**USULAN TUGAS AKHIR**

**RANCANG BANGUN SISTEM INFORMASI**

**KEUANGAN DESA DORIDUNGGA KECAMATAN DONGGO KABUPATEN BIMA BERBASIS WEB MENGGUNAKAN METODE EXTREME PROGRAMMING**

**A close up of a logo

Description automatically generated**

**Oleh:**

**M. Azhar**

**F1D016051**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MATARAM**

**2020**

## HALAMAN PENGESAHAN

**RANCANG BANGUN SISTEM INFORMASI DESA DORIDUNGGA KECAMATAN DONGGO KABUPATEN BIMA MENGGUNAKAN METODE EXTREME PROGRAMMING**

Telah diperiksa dan disetujui oleh Tim Pembimbing :

1. Pembimbing Utama

**Nadiyasari Agitha, S.Kom., M.M.T.** Tanggal: November 2021

## NIP: 19860813 201803 2 001

1. Pembimbing Pendamping

**Moh. Ali Albar, S.T., M.Eng.** Tanggal: Desember 2021

## NIP. 19831125 201504 1 002

Mengetahui,

Sekretaris Program Studi Teknik Informatika Fakultas Teknik

Universitas Mataram

## Andy Hidayat Jatmika, S.T., M.Kom.

**NIP: 19831209 201212 1 001**

# ABSTRAK

Teknologi informasi merupakan istilah umum untuk teknologi apapun yang membantu manusia dalam membuat, mengubah, menyimpan, mengkomunikasikan dan menyebarkan informasi. Kemajuan yang pesat dibidang teknologi informasi telah memberikan efisiensi pada manusia hampir disetiap aspek kehidupan. Saat ini, kecanggihan teknologi informasi dimanfaatkan diantaranya untuk mempermudah berkomunikasi dan mendapatkan informasi. Desa Doridungga merupakan salah satu desa yang berada di Kecamatan Donggo, Kabupaten Bima yang sistem pengolahan keuangannya masih menggunakan cara manual. Untuk mendapatkan informasi terkait keuangan desa, masyarakat setempat harus berjalan ke kantor desa, atau mereka harus melihat spanduk-spanduk yang dipasang di samping jalan di desa setempat. Hal tersebut dinilai tidak efisien karena memakan waktu yang cukup banyak hanya untuk mendapatkan informasi. Apalagi jika masyarakat desa setempat berada diluar daerahDalam penelitian tugas akhir ini menggunakan metode *Extreme Programming*.

**Kata Kunci:** Keuangan, Sistem Informasi, Website, *Extreme Programming*

# DAFTAR ISI

**HALAMAN PENGESAHAN ii**

**ABSTRAK iii**

**DAFTAR ISI iv**

**DAFTAR GAMBAR vi**

**DAFTAR TABEL viii**

**BAB I PENDAHULUAN 1**

1. LATAR BELAKANG 1
2. RUMUSAN MASALAH 3
3. BATASAN MASALAH 3
4. TUJUAN 3
5. MANFAAT 3
6. SISTEMATIKA PENULISAN 3

**BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI 5**

1. TINJAUAN PUSTAKA 5
2. DASAR TEORI 7
3. E-*Government* 7
4. Sistem Informasi 8
5. *Website* 8
6. *Codeigniter* 8
7. PHP 9
8. XAMPP 9
9. *Apache* 9
10. *Web server* 9
11. PHP *MyAdmin* 9
12. *MySQL* 10
13. *Personal Extreme Programming* 10

**BAB III METODE PENELITIAN 17**

1. ALAT DAN BAHAN 17
2. Alat 17
3. Bahan 17
4. DIAGRAM ALIR PENELITIAN 18
5. Studi Literatur 18
6. *Planning* 18
7. *Design* 26
8. *Usecase* Diagram 26
9. *Class Responsibilities Collaboration Card* 27
10. *Class* Diagram 28
11. *Activity* Diagram 30
12. *Sequence* Diagram 35
13. *Entity Relationship* Diagram 40
14. Desain *Interface* 41
15. *Coding* 45
16. *Testing* 46
17. *Release* 47

**DAFTAR PUSTAKA 48**

# DAFTAR GAMBAR

Gambar 2.1 Fishbone Diagram 8

Gambar 3.1 Diagram alir penelitian 19

Gambar 3.2 *Usecase* Diagram 36

Gambar 3.3 *Class* diagram pada web 39

Gambar 3.4 *Activity* diagram proses *login* admin 41

Gambar 3.23 *Activity* diagram proses menambah anggaran dana desa 53

Gambar 3.24 *Activity* diagram proses mengubah anggaran dana desa 54

Gambar 3.25 *Activity* diagram proses menghapus anggaran dana desa 55

Gambar 3.26 *Activity* diagram proses menambah berita 55

Gambar 3.27 *Activity* diagram proses mengubah berita 56

Gambar 3.28 *Activity* diagram proses menghapus berita 57

Gambar 3.29 *Activity* diagram proses mengajukan pengaduan masyarakat 57

Gambar 3.30 *Sequence* diagram proses *login* admin 58

Gambar 3.31 *Sequence* diagram proses menambah sejarah desa 59

Gambar 3.32 *Sequence* diagram proses mengubah sejarah desa 59

Gambar 3.33 *Sequence* diagram proses menghapus sejarah desa 60

Gambar 3.34 *Sequence* diagram proses pengaduan masyarakat 61

Gambar 3.35 *Entity Relationship* Diagramsystem 62

Gambar 3.36 Rancangan tampilan menu beranda pada web 65

Gambar 3.37 Rancangan tampilan login 66

Gambar 3.38 Rancangan tampilan menu profil desa 66

Gambar 3.39 Rancangan tampilan menu pemerintah desa 67

Gambar 3.40 Rancangan tampilan menu data desa 67

Gambar 3.41 Rancangan tampilan menu anggaran dana desa 68

Gambar 3.42 Rancangan tampilan menu berita 68

Gambar 3.43 Arsitektur Sistem 69

Gambar 3.44 Alur kerja TDD 70

# DAFTAR TABEL

Tabel 2.1 *Twelve Core Practices Compared* 11

Tabel 3.1 *User story cards* 21

Tabel 3.2 Pembagian iterasi *story* 34

Tabel 3.3 Penjelasan CRC-*Card* 38

Table 3.4 Tabel Admin 63

Table 3.5 Tabel Profil Desa 63

Table 3.6 Tabel Mengelola Profil Desa 63

Table 3.7 Tabel Pemerintah Desa 63

Table 3.8 Tabel Mengelola Pemerintah Desa 63

Table 3.9 Tabel Data Desa 63

Table 3.12 Tabel Mengelola Anggaran Dana Desa 64

# BAB I PENDAHULUAN

## Latar Belakang

Teknologi informasi yaitu meliputi perangkat keras, lunak untuk menjalankan satu atau sejumlah tugas pemprosesan data misalnya menangkap, mentransmisikan, menyimpan, mengambil, memanipulasi atau menampilkan data, segala jenis teknologi yang mampu mempores atau mengirim informasi dalam bentuk elektronik, seperangkat alat memperlancar penataan tugas-tugas yang berhubungan dengan pemproses data, informasi dan komunikasi, dan memecahkan masalah, membuka kreatifitas, meningkat efektifitas serta efiesensi dalam melakukan pekerjaan. Kemajuan yang pesat dibidang teknologi informasi telah memberikan efisiensi pada manusia hampir disetiap aspek kehidupan. Saat ini, kecanggihan teknologi informasi dimanfaatkan diantaranya untuk mempermudah berkomunikasi dan mendapatkan informasi[1].

Desa Doridungga merupakan salah satu desa yang berada di Kecamatan Donggo, Kabupaten Bima yang sistem pengolahan keuangannya masih menggunakan cara manual. Untuk mendapatkan informasi terkait keuangan desa, masyarakat setempat harus berjalan ke kantor desa, atau mereka harus melihat spanduk-spanduk yang dipasang di samping jalan di desa setempat. Hal tersebut dinilai tidak efisien karena memakan waktu yang cukup banyak hanya untuk mendapatkan informasi. Apalagi jika masyarakat desa setempat berada diluar daerah.

Berdasarkan hasil wawancara penulis dengan perangkat desa bahwa kondisi yang terjadi di desa adalah belum terdapat sistem informasi yang memuat informasi keuangan desa, seperti anggaran pendapatan dan belanja desa (APBDes). Dari permasalahan yang terjadi, solusi yang tepat yang dapat diambil adalah dengan membuat sebuah sistem informasi desa yang dapat mengolah dan menampilkan data terkait APBDes (berbasis *website*) sehingga dapat diakses oleh masyarakat luas, dengan adanya sistem informasi tersebut harapannya dapat membantu masyarakat luas dalam mengakses informasi terkait keuangan desa. Informasi-informasi terkait keuangan desa yang semulanya hanya dicetak dalam bentuk spanduk dapat dikelola dan ditampilkan dalam bentuk *website*, sehingga publik atau masyarakat luas tidak hanya mengakses melalui spanduk, melainkan bisa melalui *website*.

Dalam penelitian Tugas Akhir ini menggunakan metode *Extreme Programming* atau XP. Pada XP terdapat model Personal *Extreme Programming* atau PXP yang proses pengembangannya dapat dilakukan oleh *programmer* tunggal. Dalam XP menekankan konsep lebih berinteraksi dengan *client* dalam proses pengembangan sistem dan berfokus pada sistem yang dikembangkan, sehingga meminimalisir terjadinya kesalahan pada sistem dan sesuai dengan kebutuhan *client.* Penggunaan metode XP sangat cocok digunakan dalam pengembangan sistem informasi ini dikarenakan *client* sewaktu-waktu menginginkan perubahan-perubahan dari segi fitur dalam sistem informasi karena dalam penggunaan metode XP sendiri membutuhkan lebih banyak interaksi dengan *client*. Selain daripada itu, kelebihan yang dimiliki metode XP bisa dikerjakan oleh tim yang dibentuk dalam skala kecil sampai medium, ukuran tim dibatasi antara tiga dan maksimal dua puluh anggota proyek, dan *pair programming* (dua *programmer* yang melakukan pengkodean di satu komputer) merupakan karakteristik dari XP [2]. Melihat dari karakteristik metode XP, maka sangat cocok digunakan pada penelitian ini karena dilakukan oleh pengembang tunggal dengan model Personal *Extreme Programming* (PXP).Dalam kasus penelitian Tugas Akhir ini pihak pemerintah Desa Doridunggan, Kecamatan Donggo, Kabupaten Bima bertindak sebagai *client*.

Melihat uraian permasalahan di atas, penulis akan merancang dan membangun sebuah sistem informasi yang berjudul “**Rancang Bangun Sistem Informasi Keuangan Desa Doridungga Kecamatan Donggo Kabupaten Bima Berbasis Web**”. Sistem informasi ini dibuat untuk membantu pegawai desa Doridungga Kecamatan Donggo Kabupaten Bima dalam mengolah informasi terkait keuangan desa, seperti anggaran pendapatan belanja desa dengan menggunakan *web*. Dalam penggunaanya, sistem informasi ini dikelola oleh operator desa. Operator desa akan bertindak dalam memasukkan data, merubah, serta menghapus data desa. Diharapkan dengan adanya sistem informasi ini dapat membantu operator desa dalam mengolah informasi terkait keuangan Desa Doridungga Kecamatan Donggo Kabupaten Bima.

## Rumusan Masalah

Berdasarkan uraian dari latar belakang, maka rumusan masalah dalam penelitian ini adalah: “Bagaimana melakukan rancangan sistem informasi berbasis *website* yang dapat membantu pemerintah desa dalam mengolah dan menyajikan informasi terkait keuangan desa?”.

## Batasan Masalah

Adapaun Batasan-batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sistem informasi desa Doridungga kecamatan Donggo kabupaten Bima hanya berbasis *website*.
2. Sistem informasi desa hanya memiliki tiga orang user yaitu super admin, admin dan masyarakat.
3. Sistem informasi hanya memuat keuangan desa seperti anggaran pendapatan dan belanja desa.

## Tujuan Penelitian

Adapun tujuan dari penulisan tugas akhir ini adalah untuk melakukan rancangan sistem informasi berbasis *website* yang dapat membantu pemerintah desa dalam mengolah dan menyajikan informasi terkait keuangan desa seperti anggaran pendapatan dan belanja desa dengan memakai bahasa pemrograman PHP, *framework* *Codeigniter* dengan metode *Extreme Programming*.

## Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Memudahkan pemerintah desa dalam mengolah informasi terkait keuangan desa seperti anggaran pendapatan dan belanja desa.
2. Memudahkan masyarakat dalam mengakses informasi terkait keuangan desa seperti anggaran pendapatan belanja desa.

## Sistematika Penulisan

Sistematika penulisan tugas akhir adalah sebagai berikut:

## Bab 1 Pendahuluan

## Bab ini mengemukakan latar belakang, rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan yang digunakan untuk menyusun laporan tugas akhir.

## Bab 2 Tinjauan Pustaka dan Dasar Teori

## Bab ini berisi dasar teori yang digunakan dalam melakukan analisis, perencangan dan implementasi tugas akhir yang dilakukan pada bab bab selanjutnya.

## Bab 3 Metodologi Perancangan

## Bab ini berisi tentang pelaksanaan, alat, bahan, jalannya perencanaan, dan hasil yang diharapkan.

## Bab 4 Hasil dan Pembahasan

## Bab ini memuat tentang analisis dan pembahasan hasil penelitian yang telah dilakukan.

## Bab 5 Penutup

## Bab ini berisi kesimpulan dan saran-saran berdasarkan hasil pembahasan yang diproleh.

# BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

1. Tinjauan Pustaka

Pembuatan sistem informasi desa telah dilakukan oleh peneliti terdahulu. Beberapa sistem informasi yang dibuat oleh peneliti terdahulu adalah sebagai berikut:

Penelitian ini berjudul “Sistem Informasi Pelayanan Distribusi Keuangan Desa Untuk Pembangunan (Study Kasus : Dusun Srikaya)” yang bertujuan untuk membantu sekretaris Desa Sukada Tengah Kecamatan Sukadana Kabupaten Lampung Timur dalam membuat rencana anggaran biaya (RAB). Pembuatan RAB sebelumnya pada desa yang dimaksud adalah dengan menyusunnya pada Ms. Word kemudian mencetaknya lalu diajukan ke kepala desa. Masalah yang terjadi adalah ketika RAB yang diajukan tidak disetujui oleh kepala desa, maka sekretaris desa harus menyusunnya kembali, kemudian mencetak lagi lalu di ajukan kembali. Penggunaan cara seperti itu dinilai tidak efektif karena menghabiskan waktu yang lama dan menghabiskan banyak kertas. Sistem informasi ini memiliki 3 pengguna, yaitu admin, kepala desa, dan masyarkat. Admin bertugas mengelola bidang, mengelola kegiatan, mengelola rincian pendanaan, mengelola laporan kegiatan, mengelola RAB, mengelola kegiatan dan mengelola artikel. Kepala desa bertugas melihat RAB, melihat data kegiatan, dan menampilkan diagram statistik biaya anggaran pertahun. Sedangkan masyarakat dapat melihat komentar artikel, melihat komentar RAB, dan daftar kegiatan[3].

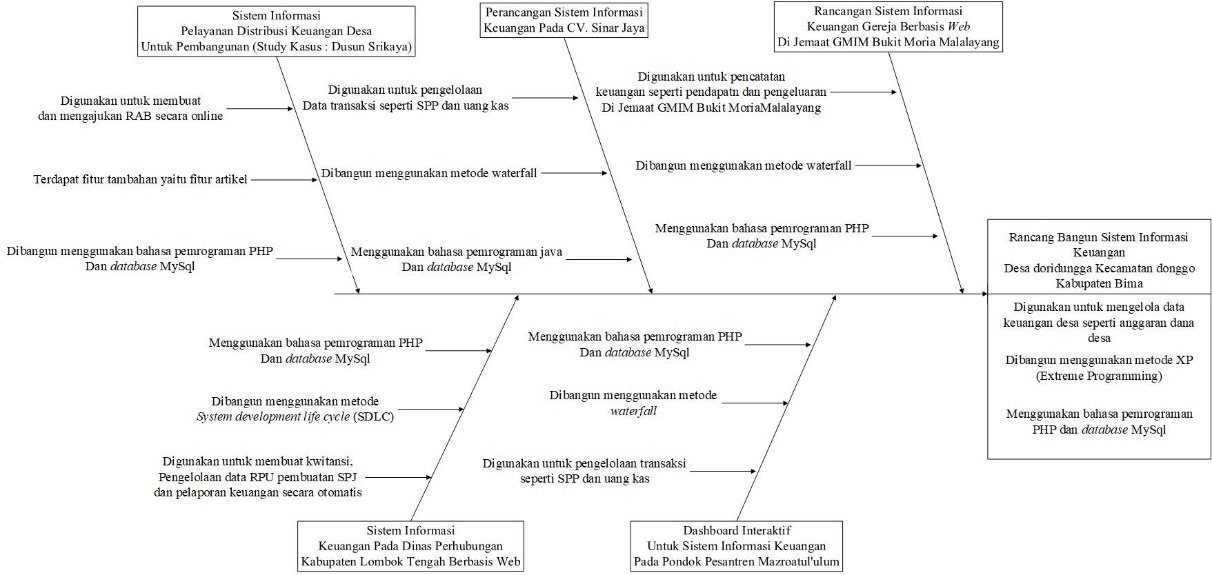
Penelitian yang berjudul “Perancangan Sistem Informasi Keuangan Pada CV. Sinar Jaya” yang bertujuan untuk untuk memudahkan karyawan CV. Sinar Jaya dalam mengolah data dan menghasilkan informasi yang lebih akurat yang sebelumnya pengolahan keuangan CV. Sinar Jaya masih menggunakan cara yang manual dimana data uang masuk dan uang keluar masih menggunakan pencatatan ke dalam buku. Penggunaan cara seperti itu sering terjadinya kesalahan dan sulitnya dalam pencarian maupun dalam menyajikan laporan keuangan. Pengembangan sistem informasi ini menggunakan metode *Waterfall.* Sistem informasi ini memiliki 2 pengguna yaitu admin dan *user*. Admin dapat melakukan *login* ke sistem, kemudian mengolah data user, mengolah data jenis pengiriman dan mengolah data pengiriman, sedangkan *user* dapat melakukan *login*, mengolah data perkiraan, mengolah data jurnal dan mengolah data laporan[4].

Penelitian ini berjudul, “Rancangan Sistem Informasi Keuangan Gereja Berbasis *Web* Di Jemaat GMIM Bukit Moria Malalayang” yang bertujuan untuk membantu proses pendaftaran dan administrasi menjadi lebih efektif dan efisien. Dalam proses pengembangan sistem informasi ini menggunakan metode *Waterfall*. System informasi ini memiliki 1 pengguna yaitu *user* yang dapat mengelola data seperti memasukkan data, memanipulasi data, menghapus data, mencetak laporan (data), dan melakukan transaksi[5].

Penelitian ini berjudul “Sistem Informasi Keuangan Pada Dinas Perhubungan Kabupaten Lombok Tengah Berbasis Web” yang bertujuan untuk membantu pegawai Dinas Perhubungan Kabupaten Lombok Tengah dalam melakukan pengolahan data Rencana Penggunaan Uang (RPU), pembuatan laporan surat pertanggungjawaban belanja, dimana pengolahan data tersebut sebelumnya belum berjalan optimal. Pengolahan data yang dimaksud melibatkan banyak karyawan, sehingga memberikan konsekuensi menambah biaya operasional serta membutuhkan banyak waktu dan dana. Dalam proses pengembangan sistem informasi ini menggunakan metode *System Developmet Life Cycle* (SDLC). System informasi ini memiliki 2 *user* yaitu admin dan kepala dinas Perhubungan Lombok Tengah. Admin bertugas mengelola data pegawai, data Rencana Penggunaan Uang (RPU), data Kwitansi, dan data Surat Tanggung Jawab (SPB). Sedangkan kepala dinas bertugas menanda tangani RPU, Kwitansi dan SPB[6].

Penelitian ini berjudul “Dashboard Interaktif Untuk Sistem Informasi Keuangan Pada Pondok Pesantren Mazroatul'ulum” yang bertujuan untuk membantu pegawai pondok pesantren Mazroatul'ulum dalam menginputkan data santri yang melakukan pembayaran SPP. Dalam mengola data tersebut, sebelumnya pondok pesantren yang dimaksud melakukannya dengan cara konvensional yaitu dengan menuliskannya pada buku. Penggunaan cara tersebut tentunya menghambat proses penyusunan laporan dan rentan terhadap kesalahan penulisan data dan kehilangan data karena hilang atau terkena air. Dalam proses pengembangan system informasi ini menggunakan metode *Waterfall*. Sistem informasi ini memiliki dua *user* yaitu admin dan pimpinan pondok pesantren. Admin bertugas mengelola data santri, memasukkan data pembayaran SPP, mengunggah bukti transfer dan validasi, mengelola kas masuk, mengelola kas keluar, dan mengelola laporan. Pimpinan pondok pesantren bertugas mengecek status pembayaran SPP dan mengelola laporan[7].

Berikut merupakan *fishbone* diagram dari tinjauan pustaka dan usulan sistem yang akan dikembangkan dapat dilihat pada gambar 2.1:



**Gambar 2.1** *Fishbone* Diagram

1. Dasar Teori

Dasar teori yang digunakan oleh penulis dalam penelitian tugas akhir ini adalah sebagai berikut:

### Sistem Informasi

Sistem merupakan kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan dalam usaha mencapai suatu tujuan. Informasi adalah hasil pemrosesan data yang diperoleh dari setiap elemen sistem tersebut menjadi bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan dan dibutuhkan oleh orang untuk menambah pemahamannya terhadap fakta-fakta yang ada. Sistem Informasi juga dapat didefinisiskan sebagai perangkat elemen yang bekerja mengumpulkan, memproses, menyimpan, dan menyebarkan informasi untuk mendukung pengambilan keputusan, koordinasi, pengawasan, analisis, dan visualisasi dalam organisasi[8].

1. ***Website***

*Website* atau situs juga dapat diartikan sebagai kumpulan halaman yang menampilkan informasi dat teks, data gambar diam atau bergerak, data animasi suara, vidio atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman[9].

1. ***Codeigniter***

*CodeIgniter* merupakan framework PHP yang dibuat berdasarkan model view Controller (MVC). CI memiliki library yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh aplikasi berbasis web misalnya mengakses *database*, memvalidasi form sehingga sistem yang dikembangkan mudah. CI juga menjadi satu-satunya *Framework* dengan dokumentasi yang lengkap dan jelas. *Source code* CI yang dilengkapi dengan comment didalamnya sehingga lebih memperjelas fungsi sebuah kode program dan CI yang dihasilkan sangat Bersih (*clean*) dan search *Engine Friendly* (SEF)[10].

1. **PHP**

PHP Merupakan singkatan *recursive* dari PHP : *Hypertext Prepocessor*. Pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1994. PHP merupakan bahasa pemograman untuk membuat web. PHP dapat digunakan untuk membuat halaman *web* yang dinamis. PHP dapat digunakan pada sistem operasi Windows, Mac OS, Linux, dan sistem operasi yang lainnya[11].

1. **XAMPP**

XAMPP adalah perangkat yang menggabungkan tiga aplikasi kedalam satu paket yaitu Apache, MySQL,dan PhpMyAdmin, Dengan Xampp pekerjaan menjadi sangat dimudahkan karena dapat menginstalasi dan mengkonfigurasi ketiga aplikasi tersebut dengan sekaligus dan otomatis. Xampp telah mengalami perkembangan dari waktu ke waktu.versi yang terbaru adalah revisi dari yang terdahulu, sehingga lebih baik dan lebih lengkap. Aplikasi utama dalam paket Xampp yakni terdiri atas *web server* Apache, MySQL, PHP, dan PhpMyAdmin.

1. **Apache**

Apache adalah sebuah *web server open source*, jadi semua orang dapat menggunakannya secara gratis, bahkan anda bisa mengedit kode programnya. fungsi utama dari *apache* yakni menghasilkan halaman *web* yang benar sesuai dengan yang dibuat oleh seorang *web programmer*, dengan menggunakan kode PHP[12].

1. ***Web* *Server***

*Web Server* merupakan sebuah perangkat lunak dalam *server* yang berfungsi menerima permintaan (*request*) berupa halaman web melalui HTTP atau HTTPS dari *klien* yang dikenal dengan *browser web* dan mengirimkan kembali (*response*) hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML[11].

1. **PHPMyAdmin**

PHPMyAdmin merupakan *front-end* MySQL berbasis *web*. PHPMyAdmin dibuat dengan menggunakan PHP. Saat ini, PHPMyAdmin banyak digunakan dalam hampir semua penyedia hosting yang ada di internet. PHPMyAdmin mendukung berbagai fitur administrasi MySQL termasuk manipulasi database, tabel, index dan juga dapat mengeksport data ke dalam berbagai format data[11].

1. **MySQL**

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL adalah sistem manajemen *database* yang sering digunakan bersama PHP. PHP juga mendukung pada Microsoft Access, Database Oracle, dBase, dan sistem manajemen database lainnya. SQL (*Structured Query Language*) adalah bahasa terstruktur yang digunakan secara khusus untuk mengolah database, dan MySQL merupakan sebuah sistem manajemen *database*[12].

## *Personal Extreme Programming*

Personal Extreme Programming atau PXP merupakan metode pengembangan perangkat lunak yang termasuk kedalam *agile software development.* PXP merupakan metode pengembangan dari *Extreme Programming* yang disesuaikan agar dapat dikerjakan oleh pengembang tunggal. Pada Tabel 2.1 menunjukkan pemetaan antara XP dan PXP dalam hal dua belas praktik inti[13].

## Tabel 2.1 *Twelve Core Practices Compared*

|  |  |  |  |
| --- | --- | --- | --- |
| No | *Core Practices* | *Extreme Programming* (XP) | *Personal Extreme Programming* (PXP) |
| 1. | *The Planning Game* | Kegiatan utama dalam *planning game* adalah menulis, estimasi, prioritas atau *user stories* yang dilakukan pengembang *client*. *Client* dan pengembangan berkumpul untuk memutuskan fitur apa dari sistem yang diperlukan yang akan menjadi nilai maksimum bagi bisnis. | Jika pengembang bekerja untuk dirinya sendiri atau cukup tahu untuk menggantikan pelanggan, maka seseorang hanya perlu berganti peran (bermain peran) selama perencanaan. Merupakan praktik yang baik untuk menulis *user stories*, memperkirakan, memprioritaskan, dan melacak saat mengembangkan sendiri. Menulis dan menggunakan cerita adalah cara yang baik untuk memecah proyek besar menjadi bagian-bagian kecil yang lebih mudah ditangani dan dikelola. |
| 2. | *Small Releases* | Sistem sederhana yang berisi serangkaian fitur yang berguna dimasukkan ke dalam produksi lebih awal dan sering diperbarui dalam siklus yang singkat. | Rilis awal dan sering kali dapat diterapkan dengan mudah dengan pemrogram tunggal seperti dengan tim pengembang. |
| 3. | *Metaphor* | Setiap proyek memiliki "nama sistem" dan deskripsi yang membantu memandu proses pengembangan dan komunikasi antara semua pihak. | Tim satu orang dapat menggunakan dan menyempurnakan metafora apa pun yang terbukti paling baik. Selalu pilih metafora yang membantu Anda memahami bagian-bagian sistem yang Anda bicarakan |
| 4. | *Simple Design* | Desain paling sederhana selalu digunakan untuk membangun aplikasi selama memenuhi persyaratan bisnis saat ini. Jangan khawatir tentang persyaratan masa depan karena persyaratan berubah seiring waktu. | Sebagai pengembang tunggal, mudah untuk mempertahankan desain yang sederhana. Solusinya adalah menggunakan desain tingkat tinggi sebagai tujuan untuk dikerjakan, tetapi mengembangkan desain alternatif yang lebih sederhana dan lebih baik tetapi tetap memberikan semua fungsionalitas. |
| 5. | *Testing* | *Test-driven development* adalah salah satu kekuatan utama XP. Perangkat lunak yang dikembangkan dengan XP divalidasi setiap saat. Sebelum fitur baru ditambahkan, tes ditulis untuk memverifikasi perangkat lunak. Perangkat lunak tersebut kemudian dikembangkan untuk lulus tes ini. | Saat menulis modul baru, programmer harus menulis antarmuka terlebih dahulu, kemudian unit test, dan baru kemudian melanjutkan untuk mengimplementasikan modul. Dalam satu orang, setelan tes tim dapat dengan mudah ditulis dan digunakan. Pengembang harus menguji dan mengkompilasi kodenya sebelum memasukkannya ke dalam baseline refactor. Sampai saat itu, kode tetap berada di baseline pengembangan |
| 6. | *Refactoring* | Merupaan teknik untuk meningkatkan desain basis kode yang sudah ada. Esensinya adalah menerapkan serangkaian transformasi kecil yang melestarikan perilaku yang meningkatkan struktur kode. Melakukannya dalam langkah-langkah kecil mengurangi risiko terjadinya kesalahan | Sebagai seorang pemrogram tunggal, seseorang dapat dan harus mempraktikkan pemfaktoran ulang ke hulu kecuali dalam situasi di mana Anda tidak memiliki izin untuk mengubah kode yang tidak Anda miliki. Kode yang difaktorkan ulang harus diintegrasikan dengan baseline produksi dan diuji. |
| 7. | *Pair Programming* | Pemrogram yang menggunakan XP dipasangkan dan menulis semua kode produksi menggunakan satu mesin per pasang. Ini membantu kode untuk terus ditinjau saat sedang ditulis. Pemrograman berpasangan telah terbukti menghasilkan kode berkualitas tinggi dengan sedikit atau tanpa penurunan produktivitas. | Saat bekerja sebagai programmer mandiri, manfaat dari pemrograman berpasangan akan hilang. Anda bisa meminta rekan kerja untuk sering memberi Anda pengingat. Panduan informal dan uji filosofi pertama dapat diterapkan. |
| 8. | *Collective Code Ownership* | Semua kode adalah milik setiap anggota tim, tidak ada satu pun anggota tim yang memiliki sepotong kode dan siapa pun dapat membuat perubahan pada basis kode kapan saja. Ini mendorong semua orang untuk menyumbangkan ide-ide baru ke semua segmen proyek. | Jika Anda adalah satu-satunya pengembang dalam proyek ini maka tidak ada masalah karena Anda memiliki semua kode sumber. Namun, Anda tidak akan mendapatkan keuntungan dari masukan pemrogram lain. Pengembang dapat mempertahankan versi kode dalam repositori dan teknik kontrol versi dapat diterapkan. |
| 9. | *Continuous Integration* | Tujuan dari integrasi berkelanjutan adalah untuk mencegah atau mengurangi penyebaran kode dari basis kode utama; semakin sering kode diintegrasikan ke dalam basis kode utama, semakin kecil kemungkinan terjadinya pengalihan. Sistem perangkat lunak dibangun dan diintegrasikan beberapa kali sehari; setidaknya semua perubahan diintegrasikan ke dalam basis kode utama setidaknya sekali sehari. Setiap build diuji menggunakan kasus uji yang disiapkan. | Jika pengembang bekerja sendiri dengan tidak ada orang lain yang membuat perubahan pada basis kode utama, maka tidak ada masalah karena basis kode berfungsi sebagai catatan linier dari pekerjaan seseorang. Integrasi tidak akan menyebabkan konflik apa pun dan bebas masalah tetapi masih cukup mudah untuk menyimpang dari basis kode utama semakin lama Anda bekerja tanpa mengintegrasikan. Oleh karena itu, integrasi berkelanjutan tetap diperlukan. Cara terbaik untuk mengembangkannya adalah dengan mengerjakan tugas, mengintegrasikan, dan kemudian pindah ke tugas berikutnya. Dengan cara ini divergensi dijaga seminimal mungkin |
| 10. | 40*-Hour Week* | Programmer dalam proyek XP biasanya mematuhi 40 jam kerja seminggu untuk menjaga produktivitas dan menghindari kelelahan | Semua yang diperlukan untuk melakukan ini adalah mematuhi 40 jam seminggu dan berhenti bekerja pada hari yang Anda butuhkan. Artinya berhenti bekerja saat tidak produktif lagi, stres atau lelah untuk mengurangi rasa lelah dan membuat Anda dalam kondisi prima. |
| 11. | *On-site Customer* | Satu atau lebih pelanggan yang akan menggunakan sistem yang dibangun dialokasikan ke tim pengembangan. Pelanggan membantu memandu pengembangan dan diberdayakan untuk memprioritaskan dan menyatakan persyaratan atau menjawab pertanyaan apa pun yang mungkin dimiliki pengembang. Hal ini memastikan bahwa ada komunikasi yang efektif dengan pelanggan dan akibatnya akan lebih sedikit dokumentasi yang diperlukan | Jika Anda adalah pelanggan Anda sendiri (setidaknya pada awalnya) maka bergumam pada diri sendiri adalah hal yang wajar. Jika Anda memiliki pelanggan lain maka komunikasi melalui email atau telepon mungkin akan meredakan masalah ini selama pelanggan terbuka untuk komunikasi. |
| 12. | *Coding Standards* | Setiap orang menggunakan standar pengkodean yang sama yang memudahkan untuk bekerja berpasangan dan berbagi kepemilikan semua kode. Seseorang seharusnya tidak dapat mengetahui siapa yang mengerjakan kode apa dalam proyek XP. | Sebagai tim satu orang, cara Anda memilih untuk membuat kode adalah standar pengkodean Anda. Tetapi beberapa standar pengkodean harus diikuti untuk menjaga konsistensi selama pengembangan |

## BAB III

**METODE PENELITIAN**

1. **Alat dan Bahan**

Dalam penelitian Tugas Akhir ini memerlukan beberapa alat yang terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*softaware*) serta bahan yang berupa data-data dan informasi yang diperlukan.

1. **Alat**

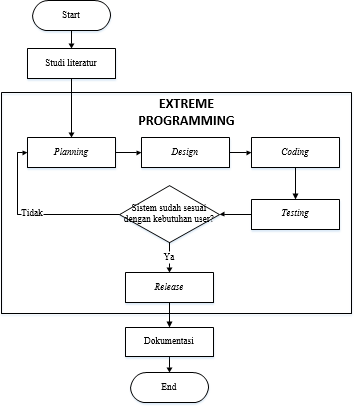
Adapun alat-alat yang digunakan dalam penelitian Tugas Akhir ini dapat dirincikan sebagai berikut.

* 1. Laptop Asus dengan processor Core i3 2.20 GHz HDD 1TB menggunakan OS (operating system) Windows 10 Pro.
  2. *CodeIgniter*, merupakan *framework* bahasa pemrograman PHP yang digunakan untuk membangun sistem berbasis web.
  3. XAMPP, sebagai penyedia *localhost* web *server*.
  4. MySQL, sebagai server *database*.
  5. Sublime text, sebagai text editor pada saat melakukan proses *coding*.
  6. Microsoft Word 2016, digunakan untuk menyusun laporan.
  7. Microsoft Visio 2016, digunakan untuk membuat rancangan diagram.

1. **Bahan**

Adapun sumber data sebagai bahan yang digunakan dalam penelitian Tugas Akhir ini adalah data keuangan desa berupa data pendapatan dan belanja Desa Doridungga Kecamatan Donggo Kabupaten Bima.

1. **Diagram Alir Penelitian**

Pada penelitian Tugas Akhir ini menggunakan metodologi pengembangan sistem *Extreme Programming* yang terdiri dari 4 tahapan yaitu : *planning, design, coding,* dan *testing.* Tahap-tahap pengembangan sistem tersebut diilustrasikan pada diagram alir penelitian pada gambar berikut:

**Gambar 3.1** Diagram alir penelitian[14]

1. **Studi Literatur**

Tahap pertama dalam penelitian Tugas Akhir ini adalah studi literatur. Studi literatur dilakukan dengan mengumpulkan data dan memahami teori-teori dari berbagai buku, skripsi, jurnal dan penelitian-penelitian sebelumnya yang berkaitan dengan pembuatan dan perancangan sistem informasi desa pada penelitian Tugas Akhir ini.

1. ***Planning***

Pada tahap ini merupakan analisis terhadap kebutuhan dalam pengembangan sistem yang akan dibuat yaitu terkait data keuangan seperti APBDes Desa Doridungga Kecamatan Donggo Kabupaten Bima. Sistem informasi ini dibuat untuk masyarakat umum, baik masyarakat desa Doridungga maupun masyarakat diluar desa Doridungga untuk mengakses informasi terkait keuangan desa seperti APBDes Desa Doridungga Kecamatan Donggo Kabupaten Bima.

Pada penelitian ini menggunakan metode wawancara dalam mengumpulkan kebutuhan. Wawancara dilakukan dengan kepala Desa Doridungga Kecamatan Donggo Kabupaten Bima, dijelaskan bahwa pengelolaan data keuangan desa seperti APBDes masih menggunakan cara yang konvensional yaitu dengan menggunakan Ms. Excel. Cara seperti itu dinilai tidak efektif dan efisien apabila suatu waktu akan dilakukan pencarian atau pengolahan data. Solusi yang ditawarkan adalah dengan membuat sebuah sistem informasi yang dapat membantu pemerintah desa dalam melakukan pengolahan dan penyimpanan data terkait keuangan desa.

1. ***User Stories***

Pada tahap ini pengguna menentukan *requirement*-nya dan ditulis dalam bentuk *user story*. *User story card* yang baik menerapkan model INVEST yaitu *Independent, Negotiable, Valuable, Estimable,* dan *Small*. *Independent* berarti tidak adanya ketergantungan antar *user story*, *Negotiable* berarti seluruh detil diperoleh melalui kolaborasi, *Valuable* berarti memiliki nilai bagi pengguna, *Estimable* berarti dapat diperkirakan baik dalam segi waktu dan biaya, *Testable* berarti dapat ditentukan *acceptance* *criteria*-nya. Sebuah *user story* dapat dikatakan “selesai” jika *acceptance criteria user story* tersebut telah dipenuhi[15].

Terdapat 4 poin dasar *user story* yaitu, nama *user story,* deskripsi tujuan *user story* berupa paragraf pendek, estimasi waktu berapa lama *user story* akan diimplementasikan, dan tingkat kepentingan *user story* (*seperti must have, should have*)[16]. *User story* pada Tabel 3.1 dikelompokkan berdasarkan pengguna aplikasi baik secara umum maupun lebih detail berdasarkan kebutuhan. Pengguna aplikasi adalah masyarakat umum yang tidak perlu login untuk dapat mengakses sistem sedangkan pemerintah Desa Doridungga Kecamatan Donggo Kabupaten Bima yang bertindak sebagai admin yang harus login untuk dapat mengakses dan mengelola sistem. Berikut merupakan rangkuman dari *user story* pada sistem yang akan dibuat dapat dilihat pada Tabel 3.1.

### Tabel 3.1 *User story cards*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Kode US | Judul | Deskripsi | *Acceptance Criteria* | Estimasi (hari) |
| 1. | US-01 | Login | Terdapat 2 jenis sub. Bagian yaitu admin dan super admin yang dapat masuk ke dalam sistem. | Terdapat tampilan *login* dimana super admin dan admin diharuskan memasukkan *username* dan *password* yang telah terdaftar dalam sistem untuk dapat mengakses sistem. Sedangkan masyarakat tidak perlu login untuk dapat mengakses sistem. | 4 |
| 2. | US-02 | Melihat data admin | Sebagai Super Admin dapat melihat daftar admin berupa *list* admin serta deskripsinya. | Setelah super admin melakukan login, terdapat tampilan daftar admin pada menu navbar dan terdapat *list* admin beserta deskripsinya. | 4 |
| 3. | US-03 | Menambah data admin | Sebagai Super Admin dapat menambah data admin baru beserta deskripsinya. | Setelah super admin melakukan login, pada bagian menu navbar daftar admin, super admin dapat menambah data admin baru beserta deskripsinya dengan cara mengisi *form* untuk tambah data admin baru. | 4 |
| 4. | US-04 | Mengubah data admin | Sebagai Super Admin dapat mengubah data admin. | Pada bagian *list* admin, super admin dapat mengubah data admin yang dinginkan dengan memasukkan data admin yang baru yang terdapat dalam *form* ubah data admin. | 2 |
| 5. | US-05 | Menghapus data admin | Sebagai Super Admin dapat menghapus data admin. | Pada bagian *list* admin, super admin dapat menghapus data admin yang dinginkan dengan memilih data admin yang akan dihapus dan kemudian menekan tombol hapus data admin. | 2 |
| 6. | US-06 | Melihat pendapatan desa | Sebagai Super Admin, Admin, dan masyarakat dapat melihat data pendapatan desa | Setelah super admin dan admin melakukan login, terdapat tampilan data pendapatan desa menu navbar, dan masyarakat langsung dapat melihat data tersebut tanpa perlu login. | 4 |
| 7. | US-07 | Menambah pendapatan desa | Sebagai Super Admin, dan Admin, dapat menambah data pendapatan desa beserta deskripsinya. | Setelah super admin dan admin melakukan login, pada bagian menu navbar data desa, super admin dan admin dapat menambah data pendapatan desa beserta deskripsinya dengan cara mengisi *form* untuk tambah data penduduk. | 4 |
| 8. | US-08 | Mengubah data pendapatan desa | Sebagai Super Admin, dan Admin, dapat mengubah data pendapatan desa beserta deskripsinya. | Pada bagian profil desa, super admin dan admin dapat mengubah data pendapatan desa yang dinginkan. | 2 |
| 9. | US-09 | Menghapus data pendapatan desa | Sebagai Super Admin, dan Admin, dapat menghapus data pendapatan desa. | Setelah super admin dan admin melakukan login, pada bagian menu navbar data desa, super admin dan admin dapat menghapus data pendapatan desa yang diinginkan dengan menekan tombol hapus. | 2 |
| 10. | US-10 | Melihat pengeluaran desa | Sebagai Super Admin, Admin, dan masyarakat dapat melihat pengeluaran desa | Setelah super admin dan admin melakukan login, terdapat tampilan pengeluaran desa pada menu navbar, dan masyarakat langsung dapat melihat pengeluaran desa tersebut tanpa perlu login. | 4 |
| 11. | US-11 | Menambah pengeluaran desa | Sebagai Super Admin, dan Admin, dapat menambah pengeluaran desa beserta deskripsinya. | Setelah super admin dan admin melakukan login, pada bagian menu navbar pengeluaran desa, super admin dan admin dapat menambah pengeluaran desa beserta deskripsinya dengan cara mengisi *form* untuk tambah data. | 4 |
| 12. | US-12 | Mengubah pengeluaran desa | Sebagai Super Admin, dan Admin, dapat mengubah pengeluaran desa beserta deskripsinya. | Pada bagian profil desa, super admin dan admin dapat mengubah pengeluaran desa yang dinginkan. | 2 |
| 13. | US-13 | Menghapus pengeluaran desa | Sebagai Super Admin, dan Admin, dapat menghapus pengeluaran desa. | Setelah super admin dan admin melakukan login, pada bagian menu navbar data desa, super admin dan admin dapat menghapus pengeluaran desa yang diinginkan dengan menekan tombol hapus. | 2 |
| Total *story points* | | | | | 40 |

Pada penelitian Tugas Akhir ini, direncanakan implementasi program dapat terselesaikan dalam waktu 40 hari aktif kerja, hari aktif kerja disini mencakup hari *weekday*. *Velocity* merupakan waktu yang diperlukan untuk menyelesaikan tiap iterasi. nilai *velocity* bisa ditentukan dengan cara memperkirakan nilai velocity itu sendiri. Nilai *velocity* ditentukan sendiri oleh pihak pengembangan[17]. Nilai *velocity* yang digunakan pada penelitian Tugas Akhir ini adalah 10.

1. ***Iteration Plan***

*Release plan* merupakan proses menentukan *user stories* mana yang akan diimplementasikan pada tiap iterasi. *Release plan* meliputi perencanaan *estimasi story* dan pembagian iterasi untuk setiap *story.* Dalam mencari jumlah iterasi digunakan dengan membagi *total story* dengan *velocity*[16]*.* Penelitian Tugas Akhir ini memiliki *story points* yang berjumlah 40 dan nilai *velocity* adalah 10, maka dapat diketahui jumlah iterasi adalah 40/10 = 4 iterasi. Tiap iterasi pada pembuatan sistem pada penelitian Tugas Akhir ini akan diselesaikan dalam kurun waktu 10 hari. Pembagian iterasi dan alokasi *story* dapat dilihat pada tabel 3.2.

Tabel 3.2 Pembagian iterasi *story*

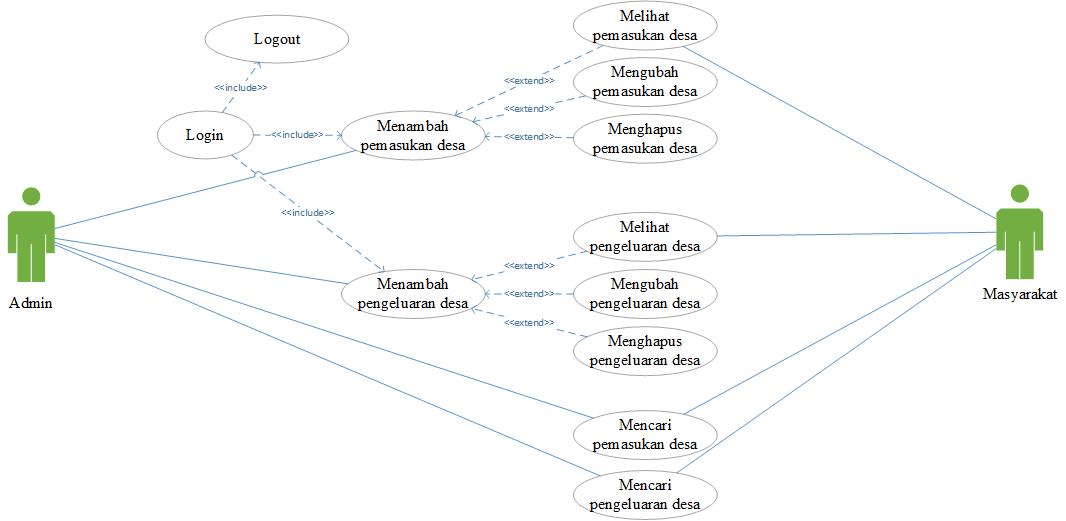
|  |  |  |  |
| --- | --- | --- | --- |
| No | Kode US | Deskripsi | Estimasi (hari) |
| *Iteration -* 1 | | | |
| 1. | US-01 | Login kedalam sistem | 4 |
| 2. | US-02 | Melihat data admin | 4 |
| 3. | US-04 | Mengubah data admin | 2 |
| *Velocity* | | | 10 |
| *Iteration -* 2 | | |  |
| 4. | US-03 | Menambah data admin | 4 |
| 5. | US-05 | Menghapus data admin | 2 |
| 6. | US-06 | Melihat pendapatan desa | 4 |
| *Velocity* | | | 10 |
| 7. | US-07 | Menambah pendapatan desa | 4 |
| 8. | US-08 | Mengubah pendapatan desa | 2 |
| 9. | US-10 | Melihat pengeluaran desa | 4 |
| *Velocity* | | | 10 |
| *Iteration -* 3 | | | |
| 10. | US-09 | Menghapus pendapatan desa | 2 |
| 11. | US-11 | Menambah pengeluaran desa | 4 |
| 12. | US-12 | Mengubah pengeluaran desa | 2 |
| 13. | US-13 | Menghapus pengeluaran desa | 2 |
| *Velocity* | | | 10 |

1. ***Design***

Desain sistem merupakan tahapan perancangan atau penggambaran tentang bagaimana sistem yang dikembangkan akan bekerja. Pada penelitian Tugas Akhir ini, sistem dirancang dengan konsep berorientasi objek menggunakan UML dan *Class Responsibilities Collaboration Card* (CRC-Card) untuk mengetahui interaksi atau hubungan antar objek yang ada pada sistem.

1. ***Usecase Diagram***

Berikut merupakan *usecase diagram* dari Sistem Informasi Desa Doridungga Kecamatan Donggo Kabupaten Bima dapat dilihat pada Gambar 3.2:



**Gambar 3.2** *Usecase* Diagram

Gambar 3.2 merupakan *usecase* dari semua pengguna sistem yaitu super admin, admin dan masyarakat. Berikut merupakan aksi-aksi yang dapat dilakukan oleh *user* admin:

* + 1. Melakukan login
    2. Mengelola pemasukan desa
    3. Mengelola pengeluaran desa
    4. Mencari pemasukan desa
    5. Mencari pengeluaran desa

Aksi-aksi yang dapat dilakukan oleh *user* masyarakat, antara lain sebagai berikut:

1. Melihat pemasukan desa
2. Melihat pengeluaran desa
3. Mengajukan pengaduan masyarakat
4. ***Class Responsibilities Collaboration Card* (CRC-*Card*)**

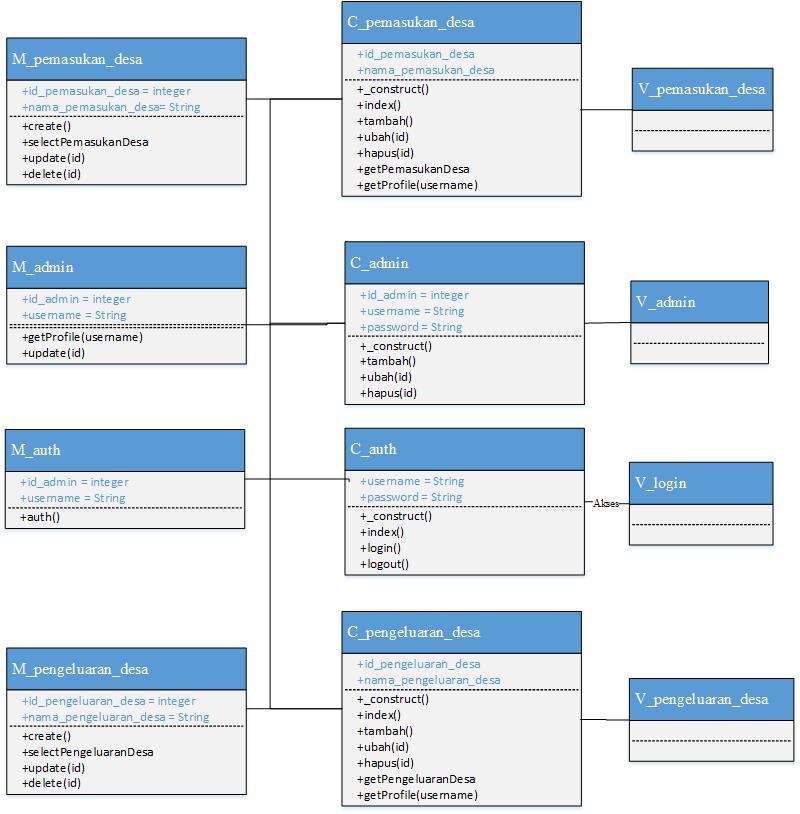
CRC-*Card* didapat dari penjabaran hubungan *responsibility* antar *class* di sistem sesuai dengan yang telah digambarkan dalam *usecase* diagram. CRC-*Card* merupakan sebuah kartu indeks yang terbagi menjadi tiga bagian yaitu nama kelas, *responsibility*, dan *collaborator* yang digunakan untuk mendeklarasikan objek-objek yang ada pada kelas dan kelas yang saling berhubungan. Digunakannya CRC-*Card* pada metode XP adalah karena mengacu pada nilai kesederhanaan yang ada pada metode XP itu sendiri[17]. Penjelasan CRC-*Card* pada penelitian Tugas Akhir ini dapat dilihat pada Tabel 3.3 :

Tabel 3.3 Penjelasan CRC-*Card*

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama *class* | *Responsibilities* | *Collaborator* |
| 1. | C PemasukanDesa | Mengatur jalannya data Pemasukan Desa yang akan dikelola maupun yang akan ditampilkan pada sistem. | M PemasukanDesa,  V PemasukanDesa |
| 2. | C Admin | Mengatur jalannya data admin yang akan dikelola maupun yang akan ditampilkan pada sistem. | M Admin,  V Admin |
| 3. | C auth | Mengatur jalannya validasi pengguna yang akan mengakses sistem. | M auth, V login |
| 4. | M PemasukanDesa | Digunakan untuk mengambil dan mengelola data Pemasukan Desa seperti menambah, mengubah serta menghapus dalam *database*. | C PemasukanDesa |
| 5. | M Admin | Digunakan untuk mengambil dan mengelola data admin seperti menambah, mengubah serta menghapus data admin dalam *database*. | C Admin |
| 6. | M auth | Digunakan untuk validasi kesesuaian *username* dan *password* pengguna yang ada pada *database*. | C auth,C PemasukanDesa |
| 7. | V PemasukanDesa | Menampilkan data Pemasukan Desa yang tersusun dalam bentuk tabel | C PemasukanDesa |
| 8. | V admin | Menampilkan data admin yang tersusun dalam bentuk tabel | C admin |
| 9. | V login | Menampilkan *form* untuk login | C auth |

1. ***Class* Diagram**

Berikut merupakan *class diagram* dari Sistem Informasi Desa Doridungga Kecamatan Donggo Kabupaten Bima dapat dilihat pada Gambar 3.3:



**Gambar 3.3** *Class* diagram pada web

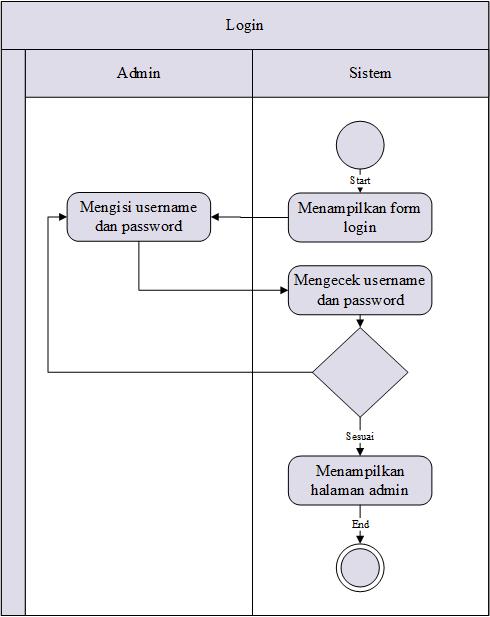
Pada Gambar 3.3 merupakan *class* diagram pada sistem informasi Desa Doridungga Kecamatan Donggo Kabupaten Bima. Sistem ini menggunakan metode analisa dan perancangan berorientasi objek dan menggunakan pola arsitektur *Model* *View* *Controller* (MVC) di setiap modul, sehingga dilakukan perancangan kelas – kelas yang ada pada sistem yang terdiri dari *view*, *controller* dan *model.* Pada *Controller* terdapat kelas “C\_pemasukan\_desa”, “C\_pengeluaran\_desa” “C\_admin”, dan “C\_auth”. Kemudian pada *Model* terdapat beberapa kelas yaitu “M\_pemasukan\_desa”, “M\_pengeluaran\_desa” “M\_admin”, dan “M\_auth”. Sedangkan *View* ini bertugas untuk menerima dan menampilkan data kepada *user*

1. ***Activity* Diagram**

Berikut merupakan *usecase diagram* dari Sistem Informasi Geografis Pemetaan Tempat Ibadah di Kota Mataram:

* 1. Proses login admin

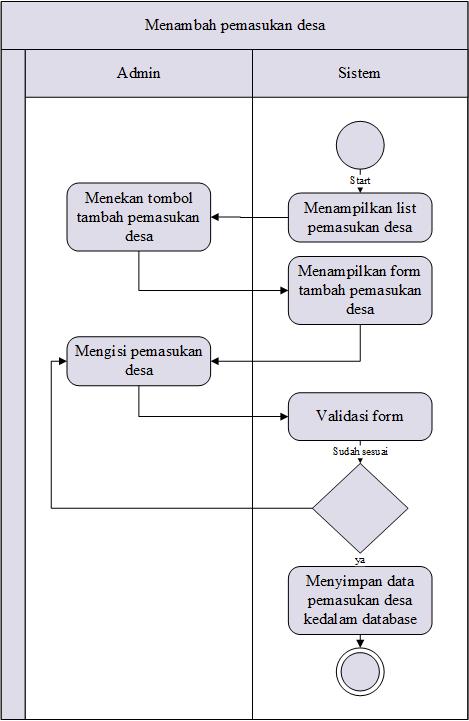
Pada sistem informasi ini, admin dapat melakukan aktivitas *login* dengan menggunakan *username* dan *password* yang sudah diberikan. Terlebih dahulu admin mengisi form *login* yang ditampilkan sistem dengan *username* dan *password* yang sudah diberikan tersebut. Kemudian sistem melakukan validasi apakah *username* dan *password* sesuai dengan yang terdapat dalam *database*. Jika sesuai maka admin dapat mengakses halaman admin, namun jika tidak sesuai maka admin harus mengisi form *login* dengan *username* dan *password* yang sesuai. *Activity* diagram proses login admin dapat dilihat pada gambar 3.4 sebagai berikut:



**Gambar 3.4** *Activity* diagram proses *login* admin

* 1. Proses menambah pemasukan desa

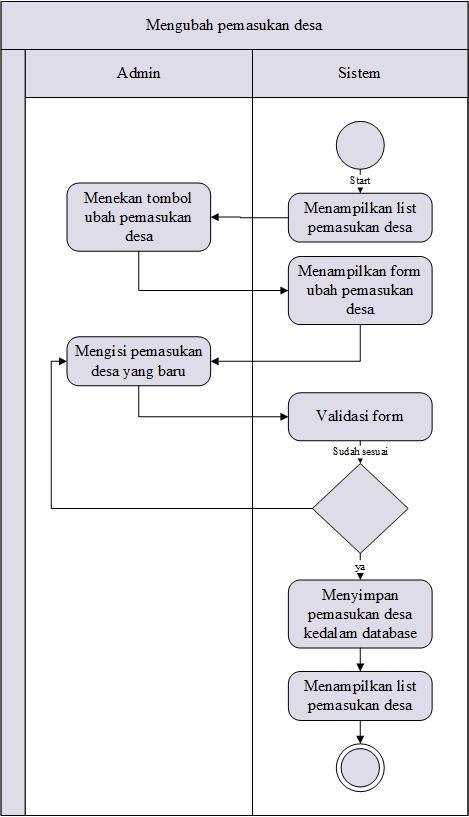
Pada sistem informasi desa ini, admin dapat melakukan aktivitas menambah pemasukan desa. Dimana terlebih dahulu sistem akan menampilkan *list* data penduduk, kemudian admin dapat menambah data dengan mengisi *form* tambah pemasukan desa yang ditampilkan sistem. Sistem melakukan validasi apakah data yang diisi sudah sesuai, jika tidak admin akan diarahkan kembali mengisi form, jika sudah sesuai maka data pemasukan desa tersebut akan masuk ke dalam *list* data pemasukan desa dan tersimpan dalam *database*. *Activity* diagram proses menambah data penduduk dapat dilihat pada gambar 3.5 sebagai berikut:



**Gambar 3.5** *Activity* diagram proses menambah data pemasukan desa

* 1. Proses mengubah pemasukan desa

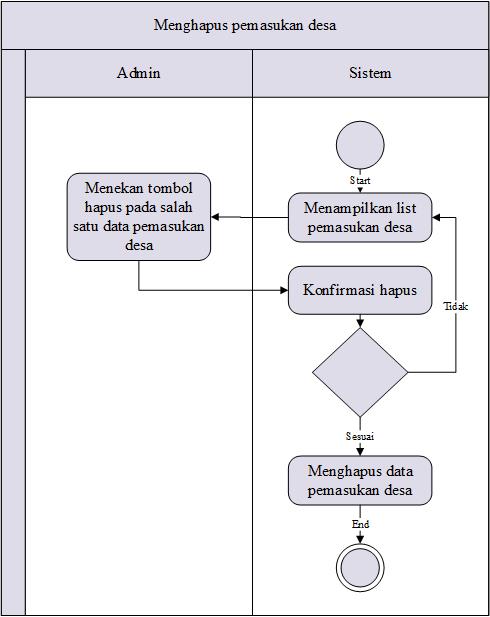
Pada sistem informasi desa ini, admin dapat melakukan aktivitas mengubah data pemasukan desa. Terlebih dahulu sistem akan menampilkan data pemasukan desa, kemudian admin dapat memilih data pemasukan desa mana yang datanya ingin diubah, setelah itu sistem akan menampilkan *form* ubah data pemasukan desa. Dalam *form* tersebut admin akan mengisi data pemasukan desa yang baru, setelah dilakukan validasi kesesuaian data maka data penduduk yang baru tersebut dapat disimpan dalam *database*. *Activity* diagram proses mengubah data penduduk dapat dilihat pada gambar 3.6 sebagai berikut:



**Gambar 3.6** *Activity* diagram proses mengubah data pemasukan desa

* 1. Proses menghapus pemasukan desa

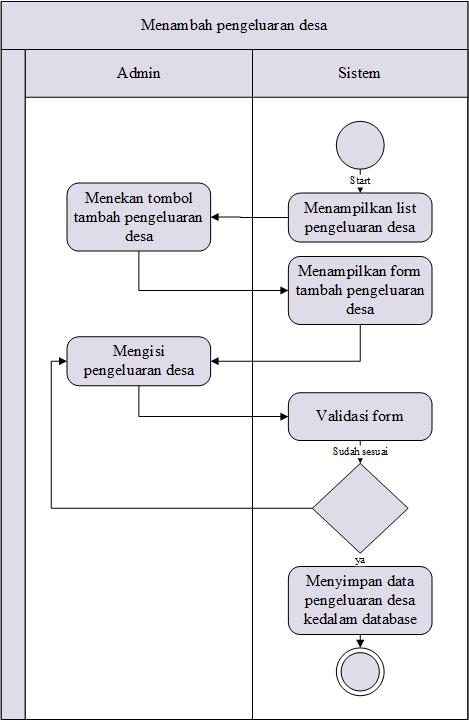
Pada sistem informasi desa ini, admin dapat melakukan aktivitas menghapus data pemasukan desa. Terlebih dahulu sistem akan menampilkan list data pemasukan desa, kemudian admin dapat memilih data pemasukan desa mana yang datanya ingin dihapus, setelah itu sistem akan menghapus data pemasukan desa tersebut dari dalam *database*. *Activity* diagram proses menghapus data pemasukan desa dapat dilihat pada gambar 3.7 sebagai berikut:



**Gambar 3.7** *Activity* diagram proses menghapus data pemasukan desa

* 1. Proses menambah pengeluaran desa

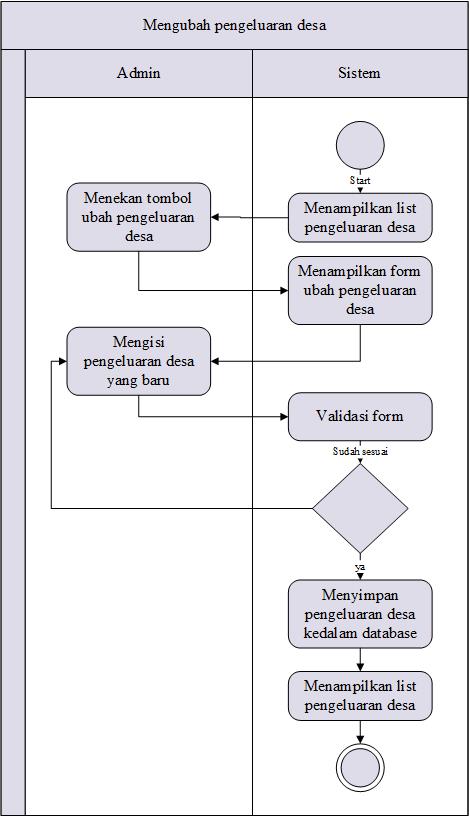
Pada sistem informasi desa ini, admin dapat melakukan aktivitas menambah pengeluaran desa. Dimana terlebih dahulu sistem akan menampilkan *list* data pengeluaran desa, kemudian admin dapat menambah data pengeluaran desa dengan mengisi *form* tambah data yang ditampilkan sistem. Sistem melakukan validasi kesesuaian data, jika sudah sesuai maka data pengeluaran desa tersebut akan masuk ke dalam *list* data pengeluaran desa dan tersimpan dalam *database*. *Activity* diagram proses menambah data pengeluaran desa dapat dilihat pada gambar 3.8 sebagai berikut:



**Gambar 3.8** *Activity* diagram proses menambah anggaran pengeluaran desa

* 1. Proses mengubah pengeluaran desa

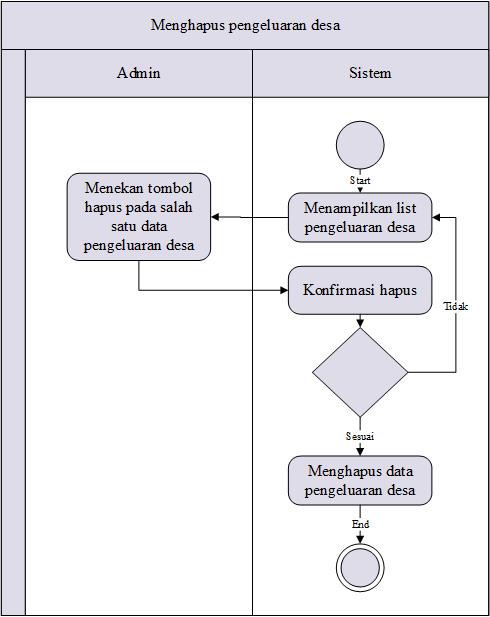
Pada sistem informasi desa ini, admin dapat melakukan aktivitas mengubah data pengeluaran desa. Terlebih dahulu sistem akan menampilkan pengeluaran desa, kemudian admin dapat memilih data mana yang ingin diubah, setelah itu sistem akan menampilkan *form* ubah pengeluaran desa. Dalam *form* tersebut admin akan mengisi data pengeluaran desa yang baru, setelah dilakukan validasi kesesuaian data maka data pengeluaran desa yang baru tersebut dapat disimpan dalam *database*. *Activity* diagram proses mengubah data pengeluaran dapat dilihat pada gambar 3.9 sebagai berikut:



**Gambar 3.9** *Activity* diagram proses mengubah pengeluaran desa

* 1. Proses menghapus pengeluaran desa

Pada sistem informasi desa ini, admin dapat melakukan aktivitas menghapus data pengeluaran desa. Terlebih dahulu sistem akan menampilkan list data pengeluaran, kemudian admin dapat memilih data mana yang datanya ingin dihapus, setelah itu sistem akan menghapus data pengeluaran desa tersebut dari dalam *database*. *Activity* diagram proses menghapus data pengeluaran desa dapat dilihat pada gambar 3.10 sebagai berikut:

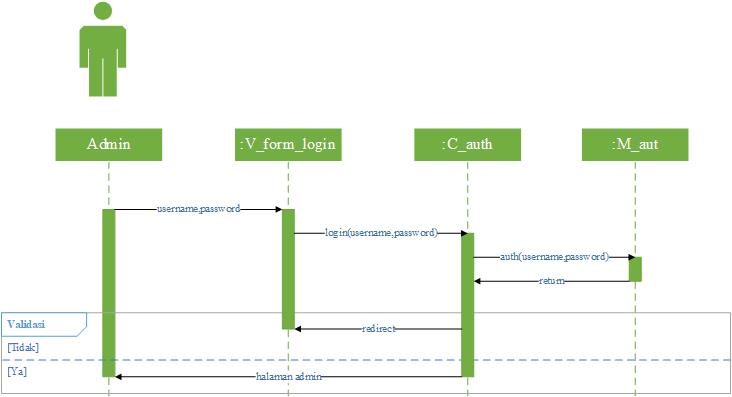


**Gambar 3.10** *Activity* diagram proses menghapus data pengeluaran desa

1. ***Sequence* Diagram**

Berikut merupakan *sequence* diagram dari Sistem Informasi Desa Doridungga Kecamatan Donggo Kabupaten Bima:

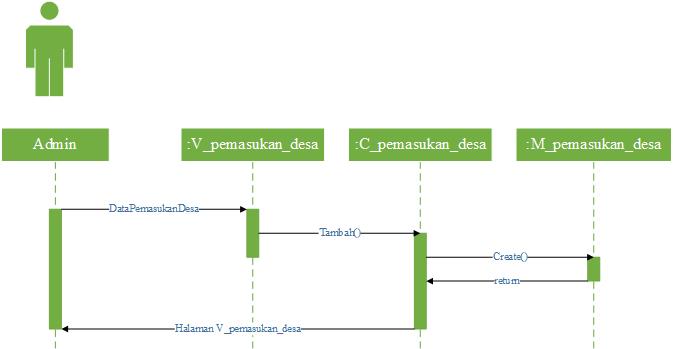
1. Proses Login



**Gambar 3.11** *Sequence* diagram proses *login* admin

Pada Gambar 3.11 merupakan *sequence* diagram proses login admin, proses *login* dimulai ketika *admin* memasukkan *username* dan *password* halaman “V\_form\_login”*.* Kemudian hasil *input form* tersebut akan dikirim dengan mengakses fungsi “login()” pada *class* “C\_auth” yang ada di *Controller* untuk diperiksa dan dilakukan validasi apakah *username* dan *password* yang dimasukkan sesuai dengan yang ada pada *database* atau tidak dengan menjalankan fungsi “auth()” yang ada di *Model*. Jika ada dan sesuai dengan data yang ada pada *database*, maka akan diarahkan ke *View* atau halaman “V\_halaman\_admin”, jika tidak sesuai maka akan diarahkan ke *View* atau halaman “V\_form\_login”.

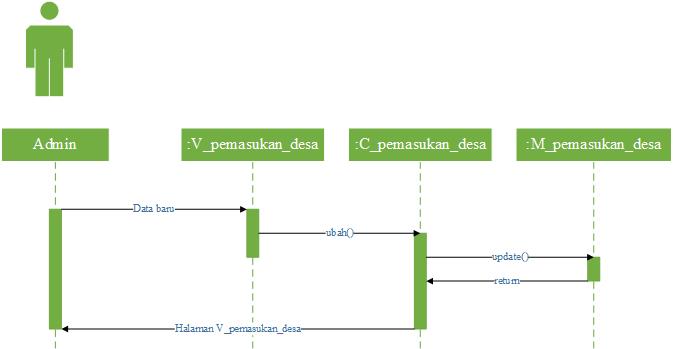
1. Proses menambah pemasukan desa



**Gambar 3.12** Proses menambah pemasukan desa

Pada Gambar 3.12 merupakan *sequence* diagram proses menambah pemasikan desa, proses tambah dimulai ketika *admin* memasukkan data pemasukan desapada *View* atau halaman “V\_pemasukan\_desa”*.* Kemudian hasil *input form* tersebut akan dikirim dengan mengakses fungsi “tambah()” pada *class* “C\_pemasukan\_desa” yang ada di *Controller*, kemudian dari *class* “C\_pemasukan\_desa” menjalankan fungsi “create()” yang ada di *class* “M\_pemasukan\_desa” pada *Model* yang berfungsi untuk menyimpan data pemasukan desa ke dalam *database*. Setelah data tersimpan admin akan langsung diarahkan ke halaman “V\_pemasukan\_desa”.

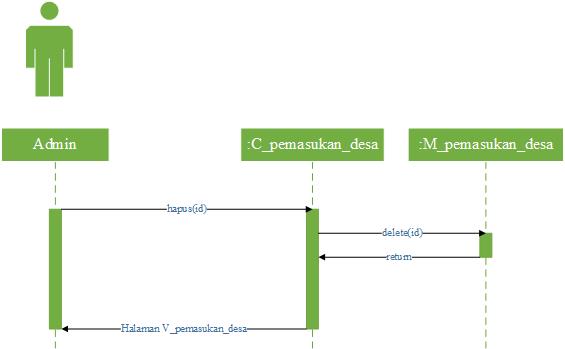
1. Proses mengubah pemasukan desa



**Gambar 3.13** Proses mengubah pemasukan desa

Pada Gambar 3.13 merupakan *sequence* diagram proses mengubah data pemasukan desa, yaitu dimuali dengan *admin* memasukkan datayangbarupada *View* atau halaman “V\_pemasukan\_desa”*.* Kemudian hasil *input form* tersebut akan dikirim dengan mengakses fungsi “ubah()” pada *class* “C\_pemasukan\_desa” yang ada di *Controller*, kemudian dari *class* “C\_pemasukan\_desa” menjalankan fungsi “update()” yang ada di *class* “M\_pemasukan\_desa” pada *Model* yang berfungsi untuk menyimpan data pemasukan desa yang baru ke dalam *database* dan setelah tersimpan admin akan diarahkan ke halaman “V\_pemasukan\_desa”.

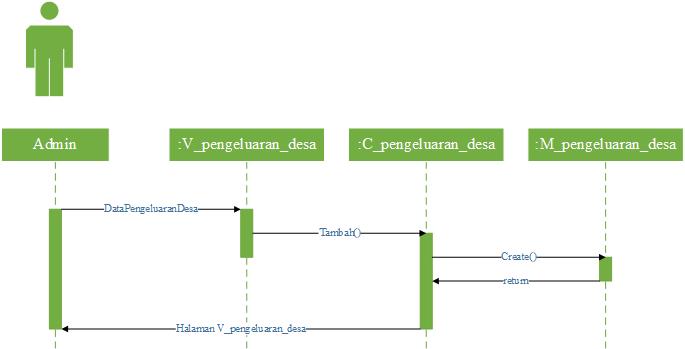
1. Proses menghapus pemasukan desa



**Gambar 3.14** Proses menghapus anggaran pendapatan belanja desa

Pada Gambar 3.14 merupakan *sequence* diagram proses menghapus data pemasukan desa, proses menghapus data dimulai ketika *admin* mengakses fungsi “hapus()” pada *class* “C\_pemasukan\_desa” yang ada di *Controller,* kemudian data akan terhapus dari dalam *database* melalui fungsi “deleted()” pada *class* “M\_pemasukan\_desa” yang ada di *Model.* Setelahdata terhapus, admin akan diarahkan ke halaman “V\_pemasukan\_desa”.

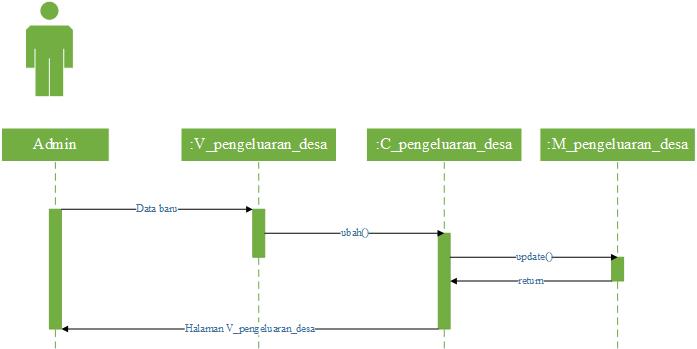
1. Proses menambah pengeluaran desa



**Gambar 3.15** Proses menambah data pengeluaran desa

Pada Gambar 3.15 merupakan *sequence* diagram proses menambah data pengeluaran desa, proses tambah dimulai ketika *admin* memasukkan data pengeluaran desapada *View* atau halaman “V\_pengeluaran\_desa”*.* Kemudian hasil *input form* tersebut akan dikirim dengan mengakses fungsi “tambah()” pada *class* “C\_pengeluaran\_desa” yang ada di *Controller*, kemudian dari *class* “C\_pengeluaran\_desa” menjalankan fungsi “create()” yang ada di *class* “M\_pengeluaran\_desa” pada *Model* yang berfungsi untuk menyimpan data penduduk ke dalam *database*. Setelah data tersimpan admin akan langsung diarahkan ke halaman “V\_pengeluaran\_desa”.

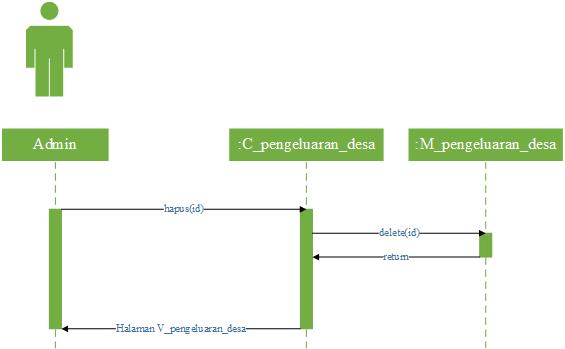
1. Proses mengubah pengeluaran desa



**Gambar 3.16** Proses mengubah data pengeluaran desa

Pada Gambar 3.16 merupakan *sequence* diagram proses mengubah data pengeluaran desa, yaitu dimulai dengan *admin* memasukkan datayangbarupada *View* atau halaman “V\_pengeluaran\_desa”*.* Kemudian hasil *input form* tersebut akan dikirim dengan mengakses fungsi “ubah()” pada *class* “C\_pengeluaran\_desa” yang ada di *Controller*, kemudian dari *class* “C\_pengeluaran\_desa” menjalankan fungsi “update()” yang ada di *class* “M\_pengeluaran\_desa” pada *Model* yang berfungsi untuk menyimpan data pengeluaran desa yang baru ke dalam *database* dan setelah tersimpan admin akan diarahkan ke halaman “V\_pengeluaran\_desa”.

1. Proses menghapus pengeluaran desa

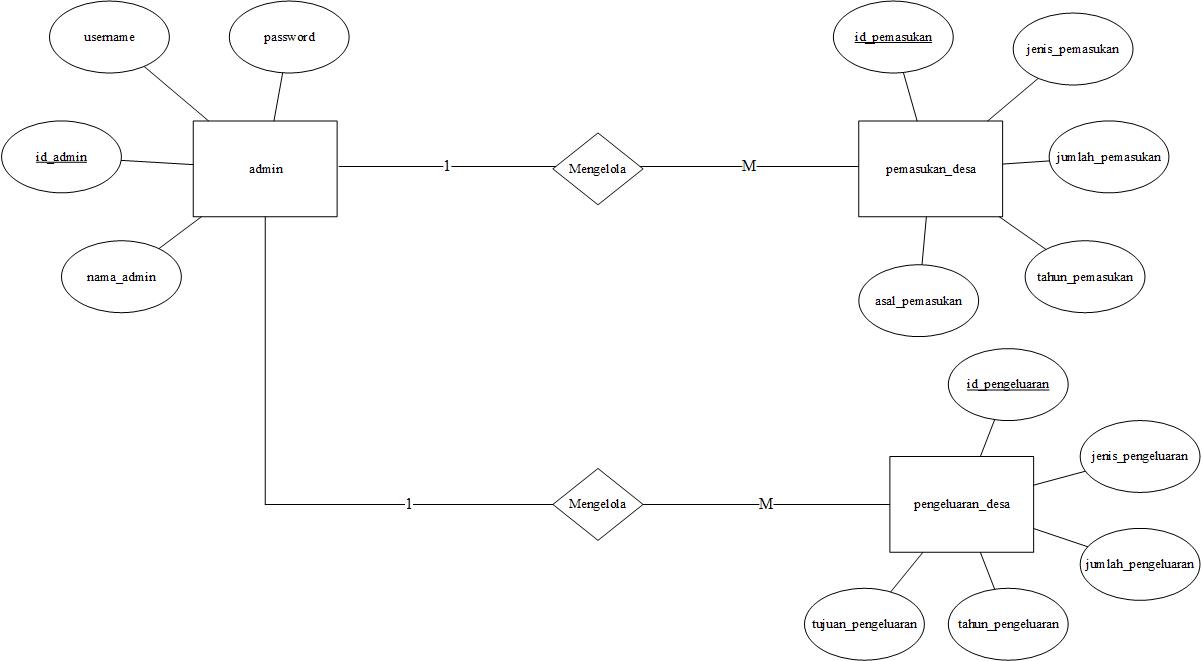


**Gambar 3.17** Proses menghapus data pengeluaran desa

Pada Gambar 3.17 merupakan *sequence* diagram proses menghapus data pengeluaran desa, proses menghapus data dimulai ketika *admin* mengakses fungsi “hapus()” pada *class* “C\_pengeluaran\_desa” yang ada di *Controller,* kemudian data akan terhapus dari dalam *database* melalui fungsi “deleted()” pada *class* “M\_pengeluaran\_desa” yang ada di *Model.* Setelahdata terhapus, admin akan diarahkan ke halaman “V\_pengeluaran\_desa”.

1. ***Entity Relationship* Diagram**

Berikut merupakan *Entity Relationship* Diagram (ERD) dari Sistem Informasi Geografis Pemetaan Tempat Ibadah di Kota Mataram dapat dilihat pada Gambar 3.18:

****

**Gambar 3.18** *Entity Relationship* DiagramSistem

Table 3.4 Tabel Admin

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama field | Type | Keterangan |
| 1. | Id\_admin | Int(5) | Primary key |
| 2. | username | Varchar(15) |  |
| 3. | password | Varchar(10) |  |
| 4. | Nama\_admin | Varchar(30) |  |

Table 3.5 Tabel Pemasukan Desa

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama field | Type | Keterangan |
| 1. | Id\_pemasukan | Int(5) | Primary key |
| 2. | Jenis\_pemasukan | Varchar(15) |  |
| 3. | Jumlah\_pemasukan | Varchar(20) |  |
| 4. | asal\_pemasukan | Varchar(8) |  |
| 5. | Tahun\_ pemasukan | Int(5) |  |

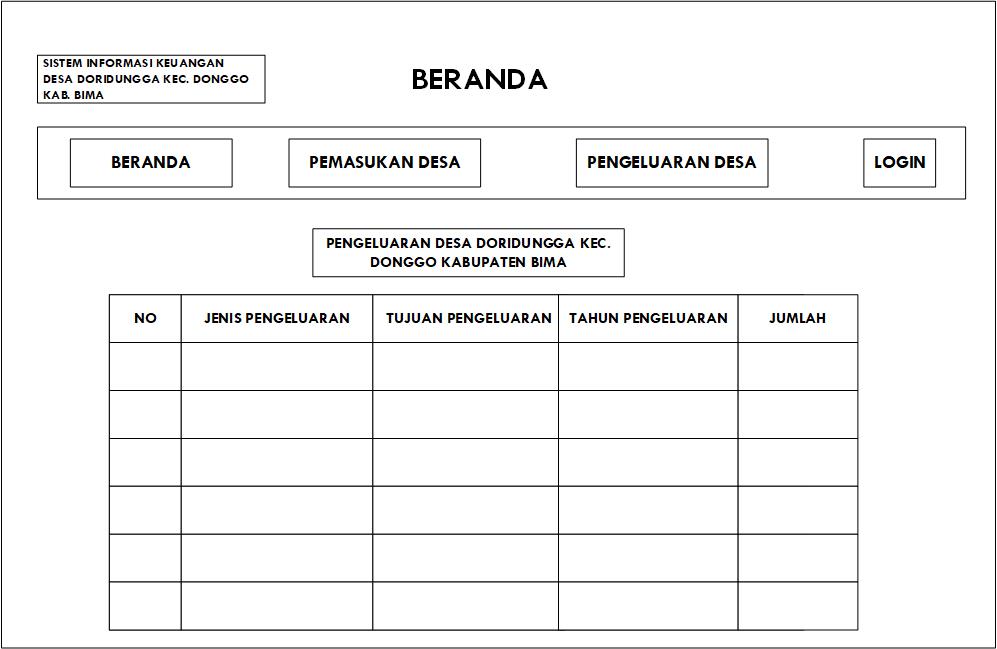
Table 3.6 Tabel Pengeluaran Desa

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama field | Type | Keterangan |
| 1. | Id\_pengeluaran | Int(5) | Primary key |
| 2. | Jenis\_pengeluaran | Varchar(15) |  |
| 3. | Jumlah\_pengeluaran | Varchar(20) |  |
| 4. | tujuan\_pengeluaran | Varchar(8) |  |
| 5. | Tahun\_pengeluaran | Int(5) |  |

1. **Desain *Interface***

Tahap selanjutnya yang akan dilakukan adalah merancang *interface* dari system informasi yang akan dibuat. Berikut merupakan *interface* yang telah dirancang:

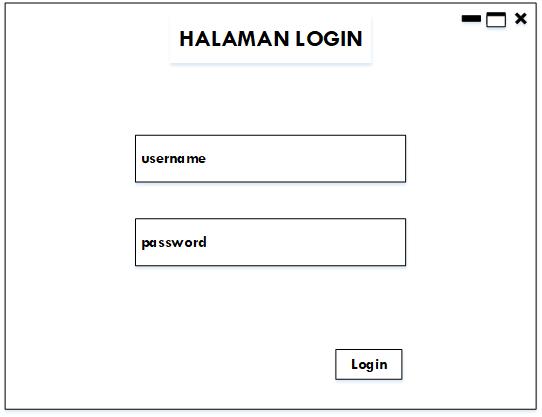
1. Halaman Beranda pada website



**Gambar 3.19** Rancangan tampilan menu beranda pada web

Pada Gambar 3.19 merupakan tampilan pada menu beranda dalam web. Pada menu ini, pengguna web dapat melihat laporan pengeluaran keuangan Desa Doridungga Kecamatan Donggo Kabupaten Bima. Pada menu beranda ini terdapat menu pemasukan desa, pengeluaran desa, dan login. Admin dapat melakukan login ke sistem dengan menekan tombol login.

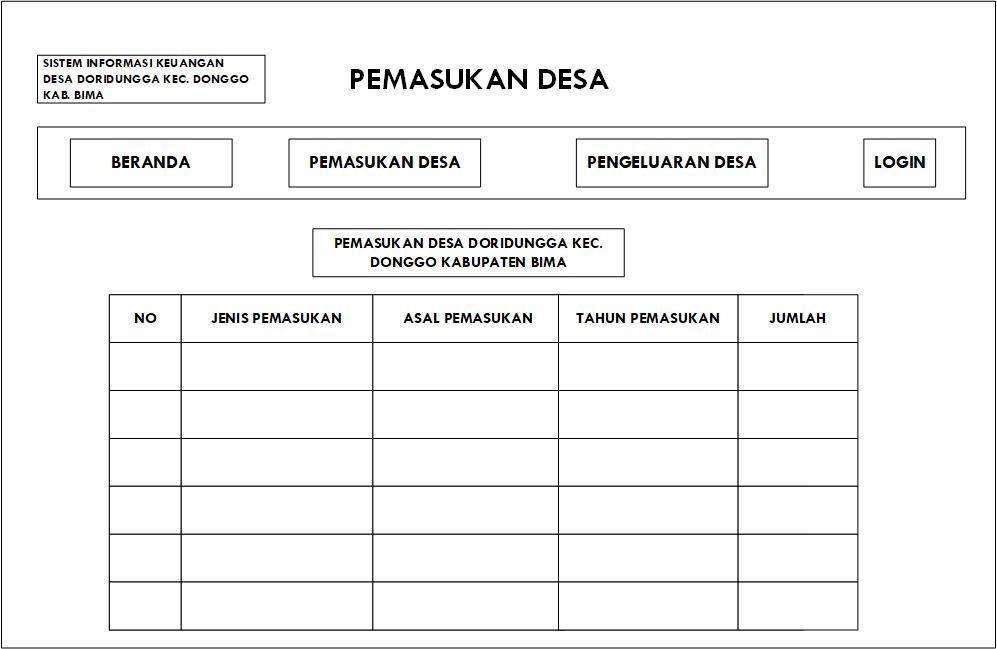
1. Halaman Login



**Gambar 3.20** Rancangan tampilan login

Pada Gambar 3.20 merupakan tampilan pada menu login. Admin dapat melakukan login ke web dengan memasukan username dan password, kemudian menekan tombol login.

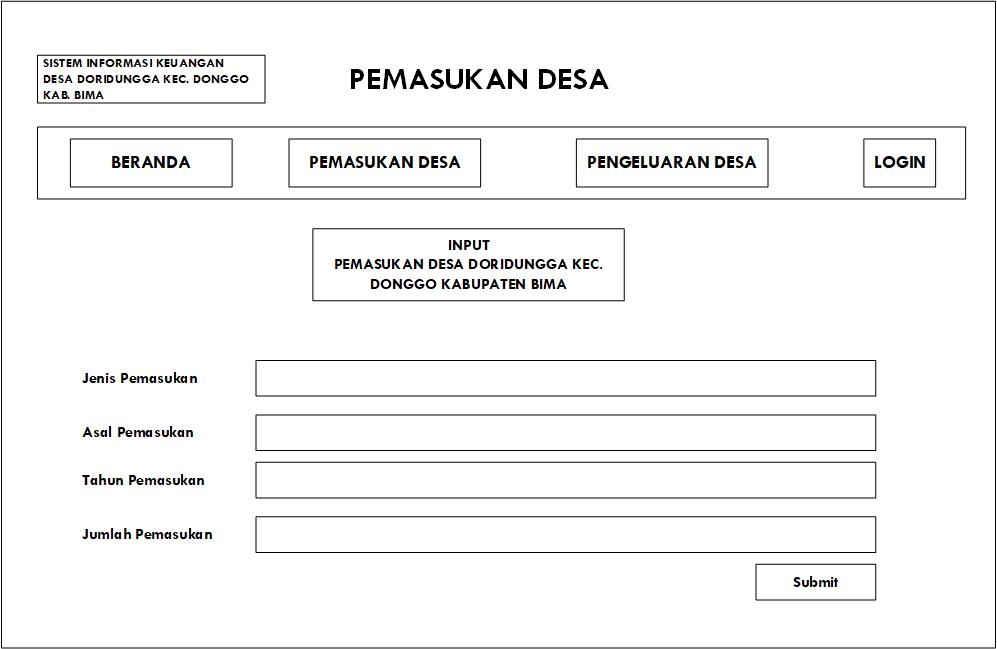
1. Halaman pemasukan desa



**Gambar 3.21** Rancangan tampilan menu pemasukan desa

Pada Gambar 3.21 merupakan tampilan dari rancangan menu pemasukan desa. Pada menu ini pengguna web dapat melihat pemasukan keuangan Desa Doridungga Kecamatan Donggo Kabupaten Bima.

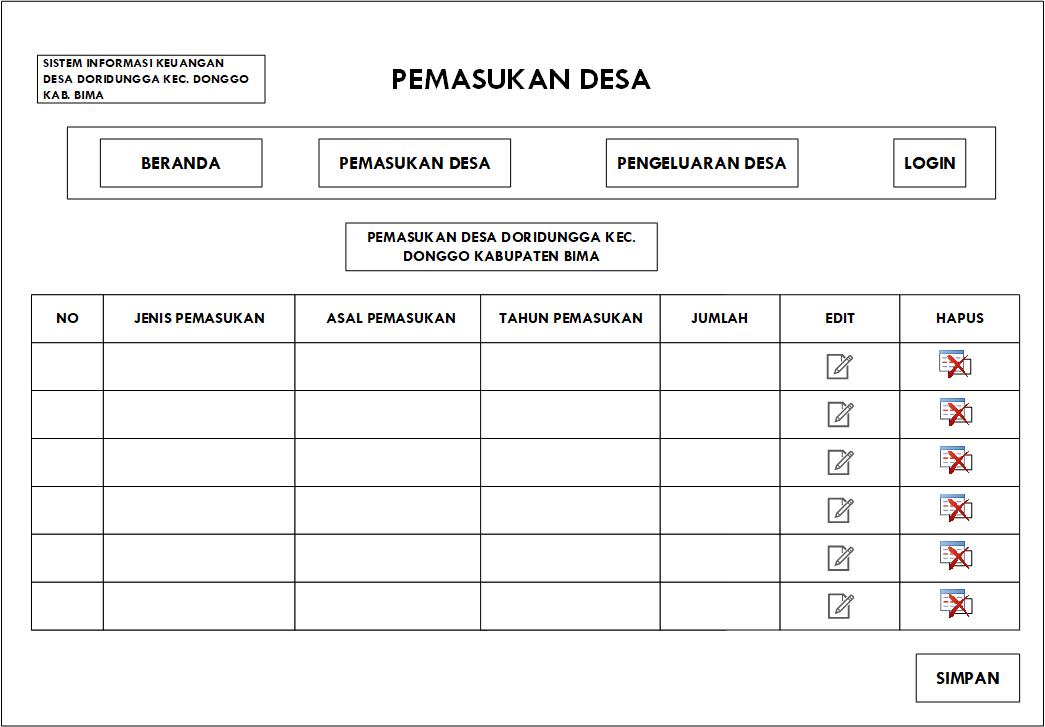
1. Halaman input data pemasukan desa



**Gambar 3.22** Rancangan tampilan menu input data pemasukan desa

Pada Gambar 3.22 merupakan tampilan dari menu input data pemasukan desa. Pada menu ini, admin dapat melakukan input data pemasukan Desa Doridungga Kecamatan Donggo Kabupaten Bima.

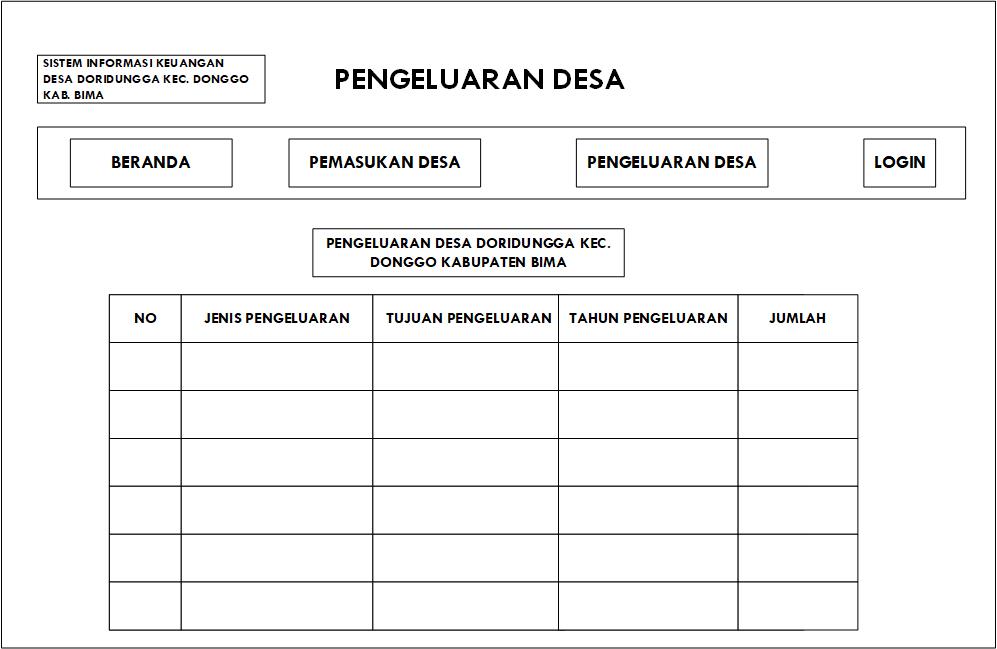
1. Halaman mengelola pemasukan desa



**Gambar 3.23** Rancangan tampilan menu mengelola pemasukan desa

Pada Gambar 3.23 merupakan tampilan dari menu mengelola pemasukan desa. Pada menu ini, admin dapat mengelola data pemasukan keuangan Desa Doridungga Kecamatan Donggo Kabupaten Bima yaitu menambah data baru, mengedit data, dan menghapus data.

1. Halaman pengeluaran desa



**Gambar 3.24** Rancangan tampilan menu pengeluaran desa

Pada Gambar 3.24 merupakan tampilan dari rancangan menu pengeluaran desa. Pada menu ini pengguna web dapat melihat pengeluaran keuangan Desa Doridungga Kecamatan Donggo Kabupaten Bima.

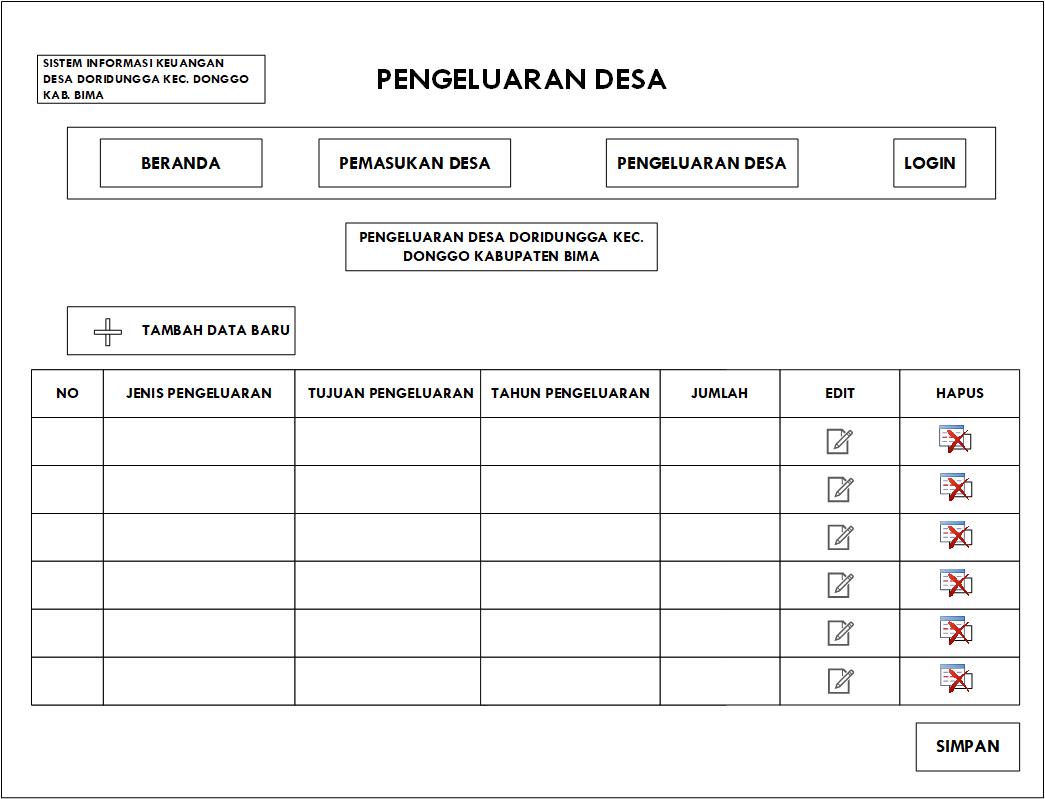
1. Halaman input data pengeluaran desa



**Gambar 3.25** Rancangan tampilan menu input data pengeluaran desa

Pada Gambar 3.25 merupakan tampilan dari menu input data pengeluaran desa. Pada menu ini, admin dapat melakukan input data pengeluaran Desa Doridungga Kecamatan Donggo Kabupaten Bima.

1. Halaman mengelola pengeluaran desa

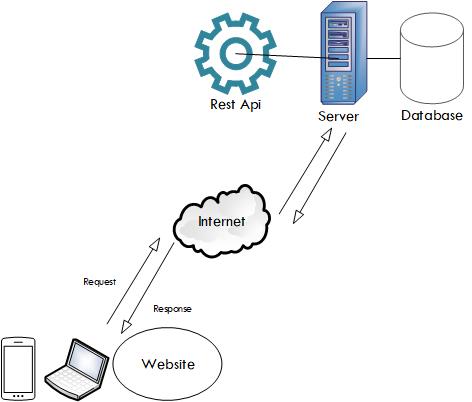


**Gambar 3.26** Rancangan tampilan menu mengelola pengeluaran desa

Pada Gambar 3.26 merupakan tampilan dari menu mengelola pengeluaran desa. Pada menu ini, admin dapat mengelola data pengeluaran keuangan Desa Doridungga Kecamatan Donggo Kabupaten Bima yaitu menambah data baru, mengedit data, dan menghapus data.

1. **Desain Arsitektur Sistem**

Berikut merupakan desain arsitektur sistemdari Sistem Informasi Desa Doridungga Kecamatan Donggo Kabupaten Bima, dapat dilihat pada Gambar 3.43:



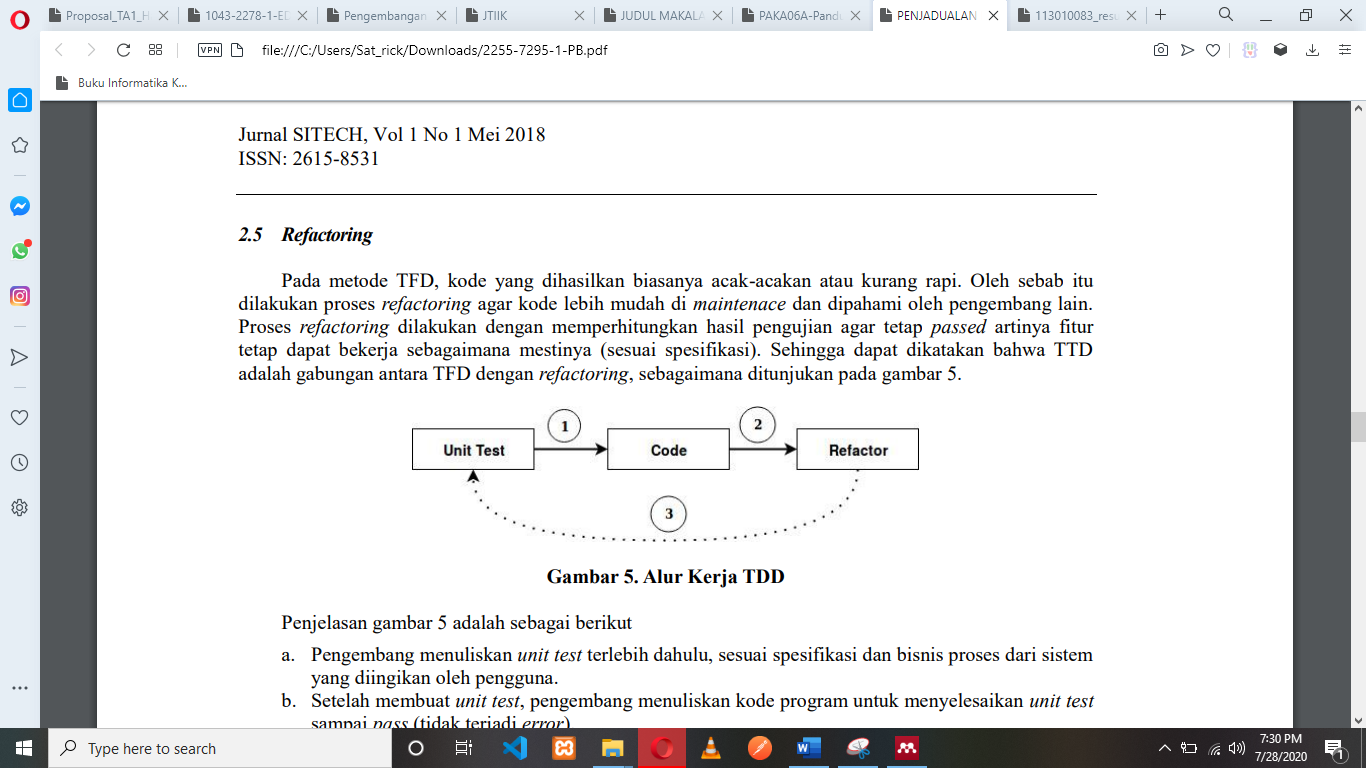
**Gambar 3.27** Arsitektur Sistem

Pada Gambar 3.27 merupakan desain arsitektur sistem yang akan dikembangkan. Pada basis *website* dapat diakses melalui *multi*-*device,* dimana *device* yang mengakses sistem mengirimkan *request* atau permintaan melalui jaringan internet dan langsung terhubung ke server kemudian akan mengakses database dan mengembalikan *response* berupa html.

1. ***Coding***

Setelah mengetahui gambaran fungsi-fungsi kebutuhan dari sistem yang akan dibuat, tahapan selanjutnya adalah proses implementasi rancangan sistem yang sudah dilakukan ke dalam bentuk sistem pemrograman (*coding*). Proses pengkodingan pada pembuatan sistem penelitian Tugas Akhir ini dilakukan oleh penulis sendiri karena mengacu pada karakteristik dari *Personal Extreme Programming* (PXP) yang dimana proses pengkodingan hanya dilakukan oleh programmer tunggal[17]. Pembuatan untuk basis *web* pada sistem penelitian Tugas Akhir ini dibangun dengan sistem pemrograman PHP menggunakan *framework* *Code Igniter*.

Pada fase *coding* dalam XP terdapat pengujian kode program menggunakan sistem *Test Driven Development* (TDD) dimana melakukan pengembangan perangkat lunak berdasarkan hasil pengujian yang telah dilakukan, hal ini bermaksud untuk dapat mengurangi waktu pengembangan dan dapat minimalisir adanya *bug* atau *error* pada aplikasi[17]. TTD memiliki alur kerja seperti Gambar 3.30 sebagai berikut:



**Gambar 3.27** Alur kerja TDD

Pengembang menuliskan *unit test* terlebih dahulu, sesuai spesifikasi dan bisnis proses dari sistem yang diinginkan oleh pengguna (*user stories*).

Setelah membuat *unit test*, pengembang menuliskan kode program untuk menyelesaikan *unit test* sampai *pass* tidak ada *error*.

Pengembang melakukan *refactoring* kode program yaitu mengubah struktur program agar lebih mudah dipahami dan dimodifikasi, tanpa harus mengubah dari *behaviour* program.

1. ***Testing***

Pada tahap ini dilakukan pengujian (*testing*) untuk mengetahui sistem telah berjalan sesuai dengan rancangan atau tidak. Pengujian merupakan kegiatan dimana suatu sistem atau komponen dieksekusi dibawah kondisi tertentu, hasilnya diamati atau dicatat untuk kemudian dievaluasi berdasarkan aspek sistem atau komponen[18]. Dalam tahap *testing* pada XP digunakan *acceptance testing* yang berguna untuk mengevaluasi kesesuaian sistem dengan kebutuhan bisnis dan menilai apakah itu dapat diterima. Dalam *acceptance testing* terdapat beberapa pengujian diantaranya yaitu *Alpha* dan *Beta testing*[19]. Pengujian *alpha* merupakan pengujian yang bertujuan untuk memastikan bahwa aplikasi yang diuji dapat berjalan dengan sistem tanpa ganguan *error* atau *bug*. Pengujian *Beta* merupakan pengujian yang bersifat langsung di lingkungan yang sebenarnya dengan penyebaran kuesioner yang akan dihitung untuk dapat diambil kesimpulan terhadap penilaian aplikasi yang dibangun.

Penekanan pada tahap ini adalah tentang pengujian, pengujian yang dilakukan menggunakan *alpha testing* dengan metode *black box* dan *beta testing* dengan penyebaran kuesioner, dimana pengguna akan menilai sejauh mana aplikasi berjalan sesuai dengan tujuan[18]. Digunakan pengujian *alpha* dan *beta* karena dirasa cocok atau sesuai dengan tujuan untuk mengetahui sistem yang dikembangkan telah sesuai atau tidak dengan kebutuhan *client*.

1. ***Release***

Langkah terakhir yang dilakukan setelah dilakukannya pengujian sistem yaitu penerapan sistem dalam kehidupan sehari-hari (*release*). Sistem dapat dikatakan *release* jika sistem sudah berjalan atau sesuai dengan kebutuhan pengguna berdasarkan dari penjabaran *user stories* yang didapat pada tabel (Tabel 3.1). Namun, aplikasi tetap harus ada dalam pendampingan jika suatu saat terdapat *bug* atau pembaharuan terhadap sistem.

## DAFTAR PUSTAKA

[1] T. 2019 Makmur, “Teknologi Informasi : Dampak dan Implikasi Bagi Perpustakaan, Perpustakawan, serta Pemustaka,” *Perpust. dan Ilmu Inf.*, vol. 1, no. 1, p. 65, 2019.

[2] Adi Supriyatna, “Metode Extreme Programming Pada Pembangunan Web Aplikasi Kerja,” 2015.

[3] S. Yana, R. D. Gunawan, and A. Budiman, “Sistem Informasi Pelayanan Distribusi Keuangan Desa Untuk Pembangunan (Study Kasus : Dusun Srikaya),” *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 1, no. 2, pp. 254–263, 2021, doi: 10.33365/jatika.v1i2.621.

[4] H. Fitriana, “Perancangan Sistem Informasi Keuangan Pada Cv. Sinar Jaya,” *Semin. Nas. Inform. 2015*, pp. 140–146, 2015, [Online]. Available: http://e-journal.potensi-utama.ac.id/ojs/index.php/SNIf/article/view/256/203.

[5] H. Riyadli, A. Arliyana, and F. E. Saputra, “Rancang Bangun Sistem Informasi Keuangan Berbasis WEB,” *J. Sains Komput. dan Teknol. Inf.*, vol. 3, no. 1, pp. 98–103, 2020, doi: 10.33084/jsakti.v3i1.1770.

[6] M. T. A. Zaen, J. Julkarnaen, and M. Saleh, “Sistem Informasi Keuangan Pada Dinas Perhubungan Kabupaten Lombok Tengah Berbasis Web,” *J. Manaj. Inform. dan Sist. Inf.*, vol. 2, no. 1, p. 50, 2019, doi: 10.36595/misi.v2i1.79.

[7] R. K. Dewi, Q. J. Ardian, H. Sulistiani, and F. Isnaini, “Dashboard Interaktif Untuk Sistem Informasi Keuangan Pada Pondok Pesantren Mazroatul’Ulum,” *J. Teknol. dan Sist. Inf.*, vol. 2, no. 2, pp. 116–121, 2021.

[8] M. Arifin and R. H. H. Hs, “Perancangan Sistem Informasi Puast Karir Sebagai Upaya Meningkatkan Relevansi Menggunakan UML,” *IC-Tech*, vol. XII, no. 2, pp. 42–49, 2017.

[9] A. Andoyo and A. Sujarwadi, “Sistem Informasi Berbasis Web Pada Desa Tresnomaju Kecamatan Negerikaton Kab. Pesawaran,” *J. TAM (Technology Accept. Model )*, vol. 3, no. 1, pp. 1–9, 2015.

[10] A. F. Sallaby and I. Kanedi, “Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework Codeigniter,” *J. Media Infotama*, vol. 16, no. 1, pp. 48–53, 2020, doi: 10.37676/jmi.v16i1.1121.

[11] A. Solichin, “Pemrograman Web dengan PHP dan MySQL,” pp. 1–122.

[12] Henri, “済無No Title No Title No Title,” *Angew. Chemie Int. Ed. 6(11), 951–952.*, pp. 3–14, 2018.

[13] R. Agarwa and D. Umphress, “Extreme programming for a single person team,” *Proc. 46th Annu. Southeast Reg. Conf. XX, ACM-SE 46*, pp. 82–87, 2008, doi: 10.1145/1593105.1593127.

[14] Yusharmen *et al.*, *Implementasi Model Personal Extreme Programming (Pxp) Dalam Pengembangan Sistem Informasi Geografis Pemetaan Flora Dan Fauna Di Taman Nasional Meru Betiri*. 2017.

[15] R. A. Azdy and A. Rini, “Penerapan Extreme Programming dalam Membangun Aplikasi Pengaduan Layanan Pelanggan (PaLaPa) pada Perguruan Tinggi,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 2, p. 197, 2018, doi: 10.25126/jtiik.201852658.

[16] M. A. F. Z and A. Wibowo, “Aplikasi gameplay edukasi pencegahan obesitas dengan menggunakan algoritma astar dan greedy pada pencarian jalur makanan,” vol. 10, no. 1, pp. 1–10, 2019, [Online]. Available: https://ejournal.undip.ac.id/index.php/jmasif/article/view/31487.

[17] A. Prasetya, A. H. Brata, and M. T. Ananta, “Pengembangan Aplikasi Pemesanan Lapangan Futsal Di Kota Malang Berbasis Android Menggunakan Metode Pengembangan Extreme Programming (Studi Kasus Champion Tidar, Zona SM Futsal, dan Viva Futsal),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 12, pp. 7293–7301, 2018.

[18] S. Masripah and L. Ramayanti, “Penerapan Pengujian Alpha Dan Beta Pada Aplikasi Penerimaan Siswa Baru,” *Swabumi*, vol. 8, no. 1, pp. 100–105, 2020, doi: 10.31294/swabumi.v8i1.7448.

[19] T. Mutiara, Achmad Benny, Muslim, A, Oswari, “Testing Implementasi Website Rekam Medis Elektronik,” *Pros. Semin. Ilm. Nas. Komput. dan Sist. Intelijen*, vol. 8, no. October, pp. 1–7, 2014.