



BAB 3 ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem yang Berjalan

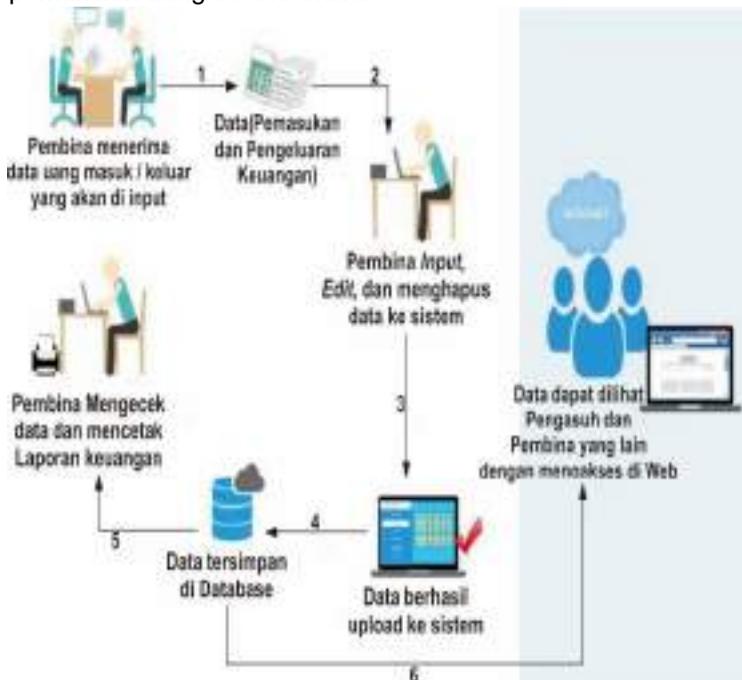
Proses administrasi keuangan yang berjalan saat ini adalah bendahara mencatat semua kegiatan keluar dan masuknya uang dengan menyertakan bukti pembayaran atau nota pada setiap transaksi yang dilakukan atas kepentingan asrama yang nantinya akan menjadi satu bentuk laporan keuangan. Sampai saat ini sistem administrasi keuangan diasrama masih dilakukan secara manual/konvensional, yaitu menggunakan kertas untuk pencatatan uang masuk ataupun uang keluar. Banyak masalah yang timbul seperti, kesalahan pencatatan uang keluar ataupun uang masuk, minim akses sehingga kurangnya transparansi keuangan dan kurang *up to date*, data hilang atau rusak yang bisa disebabkan oleh kesalahan orang ataupun lupa dalam menyimpan data, pengawasan rendah sehingga saat terjadi kecurangan oleh seseorang akan sulit untuk dilacak dan juga lama dalam mengetahuinya sehingga kerugian akan semakin besar, tidak adanya pembukuan yang rapi sehingga tidak ada acuan anggaran belanja dari yang sudah dihabiskan di tahun-tahun sebelumnya, dan juga akan memakan waktu lama dalam pembuatan laporan keuangan.

Rich Picture dibuat untuk menjelaskan alur atau proses bisnis dari sistem yang dibuat. Pada proses bisnis pembuatan *web* Administrasi Keuangan ini terdiri dari satu orang pengguna yaitu Admin. Pertama Admin mendiskusikan dengan pimpinan asrama, pembina, data atau dokumen apa saja yang akan di masukkan kedalam sistem. Kemudian apabila data sudah didapat maka Admin bisa memasukan data tersebut ke dalam sistem. Data dan dokumen yang telah ada di dalam sistem dapat dilihat oleh pengasuh dan Pembina di *web*. Pada sistem ini terdapat proses bisnis Administrasi Keuangan

Pembina akan mengelola data ulang dan Ketika sudah disetujui maka *softfile* akan di simpan di laptop atau flashdisk pembina.

3.2 Analisis Sistem yang Diusulkan

Sistem yang akan diusulkan di bangun dengan bahasa pemrograman PHP dan database MySQL dengan *framework laravel*. System ini diharapkan dapat membantu kegiatan pengelolaan uang masuk dan uang keluar agar dapat dilakukan lebih efektif dan efisien, Pengasuh atau Pembina yang lain juga dapat memantau perputaran uang sehingga terciptanya transparansi keuangan, dan pengembangan asrama dengan bisa melihat data record pemasukan uang dan pemakaian uang secara online.



Gambar 3.2 Proses Bisnis yang diusulkan

Seperti yang terlihat pada Gambar 3.2, bahwa Proses Bisnis dari Administrasi Keuangan yang baru dimana terdapat beberapa perubahan pada tugas Pembina dimana Pembina dapat mengelola data uang keluar dan uang masuk, dengan tersistemasi dan juga laporan yang terbuat secara otomatis. Dengan fitur tersebut baik pengasuh maupun Pembina dapat langsung mendownload file hasil laporan keuangan dari *website*, tanpa harus menunggu di cetak terlebih dahulu.

3.2.1 Kebutuhan Fungsional

Setelah menjelaskan usulan sistem yang baru dengan baik, maka akan dapat diidentifikasi daftar kebutuhan fungsional dari sistem yang di usulkan. Kebutuhan fungsional dapat di jabarkan melalui point – point, tabel atau bentuk lain yang sesuai. Kebutuhan fungsional Rancang Bangun Sistem Informasi Administrasi Keuangan Pondok Pesantren Berbasis Web Menggunakan Metode Scrum dapat dilihat pada tabel di bawah ini:

- 1) Sistem dapat memberikan informasi mengenai riwayat pemasukan dan pengeluaran uang sewaktu-waktu.
- 2) Sistem mampu mengelola masuk dan keluarnya keuangan.
- 3) Sistem mampu mencetak laporan keuangan setiap bulannya.

3.2.2 Kebutuhan Non Fungsional

Selain kebutuhan fungsional tentu sebuah sistem juga memiliki kebutuhan non fungsional, antara lain:

- a) Perangkat keras
Perangkat keras yang dibutuhkan dalam penggunaan system ini adalah PC dengan spesifikasi:

- Ram 4 GB
- HDD/SSD 256 GB
- Proccesor Intel I3
- LCD 16 inch

b) Perangkat lunak

Kebutuhan perangkat lunak penggunaan system ini antara lain:

- MySQL
- Google Chrome
- Windows 10
- Xampp
- Framework Laravel

c) Keamanan

Untuk keamanan dalam system, digunakan *username* dan *password* yang dimasukkan setiap kali *login* aplikasi.

3.3 Perancangan Sistem yang Diusulkan

Model penelitian untuk pengembangan sistem ini adalah menggunakan model pengembangan *scrum* yang bisa dilihat pada gambar 3.3 dibawah ini.



Gambar 3.3 Tahapan Metode Scrum

Perancangan sistem ini dibuat dengan asumsi pemodelan berorientas objek maka dari itu komponen yang ada adalah Metode Scrum, pemodelan UML (*Use Case, Activity, Sequence* dan *Class Diagram*) dan juga gambaran terkait *user interface* sistem. Aturan main dari Scrum mengikat cara, peran dan artifak serta menggambarkan hubungan dan interaksi antara satu komponen dengan yang lainnya.

3.3.1 Tim Scrum (*Inception*)

Ada beberapa anggota dalam scrum team yaitu:

1. *Product owner*

Orang yang bertanggung jawab untuk memaksimalkan nilai bisnis dari produk yang dihasilkan oleh development team. *Product owner* adalah satu-satunya orang yang bertanggung jawab dalam pengelolaan *product backlog*. Pengelolaan *product backlog* adalah:

- a. Menyampaikan isi dan *product backlog* item secara jelas
- b. Mengurutkan *product backlog* item untuk mencapai tujuan dan misi dengan cara terbaik
- c. Mengoptimalkan nilai bisnis dari pekerjaan yang dilakukan oleh *development team*
- d. Memastikan agar *product backlog* dapat dilihat, transparan, dan jelas untuk semua pihak, dan menampilkan apa yang akan dikerjakan selanjutnya oleh scrum team
- e. Memastikan development team memahami *product backlog* item hingga batas tertentu.

2. *Development team*

Terdiri dari para ahli profesi yang bekerja untuk mengantarkan increment “selesai” yang berpotensi untuk dirilis di setiap akhir sprint. Development team dibentuk untuk

menyusun dan mengelola pekerjaan mereka sendiri. *Development team* memiliki karakteristik sebagai berikut:

- a. Mereka swakelola
- b. *Development team*, bersifat lintas-fungsi
- c. Scrum tidak mengenal jabatan untuk anggota development team
- d. Scrum tidak mengenal pengelompokan didalam development team berdasarkan jenis-jenis pekerjaan ini
- e. Setiap anggota development team bisa saja memiliki keahlian khusus dan focus di bidang tertentu, tetapi tanggung gugat adalah milik seluruh anggota development team.

3. Scrum master

Scrum master bertanggung jawab untuk mengenalkan dan menyokong penggunaan scrum sebagaimana dijelaskan di dalam scrum ini, *scrum master* melakukan ini dengan membantu orang-orang agar dapat memahami teori, praktik-praktik, aturan-aturan dan tata nilai scrum. *Scrum master* adalah pemimpin yang melayani *scrum team*. Scrum master membantu orang-orang di luar *scrum team* untuk dapat memahami interaksi mana yang bermanfaat dan tidak bermanfaat. *Scrum master* membantu orang-orang untuk mengubah interaksi ini guna memaksimalkan nilai bisnis yang dihasilkan oleh *scrum team*.

Dari penjelasan di atas maka dalam penelitian ini terbentuk sebuah tim scrum seperti pada table dibawah ini :

Tabel 3.1 Tim Scrum

Tugas	Nama
<i>Product Owner</i>	Asrama 27 Al-Furqon PP. Darul Ulum
Tim Pengembang	Bendahara Asrama & M. Yusuf Affandi
<i>Scrum Master</i>	Muhammad Yusuf Affandi



3.3.2 User Story

Langkah pertama di dalam inception adalah dengan Membuat *User Story* dari Asrama Al-Furqon. Beberapa kolom didalam *User Story* sebagai berikut :

- a. *Actor* adalah pelaku yang melakukan *Story* tersebut
- b. *Objective* adalah tujuan dari *Story* tersebut dibuat
- c. *Pre-Condition* adalah keadaan sebelum *Story* tersebut akan dilaksanakan
- d. *Main Flow* adalah tahapan-tahapan didalam *Story* yang akan dilakukan oleh *Actor* untuk mencapai *Objective*.
- e. *Alternative Flow* adalah jika *Actor* tidak memenuhi *Pre-Condition* maka akan mengikuti alur tersebut
- f. *Post Condition* adalah kondisi setelah *Objective* selesai dilakukan.

1. Login User

Tabel 3.2 User Story Login

No.	Alur	Keterangan
1	<i>Actor</i>	Admin, Bendahara, Pengawas
2	<i>Objective</i>	<i>User</i> dapat melakukan <i>login</i>
3	<i>Pre-Condition</i>	<i>User</i> sudah terregistrasi di sistem
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. <i>User</i> memilih login dengan menggunakan <i>email & password</i> 2. <i>User</i> dialihkan ke dashboard awal sistem administrasi keuangan
5	<i>Alternatives Flow</i>	<i>User</i> yang belum daftar maka akan didaftarkan oleh admin sistem
6	<i>Post Condition</i>	<i>User</i> masuk ke dashboard awal dengan kondisi sudah <i>login</i>

2. Menambahkan User

Tabel 3.3 User Story Admin Menambahkan User

No.	Alur	Keterangan
1	<i>Actor</i>	Admin
2	<i>Objective</i>	Admin dapat mendaftarkan <i>user</i> baru
3	<i>Pre-Condition</i>	<i>User</i> belum terregistrasi di sistem
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Admin memilih tambah pengguna pada menu pengguna 2. Admin mendaftarkan <i>username</i> / <i>email</i> dan password yang bisa di ganti oleh <i>user</i> nanti 3. Admin dialihkan ke antarmuka menu pengguna
5	<i>Alternatives Flow</i>	Admin sudah pernah mendaftarkan <i>user</i> akan Kembali ke menu pengguna
6	<i>Post Condition</i>	Admin sudah berhasil mendaftarkan <i>user</i> dan sudah dapat <i>login</i>

3. Menambahkan Kategori

Tabel 3.4 User Story Menambahkan Kategori

No.	Alur	Keterangan
1	<i>Actor</i>	Admin & Bendahara
2	<i>Objective</i>	Admin & bendahara dapat menambahkan kategori baru
3	<i>Pre-Condition</i>	Admin & bendahara mengakses menu data kategori
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Admin & bendahara memilih tambah kategori pada menu data kategori 2. Admin & bendahara mengisi form tambah kategori



5	<i>Alternatives Flow</i>	Admin & sudah pernah menambahkan kategori dengan nama yang sama maka akan dialihkan kembali ke menu data kategori
6	<i>Post Condition</i>	Admin & bendahara sudah berhasil menambahkan data kategori baru

4. Menambahkan Transaksi

Tabel 3.5 User Story Menambahkan data transaksi

No.	Alur	Keterangan
1	<i>Actor</i>	Admin & Bendahara
2	<i>Objective</i>	Admin & bendahara dapat menambahkan transaksi baru
3	<i>Pre-Condition</i>	Admin & Bendahara mengakses menu data transaksi
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Admin & bendahara memilih tambah transaksi pada menu data transaksi 2. Admin & bendahara mengisi form tambah transaksi 3. Admin & bendahara memastikan kembali data yang sudah diisikan
5	<i>Alternatives Flow</i>	Admin & sudah pernah menambahkan transaksi dengan data yang sama maka akan dialihkan kembali ke menu data transaksi
6	<i>Post Condition</i>	Admin & bendahara sudah berhasil menambahkan data transaksi baru

5. Melihat Data Transaksi



Tabel 3.6 User Story pengawas melihat data transaksi

No.	Alur	Keterangan
1	<i>Actor</i>	Pengawas
2	<i>Objective</i>	Pengawas dapat melihat data transaksi yang sudah ditambahkan
3	<i>Pre-Condition</i>	Pengawas mengakses menu data transaksi
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengawas mengakses menu data transaksi 2. Pengawas dapat memonitoring transaksi keuangan yang berjalan
5	<i>Alternatives Flow</i>	Pengawas belum terdaftar pada sistem maka tidak bisa mengakses menu transaksi
6	<i>Post Condition</i>	Pengawas berhasil memonitoring jalanya transaksi keuangan dengan <i>up to date</i>

6. Melihat & cetak laporan keuangan

Tabel 3.7 User Story Melihat & cetak laporan keuangan

No.	Alur	Keterangan
1	<i>Actor</i>	Admin, Bendahara, Pengawas
2	<i>Objective</i>	User dapat melihat dan mencetak laporan keuangan
3	<i>Pre-Condition</i>	User mengakses menu laporan
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. User memilih menu laporan 2. User mengisi tanggal dan kategori laporan yang akan dilihat / dicetak



		3. <i>User</i> dapat melihat laporan keuangan dan dapat mencetak / mendownload laporan
5	<i>Alternatives Flow</i>	<i>User</i> salah / belum mengisi tanggal dan kategori maka akan muncul notifikasi
6	<i>Post Condition</i>	<i>User</i> dapat melihat, mencetak, dan mendownload laporan keuangan.

7. Mengganti Password

Tabel 3.8 User Story Mengganti Password

No.	Alur	Keterangan
1	<i>Actor</i>	Admin, Bendahara, Pengawas
2	<i>Objective</i>	<i>User</i> dapat mengganti password
3	<i>Pre-Condition</i>	<i>User</i> mengakses menu ganti password
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. <i>User</i> memilih menu ganti password 2. <i>User</i> mengisi password yang sekranag dan password baru 3. <i>User</i> mengkonfirmasi password baru dan menyimpan perubahan password
5	<i>Alternatives Flow</i>	<i>User</i> salah dalam memasukkan password lama atau salah dalam mengkonfirmasi password baru akan muncul notifikasi
6	<i>Post Condition</i>	<i>User</i> dapat mengganti password.

8. User Logout

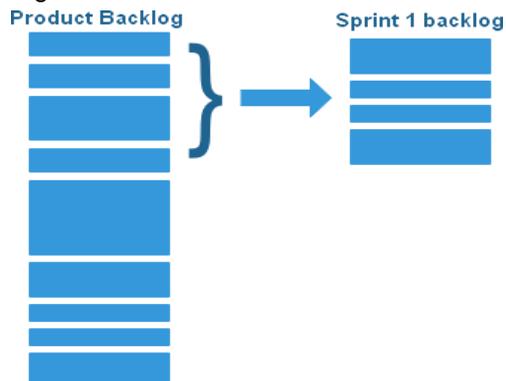


Tabel 3.9 User Story Logout

No.	Alur	Keterangan
1	<i>Actor</i>	Admin, Bendahara, Pengawas
2	<i>Objective</i>	User memilih fitur <i>logout</i> untuk keluar dari sistem administrasi keuangan
3	<i>Pre-Condition</i>	User sudah <i>login</i> sebelumnya
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. User memilih menu <i>account</i> 2. User memilih menu <i>logout</i> 3. Muncul pop up untuk konfirmasi 4. User memilih tombol “yes”
5	<i>Alternatives Flow</i>	User memilih <i>button</i> “tidak” akan tetap pada sistem
6	<i>Post Condition</i>	User <i>logout</i> dari sistem.

3.3.3 Product Backlog

Dari fitur-fitur yang telah ditentukan untuk dikembangkan pada rancang bangun sistem informasi administrasi keuangan pondok pesantren kemudian disusun *product backlog* yang merupakan daftar aktivitas pengembangan berdasarkan analisa kebutuhan dalam pengembangan sistem informasi.



Gambar 3.4 Pemecahan Product Backlog

Berdasarkan gambar di atas, setiap persegi panjang merepresentasikan *item backlog* yang diurutkan berdasarkan derajat kepentingan. Ukuran dari setiap persegi panjang merepresentasikan poin pada perkiraan awal *product backlog*. Tinggi dari kurung kurawal biru merepresentasikan perkiraan kecepatan pengerjaan tim. Kecepatan yang dimaksud adalah jumlah *item backlog* yang diperkirakan dapat dikerjakan tim selama *sprint* yang akan datang. *Sprint Backlog* pada bagian kanan gambar merupakan sebagian kecil dari *product backlog* yang akan dikerjakan dalam *sprint*.

Tabel 3.10 Daftar Product Backlog

ID	Nama	Imp, (0-100)	Est. (dalam hari)	How to Demo
1	Pembuatan UML	100	10	Memeriksa UML yang telah dibuat sesuai dengan kebutuhan aplikasi
2	Rancangan UI Aplikasi	100	5	Memeriksa Rancangan UI aplikasi
3	Konfigurasi server	100	3	<ul style="list-style-type: none"> • Proses Instalasi Server
4	Konfigurasi Database	100	1	<ul style="list-style-type: none"> • Membuat database sesuai rancangan
5	Fitur <i>Login</i> (web)	100	2	<ul style="list-style-type: none"> • Ketika memasukkan url maka halaman <i>Login</i> terbuka • Ketika menekan tombol <i>Login</i> maka akan melakukan <i>validasi</i> data • Ketika <i>email</i> dan <i>password</i> sesuai maka <i>Login</i> berhasil

ID	Nama	Imp, (0-100)	Est. (dalam hari)	<i>How to Demo</i>
6	Cek Fitur Login (web)	100	1	<ul style="list-style-type: none"> • Ketika <i>email</i> dan <i>password</i> sesuai maka <i>Login</i> berhasil • Memeriksa level <i>user</i> sesuai dengan yang terdaftar di database
7	Halaman Dashboard (web)	100	10	<ul style="list-style-type: none"> • Ketika berhasil <i>login</i> akan menampilkan halaman dashboard • Dashboard menampilkan halaman informasi singkat seputar keuangan • Dapat memilih session untuk pengeluaran, pemasukan, dan saldo
8	Fitur Data Kategori (web)	100	5	<ul style="list-style-type: none"> • Ketika <i>user</i> adalah admin / bendahara maka fitur terbuka • Ketika <i>user</i> berhasil <i>login</i> maka akan tampil halaman dashboard dan menu di sebelah kanan • <i>User</i> dapat menambahkan data kategori • Dapat mengedit data • Dapat menghapus data

ID	Nama	Imp, (0-100)	Est. (dalam hari)	<i>How to Demo</i>
9	Fitur Data Transaksi (web)	100	10	<ul style="list-style-type: none"> • Ketika memilih menu data transaksi di halaman <i>user</i> maka tampil halaman transaksi Jika <i>user</i> adalah admin / bendahara, maka : <ul style="list-style-type: none"> • Dapat melihat semua data transaksi • Dapat menambahkan data transaksi dengan memilih tombol data • Mengisi form data tambah transaksi • Dapat menyimpan transaksi yang telah ditambahkan • Dapat mengedit data • Dapat menghapus data Jika <i>user</i> adalah pengawas, maka : <ul style="list-style-type: none"> • Dapat melihat semua data transaksi • Tidak bisa menambahkan data • Tidak bisa menghapus data • Tidak dapat mengedit data



ID	Nama	Imp, (0-100)	Est. (dalam hari)	<i>How to Demo</i>
10	Fitur Laporan (web)	100	5	<ul style="list-style-type: none"> • Ketika memilih menu laporan di halaman <i>user</i> maka tampil halaman laporan • Dapat memilih tanggal dan kategori untuk laporan • Dapat menampilkan laporan keuangan sesuai dengan tanggal dan kategori yang di isikan
11	Fitur Cetak Laporan (web)	100	2	<ul style="list-style-type: none"> • Setelah berhasil menampilkan laporan keuangan tombol cetak laporan akan aktif • Dapat mencetak laporan keuangan dengan menekan tombol cetak laporan • Format laporan cetak berekstensi pdf dan berukuran A4 • Laporan cetak berisi tanggal dan kategori yang sudah di isi dalam ID 10

ID	Nama	Imp, (0-100)	Est. (dalam hari)	<i>How to Demo</i>
12	Fitur Download Laporan (web)	100	2	<ul style="list-style-type: none"> • Setelah berhasil menampilkan laporan keuangan tombol download laporan akan aktif • Dapat mendownload laporan keuangan dengan menekan tombol download laporan • Format laporan cetak berekstensi excel dan berukuran A4 • Laporan cetak berisi tanggal dan kategori yang sudah di isi dalam ID 10 • File akan tersimpan di komputer <i>user</i>
13	Fitur Menu Pengguna (Admin)	100	8	<ul style="list-style-type: none"> • Ketika <i>user</i> adalah admin maka menu pengguna akan terbuka • Dapat melakukan penambahan data pengguna ketika memilih tombol Tambah Data Pengguna • Masukkan data yang diminta

ID	Nama	Imp, (0- 100)	Est. (dalam hari)	<i>How to Demo</i>
				<ul style="list-style-type: none"> • Ketika pilih tombol simpan data, maka <i>validasi</i> data • Jika data <i>valid</i>, maka data tersimpan • Dapat menghapus data Ketika memilih tombol hapus dan <i>verify</i> hapus • Dapat mengedit data pengguna
14	Fitur Ganti Password (Web)	100	3	<ul style="list-style-type: none"> • Ketika memilih menu ganti password di halaman <i>user</i> maka tampil halaman ganti password • Dapat mengisi data sesuai dengan data yang diminta • <i>Memvalidasi</i> data password lama • <i>Verifikasi</i> password baru • Ketika pilih tombol simpan data maka <i>verify</i> data • Dapat menyimpan data setelah data <i>terverifikasi</i> dan <i>valid</i>

ID	Nama	Imp, (0-100)	Est. (dalam hari)	<i>How to Demo</i>
15	Fitur <i>Log Out User</i> (web)	100	2	<ul style="list-style-type: none"> • Ketika <i>user</i> menekan menu <i>logout</i> maka akan muncul <i>verify</i> • Ketika <i>user</i> memilih tombol “Ya” maka <i>logout</i> akan di proses • Sistem <i>memverifikasi session user</i> telah berakhir • Setelah berhasil <i>logout</i> maka akan di arahkan ke tampilan <i>login awal</i>
16	<i>Implementasi Basis Data</i> (MySql)	100	1	<ul style="list-style-type: none"> • Membuat table kebutuhan sistem • Memastikan hubungan antar table benar • Dapat memasukkan data sesuai kebutuhan • Implementasi Unit

Keterangan:

1. Nilai *importance* 1-100 dimana 100 mempunyai derajat kepentingan yang paling tinggi.
2. Nilai *Story point* sama dengan nilai *man-days* yaitu 1 (satu) hari kerja oleh satu orang.

3.3.4 *The Sprint*

Menurut (Schwaber & Scrum, 2013) menjelaskan bahwa “Jantung dari *Scrum* adalah *Sprint*, yaitu sebuah batasan waktu dengan durasi satu bulan atau kurang, dimana terdapat proses pembuatan Increment yang “Selesai”, dapat digunakan dan berpotensi untuk dirilis”. Sebuah inkremen yang “Selesai” di dalam *Sprint* harus berfungsi, berpotensi untuk dirilis dan dikembangkan. *Sprint* biasanya memiliki durasi yang konsisten sepanjang proses pengembangan produk dan dibatasi selama satu bulan menurut kalender. Bila jangka waktu *sprint* terlalu panjang, maka definisi mengenai apa yang akan dibangun akan berubah, kompleksitas dapat meningkat, dan risiko dapat bertambah. *sprint* meningkatkan prediktabilitas karena adanya peninjauan dan pengadaptasian terhadap perkembangan, setidaknya setiap satu bulan sekali. Setiap *sprint* yang dijalankan akan memuat *Scrum Events* yang terdiri dari *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective*.

Dalam pengembangan sistem informasi Administrasi Keuangan Pondok Pesantren dibagi menjadi 5 (lima) *Sprint*. Berikut penjabaran dari masing-masing *Sprint*.

3.3.5 *Sprint 1 (Satu)*

Pada *Sprint 1* (satu) ini akan dijelaskan daftar pekerjaan yang akan dikerjakan berdasarkan daftar *product backlog* yang telah di buat. Berikut adalah *scrum event* pada *Sprint 1* (satu):

1. *Sprint Planing*

Pada perancangan *Sprint 1* (satu), dihasilkan keputusan sebagai berikut:



- a) Panjang *Sprint*. 2 Minggu.
- b) Tujuan: Membuat rancangan sistem informasi dan instalasi *server*
- c) Dalam menentukan *story point* yang akan dimasukkan ke dalam *Sprint*, hal pertama yang harus dilakukan adalah memperkirakan kecepatan tim. Dari perkiraan kecepatan tim maka *Product Backlog* yang dapat dimasukkan ke dalam *Sprint* sebanyak 13 *story point*. Berikut *Sprint Backlog* pada *Sprint* 1 (satu) dan penjabaran *task* dari masing-masing *Sprint Backlog* pada tabel di bawah ini:

Tabel 3.11 *print 1 (satu)*

<i>Sprint Backlog</i>	<i>Task</i>	<i>Est. (Story Point)</i>
Pembuatan UML	<ul style="list-style-type: none"> • Membuat <i>Use Case Diagram</i> • Membuat <i>Activity Diagram</i> • Membuat <i>Sequance Diagram</i> 	10
Konfigurasi Server	<ul style="list-style-type: none"> • Instalasi <i>MySql Server</i> • Instalasi <i>Framework Laravel PHP</i> • Konfigurasi <i>Framework Laravel</i> 	3

2. *Daily Scrum*

Pada saat *Sprint* dijalankan, pertemuan harian *Scrum* diadakan hampir setiap hari. Pertemuan ini berlangsung selama kurang lebih 15 menit. Dalam pertemuan ini, tim membahas apa saja yang sudah diselesaikan dengan melihat *product backlog* dan *task*

yang sudah di selesaikan. Selain itu juga membahas apa yang akan dikerjakan selanjutnya.

3. *Sprint Review*

Dalam pertemuan evaluasi *Sprint 1* (satu) membicarakan mengenai *Sprint Backlog* yang sudah diselesaikan oleh tim. Dari semua *Sprint Backlog* yang sudah dikerjakan, semua dinyatakan selesai karena sesuai dengan hasil yang diharapkan. Hasil dari *Sprint 1* (satu) dijelaskan pada *point* selanjutnya. Selain itu, pada pertemuan ini membahas *product Backlog* yang tersisa. Dari hasil pertemuan evaluasi *Sprint 1* (satu), tidak ada revisi untuk *product Backlog* yang tersisa.

4. *Sprint Retrospective*

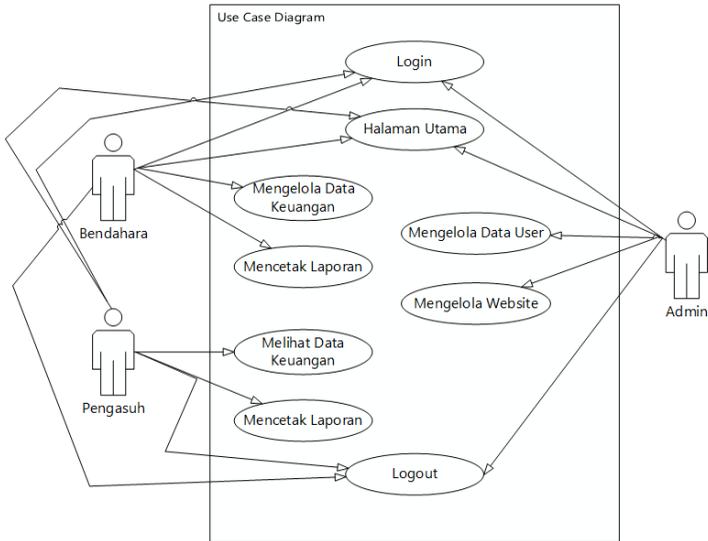
Dalam pertemuan evaluasi *Sprint* dilakukan evaluasi kinerja tim selama *Sprint 1* (satu). Perkiraan kecepatan tim pada *Sprint 1* (satu) sudah tepat. Hal ini dibuktikan dengan semua *Sprint Backlog* pada *Sprint 1* (satu) telah berhasil dikerjakan oleh tim. Kecepatan kinerja tim stabil. Hal ini dapat dilihat dari kekompakan tim yang menunjukkan *estimated work remaining (story point)* yang setiap harinya berkurang antara 1 (satu) atau 2 (dua) *story point*. Untuk *Sprint* selanjutnya diharapkan tim dapat mempertahankan stabilitas kecepatan kinerja tim dan dapat mengerjakan semua *Sprint Backlog*.

Berikut adalah hasil dari *sprint 1* (satu) yaitu pembuatan UML, dan *instalasi server* :

A. *Pemodelan Use Case Diagram*

Pada bagian ini menjelaskan perancangan sistem dalam *Use Case Diagram* serta mendeskripsikan aktor – aktor apa saja yang dapat mengakses sistem yang akan dibuat. Penjelasan *use case diagram* akan dijabarkan pada gambar di bawah ini:





Gambar 3.5 Use Case Diagram

Berikut penjelasan bagan *use case diagram* dapat dilihat pada tabel dibawah ini:

Tabel 3.12 Use Case Diagram

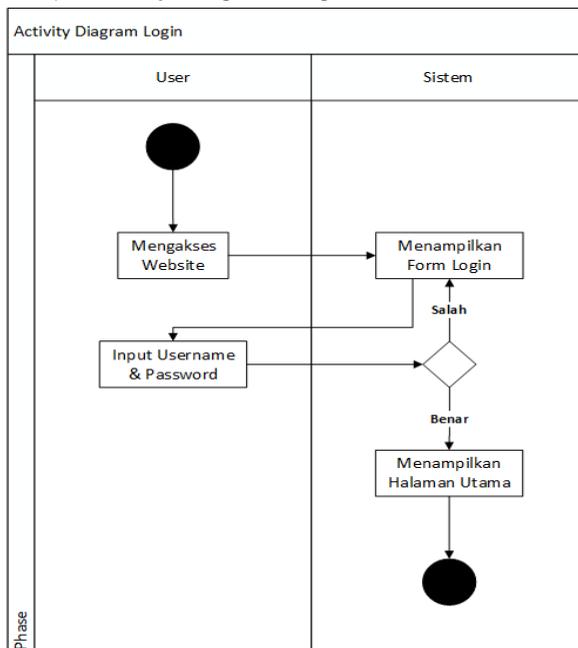
Aktor	Keterangan
Admin	Pengguna yang mempunyai hak akses melihat halaman utama, mengelola <i>user</i> , dan mengelola website.
Bendahara	Pengguna yang mempunyai hak akses untuk melihat halaman utama, mengelola keuangan, dan mencetak laporan.
Pengasuh	Pengguna yang mempunyai hak akses untuk melihat halaman utama, melihat data

	keuangan, dan mencetak laporan.
--	---------------------------------

B. Activity/Robustness Diagram

Activity Diagram menjelaskan alur kerja dan urutan aktivitas pada proses sistem. *Activity* Diagram dalam perancangan sistem ini adalah sebagai berikut:

1) *Activity Diagram Login*

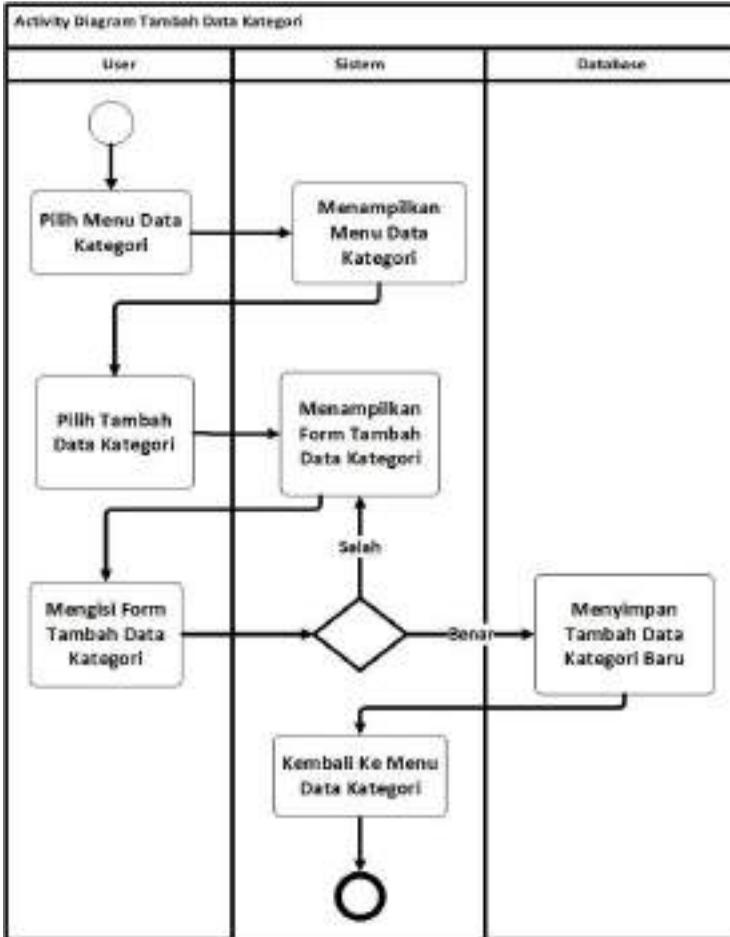


Gambar 3.6 *Activity Diagram Login*

Penjelasan *Activity* diagram diatas yaitu *user* mengunjungi website sistem untuk membuka system kemudian sistem akan menampilkan halaman form *login*, selanjutnya *user* mengisi form *login* dengan menginputkan *username* dan *password*, apabila *username* dan atau *password* benar akan

masuk ke halaman utama sistem ,tetapi jika *username* dan atau *password* salah maka akan kembali ke halaman form *login*.

2) Activity Diagram Tambah Data Kategori

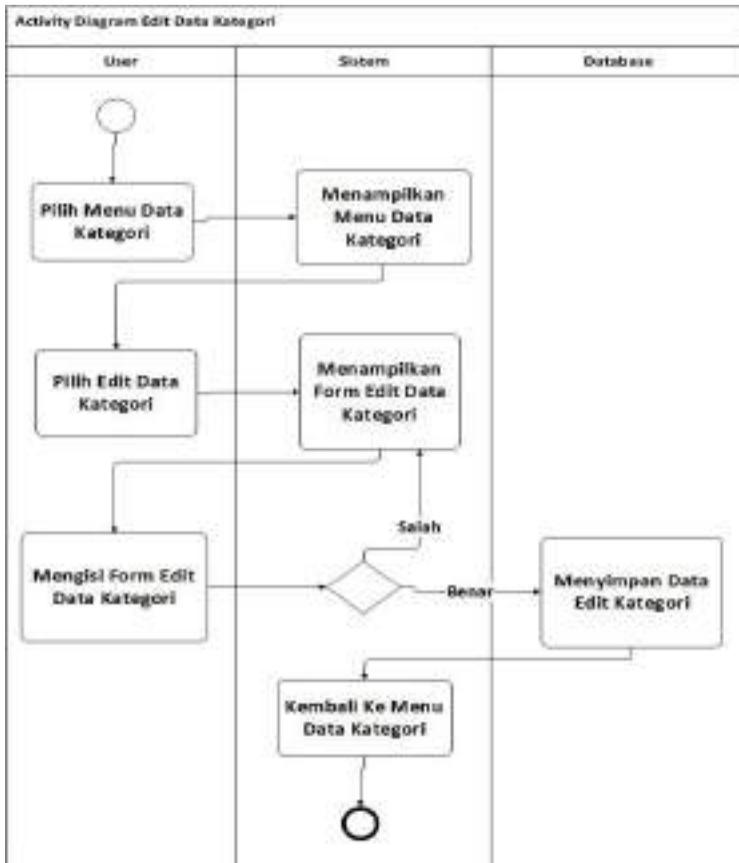


Gambar 3.7 Activity Diagram Tambah Data Kategori

Penjelasan Activity diagram diatas bagaimana untuk menambah data kategori yaitu *user login* dengan memilih menu data pemasukan dan menampilkan tabel kategori dan menekan tombol tambah pada menu kategori kemudian mengisi form tambah data kategori, jika sudah menekan tombol simpan dan

data langsung tersimpan di database, data tambah kategori berhasil disimpan.

3) Activity Diagram Edit Data Kategori

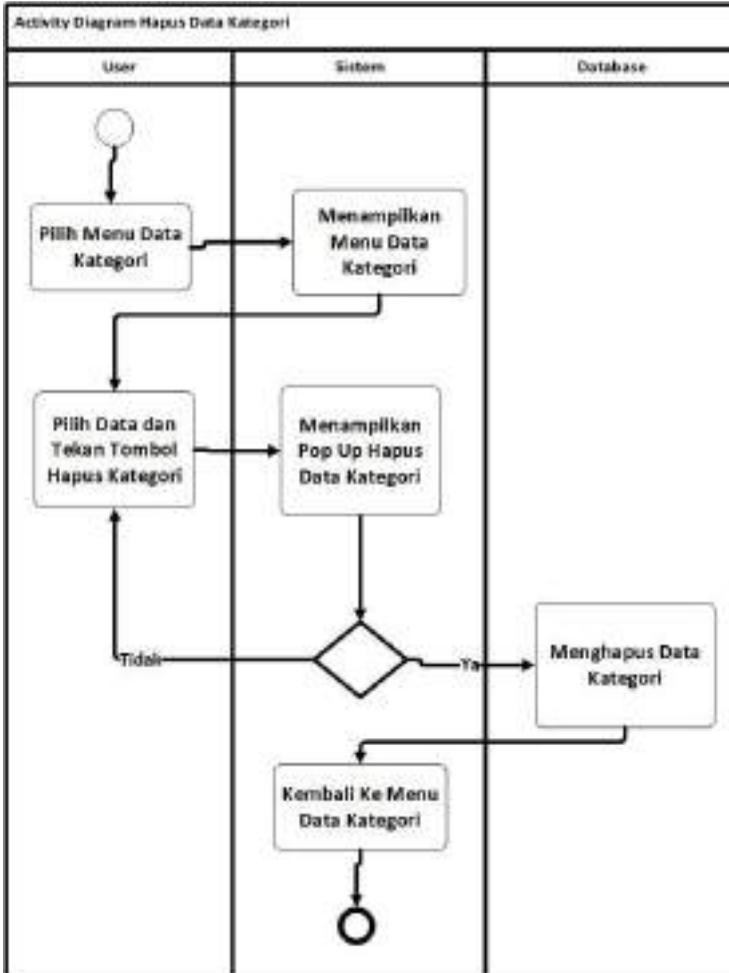


Gambar 3.8 Activity Diagram Edit Data Kategori

Penjelasan *activity* diagram di atas merupakan alur bagaimana mengedit data kategori, pertama *user login* dengan memilih menu data kategori di sistem kemudian pilih data

kategori yang akan di edit dengan menekan tombol edit kemudian data berhasil tersimpan dan teredit dari database.

4) Activity Diagram Hapus Data Kategori

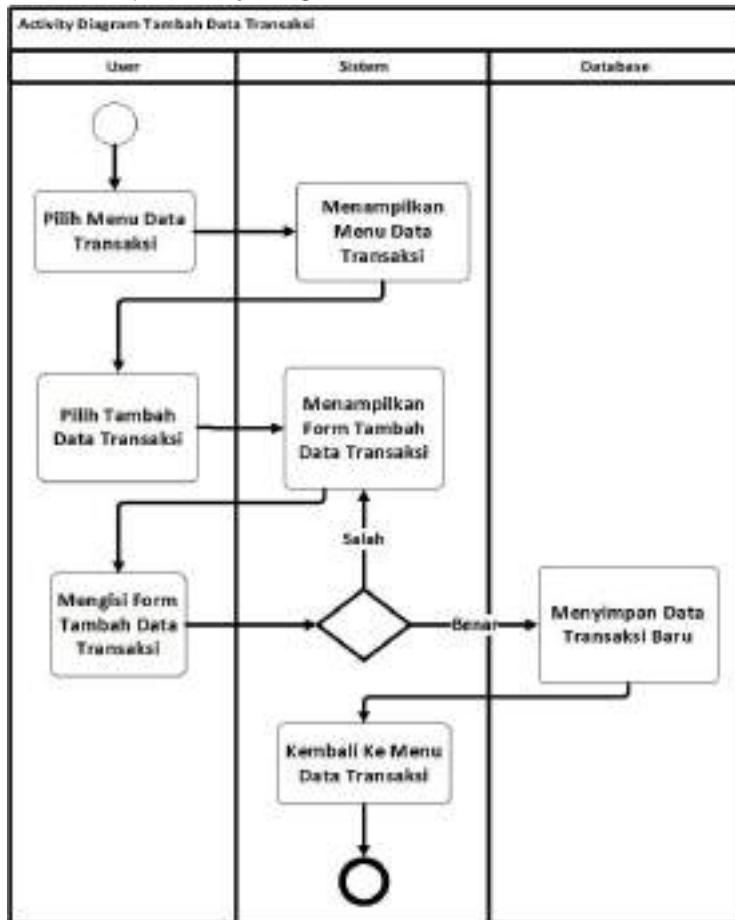


Gambar 3.9 Activity Diagram Hapus Data Kategori

Penjelasan *activity* diagram di atas merupakan alur bagaimana menghapus data kategori, pertama *user login* dengan memilih menu data kategori di sistem kemudian pilih data kategori yang akan di hapus dengan menekan tombol

hapus, kemudian data berhasil tersimpan dan terhapus dari database.

5) Activity Diagram Tambah Data Transaksi

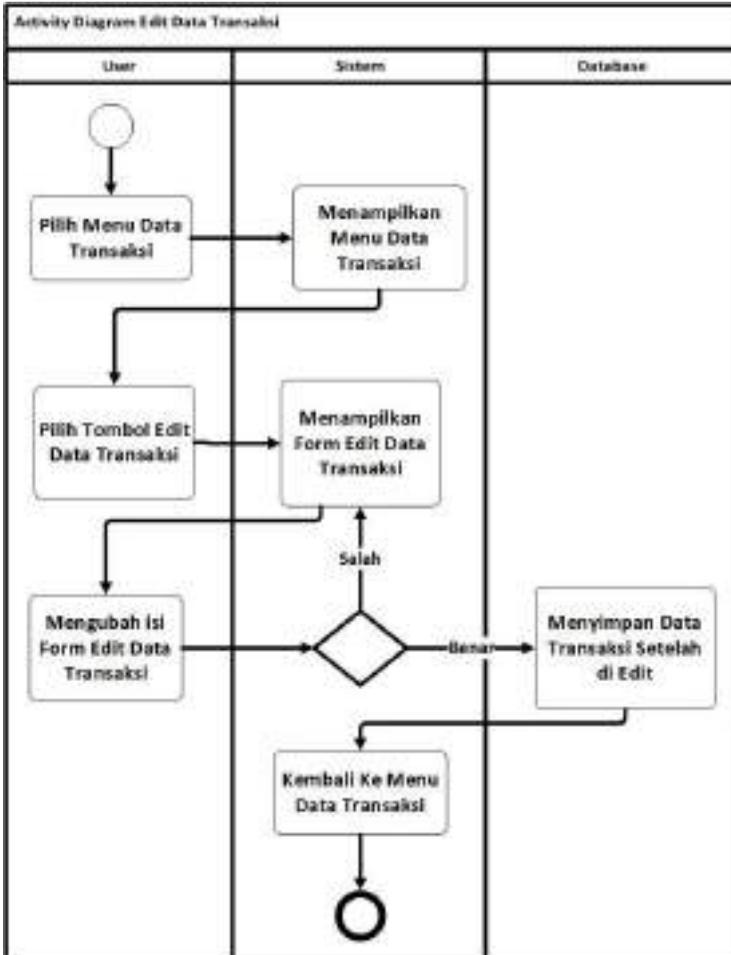


Gambar 3.10 Activity Diagram Tambah Data Transaksi

Penjelasan *activity* diagram diatas bagaimana untuk menambah data transaksi yaitu *user login* dengan memilih menu data transaksi dan menampilkan tabel transaksi dan

menekan tombol tambah pada menu transaksi kemudian mengisi form tambah transaksi, jika sudah menekan tombol simpan dan data langsung tersimpan di database, data tambah transaksi berhasil disimpan.

6) *Activity Diagram* Edit Data Transaksi

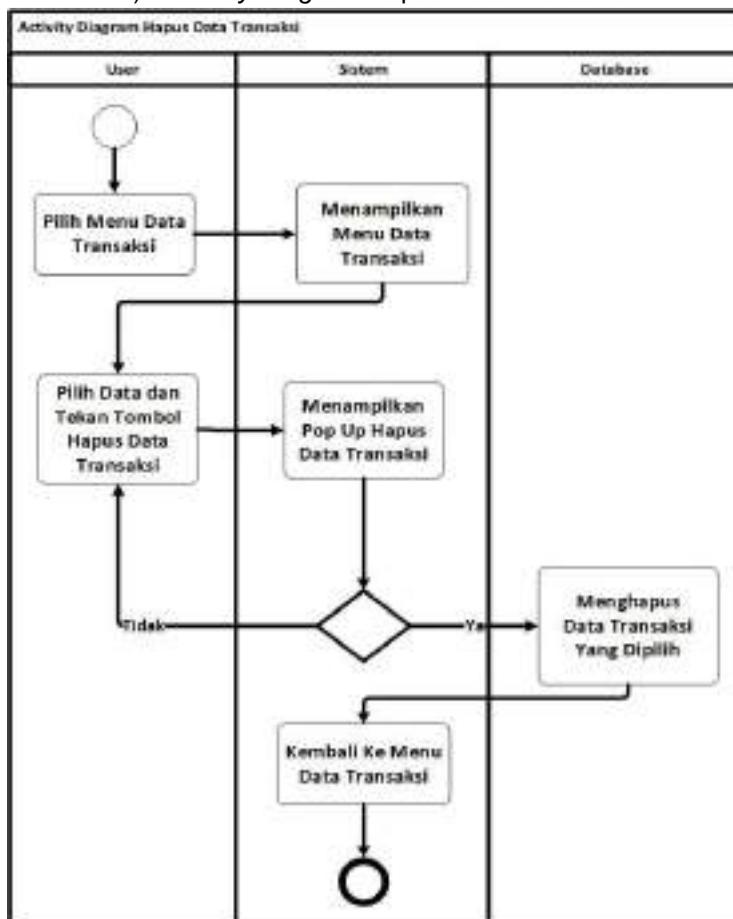


Gambar 3.11 *Activity Diagram* Edit Data Transaksi

Penjelasan *activity diagram* di atas merupakan alur bagaimana mengubah data transaksi yaitu *user login* dengan

memilih menu data transaksi kemudian pilih data transaksi yang akan di ubah dengan mengisi form, jika sudah selesai pilih tombol selesai kemudian data tersimpan di database.

7) Activity Diagram Hapus Data Transaksi

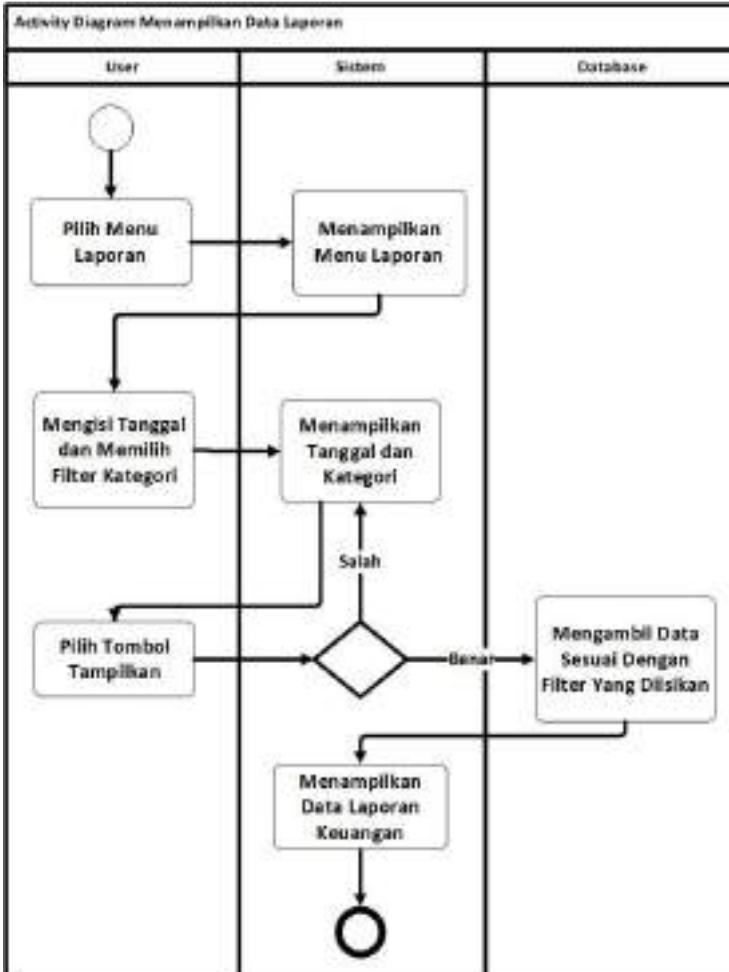


Gambar 3.12 Activity Diagram Hapus Data Transaksi

Penjelasan *activity* diagram di atas merupakan alur bagaimana menghapus data transaksi, pertama *user login*

dengan memilih menu data transaksi di sistem kemudian pilih data transaksi yang akan di hapus dengan menekan tombol hapus, kemudian data berhasil tersimpan dan terhapus dari database.

8) Activity Diagram Melihat Data Keuangan

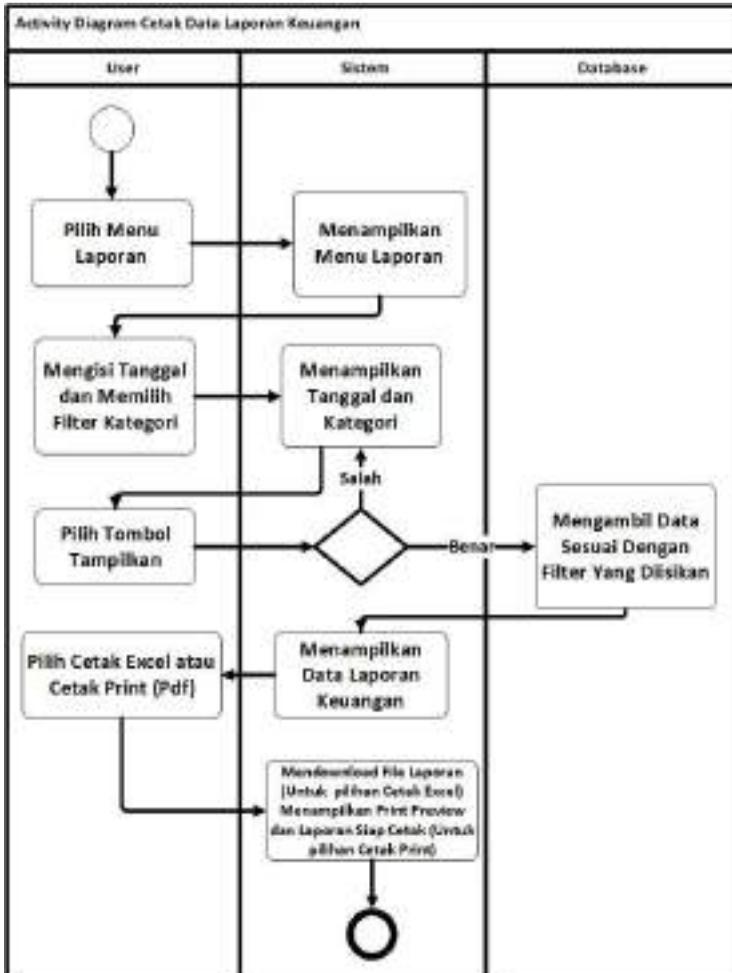


Gambar 3.13 Activity Diagram Lihat Data Keuangan

Penjelasan *activity* diagram di atas merupakan alur untuk melihat data keuangan. Pertama, *user login* ke sistem

dan memilih menu laporan lalu mengisikan tanggal dan kategori dan ambil data dari database kemudian di tampilkan di system.

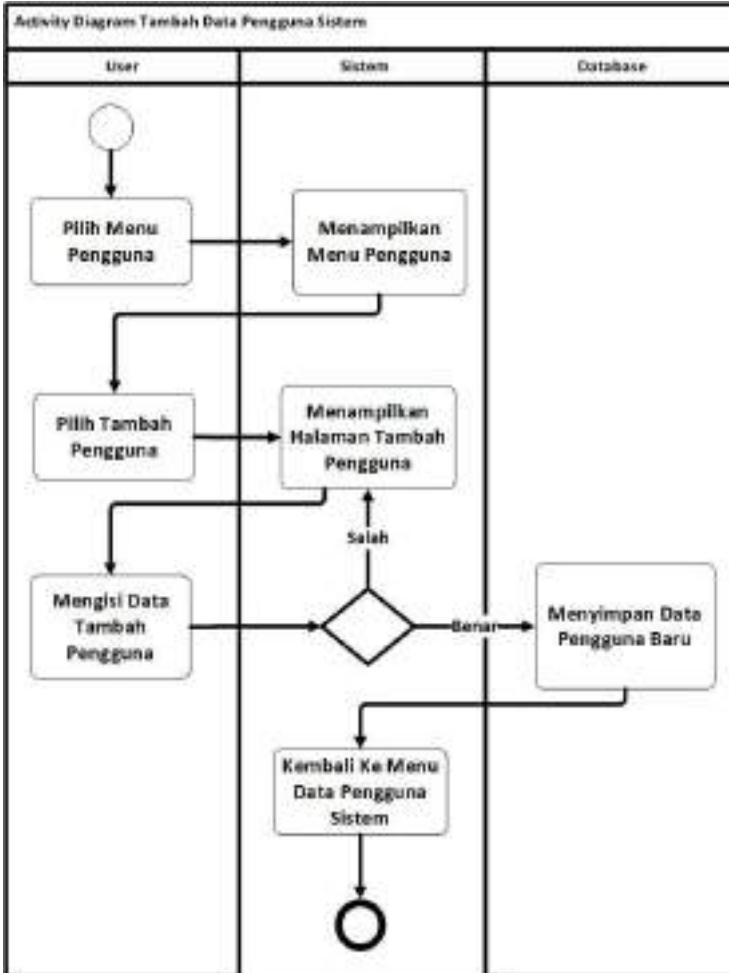
9) *Activity Diagram* Cetak Laporan



Gambar 3.14 *Activity Diagram* Cetak Laporan

Penjelasan *activity* diagram di atas merupakan alur untuk melihat dan mencetak laporan data keuangan yang dilakukan oleh bendahara/pengasuh. Pertama, *user login* ke sistem dan memilih menu laporan lalu mengisi tanggal dan kategori dan ambil data dari database kemudian di tampilkan di sistem, jika ingin mencetak laporan maka tekan tombol cetak.

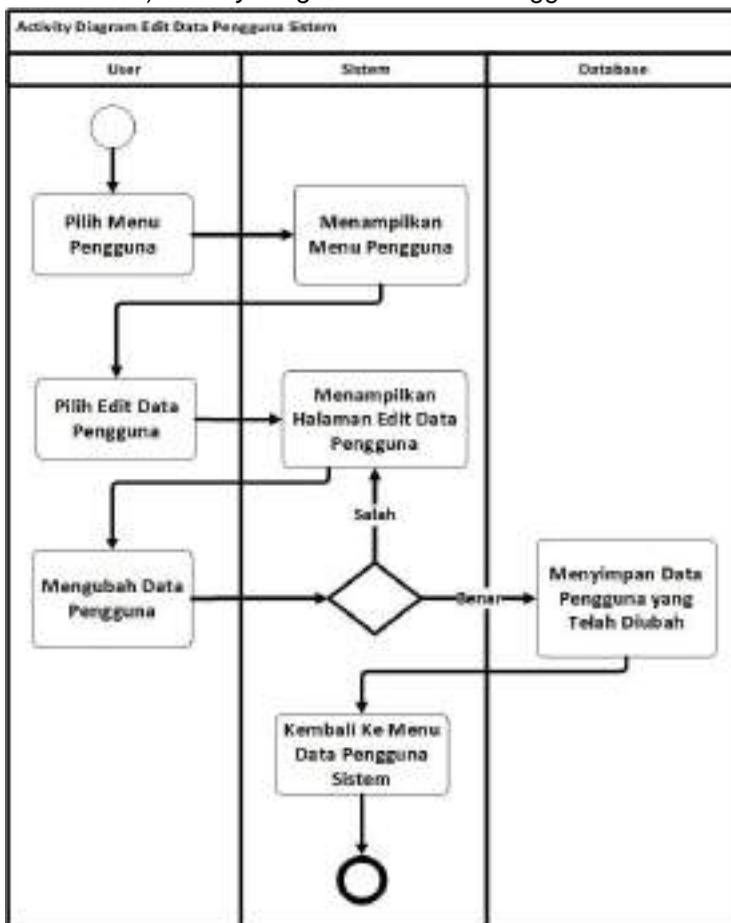
10) *Activity Diagram* Tambah Pengguna



Gambar 3.15 *Activity Diagram* Tambah Data Pengguna

Penjelasan *activity* diagram diatas bagaimana untuk menambah data pengguna yaitu *user login* dengan memilih menu pengguna dan menampilkan tabel pengguna dan menekan tombol tambah pada menu pengguna kemudian mengisi form tambah pengguna, jika sudah menekan tombol simpan dan data langsung tersimpan di database.

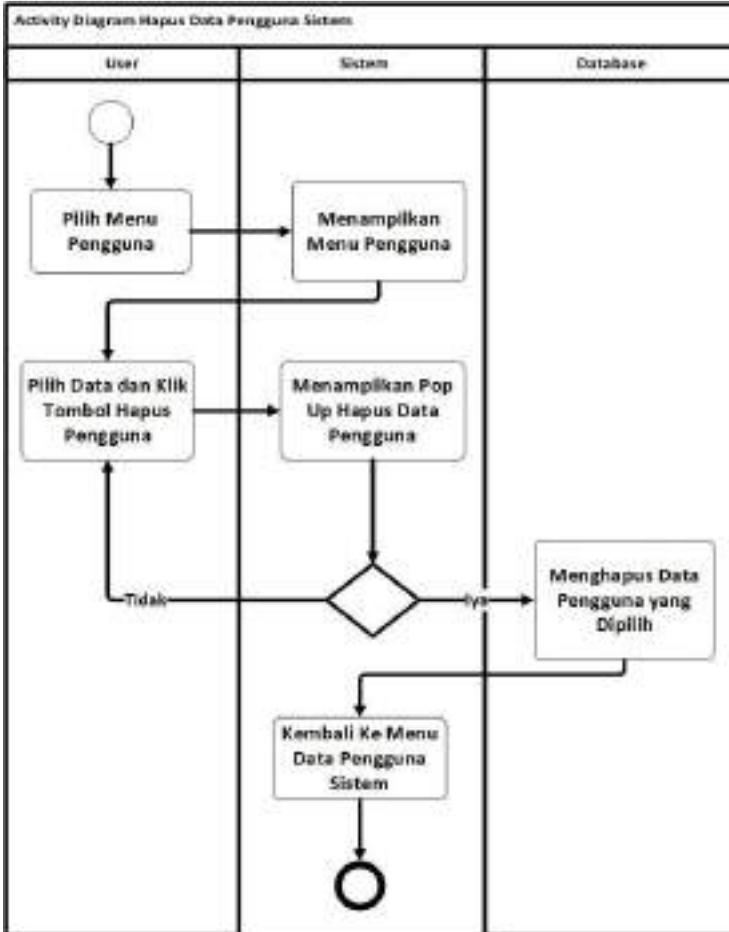
11) *Activity Diagram* Edit Data Pengguna



Gambar 3.16 *Activity Diagram* Edit Data Pengguna

Penjelasan *activity* diagram di atas merupakan alur bagaimana mengubah data pengguna yaitu *user login* dengan memilih menu pengguna kemudian pilih data pengguna yang akan di ubah dengan mengisi form, jika sudah selesai pilih tombol selesai kemudian data tersimpan di database.

12) *Activity Diagram* Hapus Data Pengguna

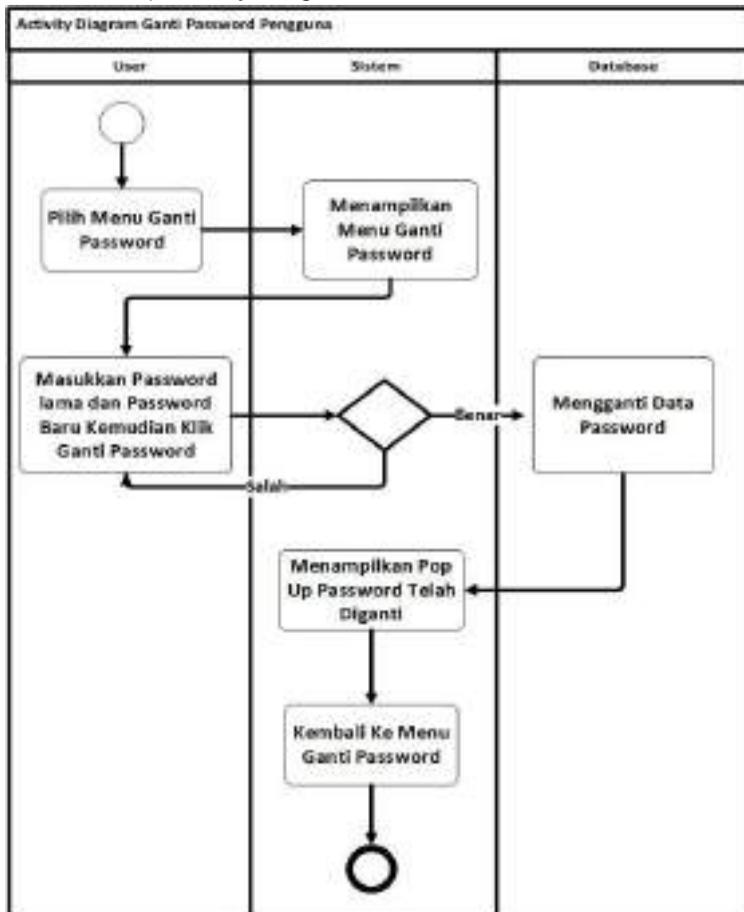


Gambar 3.17 *Activity Diagram* Hapus Data Pengguna

Penjelasan *activity* diagram di atas merupakan alur bagaimana menghapus data pengguna, pertama *user login*

dengan memilih menu pengguna di sistem kemudian pilih data pengguna yang akan di hapus dengan menekan tombol hapus, kemudian data berhasil tersimpan dan terhapus dari database.

13) Activity Diagram Ganti Password

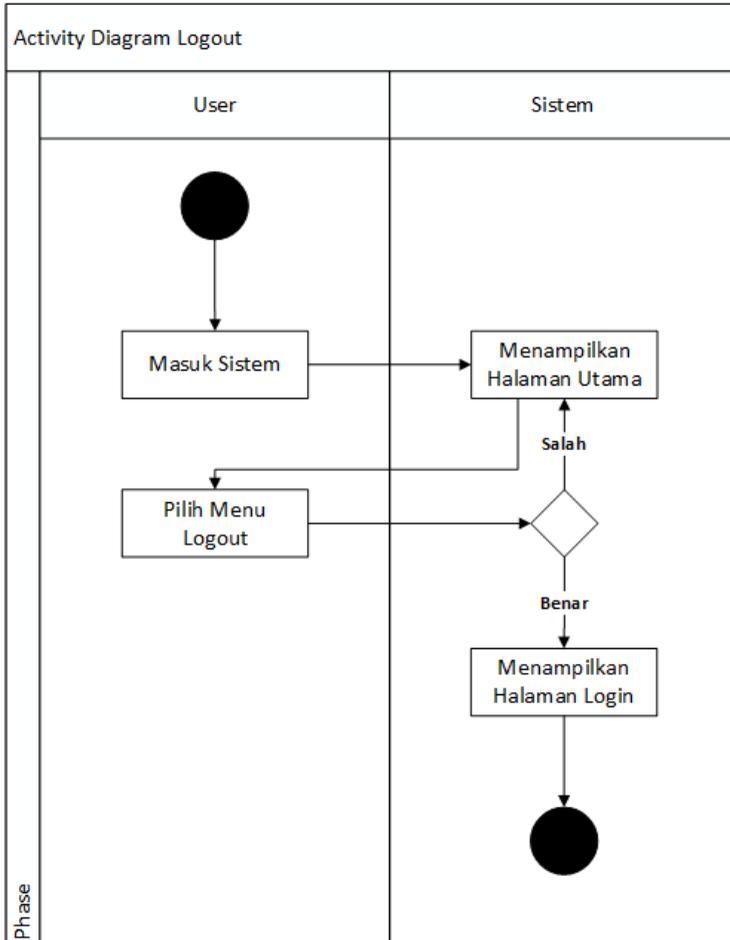


Gambar 3.18 Activity Diagram Ganti Password

Penjelasan *activity* diagram di atas merupakan alur untuk mengganti password yang di lakukan oleh semua *user*.

Pertama, *user login* ke sistem dan memilih menu ganti password dan mengisi password dulu dan password baru mengkonfirmasi password baru lalu pilih tombol simpan kemudian data tersimpan di database.

14) Activity Diagram Logout

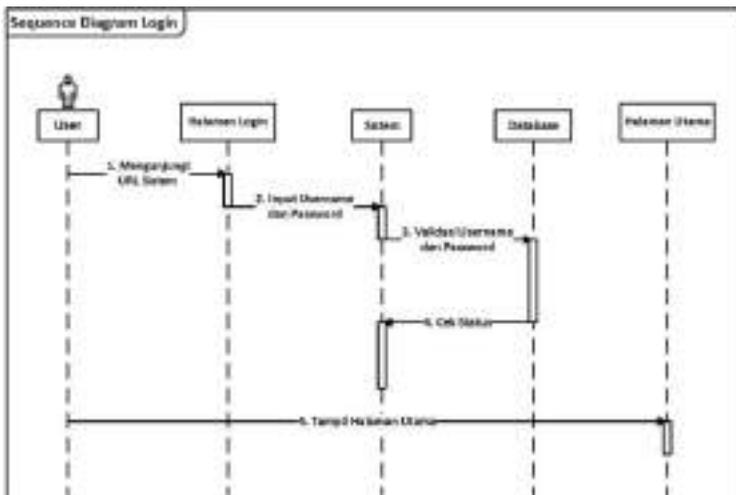


Gambar 3.19 Activity Diagram Logout

Penjelasan *activity* diagram diatas merupakan alur *user logout* dari sistem yaitu *user login* kemudian pilih menu *logout* jika berhasil maka akan menampilkan halaman *login*.

C. Sequence Diagram

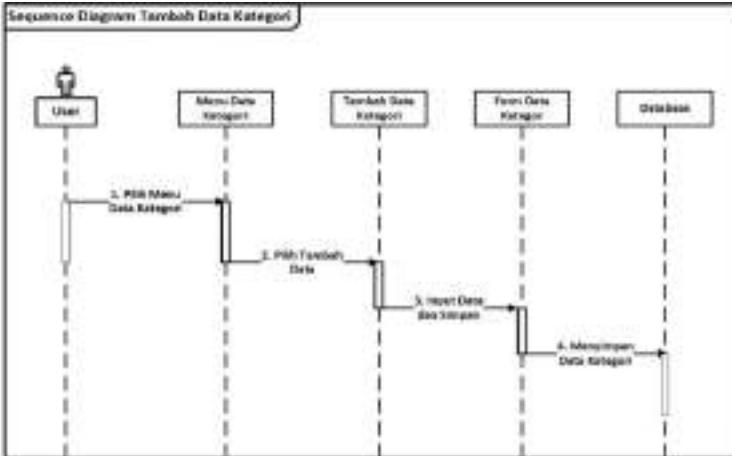
1) Sequence Diagram Login



Gambar 3.20 Sequence Diagram Login

Sequence diagram diatas menjelaskan bagaimana proses *login* dari sistem yaitu aktor mengunjungi url sistem kemudia menginputkan *username* dan *password*, jika benar maka akan masuk ke halaman utama tapi jika salah *username* maupun *password* maka akan kembali ke halaman *login*.

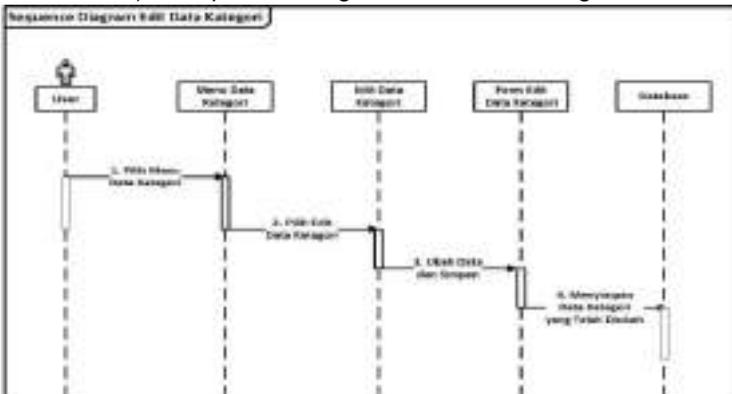
2) Sequence Diagram Tambah Data Kategori



Gambar 3.21 Sequence Diagram Tambah Data Kategori

Proses diatas menjelaskan bagaimana alur bendahara menambah data kategori yaitu bendahara *login* kemudian pilih menu data kategori pilih tombol tambah data, lalu mengisi form tambah kategori, jika selesai pilih simpan dan data akan tersimpan di database.

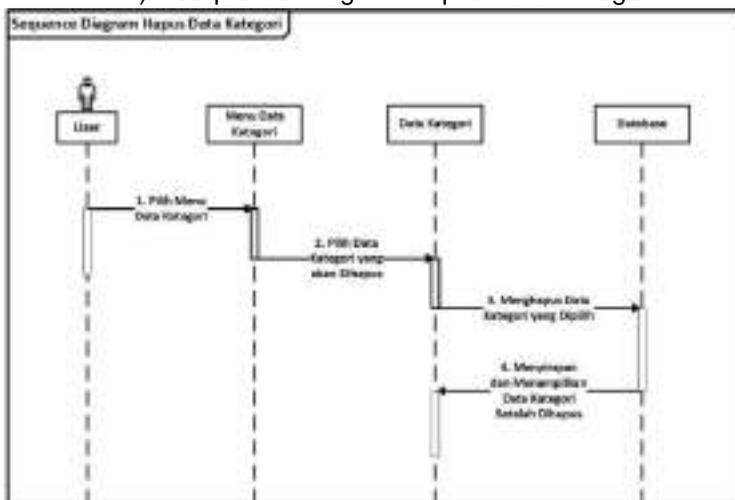
3) Sequence Diagram Edit Data Kategori



Gambar 3.22 Sequence Diagram Edit Data Kategori

Proses diatas menjelaskan alur tentang bagaimana mengubah data kategori yaitu bendahara *login* kemudian pilih menu data kategori dan pilih tombol edit dengan mengisi form data kategori kemudian pilih tombol simpan dan data akan tersimpan di database.

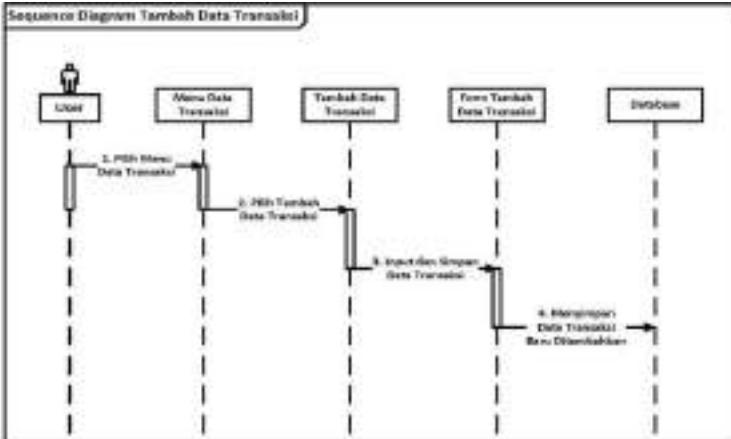
4) Sequence Diagram Hapus Data Kategori



Gambar 3.23 Sequence Diagram Hapus Data Kategori

Proses diatas menjelaskan bagaimana alur untuk menghapus data yang dilakukan oleh bendahara. *User login* ke sistem dan memilih menu data kategori kemudian pilih data kategori yang ingin di hapus dan tekan tombol hapus, kemudian data akan terhapus dari database.

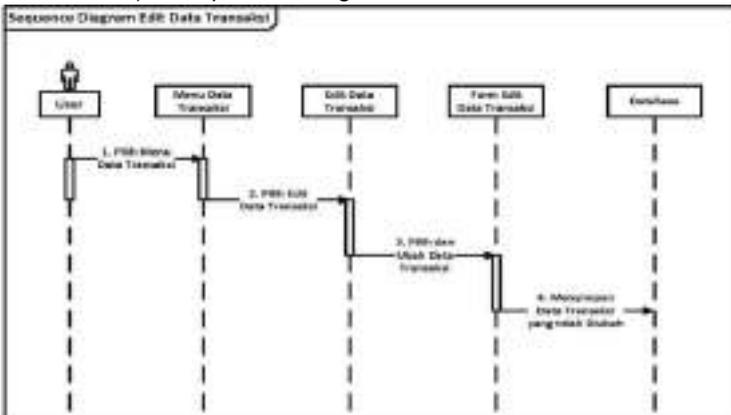
5) Sequence Diagram Tambah Data Transaksi



Gambar 3.24 Sequence Diagram Tambah Data Transaksi

Proses diatas menjelaskan bagaimana alur bendahara menambah data transaksi yaitu bendahara *login* kemudian pilih menu data transaksi pilih tombol tambah data transaksi, lalu mengisi form tambah data, jika selesai pilih simpan dan data akan tersimpan di database.

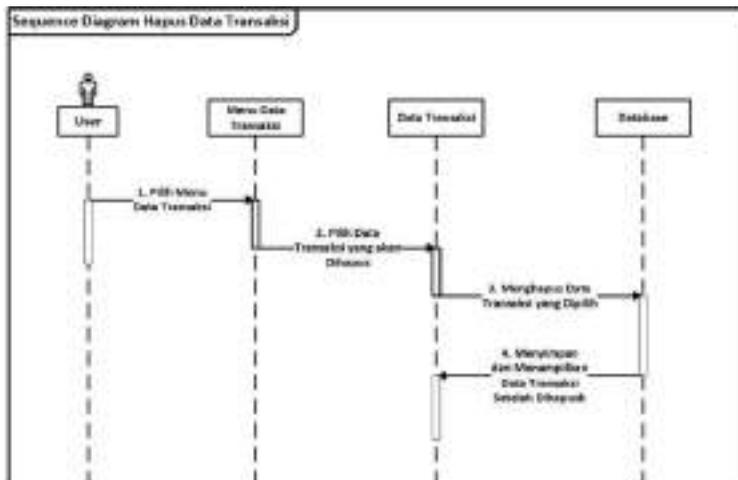
6) Sequence Diagram Edit Data Transaksi



Gambar 3.25 Sequence Diagram Edit Data Transaksi

Proses diatas menjelaskan alur tentang bagaimana mengubah data transaksi yaitu bendahara *login* kemudian pilih menu data transaksi dan pilih tombol edit dengan mengisi form data transaksi kemudian pilih tombol simpan dan data tersimpan di database.

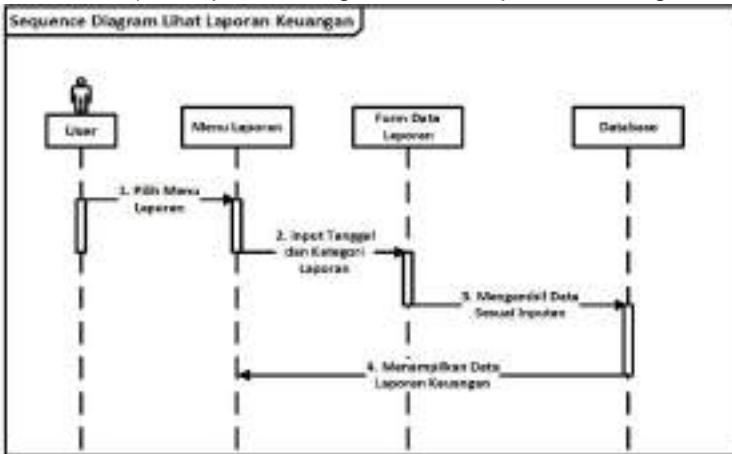
7) Sequence Diagram Hapus Data Transaksi



Gambar 3.26 Sequence Diagram Hapus Data Transaksi

Proses diatas menjelaskan bagaimana alur untuk menghapus data transaksi yang dilakukan oleh bendahara. *User login* ke sistem dan memilih menu data transaksi kemudian pilih data transaksi yang ingin di hapus dan tekan tombol hapus, kemudian data terhapus dari database.

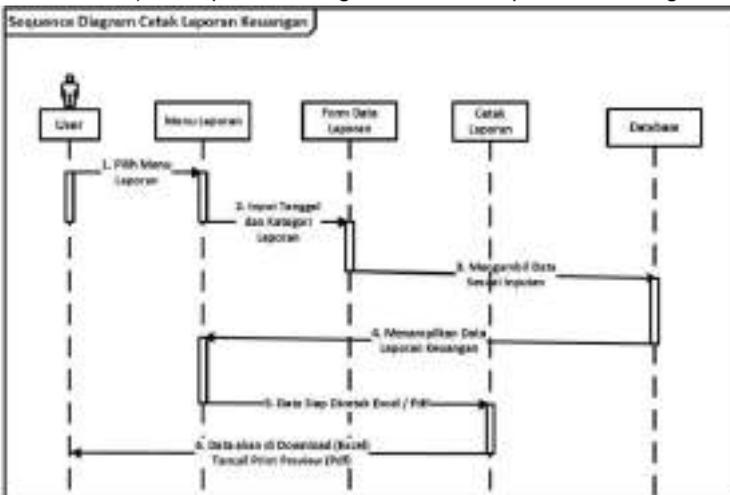
8) Sequence Diagram Lihat Laporan Keuangan



Gambar 3.27 Sequence Diagram Lihat Laporan Keuangan

Proses diatas menjelaskan bagaimana alur untuk melihat data keuangan yang berjalan. *User login* ke sistem dan memilih menu laporan kemudian isikan tanggal dan kategori, data keuangan yang tersimpan di database akan dimunculkan.

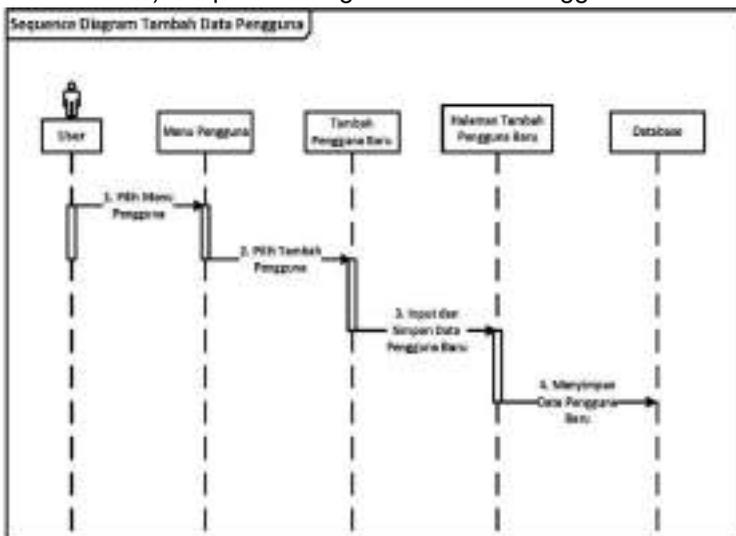
9) Sequence Diagram Cetak Laporan Keuangan



Gambar 3.28 Sequence Diagram Cetak Laporan Keuangan

Proses diatas menjelaskan tentang bagaimana alur melihat laporan dan cetak laporan yang dilakukan bendahara & pengasuh, pertama *user login* ke sistem dengan memilih menu laporan lalu mengisikan tanggal dan kategori yang menampilkan detail laporan, dan jika *user* ingin mencetak lalu pilih tombol cetak.

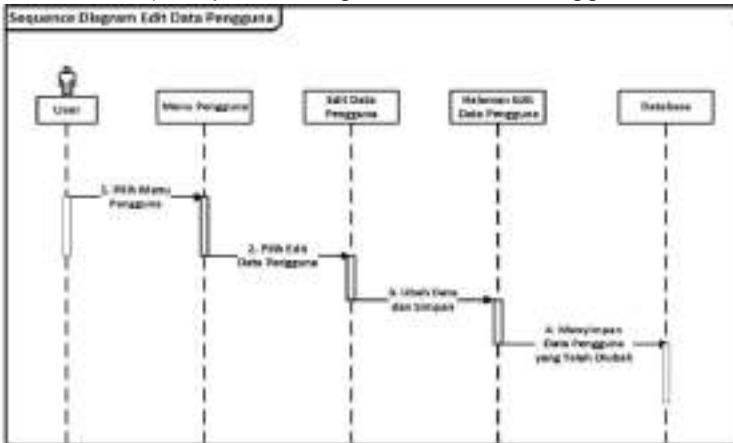
10) Sequence Diagram Tambah Pengguna Baru



Gambar 3.29 Sequence Diagram Tambah Data Pengguna

Proses diatas menjelaskan bagaimana alur admin menambah pengguna yaitu bendahara *login* kemudian pilih menu pengguna pilih tombol tambah data pengguna, lalu mengisi form tambah pengguna, jika selesai pilih simpan dan data akan tersimpan di database.

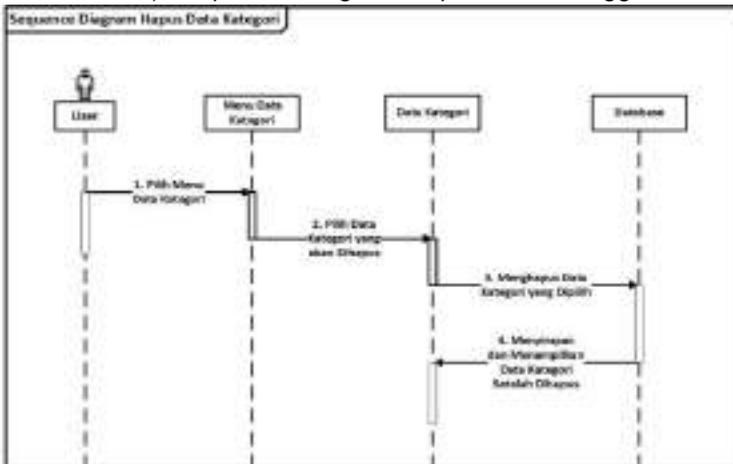
11) Sequence Diagram Edit Data Pengguna



Gambar 3.30 Sequence Diagram Edit Data Pengguna

Proses diatas menjelaskan bagaimana alur admin mengubah data pengguna yaitu bendahara *login* kemudian pilih menu pengguna pilih tombol *edit* data pengguna, lalu mengisi form *edit* data, jika selesai pilih simpan dan data akan tersimpan di database..

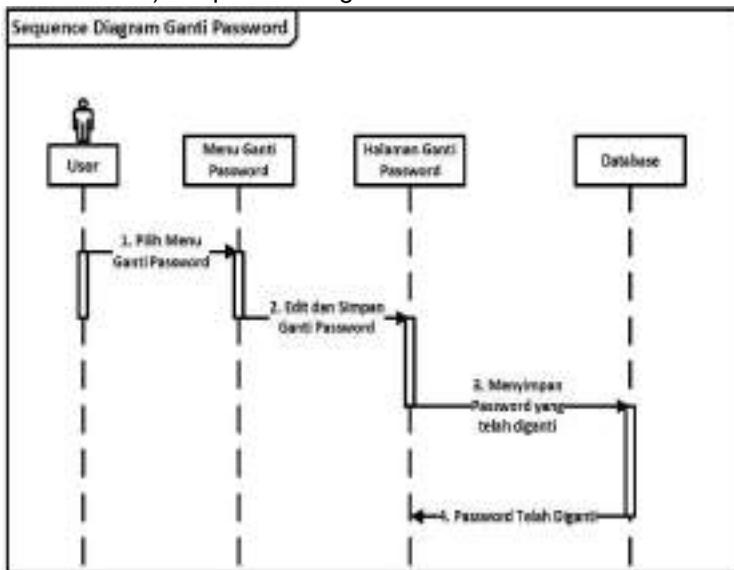
12) Sequence Diagram Hapus Data Pengguna



Gambar 3.31 Sequence Diagram Hapus Data Pengguna

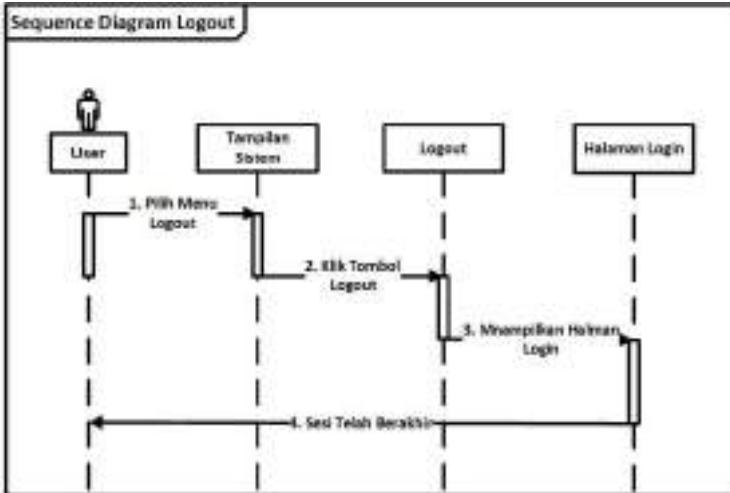
Proses diatas menjelaskan bagaimana alur admin menghapus data pengguna yaitu bendahara *login* kemudian pilih menu pengguna pilih tombol hapus data pengguna, lalu pilih data pengguna yang akan dihapus, lalu kilik hapus maka data akan dihapus dari database.

13) Sequence Diagram Ganti Password



Gambar 3.32 Sequence Diagram Ganti Password

Proses diatas menjelaskan bagaimana alur untuk menghapus data yang dilakukan oleh bendahara. *User login* ke sistem dan memilih menu pemasukan kemudian pilih data pemasukan yang ingin di hapus dan tekan tombol hapus, kemudian data terhapus dari database.

14) Sequence Diagram *Logout*

Gambar 3.33 Sequence Diagram Logout

Proses diatas menjelaskan alur tentang alur *logout* yang bisa dilakukan oleh semua *user*. Setelah *login user* dapat memilih menu *logout*. Jika sudah maka sistem akan menampilkan halaman *login*.

3.3.6 Sprint 2 (dua)

Pada *Sprint 2* (dua) ini akan dijelaskan daftar pekerjaan yang akan dikerjakan berdasarkan daftar *product backlog* yang telah di buat dan hasil pertimbangan dari *Sprint 1* (satu). Berikut adalah *scrum event* pada *Sprint 2* (dua):

1. *Sprint Planing*

Pada perancangan *Sprint 2* (dua), dihasilkan keputusan sebagai berikut:

- a) Panjang *Sprint*. 1 Minggu.
- b) Tujuan: Membuat rancangan database dan rancangan UI

- c) Dalam menentukan *story point* yang akan dimasukkan ke dalam *Sprint*, hal pertama yang harus dilakukan adalah memperkirakan kecepatan tim. Dari perkiraan kecepatan tim maka *Product Backlog* yang dapat dimasukkan ke dalam *Sprint* sebanyak 6 *story point*. Berikut *Sprint Backlog* pada *Sprint 2* (dua) dan penjabaran *task* dari masing-masing *Sprint Backlog* pada tabel di bawah ini:

Tabel 3.13 *Sprint 2* (dua)

<i>Sprint Backlog</i>	<i>Task</i>	<i>Est. (Story Point)</i>
Konfigurasi Database	<ul style="list-style-type: none"> • Membuat Class <i>Diagram</i> • Membuat Desain Basis Data 	1
Rancangan UI Aplikasi	<ul style="list-style-type: none"> • Pembuatan Rancangan UI 	5

2. *Daily Scrum*

Pada saat *Sprint* dijalankan, pertemuan harian *Scrum* diadakan hampir setiap hari. Pertemuan ini berlangsung selama kurang lebih 15 menit. Dalam pertemuan ini, tim membahas apa saja yang sudah diselesaikan dengan melihat *product backlog* dan *task* yang sudah di selesaikan. Selain itu juga membahas apa yang akan dikerjakan selanjutnya.

3. *Sprint Review*

Dalam pertemuan evaluasi *Sprint 2* (dua) membicarakan mengenai *Sprint Backlog* yang sudah diselesaikan oleh tim. Dari semua *Sprint Backlog* yang sudah dikerjakan, semua dinyatakan selesai karena sesuai dengan



hasil yang diharapkan. Hasil dari *Sprint 2* (dua) dijelaskan pada *point* selanjutnya. Selain itu, pada pertemuan ini membahas *product Backlog* yang tersisa. Dari hasil pertemuan evaluasi *Sprint 2* (dua), tidak ada revisi untuk *product Backlog* yang tersisa.

4. *Sprint Retrospective*

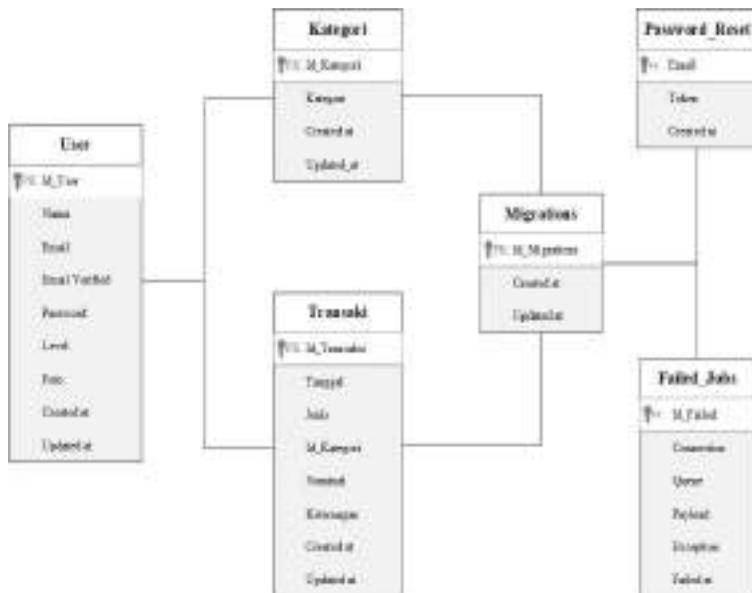
Dalam pertemuan evaluasi *Sprint* dilakukan evaluasi kinerja tim selama *Sprint 2* (dua). Perkiraan kecepatan tim pada *Sprint 2* (dua) sudah tepat. Hal ini dibuktikan dengan semua *Sprint Backlog* pada *Sprint 2* (dua) telah berhasil dikerjakan oleh tim. Kecepatan kinerja tim stabil. Hal ini dapat dilihat dari kekompakan tim yang menunjukkan *estimated work remaining (story point)* yang setiap harinya berkurang antara 1 (satu) atau 2 (dua) *story point*. Untuk *Sprint* selanjutnya diharapkan tim dapat mempertahankan stabilitas kecepatan kinerja tim dan dapat mengerjakan semua *Sprint Backlog*.

Berikut adalah hasil dari *sprint 2* (dua) yaitu rancangan database, dan rancangan UI :

A. Class Diagram

Pada dasarnya hanya ada satu class diagram. Namun apabila terlalu besar class diagram dapat dipecah sesuai kebutuhan asal tidak mengubah maksud dari relasi antar class diagram berfungsi untuk menjelaskan tipe dari objek sistem beserta hubungannya dengan objek yang lain. Objek merupakan nilai dari setiap atribut kelas maupun entitas. Class sendiri menggambarkan keadaan atribut suatu sistem dan bisa untuk memanipulasi keadaan tersebut yang disebut dengan istilah metode (*methode*) atau fungsi (*function*) Class diagram pada sistem ini bisa dilihat pada gambar 3.34





Gambar 3.34 Class Diagram

B. Desain Basis Data

Dari sistem yang akan dibangun ini dapat disimpulkan bahwa data-data yang akan digunakan dalam membangun Sistem Informasi Administrasi Keuangan Asrama XXVII Al Furqon Pondok Pesantren Darul Ulum Jombang yaitu desain perancangan basis data berbentuk tabel-tabel yang di kelompokkan berdasarkan fungsi masing-masing basis data. Desain basis data Sistem Informasi Administrasi Keuangan Asrama XXVII Al Furqon Pondok Pesantren Darul Ulum Jombang dapat dilihat pada tabel di bawah ini :

1. Tabel User

Tabel 3.14 Desain Basis Data Tabel User

Nama	Type	Keterangan
Id_user	BIGINT (20)	Primary_Key
Nama	VARCHAR (255)	
Email	VARCHAR (255)	
Email Verified	TIMESTAMP	
Password	VARCHAR (255)	
Level	ENUM	
Foto	VARCHAR (50)	
Created_at	TIMESTAMP	
Updated_at	TIMESTAMP	

2. Tabel Kategori

Tabel 3.15 Desain Basis Data Tabel Kategori

Nama	Type	Keterangan
Id_kategori	BIGINT (20)	Primary_Key
Kategori	VARCHAR (255)	
Created_at	TIMESTAMP	
Updated_at	TIMESTAMP	

3. Tabel Transaksi

Tabel 3.16 Desain Basis Data Tabel Transaksi

Nama	Type	Keterangan
Id_Transaksi	BIGINT (20)	Primary_Key
Tanggal	DATE	
Jenis	ENUM	
Kategori_id	INT (11)	
Nominal	INT (11)	
Keterangan	TEXT	
Created_at	TIMESTAMP	
Updated_at	TIMESTAMP	



4. Tabel Password Reset

Tabel 3.17 Desain Basis Data Tabel Password Reset

Nama	Type	Keterangan
Email	VARCHAR (255)	Primary_Key
Token	VARCHAR (255)	
Created_at	TIMESTAMP	

5. Tabel Failed Jobs

Tabel 3.18 Desain Basis Data Tabel Failed Jobs

Nama	Type	Keterangan
Id_Failed	BIGINT (10)	Primary_key
Connection	TEXT	
Queue	TEXT	
Payload	LONGTEXT	
Exception	LONGTEXT	
Failed_at	TIMESTAMP	

6. Tabel Migrations

Tabel 3.19 Desain Basis Data Tabel Migrations

Nama	Type	Keterangan
Id_migrations	INT (10)	Primary_Key
Migration	VARCHAR (255)	
Batch	INT (10)	

C. Desain User Interface

1) Design User Interface Login

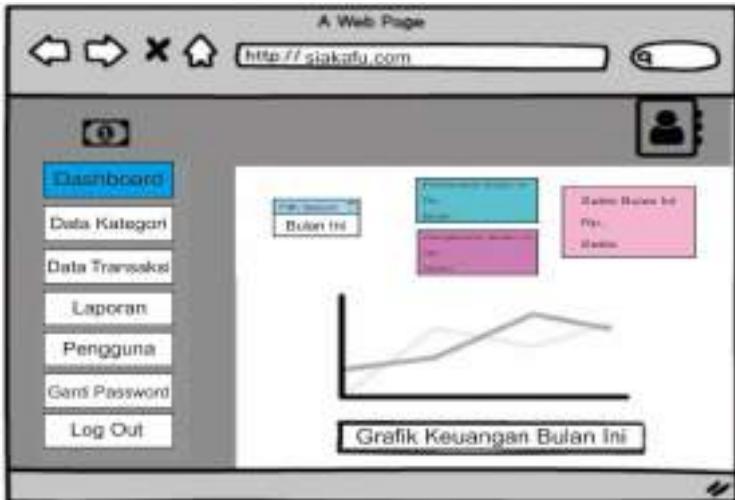
Gambar dibawah merupakan *user interface* dari *login* yang terdiri *form username*, *form password* dan tombol *login*.



Gambar 3.35 Desain User Interface Login

2) Design User Interface Dashboard

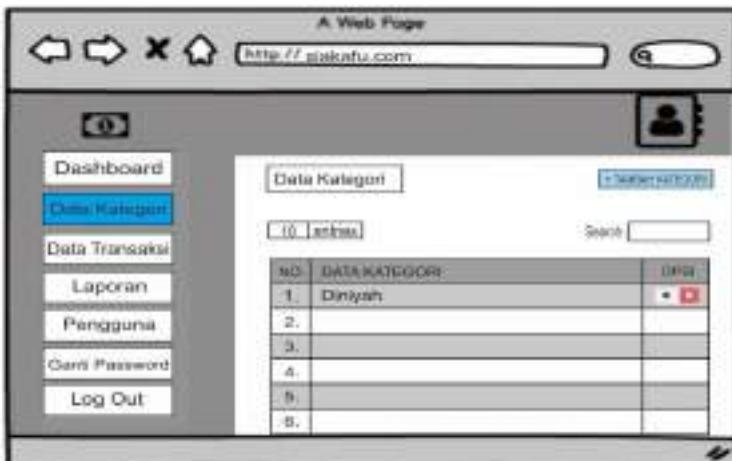
User interface dibawah merupakan *design* dari halaman *dashborad* yang merupakan gambaran tampilan menu utama setelah melakukan *login*, halaman *dashboard* berisi tentang data pemasukan, pengeluaran, dan saldo yang bisa di pilih sesuai *session* yang diinginkan. Terdapat juga grafik keuangan yang menampilkan informasi uang masuk dan uang keluar dalam bentuk hari, bulan, tahun, dan juga per kategori.



Gambar 3.36 Desain User Interface Dashboard

3) Design User Interface Data Kategori

Desain User interface dibawah merupakan halaman menu data kategori yang di gunakan user untuk melihat dan menambahkan kategori keuangan asrama.



Gambar 3.37 Desain User Interface Data Kategori

4) Design User Interface Data Transaksi

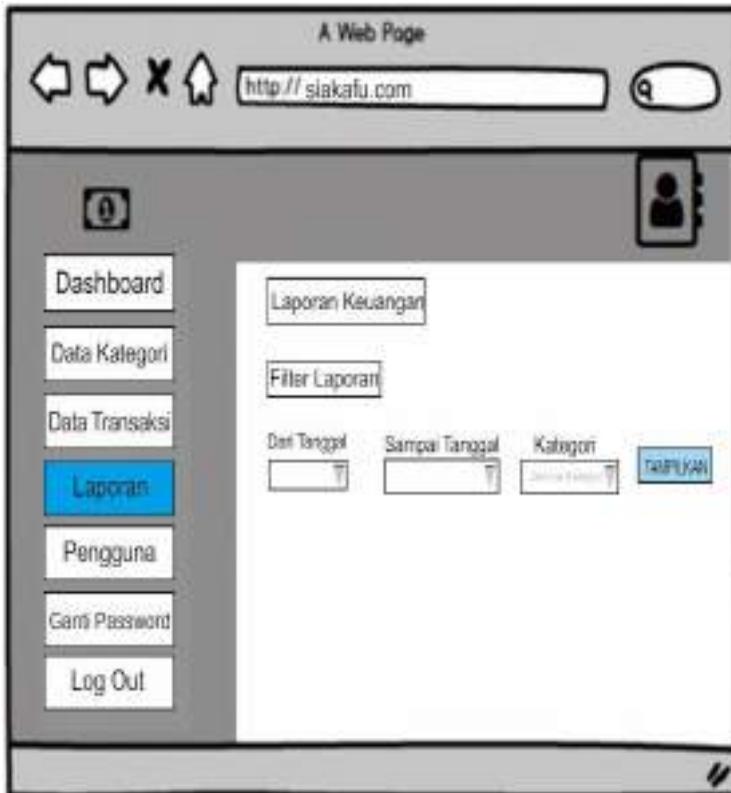
Desain User interface di bawah merupakan halaman menu data transaksi yang di gunakan bendahara untuk menambah, mengubah dan menghapus data keuangan.



Gambar 3.38 Desain User Interface Data Transaksi

5) *Design User Interface Laporan*

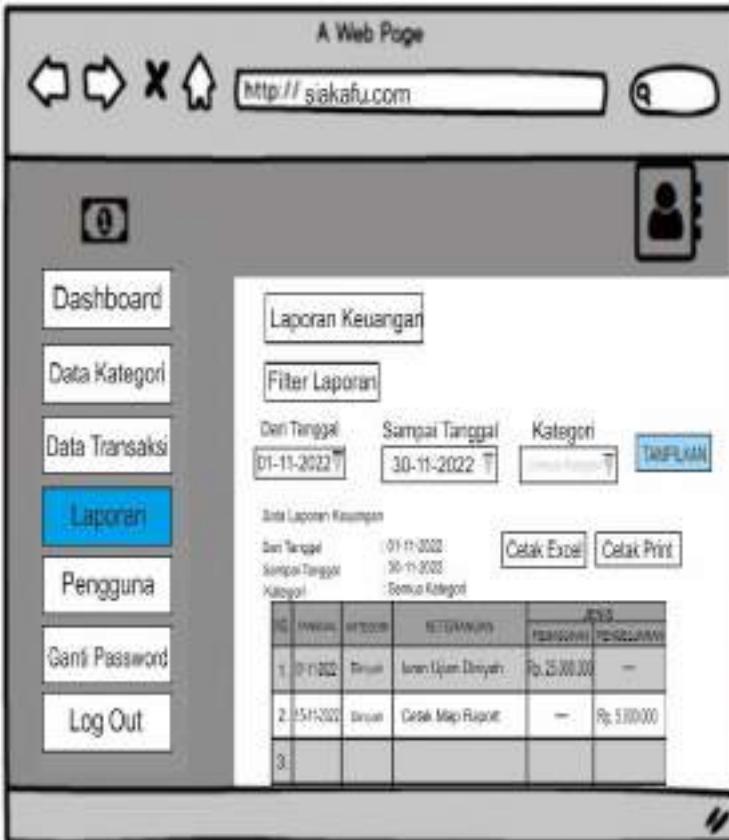
Desain User interface di bawah merupakan halaman menu laporan keuangan yang di gunakan *user* untuk melihat laporan keuangan berdasarkan tanggal dan juga kategori.



Gambar 3.39 *Desain User Interface Laporan Keuangan*

6) Design User Interface Cetak Laporan

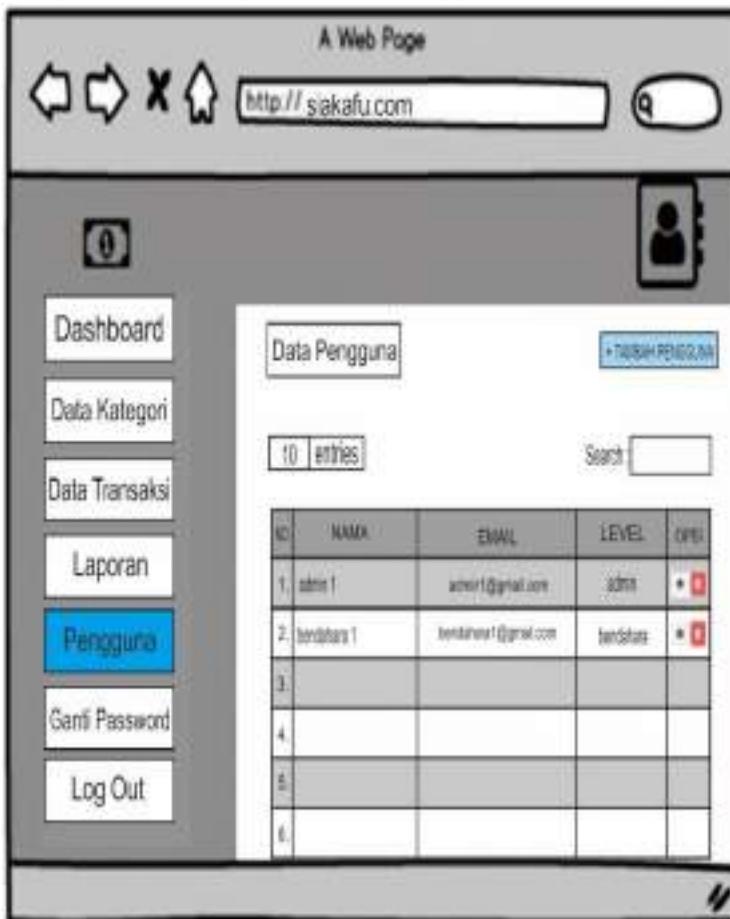
User Interface di atas merupakan halaman laporan yang ada di halaman setiap *user*. Menu halaman laporan tersebut dapat dilakukan pencarian berdasarkan per tanggal dan filter per kategori yang menghasilkan output laporan berupa file XLSX, dan CETAK (Pdf).



Gambar 3.40 Desain User Interface Cetak Laporan Keuangan

7) Design User Interface Pengguna

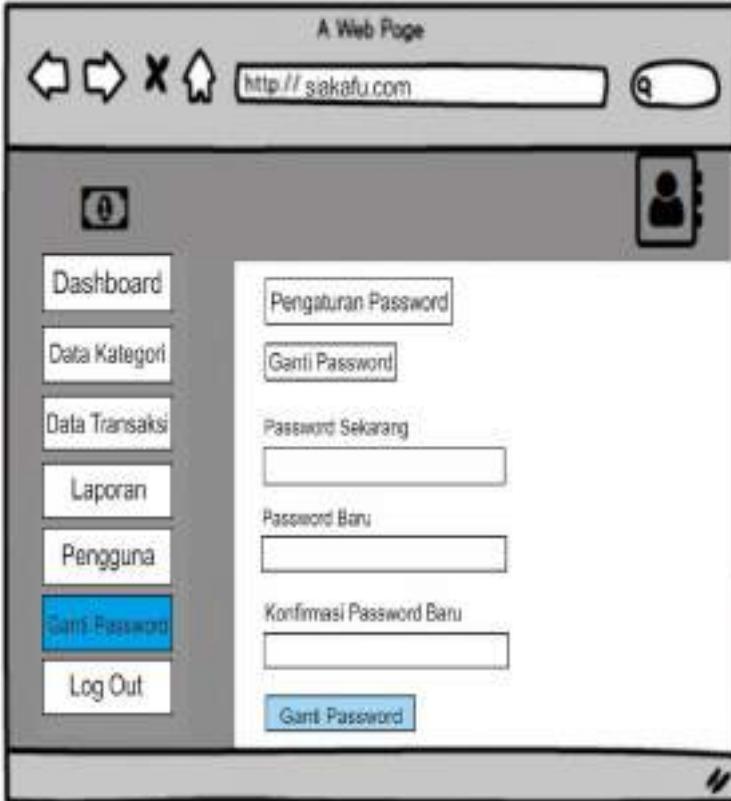
Desain User interface di bawah merupakan halaman menu pengguna yang di gunakan admin untuk menambah, mengubah dan menghapus data pengguna.



Gambar 3.41 Desain User Interface Pengguna

8) Desain User Interface Ganti Password

Desain User interface di bawah merupakan halaman menu ganti password yang di gunakan user untuk mengganti atau mengubah password user.



Gambar 3.42 Desain User Interface Ganti Password