

BAB III

METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian dilaksanakan di Panti Asuhan Ash Shiddiqiyah Kulon Progo. Sistem Informasi ini dilaksanakan dari bulan Mei 2017 – Agustus 2017.

3.2 Alat dan Bahan

Penelitian ini membutuhkan alat-alat untuk mendukung berjalannya perancangan dan implementasi sistem informasi. Alat yang digunakan berupa *hardware* dan *software*.

3.2.1 Software

- a. Visual Studio 2013 sebagai software pengembangan aplikasi.
- b. SQL Server 2014 sebagai database.
- c. Balsamiq Mockups 3 untuk merancang user interface.
- d. Microsoft Visio 2013 untuk merancang gambaran sistem

3.2.2 Hardware

- a. Personal Computer (PC) atau laptop.
 - 32/64 bit Architecture Processor.
 - 4GB Random Access Memory (RAM).
- b. Printer Untuk mencetak laporan

3.3 Metode Pengumpulan Data

Jenis data yang dikumpulkan dalam penelitian ini adalah data primer dan data sekunder, sebagai berikut:

3.3.1 Data Primer

Data primer merupakan data yang berasal dari individu, kelompok, panel atau data terselubung. Untuk mengumpulkan data primer perlu dilakukan observasi dan wawancara.

Tujuan dilakukannya observasi menurut Patton (1990) adalah mendeskripsikan *setting* yang diamati, tempat kegiatan orang-orang yang berpartisipasi didalamnya dan makna apa yang diamati menurut prespektif pengamat. Pada penelitian ini peneliti mendapatkan data meliputi sistem manajemen panti asuhan dengan mengamati objek yang akan diteliti secara langsung di tempat penelitian yaitu Panti Asuhan Ash Shiddiqiyah di Kulon Progo.

Menurut Moleong (1998) definisi dari wawancara adalah percakapan dengan maksud tertentu, percakapan dilakukan antara peneliti (mengajukan pertanyaan) dan yang di wawancarai memberikan jawaban atas pertanyaan tersebut. Proses wawancara dilakukan di Panti Asuhan Ash Shiddiqiyah yang bertujuan untuk mengetahui sistem yang dibutuhkan oleh pihak panti asuhan dari segi teknologi yang digunakan serta pemanfaatannya dengan cara melakukan dialog secara langsung dengan Bapak Nanang selaku perwakilan dari pengurus panti asuhan dan orang yang dianggap mampu memberikan informasi yang dibutuhkan.

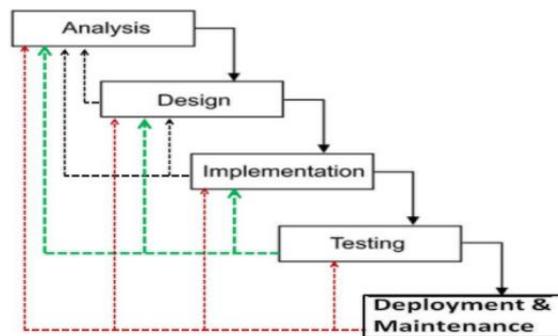
3.3.2 Data Sekunder

Data sekunder adalah data yang diperoleh melalui perantara, biasanya berupa catatan atau laporan yang telah tersusun dalam arsip yang dipublikasikan dan yang tidak dipublikasikan. Dalam penelitian ini untuk mendapatkan data yang diperlukan, data sekunder bersumber dari tinjauan pustaka dan landasan teori melalui berbagai jurnal, artikel, laporan dan referensi yang diambil dari perpustakaan maupun internet.

3.4 Langkah Penelitian

Untuk pengembangan sistem penelitian ini menggunakan model SDLC (*Software Development Life Cycle*). Model SDLC adalah proses dalam membangun sebuah Sistem Informasi yang melalui beberapa fase, yang terdiri dari *Planning, Analysis, Design, Implementation, testing* dan *maintenance*.

Model SDLC yang dipakai dalam penelitian ini adalah model *waterfall*. Model *waterfall* adalah sebuah metode pengembangan aplikasi dengan pendekatan sekuensial. Berikut adalah tahapan pembuatan aplikasi sesuai dengan model SDLC *Waterfall*.



Gambar 3. 1 Pengembangan Sistem dengan *Waterfall*

3.4.1 System / Information Engineering and Modeling

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 *software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware, database*, dan sebagainya.

3.4.2 Software Requirements Analysis

Proses ini lebih difokuskan pada pencarian kebutuhan *software*. *Software Engineer* harus mengerti tentang domain informasi dari *software* agar mengetahui sifat dari program yang akan dibuat. Misalnya fungsi yang dibutuhkan, *user interface*, dan sebagainya.

3.4.3 System Design

Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk “*blueprint*” *software* sebelum coding dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*. Setelah mendapatkan dokumentasi dari hasil analisis, maka kebutuhan-kebutuhan fungsi *software* harus diubah menjadi sebuah bentuk “*blueprint*” *software*. Sehingga hasil desain akan digunakan untuk membangun sebuah aplikasi.

3.4.4 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.

3.4.5 Testing / Verification

Tahap ini *software* hasil produksi harus diuji coba, termasuk semua fungsi-fungsinya. Tujuan pengujian ini adalah untuk meminimalisir kesalahan agar *software* bebas dari *error* dan hasilnya harus sesuai dengan kebutuhan.

3.4.6 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.

3.5 Analisis Penelitian

3.5.1 Arsitektur

SQL Server 2014 Management Studio adalah *Database server* yang digunakan pada aplikasi, yang kemudian akan diolah melewati *IIS server local*. Dan aplikasi yang digunakan adalah *Web Application* dengan metode MVC (*model, view, controller*) *ASP.NET* sebagai *controller*, dan *HTML* sebagai *view/user interface*. Pengguna dapat mengelola data, baik menyimpan, menghapus maupun mengedit data pada aplikasi dan disimpan kedalam *database*, melalui *interface* pada *web application* yang ditampilkan oleh *browser*. Dan pada saat pengguna mengakses aplikasi, *local server* memuat *interface* dan melakukan pengambilan data yang di perlukan dari *database*.

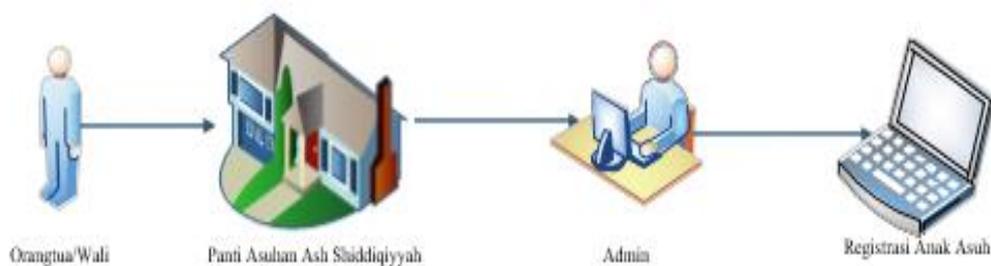
3.5.2 Analisis Pengguna

Aplikasi yang dibuat dapat diakses oleh admin, yakni:

Admin memiliki hak akses atau dapat mengakses apapun pada aplikasi, seperti mengelola (menambah, menghapus dan mengubah) data pada menu *setting master* seperti alamat, pendidikan dan sebagainya, dan pada menu *transaksi* seperti data anak asuh, data pengelola/pengurus, data doantur dan keuangan (pemasukan dan pengeluaran), serta menerbitkan/mencetak laporan untuk rekap data anak asuh, data pengelola dan keuangan.

3.5.3 Analisis Pendaftaran Anak Asuh

Proses pendaftaran anak asuh di Panti Asuhan Ash Shiddiqiyah.



Gambar 3. 2 Alur Pendaftaran anak asuh

Penjelasan tentang alur pendaftaran anak asuh pada gambar 3.2, sebagai berikut:

1. Calon anak asuh datang bersama orangtua/wali datang ke Panti Asuhan Ash Shiddiqiyah untuk melakukan registrasi sebagai anak asuh serta menyerahkan beberapa dokumen sebagai syarat registrasi.
2. Admin menginput data lengkap anak asuh baru ke dalam aplikasi.

3.5.4 Analisis Registrasi Donatur

Proses pendaftaran donatur di Panti Asuhan Ash Shiddiqiyah.



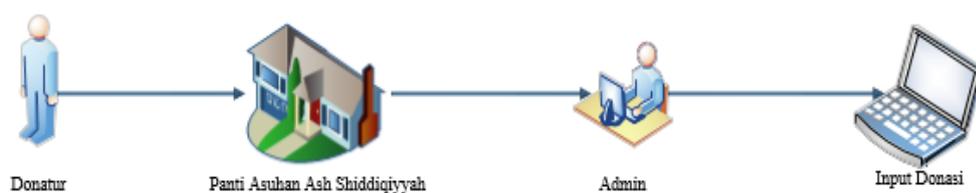
Gambar 3. 3 Alur pendaftaran donatur

Penjelasan tentang alur pendaftaran donatur pada gambar 3.3, sebagai berikut:

1. Calon donatur datang ke Panti Asuhan Ash Shiddiqiyah untuk melakukan registrasi sebagai donatur.
2. Admin menginput data lengkap donatur baru ke dalam aplikasi.

3.5.5 Analisis Donatur Melakukan Donasi

Proses donasi di Panti Asuhan Ash Shiddiqiyah.



Gambar 3. 4 Alur donatur dalam melakukan donasi

Penjelasan tentang alur pendaftaran donatur pada gambar 3.4, sebagai berikut:

1. Donatur yang sudah teregistrasi datang ke Panti Asuhan Ash Shiddiqiyah untuk melakukan donasi.
2. Admin membuat tanda terima dana.
3. Admin menginput data donasi dari donatur ke dalam aplikasi.

3.5.6 Analisis Kebutuhan

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

- a) Halaman *Login*
- b) Halaman pengaturan tabel untuk pengisian biodata anak asuh, pengelola dan donatur.
- c) Halaman donasi untuk mendata daftar donasi dari donatur.
- d) Laporan dari masing-masing data rekap keuangan, pengelola, anak asuh.

3.6 Rancangan Sistem dan Basis Data

3.6.1 Rancangan Sistem

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*.

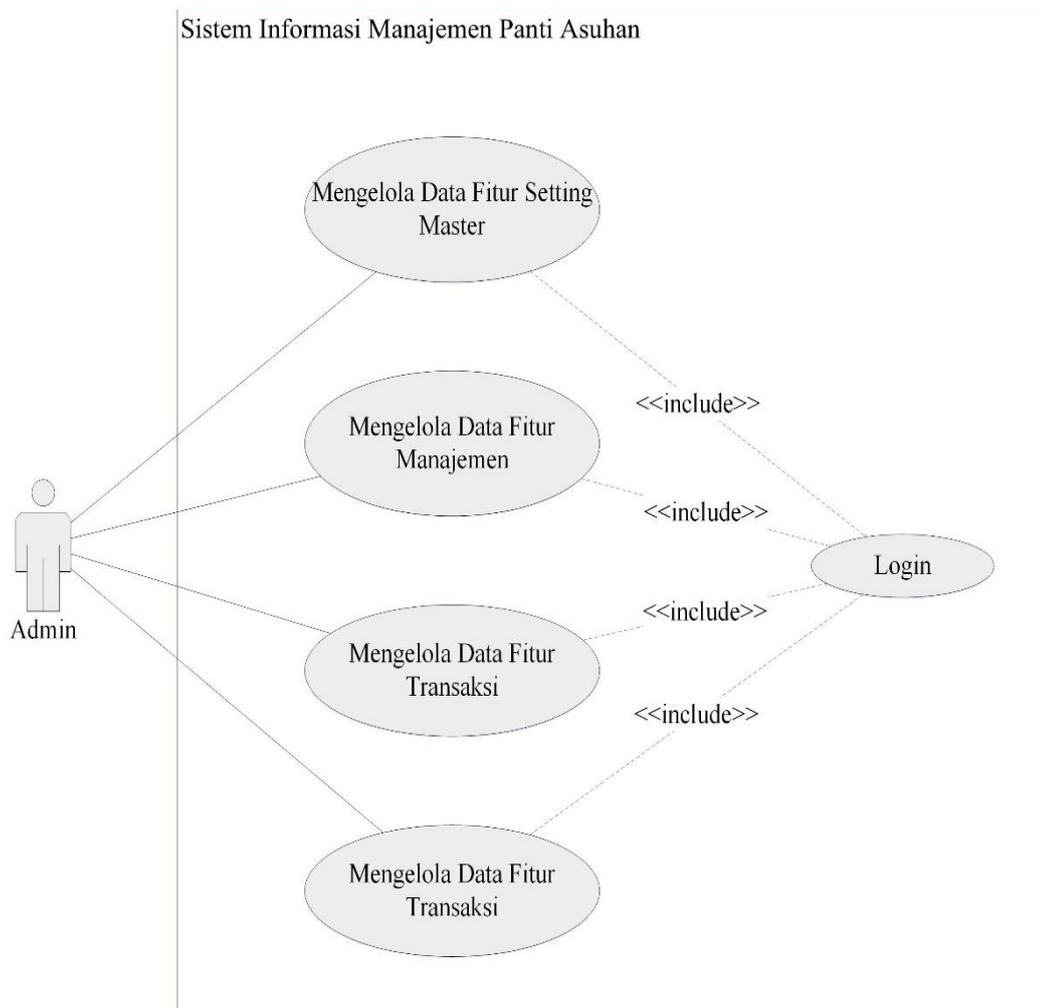
1) Use Case Diagram

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

Rancangan *Use Case diagram* yang digunakan dalam pembangunan aplikasi dapat dilihat pada gambar 3.5.

1. Di dalam sistem ini mempunyai satu aktor, yaitu admin.
2. Admin 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 Log In terlebih dahulu.



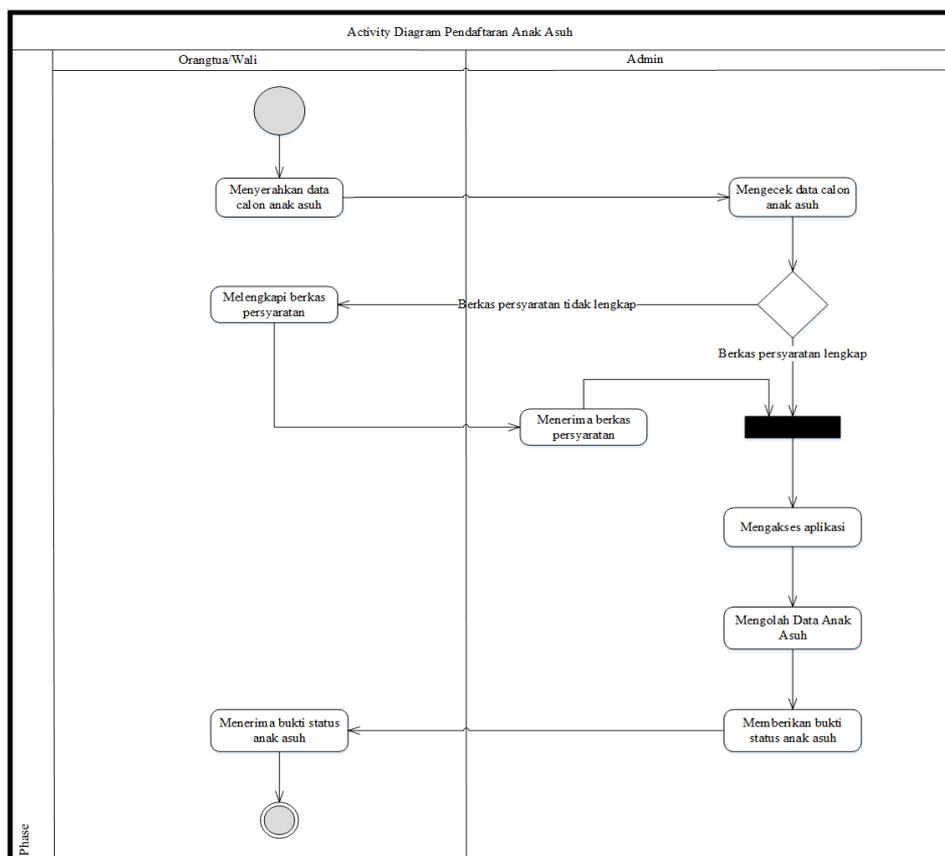
Gambar 3. 5 Use case diagram

2) Activity Diagram Pendaftaran Anak Asuh

Berikut penjelasan tentang gambar *Activity Diagram* pada gambar 3.6:

1. Terdapat alur dari kegiatan pendaftaran anak asuh, yaitu di mulai dari orangtua/wali mendaftarkan anak.
2. Admin memeriksa persyaratan pendaftaran.

3. Jika ada persyaratan yang kurang misalnya tidak di sertai akta kelahiran anak atau kartu keluarga maka orangtua/wali harus melengkapi persyaratan tersebut, dan jika orangtua/wali tersebut tidak bisa melengkapi maka akan ada kebijakan dari panti asuhan.
4. Admin menerima berkas persyaratan dan mengelola data anak asuh.
5. Admin memberikan bukti status anak asuh.
6. Orangtua/wali menerima bukti status anak asuh.



Gambar 3. 6 Activity Diagram Pendaftaran Anak Asuh

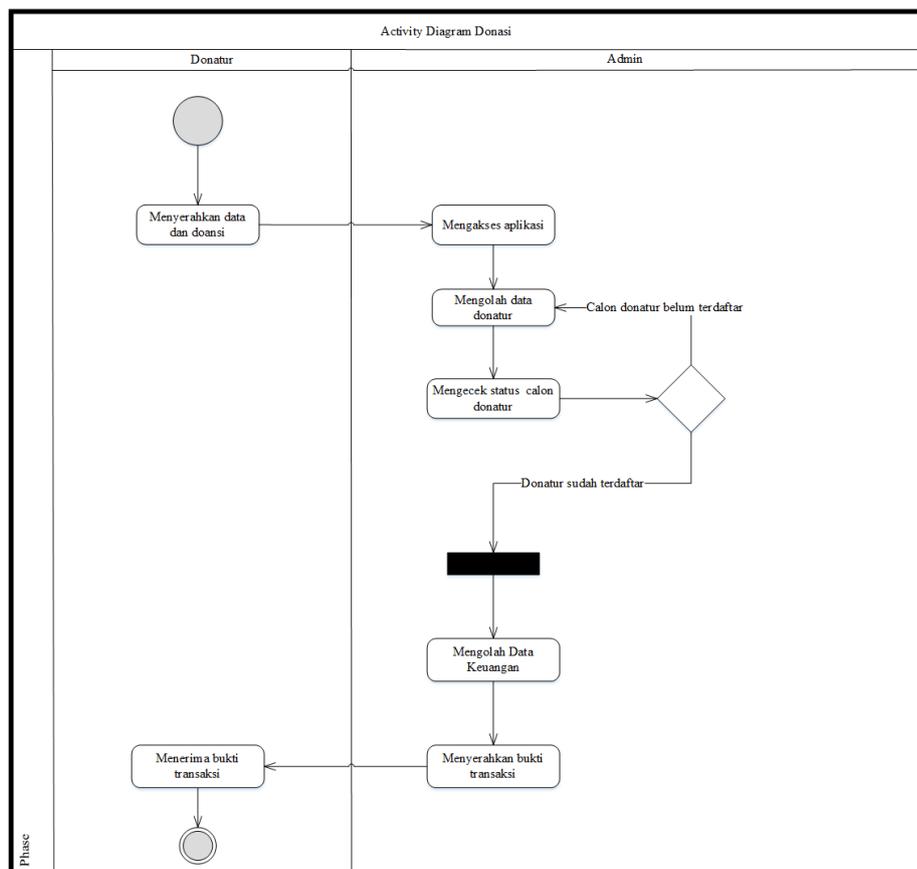
Berikut penjelasan tentang gambar Activity Diagram pada gambar 3.6:

1. Terdapat alur dari kegiatan pendaftaran anak asuh, yaitu di mulai dari orangtua/wali mendaftarkan anak.
2. Admin memeriksa persyaratan pendaftaran.
3. Jika ada persyaratan yang kurang misalnya tidak di sertai akta kelahiran anak atau kartu keluarga maka orangtua/wali harus melengkapi persyaratan

tersebut, dan jika orangtua/wali tersebut tidak bisa melengkapi maka akan ada kebijakan dari panti asuhan.

4. Admin menerima berkas persyaratan dan mengelola data anak asuh.
5. Admin memberikan bukti status anak asuh.
6. Orangtua/wali menerima bukti status anak asuh.

3) Activity Diagram Donasi



Gambar 3.7 Activity Diagram Donasi

Berikut penjelasan tentang gambar *Activity Diagram* pada gambar 3.7:

1. Terdapat alur dari kegiatan donasi, yaitu di mulai dari donatur memberikan data donatur dan donasi.
2. Admin memeriksa status donatur pada data donatur diaplikasi.

Berikut penjelasan untuk *ER Diagram* pada gambar 3.8:

1. Terdapat 22 entitas pada 6 proses yang berbeda dan masing-masing memiliki atribut dan relasi.
2. Relasi antar entitas dimiliki oleh semua entitas kecuali entitas login. Karena pada entitas login tidak terdapat relasi ke entitas lainnya,
3. Entitas anak asuh memiliki relasi dengan entitas desa, entitas orangtua/wali, entitas status anak, entitas status asal, entitas jenis kelamin dan entitas pendidikan.
4. Pada saat proses anak asuh.
 - Setiap anak asuh hanya bisa memiliki satu desa, satu orangtua/wali, satu status asal, satu jenis kelamin dan satu pendidikan.
 - Sedangkan desa, orangtua/wali, status anak, status asal, jenis kelamin dan pendidikan bisa dimiliki oleh banyak anak.
5. Entitas pengelola memiliki relasi dengan entitas jabatan, entitas status, jenis dan entitas pendidikan.
6. Pada saat proses pengelola.
 - Setiap pengelola hanya bisa memiliki satu jabatan, satu jenis dan satu pendidikan.
 - Untuk jabatan, jenis dan pendidikan bisa dimiliki oleh banyak pengelola.
7. Entitas donatur memiliki relasi dengan entitas status dan entitas desa
8. Pada saat proses donatur.
 - Setiap donatur hanya bisa memiliki satu status dan satu desa.
 - Untuk status dan desa bisa dimiliki oleh banyak donatur.
9. Entitas donasi memiliki relasi dengan entitas donatur, entitas jenis dan entitas bentuk.
10. Pada saat proses transaksi donasi.
 - Setiap transaksi donasi hanya bisa memiliki satu donatur, satu jenis dan satu bentuk.

- Sedangkan untuk donatur, jenis dan bentuk bisa di miliki oleh banyak transaksi donasi.
11. Entitas pengeluaran memiliki relasi dengan entitas donasi
 12. Pada saat proses transaksi pengeluaran.
 - Setiap transaksi pengeluaran hanya bisa memiliki satu donasi.
 - Sedangkan donasi bisa dimiliki oleh banyak transaksi pengeluaran.

5) Class Diagram

Berikut penjelasan untuk fungsi *class diagram* sistem pada gambar 3.9:

1. *Class* *asuh_statusasal*, berfungsi untuk memasukan dan menampilkan data status asal yang digunakan pada *class* *anak_asuh*.
2. *Class* *asuh_status*, berfungsi untuk memasukan dan menampilkan data status yang digunakan pada *class* *anak_asuh*.
3. *Class* *ortu_jenis*, berfungsi untuk memasukan dan menampilkan data ortu jenis yang digunakan pada *class* *ortu*.
4. *Class* *ortu*, berfungsi untuk memasukan dan menampilkan data ortu yang digunakan pada *class* *anak_asuh*.
5. *Class* *jenis_kelamin*, berfungsi untuk memasukan dan menampilkan data jenis kelamin yang digunakan pada *class* *anak_asuh*.
6. *Class* *provinsi*, berfungsi untuk memasukan dan menampilkan data provinsi yang digunakan pada *class* *kabupaten*.
7. *Class* *kabupaten*, berfungsi untuk memasukan dan menampilkan data kabupaten yang digunakan pada *class* *kecamatan* dan pada *anak_asuh*.
8. *Class* *kecamatan*, berfungsi untuk memasukan dan menampilkan data kecamatan yang digunakan pada *class* *desa*.
9. *Class* *desa*, berfungsi untuk memasukan dan menampilkan data desa yang digunakan pada *class* *anak_asuh*, *class* *ortu* dan *class* *donatur registrasi*.
10. *Class* *pendidikan*, berfungsi untuk memasukan dan menampilkan data pendidikan yang digunakan pada *class* *anak_asuh* dan *class* *pengelola data*.

11. *Class* *pengelola_jenis*, berfungsi untuk memasukan dan menampilkan data jenis untuk pengelola yang digunakan pada *class* *pengelola data*.
12. *Class* *pengelola_jabatan*, berfungsi untuk memasukan dan menampilkan data jabatan pengelola yang digunakan pada *class* *pengelola data*.
13. *Class* *pengelola_status*, berfungsi untuk memasukan dan menampilkan data status pengelola yang digunakan pada *class* *pengelola data*.
14. *Class* *donatur_status*, berfungsi untuk memasukan dan menampilkan data status donatur yang digunakan pada *class* *donatur_registrasi*.
15. *Class* *donasi_jenis*, berfungsi untuk memasukan dan menampilkan data jenis donasi yang digunakan pada *class* *donasi_transaksi*.
16. *Class* *donasi_bentuk*, berfungsi untuk memasukan dan menampilkan data bentuk donasi yang digunakan pada *class* *donasi_transaksi*.
17. *Class* *donatur_transaksi*, berfungsi untuk memasukan dan menampilkan data nama donatur yang digunakan pada *class* *donasi_pengeluaran*.

Berikut penjelasan untuk *class diagram* sistem pada gambar 3.9:

1. Kelas pada diagram digunakan untuk inialisasi ke dalam tabel model dan basis data.
2. *Class* *anak_asuh*
Memiliki *asosiasi* dengan *class* *ortu*, *class* *pendidikan* dan *class* *desa*, artinya setiap data *anak_asuh* dapat memilih satu orangtua, satu desa dan satu pendidikan untuk mengisi data.
Memiliki *composition* dengan *class* *asuh_statusasal*, *class* *anak_status* dan *class* *jenis_kelamin*, artinya *class* *anak_asuh* tidak akan berdiri jika *class-class* tersebut tidak ada.
3. *Class* *ortu*
Memiliki *composition* dengan *class* *ortu_jenis*, artinya *class* *ortu_jenis* ini merupakan bagian dari *class* *ortu*. *Class* *ortu* tidak akan berdiri jika *class* *jenis_ortu* tidak ada.
4. *Class* *desa*

Memiliki *composition* dengan *class* kecamatan. *Class* kecamatan memiliki *composition* dengan *class* kabupaten dan *class* kabupaten juga memiliki *composition* dengan *class* provinsi. Artinya *class* desa ini merupakan bagian dari *class* kecamatan. Yang mana *class* desa tidak akan berdiri jika *class* kecamatan, *class* kabupaten dan *class* provinsi tidak ada.

5. *Class* pengelola_data

Memiliki *composition* dengan *class* pengelola_jabatan, *class* pengelola_jenis, dan *class* pengelola_status artinya *class* pengelola_data tidak dapat berdiri sendiri.

Memiliki *asosiasi dengan class* pendidikan artinya setiap data pengelola dapat memilih satu pendidikan.

6. *Class* donatur_registrasi

Memiliki *asosiasi* dengan *class* desa, artinya setiap data donatur_registrasi dapat memilih satu desa untuk mengisi data.

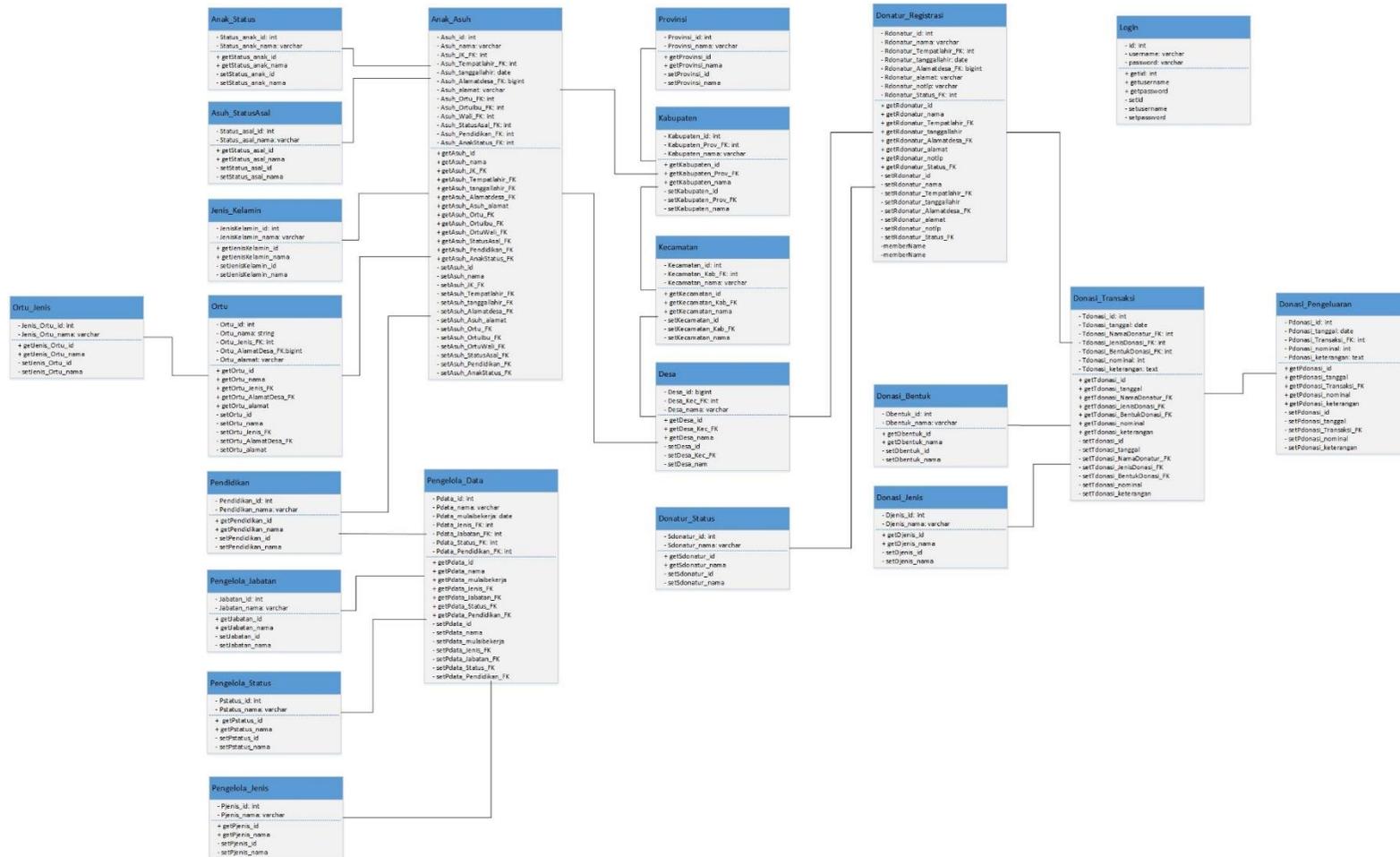
Memiliki *composition* dengan *class* donatur_status, artinya *class* donatur_registrasi tidak akan berdiri jika *class* donatur_status tersebut tidak ada.

7. *Class* donasi_transaksi

Memiliki *composition* dengan *class* donatur_registrasi, *class* donasi_jenis dan *class* donasi_bentuk, artinya *class* donatur_transaksi tidak akan berdiri jika *class-class* tersebut tidak ada.

8. *Class* donasi_pengeluaran

Memiliki *asosiasi* dengan *class* donasi_transaksi, artinya setiap data donatur_pengeluaran dapat memilih satu donasi_transaksi untuk mengisi data.



Gambar 3.9 Class Diagram

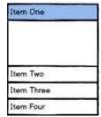
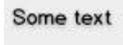
3.7 Rancangan Antarmuka

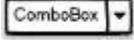
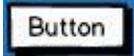
Rancangan antarmuka atau tampilan yang akan dirancang memberikan gambaran untuk setiap bagian dalam aplikasi. Rancangan ini menjadi dasar untuk membuat tampilan pada aplikasi yang akan dibuat dan menjelaskan keterkaitan setiap halaman menu dan penjelasan cara kerja dari setiap menu.

Mockup adalah program aplikasi replika atau struktur. Aplikasi *balsamiq mockup* digunakan dengan tujuan intruksional atau eksperimental. *Balsamiq mockup* merupakan program aplikasi yang digunakan dalam merancang antarmuka sebuah aplikasi. *Balsamiq mockup* dapat membantu penulis dalam merancang antarmuka aplikasi yang akan dibangun, karena *software* ini menyediakan *tools* yang dapat digunakan dalam merancang desain *prototyping* dan berfokus pada konten yang ingin dirancang dan fungsionalitas yang dibutuhkan oleh penulis.

Adapun *tools* yang digunakan dalam perancangan antarmuka pada semua rancangan aplikasi yang akan dibangun dapat dilihat pada tabel 3.1.

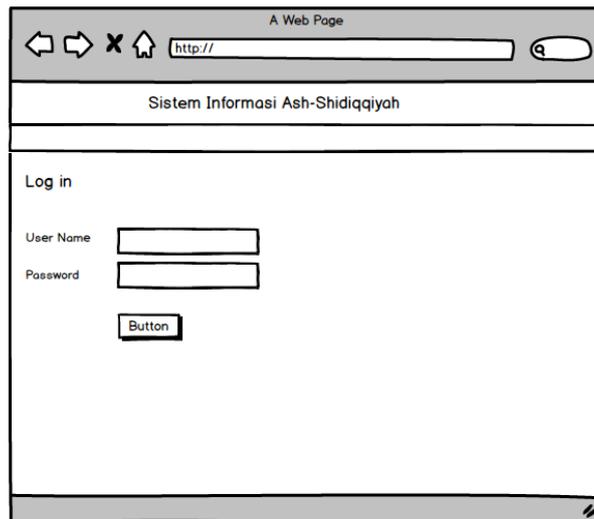
Tabel 3. 1 *Tools Balsamiq mockup* dan fungsinya

No.	<i>Tools</i>	Nama <i>Tools</i>	Fungsi
1.		Browser	Desain <i>prototyping</i> antarmuka yang ditampilkan oleh web application melalui browser.
2.		Menu Bar	Desain menu pada navbar
3.		Accordion	Desain <i>drop down</i> menu pada menu navbar
4.		Label	Digunakan untuk teks yang bersifat statis, label juga digunakan untuk memberi keterangan terhadap objek lain, ataupun judul pada form
5.		Text Input	untuk menerima input pemakai dalam bentuk satu baris teks dan menampilkan text.

No.	Tools	Nama Tools	Fungsi
6.		Combo Box	Untuk memilih satu dari sejumlah pilihan
7.		Date Chooser	Digunakan untuk menentukan waktu kejadian
8.		Data Grid	Digunakan untuk menampilkan data dalam bentuk grid.
9.		Button	Tombol yang akan melakukan suatu perintah atau kejadian pada saat komponen ini digunakan atau di klik
10.		Comment	Berfungsi untuk menambah catatan pada perancangan.

3.7.1 Rancangan Antarmuka Halaman Awal dan Login

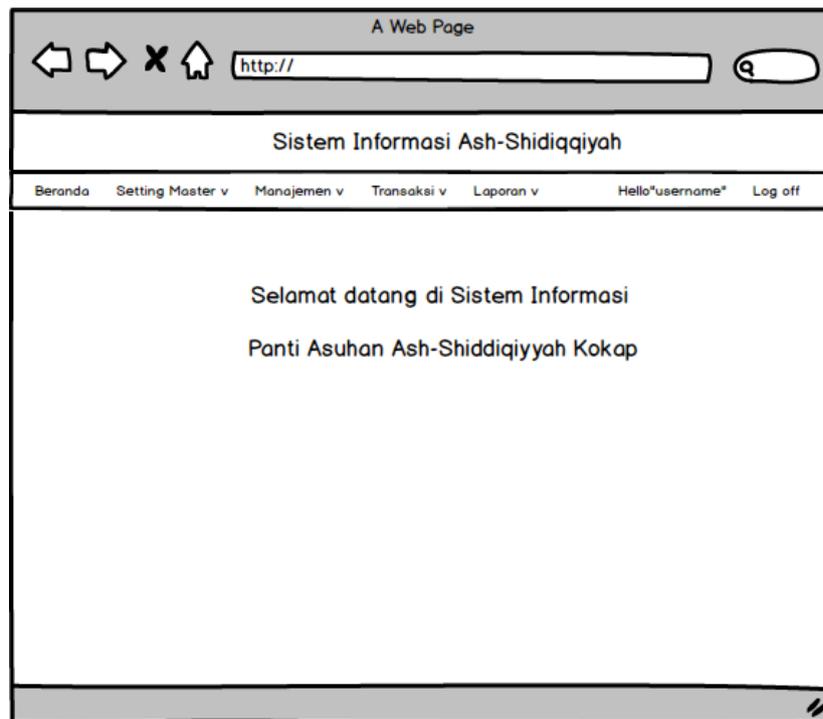
Untuk dapat mengakses fitur lain pengguna harus melakukan login terlebih dahulu. Berikut merupakan rancangan tampilan halaman log in dapat dilihat pada gambar 3.10.



The image shows a web browser window titled "A Web Page" with a URL bar containing "http://". Below the browser window, the page content is displayed. At the top, there is a header "Sistem Informasi Ash-Shidiqqiyah". Below the header, there is a "Log in" section. This section contains two input fields: "User Name" and "Password". Below these fields is a "Button" for submitting the login information.

Gambar 3. 10 Rancangan tampilan halaman *Log in*

Admin memiliki hak akses penuh. Rancangan antar muka admin ini terdiri dari fitur *setting master*, manajemen, transaksi dan laporan. Setelah pengguna *login* maka aplikasi akan menampilkan halaman utama. Gambaran rancangan dilihat pada gambar 3.11.



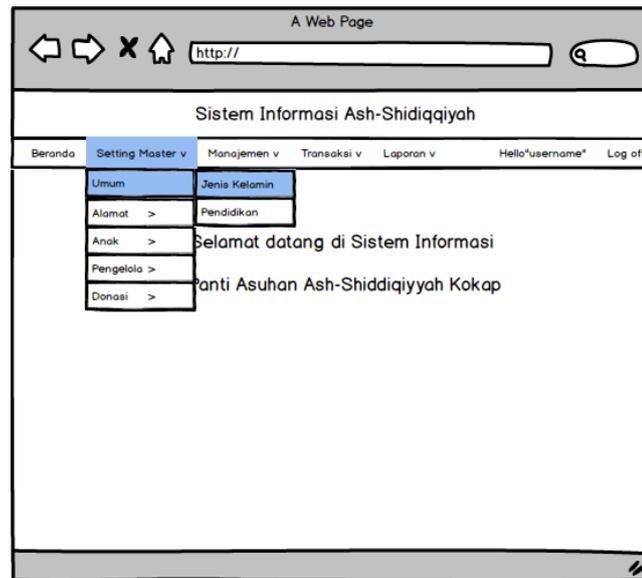
Gambar 3. 11 Rancangan tampilan halaman utama

3.7.2 Rancangan Antarmuka Fitur pada Sistem

Adapun fitur yang terdapat pada sistem sebagai berikut :

1. Fitur Manajemen

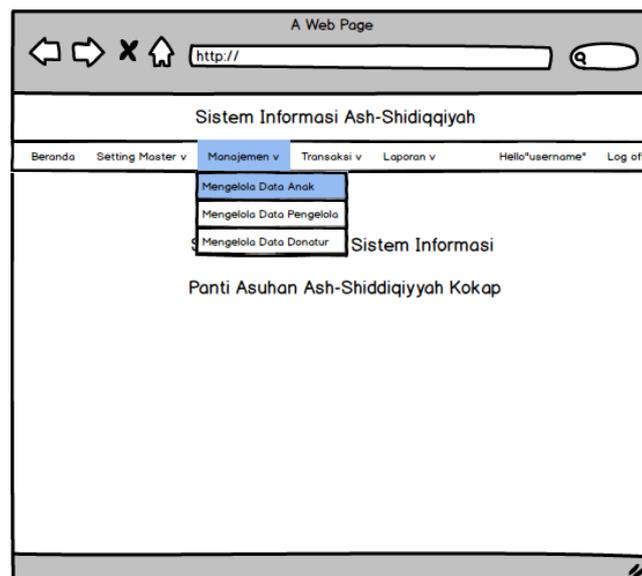
Fitur *setting master* merupakan data tunggal yang memiliki menu *dropdown* dan terdiri dari menu umum, menu alamat, menu anak, menu pengelola dan menu donasi. Pada gambar 3.12, merupakan rancangan menu *dropdown* pada fitur *setting master*.



Gambar 3. 12 Rancangan tampilan menu *dropdown* pada fitur *setting master*

2. Fitur Manajemen

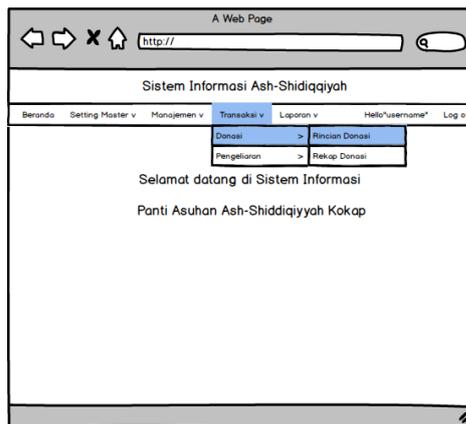
Fitur manajemen merupakan menu untuk mengelola data yang terdiri dari menu mengelola data anak, menu mengelola data pengelola dan menu mengelola data donatur. Pada gambar 3.13, merupakan rancangan menu *dropdown* pada fitur manajemen.



Gambar 3. 13 Rancangan tampilan menu *dropdown* pada fitur manajemen

3. Fitur Transaksi

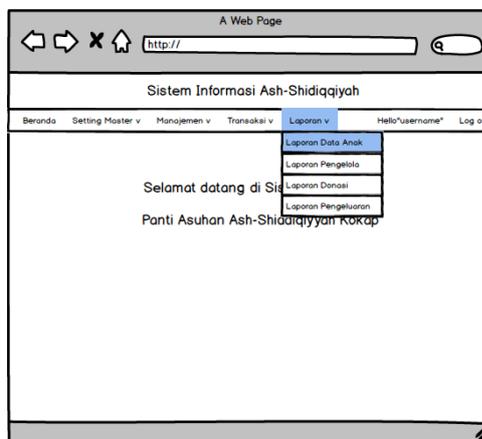
Fitur transaksi merupakan fitur untuk mengelola keuangan yang terdiri dari menu donasi dan menu pengeluaran. Pada gambar 3.14, merupakan rancangan menu dropdown pada fitur transaksi.



Gambar 3. 14 Rancangan tampilan menu *dropdown* pada fitur transaksi

4. Fitur Laporan

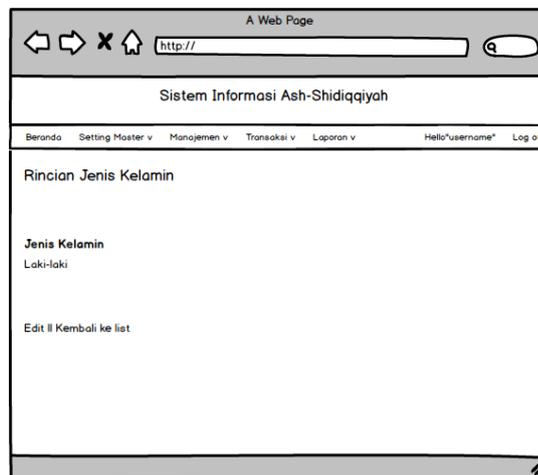
Fitur laporan merupakan menu untuk mencetak data yang terdiri dari menu laporan data anak, menu laporan data pengelola, menu laporan data pengeluaran dan menu laporan data donasi. Pada gambar 3.15, merupakan rancangan menu *dropdown* pada fitur manajemen.



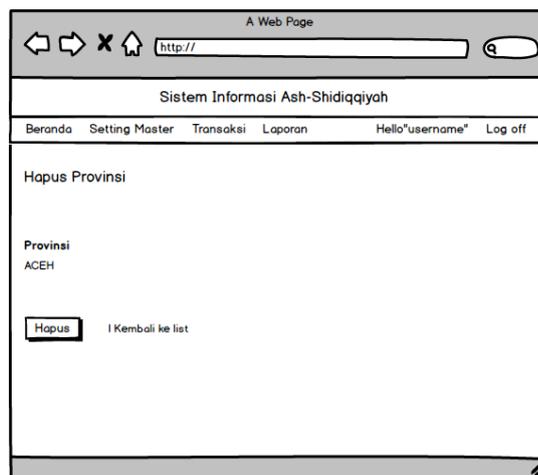
Gambar 3. 15 Rancangan tampilan menu *dropdown* pada fitur laporan

3.7.3 Rancangan Antarmuka Halaman Rincian dan Halaman Hapus

Rancangan antarmuka halaman rincian merupakan rancangan untuk menampilkan rincian data yang ada di 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*, manajemen dan transaksi. Gambaran perancangan antarmuka untuk halaman rincian dapat dilihat pada gambar 3.16 dan untuk gambaran perancangan antarmuka untuk halaman hapus dapat dilihat pada gambar 3.17.



Gambar 3. 16 Rancangan tampilan halaman rincian jenis kelamin

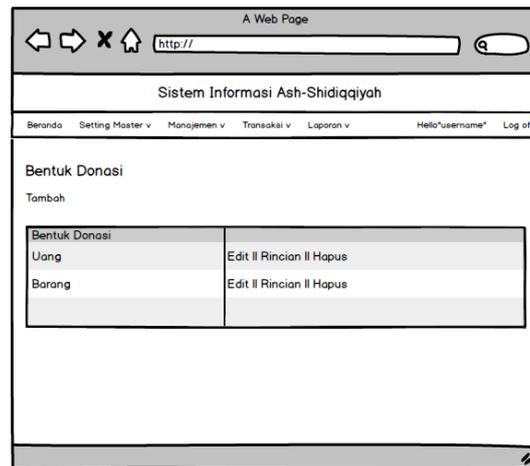


Gambar 3. 17 Rancangan tampilan halaman rincian jenis kelamin

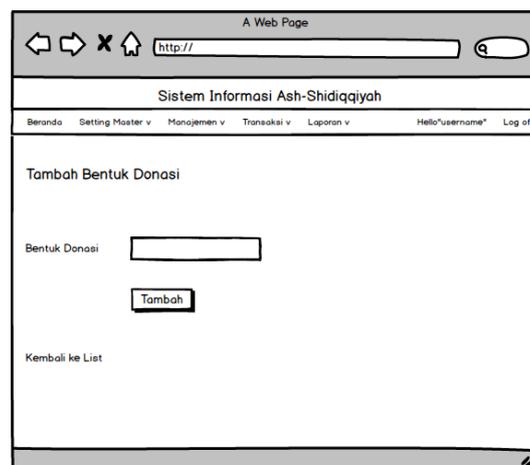
3.7.4 Rancangan Antarmuka *Setting Master* Menu Bentuk Donasi

1. Halaman Daftar Anak

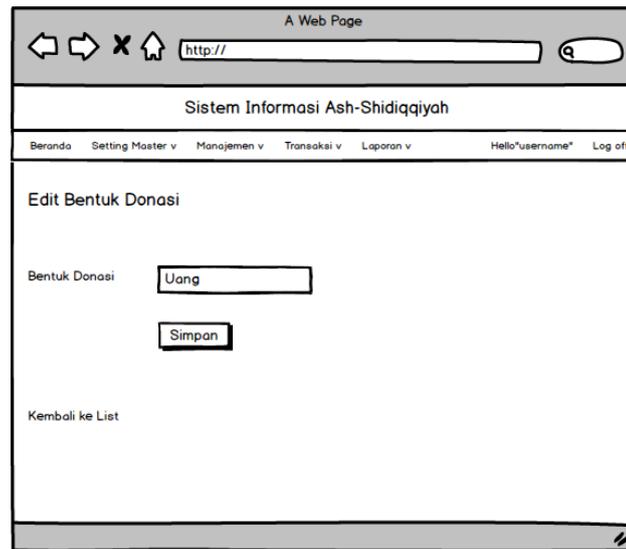
Jika menu bentuk donasi dipilih maka aplikasi akan menampilkan halaman bentuk donasi yang berfungsi untuk mengelola data bentuk donasi. Gambaran perencanaan halaman daftar bentuk donasi dapat dilihat pada gambar 3.18. Di halaman terdapat fitur untuk mengelola data bentuk donasi tambah, edit, rincian dan hapus. Gambaran untuk perancangan halaman tambah bentuk donasi dapat dilihat pada gambar 3.19 dan gambaran perancangan halaman edit bentuk donasi dapat dilihat pada gambar 3.20.



Gambar 3. 18 Rancangan tampilan halaman daftar bentuk donasi



Gambar 3. 19 Rancangan tampilan halaman tambah bentuk donasi

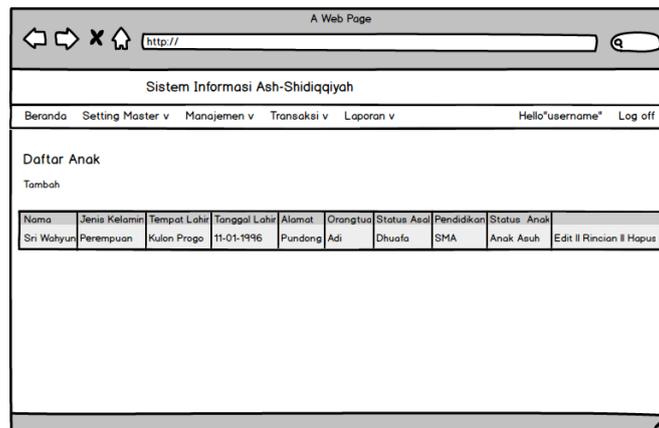


Gambar 3. 20 Rancangan tampilan halaman edit bentuk donasi

3.7.5 Rancangan Antarmuka Manajemen Menu Mengelola Data Anak

1. Daftar Data Anak

Rancangan antarmuka halaman data anak merupakan halaman yang digunakan untuk memasukan dan mengubah data anak. Admin dapat memasukan data sesuai dengan anak yang berada di panti asuhan. Selain itu juga admin dapat memasukan data orang tua melalui halaman anak asuh. Gambaran rancangan antarmuka halaman daftar data anak dapat dilihat pada gambar 3.21.



Gambar 3. 21 Rancangan tampilan halaman daftar data anak

Pada halaman ini admin dapat menambah, edit, melihat rincian dan menghapus data anak. Perancangan halaman tambah dapat dilihat pada gambar 3.22 dan perancangan halaman edit dapat dilihat pada gambar 3.23.

A Web Page

http://

Sistem Informasi Ash-Shidiqqiyah

Beranda Setting Master v Manajemen v Transaksi v Laporan v Hello*username* Log off

Tambah Anak Asuh

Nama

Jenis Kelamin

Tempat Lahir

Tanggal Lahir

Provinsi

Kota/Kabupaten

Kecamatan

Desa

Alamat Lengkap

Status Asal

Pendidikan

Status Anak

Penanggung Jawab

Ayah

Ibu

Kembali ke List

Case,

- Penanggung Jawab pilih Orang tua, otomatis menampilkan text input Ayah dan ibu. ini juga berlaku jika pengguna memilih Wali sebagai penanggung jawab, maka yang akan muncul text input Wali.
- Ayah adalah contoh untuk menambah orang tua baru.
- Ibu adalah contoh untuk memilih penanggung jawab ibu yang namanya suda ada di database

Gambar 3. 22 Rancangan tampilan halaman tambah data anak

Pada halaman tambah data anak, sebelum admin memilih provinsi *tools combo box* untuk kota/kabupaten, kecamatan dan desa diberi kondisi disable. Di halaman tambah ini juga dapat menambah data orangtua. Langkah awal dalam menambah data orangtua/wali admin memilih penanggung jawab anak, setelah dipilih maka form untuk *input* data orangtua/wali akan ditampilkan.

A Web Page

http://

Sistem Informasi Ash-Shidiqqiyah

Beranda Setting Master Transaksi Laporan Hello "username" Log off

Edit Anak Asuh

Nama: Haryono

Jenis Kelamin: Laki-laki

Tempat Lahir: Kulon Progo

Tanggal Lahir: 22/04/1990

Provinsi: DI YOGYAKARTA

Kota/Kabupaten: KULON PROGO

Kecamatan: KOKAP

Desa: HARGOTIRTO

Alamat Lengkap: Hargotirto Rt01/Rw04

Status Asal: Piatu

Pendidikan: SLTA

Status Anak: Alumni

Ayah: Marto

Ibu: Sinem(alm)

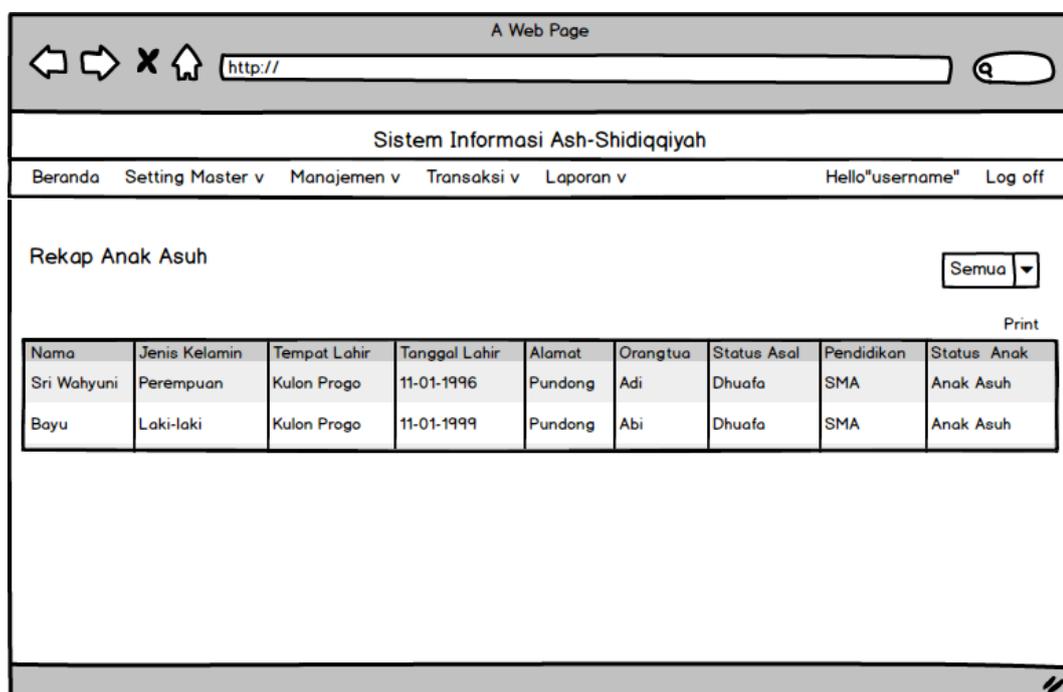
[Kembali ke List](#)

Gambar 3. 23 Rancangan tampilan halaman edit data anak

3.7.6 Rancangan Antarmuka Laporan

2. Laporan Data Anak

Fitur laporan merupakan halaman rekap untuk beberapa data. Fitur ini memiliki 4(empat) yaitu, menu laporan data anak, menu laporan data pengelola, menu laporan data donasi dan menu laporan data pengeluaran. Rancangan halaman laporan data anak dapat dilihat pada gambar 3.24.



Gambar 3. 24 Rancangan tampilan halaman