

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Alat dan Bahan**

Penelitian ini membutuhkan alat untuk mendukung berjalannya perancangan dan implementasi sistem. Alat yang digunakan berupa perangkat keras dan perangkat lunak

##### **3.1.1 Perangkat Keras**

Spesifikasi laptop yang digunakan pengembang adalah sebagai berikut:

- *Processor Intel(R) Core(TM)i5-3317U CPU @1.70 GHz*
- *Ram 8 GB*
- *Hardisk 500 GB*
- Sistem Operasi Windows 8.1 Enterprise

##### **3.1.2 Perangkat Lunak**

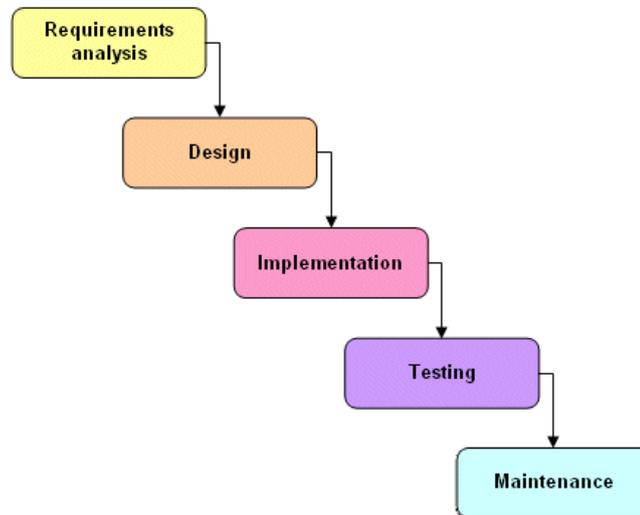
Daftar perangkat lunak yang digunakan pengembang adalah sebagai berikut:

- Visual Studio 2015 sebagai software pengembangan aplikasi.
- SQL Server 2014 sebagai database.
- Balsamiq Mockups 3 untuk merancang user interface.
- Microsoft Visio 2013 untuk merancang gambaran sistem
- Google Chrome untuk browser internet.

#### **3.2 Langkah Penelitian**

Dalam penelitian ini, pengembangan aplikasi menggunakan metode pengembangan *software* model *waterfall*. Model ini merupakan turunan dari metode *SDLC (Software Developing Life Cycle)* dan merupakan model klasik yang mengusung pengembangan perangkat lunak yang sistematis/sekuensial. Inti dari model *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan

atau secara linear. Metode *waterfall* ini memiliki 5 tahapan, yaitu *requirements analysis*, *system design*, *implementation*, *testing*, dan *maintenance*.



**Gambar 3.1** Pengembangan Sistem dengan *Waterfall*

### 3.2.1 Analisa Kebutuhan

Merupakan tahap awal dalam model *waterfall*. Permodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk *software*. Pada tahap ini seluruh kebutuhan *software* harus bisa didapatkan, termasuk didalamnya kegunaan *software* yang diharapkan pengguna dan batasan *software*. Informasi ini biasanya dapat diperoleh melalui wawancara, survey atau diskusi. Informasi tersebut dianalisis untuk mendapatkan dokumentasi kebutuhan pengguna untuk digunakan pada tahap selanjutnya.

### 3.2.2 Perancangan Sistem

Pada tahap ini kebutuhan-kebutuhan yang didapat pada tahap analisa kebutuhan diubah menjadi representasi ke dalam bentuk "*blueprint*" *software*. Gunanya untuk menspesifikasikan kebutuhan hardware dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan. Dengan bantuan *software* Microsoft Visio 2013, perancangan sistem dilakukan dengan membangun *use case diagram*, *class diagram*, dan *activity diagram*. Setelah itu, dapat dilanjutkan dengan perancangan antar muka dengan menggunakan *software* Balsamiq.

### 3.2.3 *Implementation*

Tahap ini merupakan implementasi dari tahap desain yang secara teknis dikerjakan oleh *programmer* menggunakan bahasa ASP.NET sebagai *controller* dan HTML sebagai *View*. Pengembangan aplikasi ini dilakukan dari awal hingga aplikasi siap dijalankan.

### 3.2.4 *Testing*

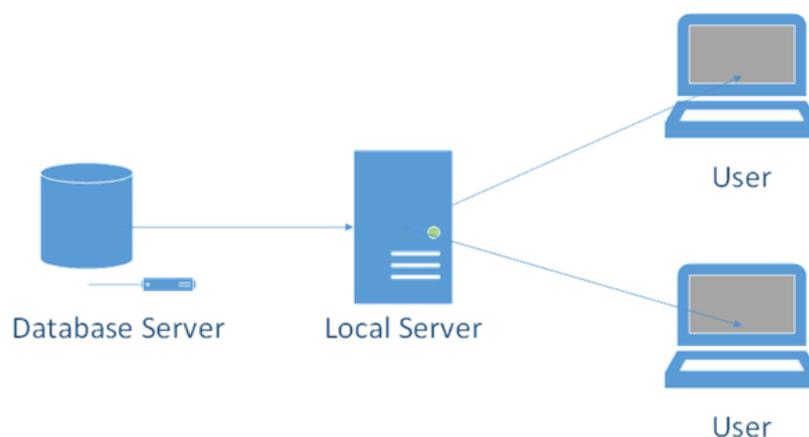
Tahap ini *software* hasil implementasi harus diuji coba. Tujuannya adalah untuk meminimalisir terjadi *error* pada *software* dan hasilnya sesuai dengan kebutuhan.

### 3.2.5 *Maintenance*

Proses *maintenance* merupakan tahap pemeliharaan *software*. *Software* yang dibuat harus memiliki tahap pemeliharaan atau pembaharuan, karena *software* yang dibuat memungkinkan untuk penambahan fitur-fitur baru dan juga perbaikan apabila terdapat error pada sistem yang dikembangkan. Pengembangan diperlukan ketika adanya perubahan dari eksternal seperti ketika ada pergantian sistem operasi atau perangkat lainnya.

## 3.3 Analisis Penelitian

### 3.3.1 Arsitektur



**Gambar 3.2** Arsitektur Riil

*Database server* yang digunakan adalah SQL Server 2014 Management Studio yang kemudian diolah melalui IIS *server local*. Aplikasi yang digunakan adalah *Web Application* dengan metode MVC (*model, view, controller*) ASP.NET. *User* dapat mengelola data, baik menyimpan, menghapus maupun mengedit data pada aplikasi dan disimpan kedalam *database*, melalui *interface* pada *web application* yang ditampilkan oleh *browser*.

### **3.3.2 Analisis Pengguna**

Aplikasi yang dibuat hanya digunakan administrator yang ditunjuk ketua takmir. Administrator dapat mengakses apapun pada aplikasi, seperti menambah, menghapus, dan mengubah data jamaah, takmir, transaksi arus kas, kegiatan dan inventaris.

### **3.3.3 Analisis Kebutuhan**

Analisis kebutuhan didapat dari studi *literatur* dan diskusikan dengan dosen pembimbing. Berikut ini merupakan analisis kebutuhan pada aplikasi sistem informasi masjid:

- a) Halaman untuk *Login*
- b) Halaman pengaturan tabel untuk pengisian jamaah, takmir, kegiatan, transaksi arus kas, inventaris, dan jadwal khotib.
- c) Halaman laporan dari data jamaah, kegiatan dan arus kas.

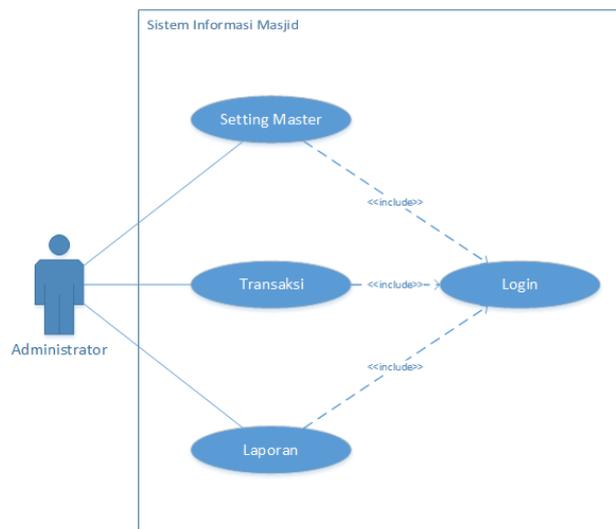
## **3.4 Rancangan Sistem dan Basis Data**

Metode perancangan yang digunakan pada sistem adalah *United Markup Language* (UML), dengan model UML yang dipakai dalam pengembangan aplikasi yaitu model *Use Case Diagram*, *Class Diagram* dan *Activity Diagram*. Sedangkan metode perancangan database menggunakan Diagram ER.

### 3.4.1 Use Case Diagram Sistem Informasi Masjid

*Use Case Diagram* merupakan teknik permodelan untuk menjelaskan setiap aktifitas yang mungkin terjadi. Aktifitas ini menggunakan sudut pandang dari aktor sebagai administrator.

Gambaran *use case diagram* yang digunakan dalam pada aplikasi dapat dilihat pada gambar 3.3.



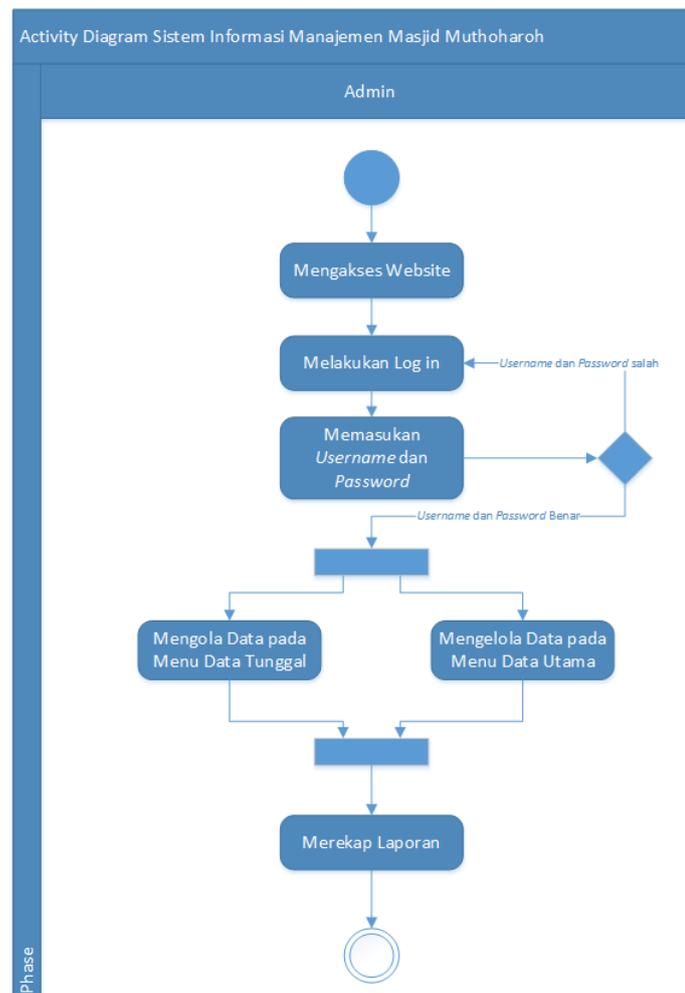
**Gambar 3.3** Use Case Diagram Sistem Informasi Masjid

Berikut penjelasan tentang *use case* diagram pada Gambar 3.4:

1. Di dalam sistem ini mempunyai satu aktor, yaitu administrator.
2. Administrator dapat mengakses semua fitur-fitur yang ada pada aplikasi.
3. Untuk melakukan semua aktivitas (*input*, lihat, ubah, hapus dan cetak) pada aplikasi ini maka kedua aktor harus melakukan proses *Login* terlebih dahulu.

### 3.4.2 Activity Diagram Sistem Informasi Masjid

Berdasarkan *use case* yang dibuat, diperoleh activity diagram berdasarkan actor yang terlibat dalam *use case* diagram tersebut. Gambaran *activity diagram* yang digunakan dalam pada aplikasi dapat dilihat pada gambar 3.4.



**Gambar 3.4** Activity Diagram Sistem Informasi Masjid

Berikut penjelasan tentang *activity diagram* pada Gambar 3.4:

1. Administrator mengakses *website* pada browser.
2. Administrator diharuskan untuk melakukan login terlebih dahulu.
3. Jika *username* dan *password* benar maka akan masuk ke halaman *Home website*, jika salah maka sistem akan meminta untuk memasukkan *username* dan *password* kembali.
4. Setelah login berhasil, administrator dapat mengelola data yang pada *website*
5. Setelah mengelola data, administrator dapat merekap laporan

### 3.4.3 Class Diagram Sistem Informasi Masjid



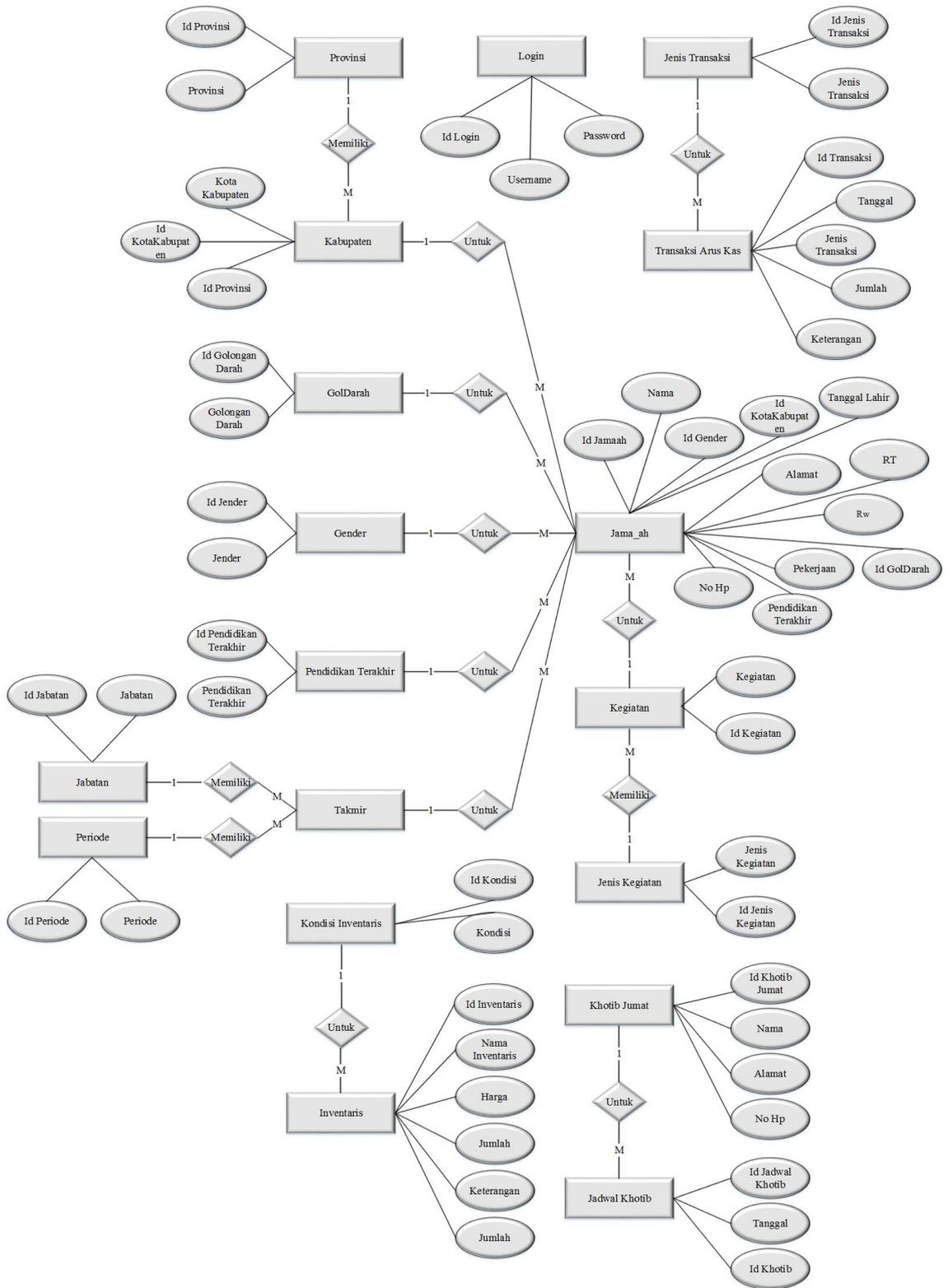
Gambar 3.5 Class Diagram Sistem Informasi Masjid

Berikut penjelasan tentang *class diagram* pada Gambar 3.5:

1. Seluruh kelas yang ada di dalam sistem ini terinisialisasi menjadi model tabel yang didasarkan pada database yang dibangun.
2. Kelas *Gender*, *Provinsi*, *GolDarah*, *PendidikanTerakhir*, *KhotibJumat*, *KondisiInventaris*, *JenisKegiatan*, *JenisTransaksi*, *Jabatan*, dan *Periode* merupakan kelas yang berdiri sendiri dan diasosiasi oleh kelas lain.
3. Kelas *KotaKabupaten*, *Jamaah*, *Takmir*, *TransaksiArusKas*, *Inventaris*, *Kegiatan* dan *Jadwal Khotib Jum'at* merupakan kelas yang berdiri karena berasosiasi dengan kelas lain.
4. Kelas *JamaahController* merupakan kelas untuk proses mengolah data Kelas *Jamaah*.
5. Kelas *TakmirController* merupakan kelas untuk proses mengolah data Kelas *Takmir*.
6. Kelas *KegiatanController* merupakan kelas untuk proses mengolah data Kelas *Kegiatan*.
7. Kelas *InventarisController* merupakan kelas untuk proses mengolah data Kelas *Inventaris*.
8. Kelas *TransaksiArusKasController* merupakan kelas untuk proses mengolah data Kelas *TransaksiArusKas*.
9. Kelas *JadwalKhotibController* merupakan kelas untuk proses mengolah data Kelas *JadwalKhotib*.

#### **3.4.4 ER Diagram Sistem Informasi Masjid**

Tujuan ER Diagram dalam pengembangan aplikasi adalah dapat memberikan gambaran umum tentang sistem yang dikembangkan sehingga memudahkan dalam merancang *database*. Gambaran *ER diagram* yang digunakan dalam pada aplikasi dapat dilihat pada gambar 3.6.



Gambar 3.6 ER Diagram Sistem Informasi Masjid

Berikut penjelasan tentang *ER diagram* pada Gambar 3.5:

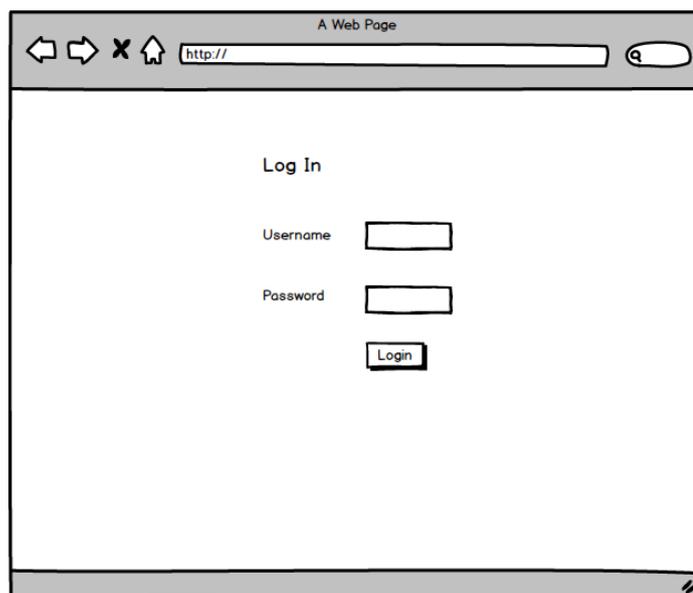
1. Terdapat 18 entitas pada 5 proses yang berbeda dan tiap entitas memiliki atribut dan relasi.
2. Entitas *log in* tidak memiliki relasi dengan entitas lainnya.
3. Entitas *Jamaah* memiliki relasi *one to many* dengan entitas kabupaten, entitas gol darah, entitas *gender*, entitas pendidikan, entitas takmir dan entitas kegiatan.
4. Entitas kegiatan memiliki relasi *one to many* dengan entitas jenis kegiatan.
5. Entitas kabupaten memiliki relasi *one to many* dengan entitas provinsi.
6. Entitas inventaris memiliki relasi *one to many* dengan entitas kondisi inventaris.
7. Entitas transaksi arus kas memiliki relasi *one to many* dengan entitas jenis transaksi.
8. Entitas jadwal khotib memiliki relasi *one to many* dengan entitas khotib jum'at.

### **3.5 Rancangan Antarmuka**

Rancangan antarmuka atau tampilan yang akan dirancang memberikan gambaran untuk setiap bagian dalam *website*. Rancangan antarmuka ini menjelaskan keterkaitan setiap halaman menu dan penjelasan cara kerja dari setiap menu dari *website*. Rancangan ini menjadi dasar untuk membuat tampilan pada aplikasi yang akan dibuat dan menjelaskan keterkaitan setiap halaman menu dan penjelasan cara kerja dari setiap menu.

#### **3.5.1 Rancangan Antarmuka Halaman Awal dan Login**

Halaman awal merupakan tampilan awal saat aplikasi dijalankan. Namun, sebelum masuk ke halaman awal, diharuskan untuk melakukan login terlebih dahulu. Rancangan antarmuka halaman login dapat dilihat pada Gambar 3.7, dan rancangan antarmuka halaman awal dapat dilihat pada Gambar 3.8.



A Web Page

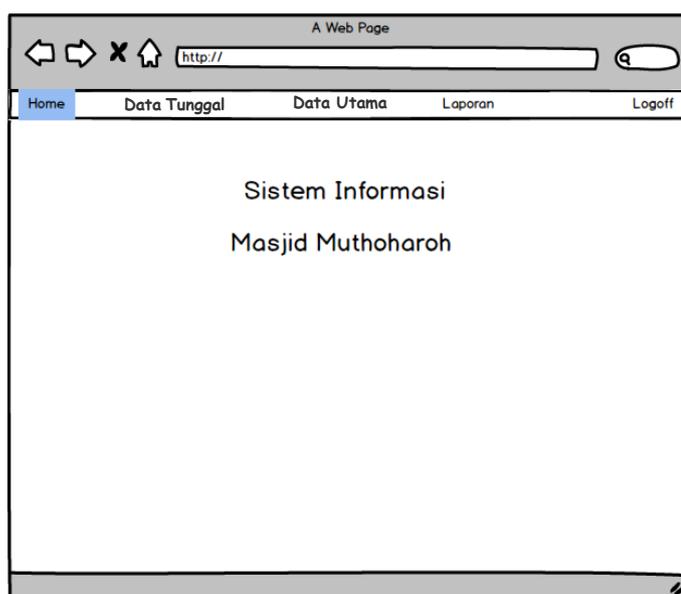
http://

Log In

Username

Password

**Gambar 3.7** Rancangan Antarmuka Halaman Login



A Web Page

http://

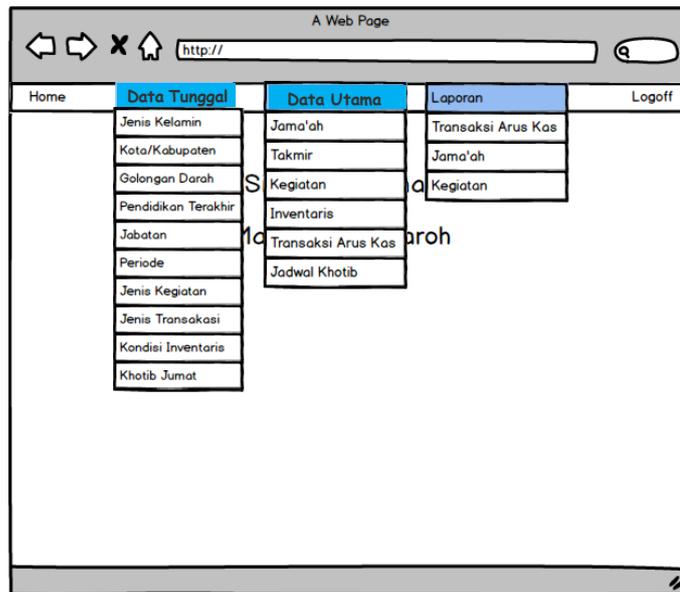
Home Data Tunggal Data Utama Laporan Logoff

Sistem Informasi  
Masjid Muthoharoh

**Gambar 3.8** Rancangan Antarmuka Halaman Awal.

### 3.5.2 Rancangan Antarmuka Menu Navigasi

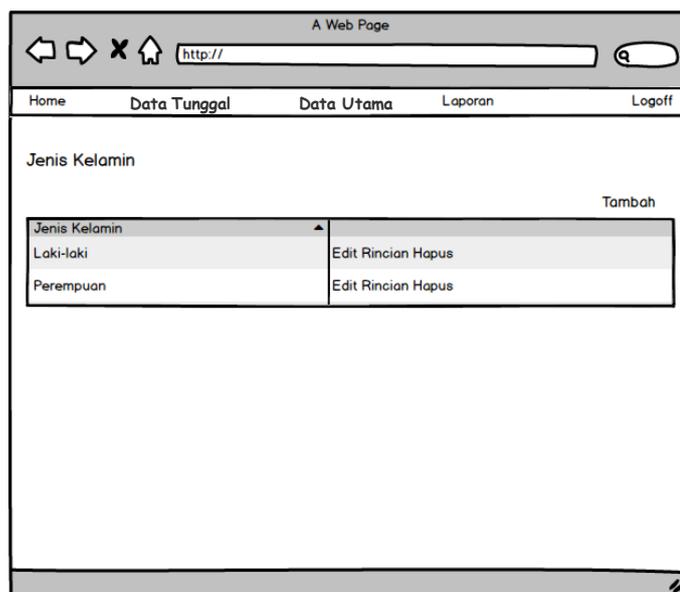
Menu navigasi mempermudah dalam mengakses semua halaman yang ada pada website. Rancangan antarmuka menu navigasi dapat dilihat pada Gambar 3.9.



**Gambar 3.9** Rancangan Antarmuka Menu Navigasi.

### 3.5.3 Rancangan Antarmuka Halaman Data Tunggal

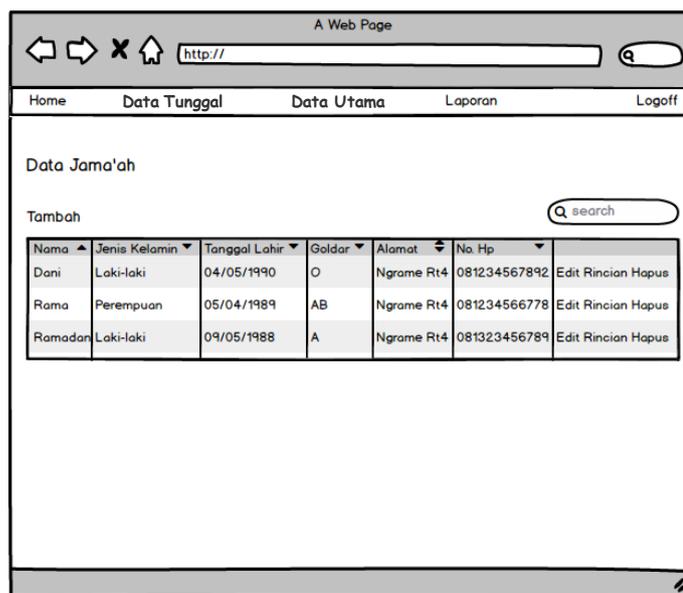
Halaman data tunggal adalah halaman untuk data yang tidak memiliki asosiasi atau relasi dengan data lain. Pada menu navigasi, semua data tunggal terdapat pada menu Setting Master. Rancangan antarmuka halaman data tunggal dapat dilihat pada Gambar 3.10.



**Gambar 3.10** Rancangan Antarmuka Halaman Data Tunggal.

### 3.5.4 Rancangan Antarmuka Halaman Transaksi

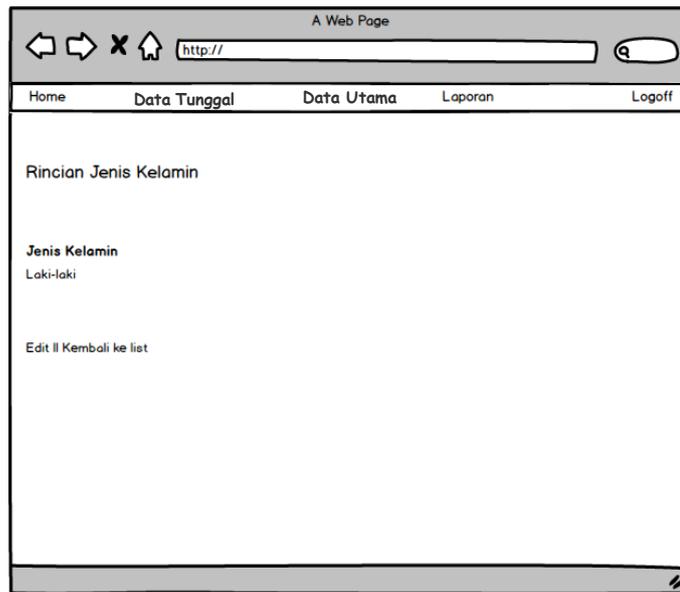
Halaman data transaksi adalah halaman untuk data yang memiliki asosiasi atau relasi dengan data lain. Pada menu navigasi, semua data transaksi terdapat pada menu Transaksi. Rancangan antarmuka halaman data transaksi dapat dilihat pada Gambar 3.11.



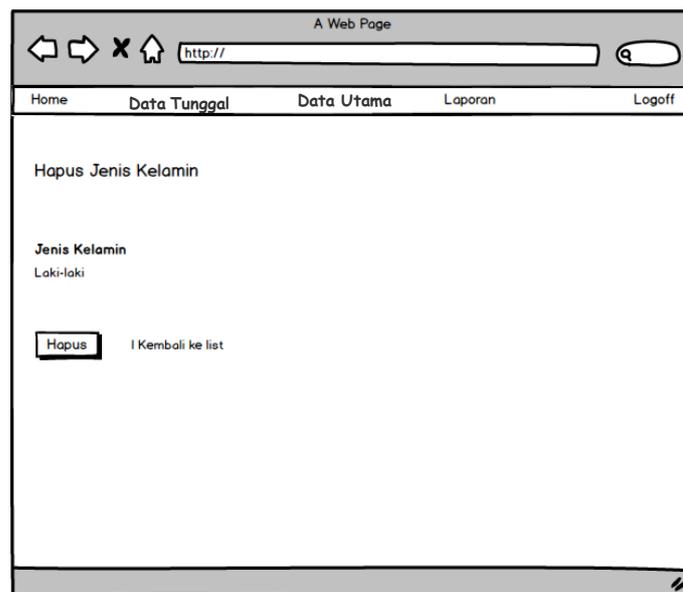
**Gambar 3.11** Rancangan Antarmuka Halaman Data Transaksi

### 3.5.5 Rancangan Antarmuka Halaman Rincian dan Halaman Hapus

Rancangan antarmuka halaman rincian merupakan rancangan untuk menampilkan rincian data yang ada pada data grid. Rancangan antarmuka halaman hapus merupakan rancangan untuk menghapus data yang ada di data grid. Untuk kedua halaman ini diterapkan pada menu yang ada difitur *setting master* dan transaksi. Gambaran perancangan antarmuka untuk halaman rincian dapat dilihat pada gambar 3.12 dan untuk gambaran perancangan antarmuka untuk halaman hapus dapat dilihat pada gambar 3.13.



**Gambar 3.12** Rancangan Antarmuka Halaman Rincian



**Gambar 3.13** Rancangan Antarmuka Halaman Hapus