatak surat kependudukan, surat keterangan pindah, surat keterangan meninggal



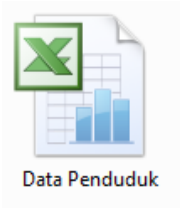
Gambar 4.57 *Form* Formulir data Kependudukan

15. Hasil dari output-output yang berupa laporan-laporan kependudukan dan file excel hasil export data penduduk dari database. Berikut merupakan hasil output dari data penduduk yang diolah.

DATA PENDUDUK

NO	NAMA LENGKAP	JENIS KELAMIN	STATUS PERKAWINAN	TEMPAT & TANGGAL LAHIR	AGAMA	PENDIDIKAN	PEKERJAAN	KETERANGAN
1	sun-sun	LK	Kawin	bandung 2014-12-06	Islam	Tidak Sekolah	Belum / Tidak bekerja	Hidup
2	oki kurniawan	LK	Belum Kawin	bandung 2014-12-06	Islam	Tidak Sekolah	Belum / Tidak bekerja	Hidup
3	Tessy	PR	Kawin	bandung 1992-12-03	Katolik	Tidak Sekolah	Belum / Tidak bekerja	Hidup
4	Budi	LK	Belum Kawin	bandung 1992-12-04	Islam	Tidak Sekolah	Belum / Tidak bekerja	Hidup
5	Doni waluya	LK	Belum Kawin	bandung 1992-12-03	Islam	Tidak Sekolah	Belum / Tidak bekerja	Hidup
6	Susi	PR	Belum Kawin	bandung 1993-12-08	Islam	Tidak Sekolah	Belum / Tidak bekerja	Pindah
7	Susana	LK	Belum Kawin	bandung 1998-12-10	Islam	Diploma III	PNS	Meninggal
8	Bayi	LK	Belum Kawin	bandung 2014-12-06	Islam	Tidak Sekolah	Belum / Tidak bekerja	Hidup
9	Bagus	LK	Belum Kawin	Bandung 1980-12-27	Islam	Strata I	PNS	Meninggal
10	Sulaeman Nugraha	LK	Kawin	Bandung 1950-01-13	Islam	null	Pensiunan	Hidup

Gambar 4.58 Laporan Data Penduduk



Gambar 4.59 File Excel hasil Export dari database

4.5 Pengujian

Proses pengujian berfokus pada logika internal *software*, memastikan bahwa semua pernyataan sudah diuji. Dan pada eksternal fungsional mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa masukan yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil

yang dibutuhkan.

4.5.1 Rencana Pengujian

Pengujian dilakukan melalui menggunakan data uji yang dimasukkan oleh user. Berikut adalah rencana pengujian yang dilakukan:

Tabel 4.16 Rencana Pengujian

Item Uji	Deskripsi	*Form*	Jenis Pengujian
Login	Melakukan *login* ke dalam sistem	FrameLogin.java	*Black Box*
Olah Data Penduduk	Melakukan pengolahan data	penduduk.java	*Black Box*
		keluarga.java	
		laporan.java	
Proses Kependudukan	memproses Data Kependudukan	Kependudukan.java	*Black Box*

4.5.2 Kasus dan Hasil Pengujian

Pada bagian ini, item-item yang telah ditentukan di atas akan diujikan. Kasus-kasus dalam pengujian akan dipaparkan berikut dengan hasil pengujian tersebut.

1. Pengujian Validalitas *Login*

   Pada pengujian *login*, data antara *username* dan *password* harus sesuai, untuk pengujian diasumsikan data *username* “admin” dan *password* inputan “rahasia” adalah benar, selain dari itu salah. Pada pengujian ini akan diperlihatkan pengujian pada tingkatan Validitas dari Black Box.

Tabel 4.17 Pengujian *Login*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Username: admin *Password*: rahasia	Menampilkan menu utama	Tampil menu Utama	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Username: salah *Password*: salah	Menampilkan pesan peringatan	Tampil pesan peringatan	[X] Diterima [ ] Ditolak

1. Pengujian Data Penduduk

Tabel 4.18 Pengujian Data Penduduk

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan lengkap	Menampilkan pesan data tersimpan dan data akan terlihat di table pada *form*.	Data tampil pada table	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan tidak	Menampilkan pesan	Tampil	[X] Diterima

lengkap	data tidak boleh kosong.	Pesan Peringatan	[ ] Ditolak
Masukan data Sama dengan yang telah dimasukan	Menampilkan pesan data telah ada.	Tampil Pesan Peringatan	[X] Diterima [ ] Ditolak

2. Pengujian Data Kartu Keluarga

Tabel 4.19 Pengujian Data Kartu Keluarga

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan lengkap	Menampilkan pesan data tersimpan dan data akan terlihat di table pada *form*.	Data tampil pada table	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan tidak lengkap	Menampilkan pesan data tidak boleh kosong.	Tampil Pesan Peringatan	[X] Diterima [ ] Ditolak
Masukan data Sama dengan yang	Menampilkan pesan data telah ada.	Tampil Pesan Peringatan	[X] Diterima [ ] Ditolak

telah dimasukan			

3. Pengujian Proses Kependudukan

Tabel 4.20 Pengujian Proses Kependudukan

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan lengkap	Menampilkan pesan data tersimpan dan data akan terlihat di table pada *form*.	Data tampil pada table	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan tidak lengkap	Menampilkan pesan data tidak boleh kosong.	Tampil Pesan Peringatan	[X] Diterima [ ] Ditolak

4. Pengujian laporan

Tabel 4.21 Pengujian Penjadwalan

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan

Masukan tanggal awal dan akhir penduduk pindah	Menampilkan laporan penduduk pindah per periode	Laporan perpindahan penduduk muncul	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak memasukan tanggal awal dan akhir	Laporan tidak muncul dan muncul pesan error	Laporan tidak muncul dan muncul pesan error	[X] Diterima [ ] Ditolak

4.5.3 Kesimpulan dan Hasil Pengujian

Pengujian yang telah dilakukan merupakan beberapa proses yang terdapat dalam aplikasi. Berdasarkan hasil pengujian diatas dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan sintaks dan secara fungsional mengeluarkan hasil yang diharapkan. Namun demikian pengujian tersebut dapat dikatakan belum sempurna, dikarenakan hanya dilakukan pada satu sisi pengujian. Dari semua yang telah dilakukan dalam pengujian ini diharapkan dapat mewakili pengujian fungsi yang lainnya. Secara garis besar dari pengujian dan hasil implementasi perangkat lunak yang dibangun sudah sesuai dengan spesifikasi kebutuhan dan hasil perancangan perangkat lunak