APLIKASI SISTEM INFORMASI KEUANGAN BERBASIS WEB MENGGUNAKAN KONSEP MVC (*MODEL, VIEW, CONTROL*) DENGAN FRAMEWORK LARAVEL UNTUK MENGELOLA KEUANGAN DI KELOMPOK PETANI SAYUR HIDROPONIK ASRI 12 KAUMAN YOGYAKARTA.



Di Susun Oleh :

SAFRI ADAM (12018175)

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS AHMAD DAHLAN

YOGYAKARTA

2016

DAFTAR ISI

HALAMAN JUDULi

DAFTAR ISIii

DAFTAR GAMBAR DAN RUMUSiv

DAFTAR TABEL v

ABSTRAKvi

BAB I PENDAHULUAN1

1. Latar Belakang Masalah1
2. Identifikasi Masalah4
3. Batasan Masalah 4
4. Rumusan Masalah4
5. Tujuan Penelitian5
6. Manfaat Penelitian5

BAB II KAJIAN PUSTAKA6

1. Kajian Penelitian Terdahulu6
2. Landasan Teori 12
3. Sistem Informasi12
4. Laporan Laba Rugi13
5. Harga Pokok14
6. Biaya Aset dan Biaya Produksi 15
7. Sistem Informasi Keuangan 15
8. PHP17
9. MySQL19
10. Framework Laravel20

BAB III METODOLOGI PENELITIAN22

1. Subyek Penelitian22
2. Metode Pengumpulan Data 22
3. Metode Wawancara22
4. Survey Lokasi23
5. Alat dan Bahan23
6. Analisis Kebutuhan Sistem24
7. Desain Sistem (Pemodelan)24
8. Implementasi25
9. Model 25
10. View 26
11. Controller 26
12. Pengujian Sistem26

DAFTAR PUSTAKA28

DAFTAR GAMBAR DAN RUMUS

Gambar 2.0 gambaran umum sistem informasi12

Gambar 2.1 Rumus pembuatan laporan laba rugi14

Gambar 2.2 Rumus Penentuan Harga Pokok 14

DAFTAR TABEL

Tabel 2.1. tabel perbandingan kajian penelitian terdahulu9

Aplikasi Sistem Informasi Keuangan Berbasis Web Menggunakan Konsep MVC (*Model, View, Control*) Dengan Framework Laravel Untuk Mengelola Keuangan Di Kelompok Petani Sayur Hidroponik ASRI 12 Kauman Yogyakarta

Safri Adam

12018175

ABSTRAK

Kelompok petani sayur hidroponik di RW 12 kauman, Yogyakarta merupakan kelompok tani yang memiliki masalah dalam pengelolaan keuangan. Sehingga dampaknya teerlihat ketika melihat hasil penjualan yan selalu merugi. Hal in diakibatkan tidak adanya sistem terkomputerisi yang menangani masalah keuangan di kelompok tani hidroponik ini. Sehingga diperlukan sebuah sistem yang terkomputerisasi untuk menangani masalah keuangan yang ada di kelompok tani hidroponik kauman dapat terselesaikan.

Sistem yang akan dibuat berbasis web dan dibangun menggunakan *framework* laravel dengan konsep MVC (*Model, View, Control*). Fitur yang ada pada sistem ini antara lain dapat merancang anggaran bulanan, mendukeentasikan pengeluaran,dan pemasukan, serta dapat mencetak laporan setiap periode yang telah ditentukan. Hasil yang diharapkan nantinya sistem ini dapat memudahkan bendahara di kelompok petani hidroponik untuk mengelola keuangan.

Kata kunci : Sistem Informasi, Web, Keuangan, hidroponik , sayur hidroponik.

**Bab I**

**Pendahuluan**

Latar Belakang Masalah

Memasuki tahun 2016 dimana MEA secara menyeluruh akan diterapkan, MEA (Masyarakat Ekonomi ASEAN) merupakan sebuah gagasan dari pemimpin ASEAN dan seluruh Negara-negara ASEAN yang bertujuan untuk menciptakan pembangunan di negara ASEAN dengan melakukan integrasi ekonomi yaitu aliran bebas barang, jasa, investasi, dan tenaga kerja terdidik antar negara ASEAN. Indonesia, dengan penduduknya yang berjumlah 252 juta jiwa lebih akan menjadi sasaran perdagangan karena sangat berpotensi menjadi basis konsumsi yang besar. Oleh karena itu, jika Indonesia memiliki strategi yang baik, MEA 2016 dapat menjadi peluang besar Indonesia untuk membangun perekonomiannya.

Salah satu cara Indonesia untuk membangun perekonomiannya adalah dengan menambah sektor UMKM. UMKM (Usaha Mikro, Kecil dan Menengah) memiliki peran yang sangat besar dalam perekonomian Indonesia. Jumlah usaha mikro kecil dan menengah (UMKM) di Indonesia saat ini sekitar 55 juta, dan menyerap 97% tenaga kerja Indonesia. Meski secara kuantitas sangat besar dan menyerap banyak tenaga kerja, pangsa dalam pendapatan nasional masih sekitar 57%. Keadaan ini disebabkan oleh masalah internal yang dihadapi UMKM yaitu: rendahnya kualitas SDM UMKM dalam manajemen, organisasi, penguasaan teknologi, dan pemasaran, lemahnya kewirausahaan dari para pelaku UMKM, dan terbatasnya akses UMKM terhadap permodalan, informasi, teknologi dan pasar, serta faktor produksi lainnya.(Anggarani, Prasetyoning, & Safitri, 2015)

Salah satu masalah internal terhambatnya pertmbuhan UMKM yaitu rendahnya penguasaan SDM terhadap manajemen dan penguasaan teknologi. Dari sisi kelemahan inilah dapat dibangun pondasi kekuatan untuk menyelesaikan kendala terhadap manajemen. Salah satu manajemen yang terpenting adalah manajemen keuangan. Yang terpenting dalam menjalankan usaha adalah menejemen keuangan yang baik. Salah satu caranya yaitu mengelola keuangan dengan ilmu akuntansi yang kemudian disinergikan dengan penguasaan teknologi informasi.

Kauman merupakan sebuah kampung yang terletak di pusat kota Yogyakarta, berada di kelurahan Ngupasan dan kecamatan Gondomanan, selatan malioboro dan barat Kraton Ngayogyakarta. Saat ini warga Kauman yang meliputi 7 RT sedang menggalakkan budidaya tanaman menggunakan hidroponik meliputi sayuran selada, kangkung, sawi dan seledri. Setiap RW di Kauman memiliki beberapa kelompok tani yang masing-masing memiliki nama dan keunggulan dalam hal bercocok tanam menggunakan hidroponik. Salah satu kelompok petani hidroponik yang sudah sukses adalah kelompok tani Sakinah di RW 11. Bercermin kepada kelompok petani yang sudah sukses, maka membuat kelompok tani lain ingin memajukan kelompok tani masing-masing. Salah satunya yaitu kelompok tani RW 12.

Kelompok petani sayur hidroponik di RW 12 kauman, Yogyakarta merupakan kelompok tani yang berdiri atas gagasan kelompok ibu-ibu PKK yang berinisiatif menjadikan usaha sayur hidroponik sebagai UMKM bersama. Kelompok tani RW 12 selama bercocok tanam dengan hidroponik sudah 2 kali panen, namun hasil panen belum dapat dipasarkan secara optimal, pemasaran hanya di warung sekitar dengan harga kekeluargaan sehingga harga yang didapat sangat ekonomis atau murah.

Hal ini mengakibatkan hasil dari penjualan sayur hidroponik ini tidak memberikan keuntungan bagi petani dan bahkan mengalami kerugian. Sehingga untuk membuka masa tanam baru, kelompok ini harus membeli lagi bibit tanaman dengan dana pribadi bukan menggunakan dana hasil penjualan sebelumnya. Hal ini disebabkan belum terdokumentasinya dana yang sudah dikeluarkan untuk modal usaha sehingga para petani tidak bisa melihat keadaan keuangan kelompok saat itu berdampak salah satunya tidak dapat menentukan harga jual yang pantas agar modal yang dikeluarkan bisa tertutupi dan memberi keuntungan. Harga pantas yang di maksud adalah harga yang sesuai dengan harga pasar dan dapat menutupi modal yang telah dikeluarkan selama masa tanam.

Bedasarkan hal tersebut , maka kelompok tani hidroponik RW 12 memerlukan sebuah pembukuan keuangan yang nantinya diharapkan akan menyelesaikan masalah diatas. Mengingat pada tahun ini sudah diterapkan MEA (Masyarakat Ekonomi Asean) yang diharapkan masyarakat menguasai teknologi informasi dalam bersaing dalam MEA, maka dibangunlah sebuah system informasi keuangan terkomputerisasi yang kemudian diharapkan dengan sistem ini, petani dapat mendokumentasikan arus kas sehingga mempermudah kelompok petani dalam mengelola keuangan. Dengan sistem ini petani juga dapat mengevaluasi *history* keuangan kelompok sebagai acuan perencanaan di masa tanam yang akan datang. Selain itu dengan sistem ini juga akan menambah daya saing keunggulan teknologi dibidang manajemen dan teknologi informasi.

Salah satu teknologi yang berkembang pesat saat ini adalah internet. Mengingat kebutuhan akan internet saat ini sudah mendekati kebutuhan primer di masyarakat kota khususnya, maka dibangun sistem informasi keuangan berbasis website untuk kelompok petani hidroponik RW 12 Kauman. Dengan demikian, diharapkan kelompok tani ini dapat berkembang pesat dan siap menuju persaingan Masyarakat Ekonomi Asean 2016. Untuk membangun sebuah website dibutuhkan sebuah *framework* yang mana *framework* ini fungsinya sebagai kerangka kerja yang dapat mempermudah *programmer* dalam membangun *website*. *Framework* yang digunakan adalah *Laravel* karena proses penulisan PHP akan lebih ringkas dan struktur yang dibuat mengacu pada prosedur MVC (Model, View, Control) yang digunakan oleh framework lainnya sehingga diharapkan pengembang (*developer*) atau pemrograman lainnya yang akan memperbaiki kesalahan (*bug*) dapat mengetahui strukur yang telah baku. (Ariefin & Ilhamsyah, 2015).

Identifikasi Masalah

Bedasarkan latar belakang masalah diatas didapat beberapa masalah yang ada di kelompok tani RW 12 Kauman ini antara lain :

# Kelompok petani hidroponik tidak memiliki sistem informasi keuangan sehingga tidak dapat memantau keadaan keuangan kelompok.

# Tidak memiliki catatan pembukuan keuangan yang menyebabkan petani kesulitan menentukan harga jual sayuran hidroponik sehingga selalu menyebabkan kerugian.

Batasan Masalah

Batasan masalah pada penelitian ini adalah pembuatan sistem informasi keuangan berbasis web agar kelompok tani khususnya ketua kelompok dan bendahara dapat memantau keuangan kelompok dan merekomendasikan harga jual sayuran hidroponik.

Rumusan Masalah

Melihat latar belakang masalah tersebut, serta melihat referensi mengenai Framework Laravel dan sistem informasi keuangan melalui media digital dan media non-digital, maka didapat rumusan masalah antara lain:

# Bagaimana merancang dan membangun sebuah system informasi keuangan berbasis web yang dapat digunakan untuk mengelola keuangan kelompok tani hidroponik RW 12 Kauman.

# Bagaimana mengintegrasikan ilmu akuntansi ke *framework Laravel* agar menjadi sebuah sistem informasi keuangan yang terkomputerisasi berbasis web.

Tujuan Penelitian

1. Meningkatkan kinerja bendahara dalam mengelola keuangan.
2. Memberikan laporan keuangan yang akurat kepada ketua kelompok petani untuk di jadikan dasar pengambilan kebijakan.

Manfaat Penelitian

Manfaat bagi kelompok petani hidroponik yaitu:

1. Dapat mendokumentasikan pengeluaran dan pemasukan dana dengan sistem yang telah terkomputerisasi. Sehingga dapat menentukan harga yang sesuai untuk menjual hasil pertanian
2. Memperkecil kesalahan yang terjadi saat perhitungan keuangan.
3. Dapat menginput pengeluaran maupun pemasukan dana setiap saat karena sistem yang dibuat berbasis website dan dapat diakses secara online
4. Mempercepat perhitungan keuangan mulai dari modal hingga hasil penjualan.
5. Dapat dijadikan sebagai bahan evaluasi serta memeriksa *history* dari perolehan keuntungan dari waktu ke waktu.

**Bab II**

**Kajian Pustaka**

# Kajian Penelitian Terdahulu

Kajian terdahulu melihat dari penelitian Nugroho (2009) tentang peranan teknologi informasi dalam audit sistem informasi komputerisasi akuntansi. Mengatakan bahwa Peranan teknologi informasi pada aktivitas manusia saat ini memang begitu besar. Teknologi informasi telah menjadi fasilitator utama bagi kegiatan-kegiatan bisnis yang, memberikan andil besar terhadap perubahan perubahan mendasar bagi struktur, operasi dan manajemen organisasi. Kemudian menyebutkan permasalahan yang terjadi yaitu peningkatan kompleksitas kegiatan bisnis mengakibatkan semakin tingginya risiko kesalahan interprestasi dan penyajian laporan keuangan yang hal ini menyulitkan para *users* laporan keuangan dalam mengevaluasi kualitas laporan keuangan, harus mengandalkan laporan auditor independen atas laporan keuangan yang diaudit untuk memastikan kualitas laporan keuangan yang bersangkutan.

Namun ironisnya, pada kondisi di lapangan tidak banyak para auditor yang bisa memanfaatkan akses dari peranan teknologi informasi dalam mengaudit sistem informasi yang berbasis pada komputerisasi akuntansi baik pada saat input, proses sampai dengan output mengingat *brainware* dibidang auditor yang mengenal teknologi informasi masih relatif sedikit karena walaupun teknologi informasi sudah generalisasi dalam dunia bisnis namun tidaklah banyak yang sesuai dapat menjawab standar keilmuan misalnya dalam memenuhi kebutuhan audit sistem informasi komputerisasi akuntansi dimana peluang ini masih jarang diminati para programmer yang benar benar memahami ilmu ekonomi dan akuntansi yang juga diberikan keahlian dalam bidang pemrograman komputer sehingga walaupun ada harga software program aplikasi yang digunakan untuk mengaudit tersebut masih relatif tinggi.(Nugroho, 2009)

Penelitian yang dilakukan oleh Kabuhung (2013) tentang sistem informasi akuntansi penerimaan dan pengeluaran kas untuk perencanaan dan pengendalian keuangan pada organisasi nirlaba keagamaan mengatakan bahwa tanpa adanya sistem informasi akuntansi yang mengawasi aktivitas- aktivitas yang berlangsung, perusahaan atau organisasi akan mengalami kesulitan untuk menentukan seberapa baik kinerjanya dan juga akan mengalami kesulitan dalam menelusuri bagaimana pengaruh-pengaruh dari berbagai aktivitas atas sumberdaya-sumberdaya yang ada dibawah pengawasannya. Oleh karena itu, sistem informasi akuntansi yang efektif sangatlah penting bagi keberhasilan jangka panjang organisasi manapun.(Kabuhung, 2013)

Bedasarkan penelitian dari Satya (2015) tentang pembangunan sistem informasi keuangan ,mengatakan bahwa organisasi yang besar pasti membutuhkan sistem keuangan yang besar pula. Hal ini disebabkan karena proses transaksi yang terjadi sangat sering dan melibatkan nominal yang besar. Sistem yang besar mempunyai prosedur dan proses bisnis yang banyak dan rumit. Oleh karena itu, dibutuhkan suatu model untuk membantu kinerja keuangan menjadi lebih baik dan dapat terorganisir dengan rapi. Model tersebut dapat disebut dengan sistem informasi keuangan. (Satya, 2015)

Sugiantina (2011) dalam penelitiannya yaitu penerapan sistem informasi dalam praktek laporan keuangan sederhana menyatakan terkait dengan metode analisis laporan keuangan dan penerapan teknologi informasi dalam berbagai analisis keungan, hal ini merupakan suatu tolak ukur bagi kemajuan pelaporan keuangan. Terutama terhadap upaya setiap institusi bisnis, untuk menyeimbangkan perkembangan teknologi Informasi, yang selaras dengan pekembangan analisis pelaporan keuangan. Dengan perkembangan tersebut, institusi bisnis dituntut untuk bisa lebih cepat dan kreatif dalam bekerja khususnya dalam pelaporan keuangan. Hal ini dikarenakan adanya shopware sistem informasi yang sudah menjadi trend, dalam dunia bisnis. Oleh sebab itu, sekarang telah menjadi trend dalam pembuatan laporan keuangan berbasis web internet hal ini dapat memberi kemudahan dalam hal financial controlling dan Manajemen bagi semua lapisan level manajemen intitusi bisnis maupun secara terbuka bagi pelanggan atau masyarakat.(Sugiantina, 2011)

Bedasarkan penelitian yang dilakukan Andreas Handojo dkk (2004) tentang pembuatan sistem informasi akuntansi terkomputerisasi atas siklus pembelian dan penjualan pada cv. x. Mengatakan bahwa Sistem akuntansi pada CV. X masih menggunakan cara manual yaitu pencatatan piutang pada kartu piutang dan pencatatan hutang pada kartu hutang. Perhitungan total penjualan dan keuntungan dilakukan manual, sehingga membutuhkan waktu yang lama untuk mendapatkan informasi yang diinginkan. Oleh karena itu dibutuhkan sistem informasi akuntansi yang memadai, untuk memproses pencatatan secara otomatis dan menghasilkan infomasi yang cepat dan tepat. Sehingga setelah sistem diterapkan kepuasan pengguna terhadap keseluruhan aplikasi sebesar 72.5%, dan dapat disimpulkan bahwa aplikasi sudah memenuhi dan sesuai dengan kebutuhan sistem, tapi masih ada beberapa kelemahan yang perlu untuk diperbaiki. Aplikasi ini membantu proses pencatatan dengan melakukan otomatisasi penjurnalan dan posting serta membantu mempercepat proses pembuatan laporan laba rugi perusahaan. (Andreas Handojo et al., 2004)

perbandingan dari keseluruhan penelitian dapat dilihat pada tabel 2.1

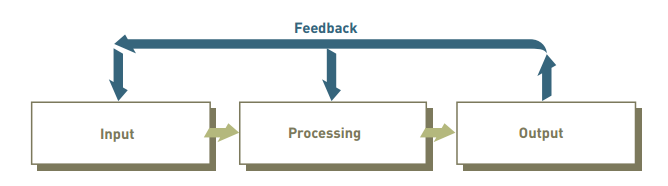
Tabel 2.1: Tabel Perbandingan Kajian Penelitian Terdahulu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Peneliti | Kajian 1 | Kajian 2 | Kajian 3 | Kajian 4 | Kajian 5 | Topik Yang Akan Dikembangkan |
| Judul | PERANAN TEKNOLOGI INFORMASI DALAM AUDIT SISTEM INFORMASI KOMPUTERISASI AKUNTANSI | PEMBUATAN SISTEM INFORMASI AKUNTANSI TERKOMPUTERISASI  ATAS SIKLUS PEMBELIAN DAN PENJUALAN PADA CV. X | PEMBANGUNAN SISTEM INFORMASI KEUANGAN STUDI KASUS SMK MARSUDI LUHUR I YOGYAKARTA | PENERAPAN SISTEM INFORMASI  DALAM PRAKTEK LAPORAN KEUANGAN SEDERHANA | SISTEM INFORMASI AKUNTANSI PENERIMAAN DAN PENGELUARAN KAS UNTUK PERENCANAAN DAN PENGENDALIAN KEUANGAN PADA  ORGANISASI NIRLABA KEAGAMAAN | APLIKASI SISTEM INFORMASI LABA RUGI USAHA BAGI KELOMPOK PETANI SAYUR HIDROPONIK DI RW 12 KAUMAN YOGYAKARTA. |
| Teknologi | Computer Based System | Desktop Application  (Borland Delphi 7.0.) | Desktop Application  (C#) | Web based application (PHP) | Computer Based System | Web based application (PHP) laravel framework |
| Database |  | SQL server 7.0. | DBMS SQL Server 2008 R2 | MySQL |  | MySql |
| Fitur Umum | Melakukan entri transaksi (seperti: transaksi penjualan dalam toko pengecer, pengambilan kas di dalam suatu bank, dan pengiriman barang dalam suatu pabrik).  Melakukan permintaan keterangan (seperti informasi tentang account atau saldo terkini customer).  Meminta laporan (seperti daftar unsur sediaan yang ada di gudang, yang kuantitasnya menunjukkan angka negatif).  Melakukan up-dating terhadap master file (seperti pembuatan account bagi customer baru dan pengubahan kode account buku besar). Tipe Sistem Komputer On-line | modul penjualan,  modul pembelian dan akuntansi (pencatatan). Penjualan meliputi pembuatan sales order, pengecekan inventori dan pengecekan kredit limit pelanggan. Pembelian meliputi pembuatan purchase order sampai penerimaan purchase invoice.  Pencatatan meliputi penjurnalan, posting dan pembuatan laporan laba rugi. | Pengelolaan mata akun pemasukan, pengelolaan mata akun pengeluaran, pengelolaan anggaran pemasukan, pengelolaan anggaran pengeluaran, pengelolaan transaksi, pembuatan reporting. | 1. Pembuatan User Account oleh Administrator.  2. Log in oleh Direktur Utama, untuk entry Data Master.  3. Log in oleh Operator, untuk entry Data Transaksi. Sebagai bahan pembuatan laporan  keuangan.  4. Log in oleh Komisaris, untuk melihat Laporan |  | Mini POS (Point Of sales ) penjualan sayur, Pengelolaan buku harian (debet-kredit), pencetakan laporan laba rugi per periode, pencetakan laporan arus kas, |
| Tempat Penelitian |  | CV. X | SMK Marsudi Luhur I Yogyakarta |  | GMIM Nafiri Malalayang | Kelompok tani hidroponik RW 12 Kauman yogyakarta. |
| Hasil |  | Sebuah system informasi akuntansi berbasis desktop untuk CV.X | Sebuah system informasi keuangan yang membantu petugas TU dalam mengelola keuangan | Sebuah system informasi keuangan sederhana berbasis web agar dapat diakses dari setiap unit usaha. |  |  |

# Landasan Teori

1. Sistem Informasi

Sistem informasi (SI) adalah seperangkat elemen yang saling terkait atau komponen yang mengumpulkan (input), memanipulasi (proses), menyimpan, dan menyebarkan (output) data dan informasi, dan memberikan reaksi korektif (mekanisme umpan balik) untuk memenuhi tujuan sesuai yang tertera pada gambar 2.0 . Mekanisme umpan balik merupakan komponen yang membantu organisasi mencapai tujuan mereka, seperti meningkatkan keuntungan atau meningkatkan layanan pelanggan (Ralph M. Stair, 2010)

(Gambar 2.0) Gambaran umum sistem informasi

* 1. INPUT

Dalam sistem informasi, input adalah kegiatan pengumpulan dan menangkap data mentah. Dalam memproduksi gaji misalnya, jumlah jam setiap karyawan bekerja harus dikumpulkan sebelum gaji dapat dihitung atau dicetak. Dalam sistem peringkat universitas, instruktur harus menyerahkan nilai siswa sebelum ringkasan nilai untuk satu semester dapat dikompilasi dan dikirim ke siswa.(Ralph M. Stair, 2010)

* 1. PROSES

Dalam sistem informasi, proses berarti mengkonversi atau mengubah data menjadi output yang berguna. proses dapat melibatkan membuat perhitungan, membandingkan data dan mengambil tindakan alternatif, dan menyimpan data untuk penggunaan kedepannya. Pengolahan dapat dilakukan secara manual atau dengan bantuan komputer. Setelah perhitungan dan perbandingan yang dilakukan, hasilnya biasanya disimpan. Penyimpanan termasuk menjaga data dan informasi yang tersedia untuk digunakan di masa depan, termasuk output, dibahas berikutnya.(Ralph M. Stair, 2010)

* 1. OUTPUT

Dalam sistem informasi, output melibatkan produksi informasi yang berguna, biasanya dalam bentuk dokumen dan laporan. Output dapat mencakup gaji bagi karyawan, laporan untuk manajer, dan informasi yang diberikan kepada pemegang saham, bank, instansi pemerintah, dan kelompok-kelompok lainnya. Dalam beberapa kasus, output dari satu sistem dapat menjadi masukan bagi yang lain. Misalnya, output dari sistem yang memproses order penjualan dapat digunakan sebagai masukan untuk penagihan pelanggan sistem.(Ralph M. Stair, 2010)

* 1. UMPAN BALIK

Dalam sistem informasi, umpan balik adalah informasi dari sistem yang digunakan untuk membuat perubahan kegiatan masukan atau pengolahan. Misalnya, kesalahan atau masalah yang perlu diperbaiki dbagian input data atau mengubah suatu proses.(Ralph M. Stair, 2010)

1. Laporan Laba Rugi

Dalam literatur akuntansi, laporan laba rugi diturunkan dari stilah *profit and loss statement, earning statement, operations statement, atau income statement.* Setiap jangka waktu tertentu, perusahaan perlu memperhitungkan hasil usaha perusahaan yang dituangkan dalam bentuk laporan laba rugi. Hasil usaha didapat dengan cara membandingkan penghasilan dan biaya selama jangka waktu tertentu. Besarnya laba atau rugi akan diketahui dari hasil perbandingan tersebut. (Jumingan, 2005)

Rumus sederhana untuk menentukan laba rugi yaitu :

Laba rugi = pendapatan – biaya .............................................................. (1)

Laba = Pendapatan > biaya ................................................................(2)

Rugi = Pendapatan < biaya .................................................................(3)

(1) Rumus laporan laba rugi. (2) Rumus laba. (3) Rumus rugi.

1. Harga Pokok

Untuk menjamin bahwa suatu kegiatan usaha akan menghasilkan nilai keluaran yang lebih tinggi dari pada nilai masukan diperlukan alat untuk mengukur nilai masukan yang dikorbankan dalam menghasilkan keluaran tersebut. Informasi biaya merupakan alat yang berfungsi untuk mengukur pengorbanan nilai masukan, guna menghasilkan informasi yang bermanfaat untuk mengukur apakah kegiatan usahanya akan menghasilkan laba atau tidak. Adapun biaya - biaya yang merupakan pengorbanan sumber ekonomi untuk memperoleh aktiva disebut dengan istilah Harga Pokok*.* (Sarfi’ah, 2013)

Salah satu tujuan dari penentuan harga pokok adalah untuk menentukan biaya yang seharusnya dikeluarkan untuk memproduksi satu satuan produk. Dengan demikian perusahaan dapat menetapkan harga jual produk per unitnya, setelah mempertimbangkan biaya produksi per unit serta biaya-biaya lain yang di bebankannya yaitu dengan rumus :

Harga Pokok

keuntungan yang di inginkan tiap unit

Total Hasil Produksi

Total biaya produksi

÷ + = .................(4)

(4) Rumus Harga pokok.

1. Biaya Aset Dan Biaya Produksi

Pada sistem yang dibuat, biaya akan di bagi menjadi 2, yaitu biaya aset dan biaya produksi yang mana pembagian dua biaya ini akan memudahkan sistem untuk membuat laporan keuangan bulanan dan laporan laba-rugi.

Dari segi pengertian, kedua biaya ini berbeda dari sisi penggunaan, tujuan dan manfaat. Menurut *Financial Accounting Standards Board* (FASB) dalam Suwardjono (2005) mengatakan aset adalah manfaat ekonomik masa datang yang cukup pasti yang diperoleh atau dikuasai atau dikendalikan oleh suatu entitas sebagai akibat transaksi atau kejadian masa lalu.

Sedangkan biaya produksi dalam bahasa akuntansi biasa di sebut harga pokok produksi. Menurut mulyadi dalam Lasena (2013) mengatakan bahwa metode penentuan harga pokok produksi adalah cara perhitungan unsur-unsur biaya ke dalam harga pokok produksi. Dalam memperhitungkan unur-unsur biaya ke dalam harga pokok produksi terdapat dua pendekatan yaitu *full costing* dan *variabel costing*. *Full costing* merupakan metode penentuan harga pokok produksi yang memperhitungkan semua unsur biaya produksi ke dalam harga pokok produksi, yang terdiri dari biaya bahan baku, biaya tenaga kerja langsung, dan biaya overhead pabrik, baik yang yang berperilaku variabel maupun tetap, dengan demikian harga pokok produksi menurut *full costing* terdiri dari unsur biaya produksi.

1. Sistem Informasi Keuangan

Laporan keuangan berkaitan erat dengan bidang akuntansi. Kegiatan akuntansi pada dasarnya merupakan kegiatan mencatat, menganalisis, menyajikan, dan menafsirkan data keuangan dari lembaga perusahaan dan lembaga lainnya dimana aktifitasnya berhubungan dengan produksi dan pertukaran barang dan jasa. Bagi lembaga yang bertujuan memperoleh keuntungan, akuntansi memberikan metode untuk menentukan apakah lembaga tersebut memperoleh keuntungan atau kerugian sebagai hasil dari transaksi yang dilakukannya.

Akuntansi dapat memberi informasi tentang kondisi keuangan dari hasil operasi perusahaan seperti tercermin pada laporan keuangan yang bersangkutan. Oleh karena itu, akuntansi (laporan keuangan) dapat dipakai sebagai alat berkomunikasi dengan pihak pihak yang berkepentingan dengan data keuangan perusahaan. (Jumingan, 2005)

* 1. Peran Komputer dalam Akuntansi

Proses pengolahan data akuntansi dapat dilakukan dengan lebih cepat bila digunakan komputer. Hal ini dapat terjadi karena kemampuan komputer untuk mengolah data jauh melebihi kecepatan manusia. Dengan adanya perkembangan teknologi komputer, semakin banyak perusahaan yang mengguakan jasa komputer untuk memproses data akuntansinya. Disatu pihak, komputer merupakan alat bantu yang sangat bermanfaat dalam sistem informasi akuntansi. Akan tetapi, dipihak lain diperlukan teknik-teknik pengawasan yang berbeda dengan yang digunakan dalam cara manual untuk menjamin ketelitian dan keamanan dalam memproses data . Beberapa tahapan proses pengo;ahan data yang memperoleh manfaat besar dari penggunaan komputer adalah :

1. Verifikasi

Komputer dapat mengecek kebenaran maupun kelayakan angka-angka yang menjadi input dalam suatu proses. Misalnya pengecekan kebenaran kode yang digunakan, pengecekan kelayakan jumlah rupiah dari transaksi, dan lain-lainnya.

1. Sortir

Komputer memungkinkan untuk dilakukannya pensortiran data ke dalam beberapa klasifikasi yang berbeda dengan cepat. Misalnya, kumpulan faktur penjualan dapat disortir ke dalam klasifikasi langgananm jenis produk, daerah penjualanm dan lain sebagainya.

1. Transmission

Komputer dapat memindahkan lokasi data dari suatu tempat ke tampat lainnya dengan cepat. Misalnya , data dari suatu file dipindahkan ke file lainnya.

1. Perhitungan

Dengan komputer, perhitungan-perhitungan dapat dilakukan dengan cepat. Misalnya, menghitung saldo rekening sesudah adanya posting, menghitung jumlah sekelompok transaksi, dan lain sebagainya. (Tata Sutabsi, 2003)

1. PHP

PHP adalah kependekan dari *PHP Hypertext Prepocessor*. Bahasa interpreter yang mirip dengan bahasa C dan Perl yang dimiliki kesederhanaan dalam perintah. PHP dapat digunakan bersama dengan HTML. Sehingga memudahkan dalam pembangunan aplikasi *web* dengan cepat. PHP dapat digunakan untuk meng-*update* basis data dan menciptakan basis data. (Iswanto, 2006)

Seperti halnya program open source lainnya, PHP dibuat dibawah lisensi GNU, *General Public License,* yang dapat di-*download* gratis melalui situs <http://www.php.net>. Awalnya, PHP diciptakan oleh Andi Gutmans untuk menghitung jumlah pengunjung yang mengakses *homepage* yang dibuatnya. Seiring perkembangan internet, diliris PHP/FI, PHP2, PHP3, PHP4, dan selanjutnya PHP5 yang telah mampu membangun aplikasi web dengan koneksi basis data yang cukup banyak.

* 1. Penulisan PHP

Penulisan *script* PHP diawali dan diakhiri dengan sintaks khusus. Ada beberapa cara yang dapat digunakan untuk menuliskan *script* PHP yaitu :

1. Dengan sintaks “<?php” dan “?>”; sintaks tersebut adalah yang paling umum dari PHP.
2. Dengan sintaks “<%” dan “%>”, sintaks tersebut sebenarnya adalah sintaks dari ASP, tetapi dapat digunakan sebagai sintaks PHP apabila konfigurasi PHP pilihan asp\_tag diaktifkan.

Dengan sintaks “<SCRIPT LANGUAGE = “php” dan diakhiri dengan </SCRIPT>; sintaks itu digunakan untuk mendeklarasikan sebuah perintah pada html.

* 1. Variabel pada PHP

Variabel merupakan suatu “tempat” di dalam memori komputer yang dialokasikan untuk menyimpan data. Pada PHP, variabel ditandai dengan tanda dollar ($) dan diikuti nama variabelnya. Penamaan variabel mengikuti aturan sebagai berikut:

1. Harus diawali dengan huruf atau *underscore* ( \_ ), dapat diikuti oleh huruf, angka,*underscore,* atau sembarang karakter ASCII dari 127 hingga 255.
2. Huruf besar dan kecil dibedakan (*case sensitve*).
3. Tidak boleh mengandung spasi.

Variabel dalam PHP tidak harus di deklarasikan terlebih dahulu sebelum digunakan, tidak seperti bahasa pemograman tertentu yang harus mendeklarasikan variabel terlebih dahulu.

* 1. Fungsi pada PHP

Fungsi adalah konstruksi pemograman untuk melakukan suatu proses. Dalam pemograman, kita biasa melakukan proses secara berulang kali. Oleh karena itu, fungsi sangat membantu *programmer* untuk menangani hal tersebut supaya penulisan kode programnya lebih efisien.

Bentuk umum fungsi:

Function nama\_fungsi (argumen){

Kode perintah }

Fungsi harus didefinisikan terlebih dahulu sebelum digunakan. Pengembalian nilai dari suatu sungsi menggunakan perintan return. Semua jenis data bisa digunakan sebagai nilainya, termasuk *list*  dan objek.

1. My SQL

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 AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. (Solichin, 2010)

Fitur-fitur MySQL antara lain :

1. Relational Database System. Seperti halnya software database lain yang ada di pasaran, MySQL termasuk RDBMS.
2. Arsitektur Client-Server. MySQL memiliki arsitektur client-server dimana server database MySQL terinstal di server. Client MySQL dapat berada di komputer yang sama dengan server, dan dapat juga di komputer lain yang berkomunikasi dengan server melalui jaringan bahkan internet.
3. Mengenal perintah SQL standar. SQL (Structured Query Language)
4. Merupakan suatu bahasa standar yang berlaku di hampir semua software database. MySQL mendukung SQL versi SQL:2003.
5. Mendukung Sub Select. Mulai versi 4.1 MySQL telah mendukung select dalam select (sub select).
6. Mendukung Views. MySQL mendukung views sejak versi 5.0
7. Mendukung Stored Prosedured (SP). MySQL mendukung SP sejak versi 5.0
8. Mendukung Triggers. MySQL mendukung trigger pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan trigger pada versi 5.1.
9. Mendukung replication.
10. Mendukung transaksi.
11. Mendukung foreign key.
12. Tersedia fungsi GIS.
13. Free (bebas didownload)
14. Stabil dan tangguh
15. Fleksibel dengan berbagai pemrograman
16. Security yang baik
17. Dukungan dari banyak komunitas
18. Perkembangan software yang cukup cepat.
19. Framework LARAVEL

Framework laravel adalah framawork yang dikembangkan oleh komunitas PHP yang menggunakan konsep MVC (Model, View, Contoller). Ketika melihat di dunia pengembangan saat ini, konsep MVC digunakan dimana saja. ASP.NET, dan PHP Frameworks seperti *CakePHP* dan *Code Igniter* semua menggunakannya. Jadi yang membuat MVC penting dalam sebuah framework adalah pemisahan Logika dari layer presentasi (Views / HTML). (Dangar, 2013)

MVC memungkinkan Anda untuk menulis kode yang dapat dibagi atas dasar tiga hal:

1. Model

Model adalah cara dimana aplikasi dapat berinteraksi dengan data. Ini adalah *layer* antara data dan aplikasi. Data yang disimpan bisa dalam sistem database seperti MySQL, MSSQL, atau fies XML

1. View

Views adalah representasi visual dari aplikasi . view juga memungkinkan untuk menulis representasi dari logika proses bisnis yang ada pada aplikasi.

1. Controller

Controller adalah hubungan antara Model dan Views. Tanggung jawab utama Controller adalah untuk menangani permintaan dan melewatkan data dari Model untuk views.

**BAB III**

**METODOLOGI PENELITIAN**

1. **Subjek Penelitian**

Subjek penelitian pada tugas akhir yang berjudul “Aplikasi Sistem Informasi Keuangan Berbasis Web Menggunakan Konsep MVC (Model, View, Control) dengan Framework Laravel Untuk Mengelola Keuangan di Kelompok Petani Sayur Hidroponik di RW 12 Kauman Yogyakarta “ adalah kelompok petani hidroponik di Kauman , Yogyakarta. Bendahara kelompok petani di UMKM ini adalah pengguna utama dari aplikasi sistem informasi keuangan untuk membantu mengelola keuangan di UMKM kelompok tani RW 12 Kauman ini. Sehingga kelompok UMKM petani hidroponik di RW 12 mampu mengelola keuangan dengan baik, dapat memantau keadaan keuangan pada periode tertentu sehingga dapat menentukan harga yang pantas untuk menjual hasil panen agar tidak merugi. Aplikasi nantinya dapat digunakan oleh UMKM lain karena sistem yang dibuat adalah sistem pengelolaan keuangan yang ada pada umumnya.

1. **Metode Pengumpulan Data**

Dalam mengambil beberapa data yang dibutuhkan dalam pembuatan Aplikasi Sistem Informasi Kuangan maka dalam melakukan pengumpulan data digunakan beberapa metode sebagai berikut :

1. Metode Wawancara

Metode wawancara dilakukan dengan mewawancarai ketua kelompok petani sayur hidroponik terkait permasalahan yang akan diteliti. Karena ketua kelompok banyak mengetahui informasi tentang permasalahan yang ada pada kelompoknya. Serta wawancara dilakukan kepada bendahara kelompok. Wawancara kepada bendahara dilakukan karena mengetahui permasalahan keuangan yang ada pada kelompok serta memberi saran terkait fitur-fitur yang akan dibuat pada sistem informasi keuangan ini. Dalam wawancara berisi 9 pertanyaan yang mendasar. Pertanyaan tiap butirnya menggali informasi tentang tingkat penguasaan anggota kelompok terhadap penggunaan komputer maupun *smartphone* . serta mengukur seberapa besar manfaat sistem informasi keuangan yang akan dibuat pada penelitian ini untuk kelompok petani sayur hidroponik RW 12 Kauman.

1. Survey Lokasi

Peninjauan dilakukan dilokasi penanaman sayur hidroponik milik kelompok tani RW 12 Kauman. Untuk mengetahui lokasi penanaman sayuran hidroponik.

1. **Alat dan Bahan**
2. Perangkat keras (*Hardware*)

Perangkat keras atau *hardware* yang digunakan untuk membangun aplikasi sistem informasi keuangan ini antara lain :

1. Laptop dengan spesifikasi :
2. Intel® Core i5-450M Processor 2.4GHz, 3MB L3 Cache
3. RAM 4Gb DDR3
4. 500Gb SATA HDD
5. Intel® HD Graphics
6. 14” HD LED LCD
7. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk membuat aplikasi sistem informasi keuangan ini antara lain :

1. Sistem Operasi : Windows 10
2. Bahasa pemograman : *PHP v5.6.15, HTML 5.*
3. *Case Tools :* Sublime text 3, Framework LARAVEL 5, MySQL 4.5.1.
4. **Analisis Kebutuhan Sistem**

Analisis kebutuhan sistem ini merupakan tahapan untuk menentukan spesifikasi kebutuhan sistem (*system requirement spesification*). Pada tahap ini didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Dalam analisis kebutuhan sistem, mengunakan *use case diagram.* Yang bertujuan untuk menggambarkan proses yang dapat dilakukan oleh *user.*

1. **Desain Sistem (Pemodelan)**

Setelah dibuat gambaran yang jelas tentang sistem yang akan dibangun, selanjutnya diakukan desain sistem(Pemodelan). Tahap desain sistem ini dilakukan untuk memberikan gambaran umum yang jelas kepada pengguna dan rancang bangun yang lengkap tentang sistem yang akan dikembangkan kepada pihak-pihak yang terlibat dalam pengembangan sistem ini. Adapun perancangan dibagi menjadi :

1. Desain Aktfitas

Desain aktifitas digunakan untuk menggabarkan proses yang terjadi pada aplikasi sistem informasi keuangan . *Activity diagram* digunakan untuk menggambarkan alur proses bisnis dan ururtan aktifitas dalam sistem.

1. Desain Data

Desain data digukana untuk manggembarkan pemodelan basis data yang ada pada aplikasi sistem informasi keuangan. *Entity Relationship Diagram (ERD)* digukana dalam tahap ini untuk memodelkan struktur data dan hubungan antar tabel yang ada dalam sistem informasi keuangan ini.

1. Desain Antarmuka

Perancangan antarmuka bertujuan untuk memberikan gambaran aplikasi sistem informasi keuangan kepada *user* . perancangan antarmuka meliputi struktur menu, inputm dan output aplikasi, dan fungsi-fungsi yang ada didalam aplikasi. Hasil dari desain antarmuka ini adalah *prototype* aplikasi sistem informasi keuangan.

1. Arsitektur Sistem

Arsitektur aplikasi sistem informasi keuangan ini memanfaatkan internet untuk menjalankan fungsinya seperti : mengakses data, menginput data serta melihat informasi hasil dari pengolahan sistem. Basis data yang disimpan di server digunakan untuk menyimpan seluruh data dan dapat diakses kapan saja selama terkoneksi internet.

1. **Implementasi**

Implementasi Aplikasi Sistem Informasi Keuangan ini dibangun menggunakan *framework* laravel 5.1 dengan bahasa pemograman *HTML* dan *PHP* yang berbasis web. Dengan beberapa bahasa pemograman yang menambah daya tarik tampilan web antara lain *Bootstrap, Javascript.* Data aplikasi ini disimpan di web server salah satu penyedia jasa hosting. Dan *database* yang dibuat menggunakan MySQL. Pembangunan web dengan *framework* Laravel menggunakan konsep MVC yaitu :

1. Model

Yaitu berisi *class* *php*  yang menghubungkan *controller* dengan *database MySql.* Selain itu model juga dapat memproteksi atau mengizinkan *controller* dalam mengakses database, *query* data dari *database* untuk disampaikan ke *controller,* serta mengatur integritas dan konsistensi data.

1. View

Yaitu berisi *script* yang digunakan untuk menerima dan mengirim data dari *controller* dan membuat tampilan aplikasi yang mudah dibaca langsung oleh *user*.

1. Controller

yaitu kode yang bertugas menerima data, validasi data, membuat data baru dan memperbaharui data. Selain itu juga dapat memodifikasi kueri *database*. Semua kode disusun di masing- masing *function* sesuai dengan yang diminta oleh *view.* Di *controller*  juga berisi kondisi yang digunakan untuk menyimpan setiap data yang masuk.

Selain dari konsep MVC, terdapat satu lagi konsep yang cukup penting di *framework* laravel yaitu *routing. Routing* berfungsi menyimpan daftar alamat yang di kirim oleh aplikasi dan setiap alamat memiliki *request* pengembalian berbeda. Request yang dipanggil kemudian mengarahkan aplikasi untuk memanggil  *view* tertentu.

1. **Pengujian Sistem**

Untuk melakukan pengujian sistem terhadap aplikasi sistem informasi keuangan, metode yang digunakan antara lain :

1. Pengujian *Black Box*

Pengujian *black box* dilakukan untuk menguji antarmuka aplikasi, menyediakan input dan memeriksa outputnya, dan untuk menguji fitur-fitur yang ada pada aplikasi sudah berjalan dengan benar. Pengujian ini dilakukan oleh pengguna dengan berinteraksi langsung dengan aplikas dan menjalankan beberapa fitur. Hal ini bertujuan untuk mengetahui apakah fitur-fitur yang ada sudah bekerja dengan baik dan tepat.

1. T-Test

Pengujian T-Test dilakukan dengan tujuan untuk membandingkan saat pengguna sebelum dan sesudah menggunakan aplikasi yang dibuat. Hasil pretest dan postest akan dibuat grafik untuk menggambarkan tingkat kepuasan user dengan aplikasi yang dibuat. Pretest dilakukan sebelum *user* menggunakan aplikasi yang dibuat, dalam hal ini Aplikasi sistem informasi keuangan. pengguna diberikan daftar pertanyaan seperti kuisioner terkait dengan aplikasi yang dibangun. Setelah pengguna selesai mengisi kuisioner tersebut, pengguna dipersilahkan untuk mencoba aplikasi yang dibuat. Setelah pengguna selesai memakai aplikasi tersebut, maka postest mulai dilakukan. pengguna diberikan kuisioner yang sama seperti pada saat melakukan pretest. Jawaban pretest dan postest dikumpulkan untuk dilakukan perbandingan antar keduanya.

**Bab IV**

**Hasil Dan Pembahasan**

1. Analisis Kebutuhan Sistem

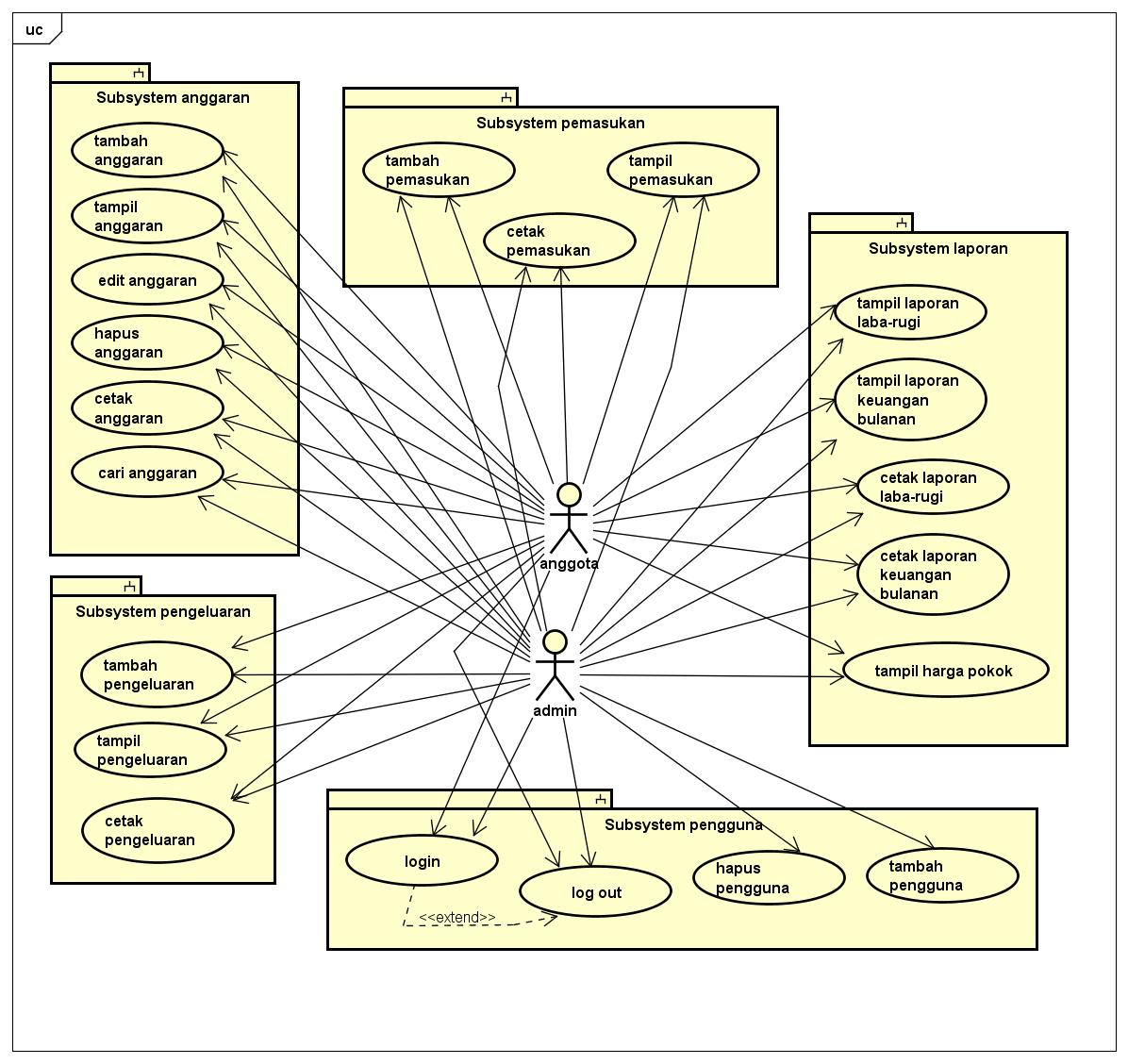
Dari hasil pengumpulan data yang dilakukan dengan metode wawancara, di dapatkan data-data yang yang diperlukan untuk membangun sistem informasi keuangan berbasis web. Data yang telah didapat kemudian di analisis dan menghasilkan spesifikasi kebutuhan sistem (*system requirement spesification*).

Bedasarkan hasil analisis kebutuhan sistem di kelompok petani hidroponik ASRI 12 Kauman, diperoleh *system requirement spesification (*SRS*)* pada tabel 4.1.

Tabel 4.1 SRS Fungsional

|  |  |
| --- | --- |
| Kode SRS | Deskripsi |
| SRS-F-01 | Aplikasi mempunyai 2 hak akses pengguna yaitu Admin dan Anggota. |
| SRS-F-02 | Aplikasi mampu membuat dan menghapus data pengguna yang hanya bisa di akses oleh admin. |
| SRS-F-03 | Aplikasi mampu membuat, menampilkan, mengubah, menghapus dan mencetak data anggaran keuangan yang bisa di akses admin dan anggota. |
| SRS-F-4 | Aplikasi mampu membuat, menampilkan, dan mencetak data pemasukan keuangan yang bisa di akses admin dan anggota. |
| SRS-F-05 | Aplikasi mampu membuat, menampilkan, dan mencetak data pegeluaran keuangan yang bisa di akses admin dan anggota. |
| SRS-F-06 | Aplikasi mampu menampilkan dan mencetak laporan laba rugi yang bisa di akses admin dan anggota. |
| SRS-F-07 | Aplikasi mampu menampilkan dan mencetak laporan keuangan bulanan yang bisa di akses admin dan anggota. |
| SRS-F-08 | Aplikasi mampu menampilkan harga pokok yang bisa di akses admin dan anggota. |

Setelah menganalisis setiap SRS (*system requirement system*) maka dapat didefinisikan kebutuhan yang harus dipenuhi oleh sistem yang dibangun menggunakan diagram *use case* yang bertujuan untuk menggambarkan proses apa saja yang bisa dilakukan *user*. Diperoleh diagram *use case* pada gambar 4.1.



Gambar 4.1. *use case diagram*

Bedasarkan *use case* yang telah di buat , maka penjelasan skenario *use case* sebagai berikut :

1. Login

Tabel 4.2 *use case scenario* login

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Login | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* login digunakan untuk memeriksa keamanan dan memastikan bahwa yang mengakses aplikasi adalah aktor yang mendapatkan hak akses | |
| *Pre Contidion* | Admin dan anggota memasukan email dan password | |
| *Trigger* | saat pengguna ingin mengakses aplikasi sesuai dengan hak akses | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Masukkan email dan password | * Cek email dan password * Menampilkan halaman dashboard |
| *Alternate Course* | Jika email dan password salah maka aplikasi menampilkan pesan kesalahan dan kembali ke halaman login | |
| *Conclusion* | User login ke dalam aplikasi sistem informasi keuangan | |
| *Post Condition* | Aplikasi berhasil di akses | |

1. tambah pengguna

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Tambah Pengguna | |
| *Actor* | Admin | |
| *Description* | *Use case* input pengguna menggambarkan saat admin akan menambahkan pengguna aplikasi sistem informasi keuangan. | |
| *Pre Contidion* | actor menambahkan informasi pengguna | |
| *Trigger* | saat actor ingin menambah pengguna sistem | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Masukkan nama * Masukan email * Masukan password * Masukan konfirmasi password * Memilih jabatan * Klik tombol “tambah” | * Cek form yang kosong * Cek kecocokan password * Menampilkan halaman tabel pengguna |
| *Alternate Course* | * Jika ada form yang kosong, maka akan muncul pesan kesalahan * Jika pada form email tidak diisi format email, maka muncul pesan kesalahan. | |
| *Conclusion* | Pengguna berhasil di tambahkan | |
| *Post Condition* | Data pengguna baru disimpan ke tabel pengguna | |

Tabel 4.3 *use case scenario* input pengguna

1. Tambah anggaran

Tabel 4.4 *use case scenario* input anggaran

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | tambah anggaran | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* input anggaran menggambarkan saat pengguna menambahkan data anggaran | |
| *Pre Contidion* | Actor memasukan data anggaran | |
| *Trigger* | - | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Memasukan data detail anggaran * Klik ‘simpan data’ | * Memeriksa form yang kosong * Menyimpan ke database * Meredirect ke halaman tabel anggaran |
| *Alternate Course* | * Jika ada form yang kosong maka akan muncul pesan form diperlukan * Klik ‘kembali’ jika ingin membatalkan menambah anggaran | |
| *Conclusion* | Data anggaran berhasil ditambahkan | |
| *Post Condition* | Data anggaran berhasil di simpan tabel anggaran | |

1. Edit anggaran

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Edit anggaran | |
| *Actor* | Admin ,Anggota | |
| *Description* | *Use case* edit anggaran digunakan untuk menggambarkan proses mengedit data anggaran yang sudah ada | |
| *Pre Contidion* | Data anggaran yang di edit harus sudah ada sebelumnya | |
| *Trigger* | jika ingin mengubah data anggaran. | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Klik tombol ‘edit’ * Mengganti data di form yang ingin diubah * Klik tombol ‘simpan data’ | * Memeriksa form yang kosong * Menyimpan ke database * Meredirect ke halaman tabel anggaran |
| *Alternate Course* | * Jika ada form yang kosong maka akan muncul pesan form diperlukan * Klik ‘kembali’ jika ingin membatalkan mengubah anggaran | |
| *Conclusion* | Data anggaran berhasil di ubah | |
| *Post Condition* | Data anggaran yang diubah sudah di perbaharui di tabel anggaran | |

Tabel 4.5 *use case scenario* edit anggaran

1. Cetak anggaran

Tabel 4.6 *use case scenario* edit anggaran

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Cetak anggaran | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* menggambarkan sistem mencetak data anggaran ke dalam bentuk PDF | |
| *Pre Contidion* | Actor memilih periode anggaran yang ingin di cetak | |
| *Trigger* | Jika actor ingin mencetak data anggaran dalam bentuk PDF | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Memilih tahun periode * Memilih bulan periode * Klik tombol ‘cetak PDF’ | * Men*download* data anggaran dalam bentuk PDF |
| *Alternate Course* |  | |
| *Conclusion* | Data anggaran berhasil dicetak sesuai periode yang dipilih | |
| *Post Condition* | Dokumen anggaran sesuai periode berhasil dicetak | |

1. Cari anggaran

Tabel 4.7 *use case scenario* cari anggaran

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Cari anggaran | |
| *Actor* | Admin , Anggota | |
| *Description* | *Use case* menggambarkan proses pencarian data anggaran bedasarkan *keyword* | |
| *Pre Contidion* | Data anggaran sudah ada sebelumnya di tabel anggaran | |
| *Trigger* | Jika actor ingin mencari item anggaran | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Mengisi form pencarian * Klik tombol ‘cari item’ | * Mencari *keyword* yang dicari di tabel anggaran * Menampilkan data anggaran yang sesuai atau mirip dengan *keyword* |
| *Alternate Course* | Jika *keyword* tidak ditemukan, maka sistem menampilkan tabel kosong | |
| *Conclusion* | Data anggaran berhasil ditemukan | |
| *Post Condition* | Data anggaran berhasil ditemukan sesuai *keyword*  yang dicari | |

1. tambah pengeluaran

Tabel 4.8 *use case scenario* tambah pengeluaran

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Tambah pengeluaran | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* tambah pengeluaran menggambarkan saat pengguna menambahkan data pengeluaran | |
| *Pre Contidion* | Actor memasukan data pengeluaran | |
| *Trigger* | Saat actor ingin menambah data pengeluaran | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Memasukan data detail pengeluaran * Memilih jenis pengeluaran * Klik ‘simpan data’ | * Memeriksa form yang kosong * Jika jenis pengeluaran = produksi, maka data juga disimpan ke tabel laba-rugi * Menyimpan ke tabel pengeluaran dan transaksi * Meredirect ke halaman tabel pengeluaran |
| *Alternate Course* | Jika ada form yang kosong maka akan muncul pesan form diperlukan  Klik ‘kembali’ jika ingin membatalkan menambah pengeluaran | |
| *Conclusion* | Data pengeluaran berhasil ditambahkan | |
| *Post Condition* | Data pengeluaran berhasil di simpan tabel pengeluaran | |

1. Tampil pengeluaran

Tabel 4.9 *use case scenario* tampil pengeluaran

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Tampil pengeluaran | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* tampil pengeluaran menjelaskan saat sistem menampilkan data pengeluaran sesuai periode yang di pilih | |
| *Pre Contidion* | Actor memilih periode | |
| *Trigger* | Saat actor ingin menampilkan tabel pengeluaran pada periode yang diinginkan | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Memilih tahun periode * Memilih bulan periode * Klik ‘tampilkan’ | * Menampilkan tabel pengeluaran sesuai dengan periode yang dipilih. |
| *Alternate Course* | Jika actor tidak memilih tahun dan bulan periode, maka sistem menampilkan seluruh data pengeluaran di tabel pengeluaran. | |
| *Conclusion* | Data pengeluaran berhasil ditampilkan | |
| *Post Condition* | Data pengeluaran berhasil ditampilkan sesuai periode | |

1. Cetak pengeluaran

Tabel 4.10 *use case scenario* cetak pengeluaran

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | cetak pengeluaran | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* cetak pengeluaran menjelaskan saat sistem mencetak data pengeluaran sesuai periode yang di pilih | |
| *Pre Contidion* | Actor memilih periode | |
| *Trigger* | Saat actor ingin mencetak tabel pengeluaran pada periode yang diinginkan | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Memilih tahun periode * Memilih bulan periode * Klik ‘Cetak PDF | * Men*download* data pengeluaran dalam bentuk PDF |
| *Alternate Course* | Jika actor tidak memilih tahun dan bulan periode, maka sistem mencetak pengeluaran pada periode sekarang. | |
| *Conclusion* | Data pengeluaran berhasil dicetak | |
| *Post Condition* | Data pengeluaran berhasil dicetak sesuai periode | |

1. tambah pemasukan

Tabel 4.11 *use case scenario* tambah pemasukan

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | tambah pemasukan | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* tambah pemasukan menggambarkan saat pengguna menambahkan data pemasukan | |
| *Pre Contidion* | Actor memasukan data pemasukan | |
| *Trigger* | Saat actor ingin menambah data pemasukan | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Memasukan data detail pemasukan * Memilih jenis pemasukan * Klik ‘simpan data’ | * Memeriksa form yang kosong * Jika jenis pemasukan = produksi, maka data juga disimpan ke tabel laba-rugi * Menyimpan ke tabel pemasukan dan transaksi * Meredirect ke halaman tabel pemasukan |
| *Alternate Course* | Jika ada form yang kosong maka akan muncul pesan form diperlukan  Klik ‘kembali’ jika ingin membatalkan menambah data pemasukan | |
| *Conclusion* | Data pemasukan berhasil ditambahkan | |
| *Post Condition* | Data pemasukan berhasil disimpan tabel pemasukan | |

1. Tampil pemasukan

Tabel 4.12 *use case scenario* tampil pemasukan

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Tampil pemasukan | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* tampil pemasukan menjelaskan saat sistem menampilkan data pemasukan sesuai periode yang di pilih | |
| *Pre Contidion* | Actor memilih periode | |
| *Trigger* | Saat actor ingin menampilkan tabel pemasukan pada periode yang diinginkan | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Memilih tahun periode * Memilih bulan periode * Klik ‘tampilkan’ | * Menampilkan tabel pemasukan sesuai dengan periode yang dipilih. |
| *Alternate Course* | Jika actor tidak memilih tahun dan bulan periode, maka sistem menampilkan seluruh data pemasukan di tabel pengeluaran periode saat ini. | |
| *Conclusion* | Data pemasukan berhasil ditampilkan | |
| *Post Condition* | Data pemasukan berhasil ditampilkan sesuai periode | |

1. Cetak pemasukan

Tabel 4.13 *use case scenario* cetak pemasukan

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | cetak pemasukan | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* cetak pemasukan menjelaskan saat sistem mencetak data pemasukan sesuai periode yang di pilih | |
| *Pre Contidion* | Actor memilih periode | |
| *Trigger* | Saat actor ingin mencetak tabel pemasukan pada periode yang diinginkan | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Memilih tahun periode * Memilih bulan periode * Klik ‘Cetak PDF | * Men*download* data pemasukan dalam bentuk PDF |
| *Alternate Course* | Jika actor tidak memilih tahun dan bulan periode, maka sistem mencetak pemasukan pada periode sekarang. | |
| *Conclusion* | Data pemasukan berhasil dicetak | |
| *Post Condition* | Data pemasukan berhasil dicetak sesuai periode | |

1. Tampil laporan laba-rugi

Tabel 4.14 *use case scenario* tampil laporan laba-rugi

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Tampil laporan laba-rugi | |
| *Actor* | Admin , Anggota | |
| *Description* | *Use case* tampil laporan laba-rugi menjelaskan saat sistem menampilkan tabel laba-rugi sesuai periode yang dipilih | |
| *Pre Contidion* | Sudah ada data di pengeluaran dan pemasukan yang berjenis ‘produksi’ sebelumnya. | |
| *Trigger* | Saat actor ingin melihat laporan laba-rugi setiap periode | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Pilih tahun periode * Pilih bulan periode * Klik ‘tampilkan’ | * Menampilkan tabel laba rugi sesuai periode yang di pilih * Menampilkan jumlah laba-rugi di form laba-rugi |
| *Alternate Course* | Jika actor tidak memilih tahun dan bulan periode, maka sistem akan menampilkan tabel laba rugi pada periode sekarang | |
| *Conclusion* | data laba-rugi berhasil ditampilkan | |
| *Post Condition* | data laba-rugi berhasil ditampilkan sesuai periode yang di pilih | |

1. Tampil laporan keuangan bulanan

Tabel 4.15 *use case scenario* tampil laporan keuangan bulanan

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Tampil laporan keuangan bulanan | |
| *Actor* | Admin , Anggota | |
| *Description* | *Use case* tampil laporan keuangan bulanan menjelaskan saat sistem menampilkan tabel keuangan bulanan sesuai periode yang dipilih | |
| *Pre Contidion* | Sudah ada data pengeluaran dan pemasukan sebelumnya. | |
| *Trigger* | Saat actor ingin melihat laporan keuangan bulanan setiap periode | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Pilih tahun periode * Pilih bulan periode * Klik ‘tampilkan’ | * Menampilkan tabel keuangan bulanan sesuai periode yang di pilih * Menampilkan jumlah saldo saat ini. |
| *Alternate Course* | Jika actor tidak memilih tahun dan bulan periode, maka sistem akan menampilkan tabel keuangan bulanan pada periode sekarang | |
| *Conclusion* | data keuangan bulanan berhasil ditampilkan | |
| *Post Condition* | data keuangan bulanan berhasil ditampilkan sesuai periode yang di pilih | |

1. Cetak laporan laba-rugi

Tabel 4.16 *use case scenario* cetak laporan laba-rugi

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | cetak laporan laba-rugi | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* cetak laporan laba-rugi menjelaskan saat sistem mencetak data laporan laba-rugi sesuai periode yang di pilih | |
| *Pre Contidion* | Actor memilih periode | |
| *Trigger* | Saat actor ingin mencetak tabel laporan laba-rugi pada periode yang diinginkan | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Memilih tahun periode * Memilih bulan periode * Klik ‘Cetak PDF | * Men*download* data pemasukan dalam bentuk PDF |
| *Alternate Course* | Jika actor tidak memilih tahun dan bulan periode, maka sistem mencetak laporan laba-rugi pada periode sekarang. | |
| *Conclusion* | Data laporan laba-rugi berhasil dicetak | |
| *Post Condition* | Data laporan laba-rugi berhasil dicetak sesuai periode | |

1. Cetak laporan keuangan bulanan

Tabel 4.17 *use case scenario* cetak laporan keuangan bulanan

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | cetak laporan keuangan bulanan | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* cetak laporan keuangan bulanan menjelaskan saat sistem mencetak data laporan keuangan bulanan sesuai periode yang di pilih | |
| *Pre Contidion* | Actor memilih periode | |
| *Trigger* | Saat actor ingin mencetak tabel laporan keuangan bulanan pada periode yang diinginkan | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Memilih tahun periode * Memilih bulan periode * Klik ‘Cetak PDF | * Men*download* data pemasukan dalam bentuk PDF |
| *Alternate Course* | Jika actor tidak memilih tahun dan bulan periode, maka sistem mencetak laporan keuangan bulanan pada periode sekarang. | |
| *Conclusion* | Data laporan keuangan bulanan berhasil dicetak | |
| *Post Condition* | Data laporan keuangan bulanan berhasil dicetak sesuai periode | |

1. Tampil harga pokok

Tabel 4.18 *use case scenario* tampil harga pokok

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Tampil harga pokok | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* tampil harga pokok menjelaskan proses menampilkan harga pokok | |
| *Pre Contidion* | Sudah ada data di tabel pengeluaran berjenis ‘produksi’ | |
| *Trigger* | Saat actor ingin menampilkan harga pokok produk yang akan di jual sesuai dengan pengeluaran produksi | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Memilih tahun periode pengeluaran produksi * Memilih bulan periode pengeluaran produksi * Mengisi form total hasil panen * Memilih margin * Klik ‘tampil harga pokok’ | * Menghitung harga pokok denga rumus penentuan harga pokok * Menampilkan harga pokok di form harga pokok |
| *Alternate Course* | Jika actor tidak memilih tahun dan bulan periode pengeluaran produksi, maka sistem menghitung bedasarkan harga pokok bedasarkan pengeluaran produksi pada periode sekarang. | |
| *Conclusion* | Harga pokok berhasil ditampilkan | |
| *Post Condition* | Harga pokok berhasil ditampilkan | |

1. Tampil anggaran

Tabel 4.19 *use case scenario* tampil anggaran

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Tampil anggaran | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* tampil anggaran menjelaskan proses menampilkan data anggaran sesuai periode yang diinginkan | |
| *Pre Contidion* | Sudah ada data di tabel anggaran | |
| *Trigger* | Saat actor ingin menampilkan anggaran per periode | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Memilih tahun periode * Memilih bulan periode * Klik ‘tampilkan’ | * Merequest database untuk menampilkan data anggaran sesuai tahun dan bulan * Menampilkan data anggaran |
| *Alternate Course* | Jika actor tidak memilih tahun dan bulan periode anggaran, maka sistem akan menampilkan data anggaran pada periode sekarang. | |
| *Conclusion* | Data anggaran berhasil ditampilkan | |
| *Post Condition* | Data anggaran berhasil ditampilkan sesuai periode. | |

1. Hapus anggaran

Tabel 4.20 *use case scenario* hapus anggaran

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Hapus anggaran | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* hapus anggaran menjelaskan proses menghapus data anggaran | |
| *Pre Contidion* | Tampil data tabel anggaran | |
| *Trigger* | Saat actor ingin menghapus salah satu data anggaran | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Klik ‘hapus’ pada data yang ingin di hapus | * Meminta *id* anggaran yang di hapus * Menghapus data sesuai *id* yang di minta |
| *Alternate Course* |  | |
| *Conclusion* | Dat anggaran berhasil dihapus | |
| *Post Condition* | Data anggaran berhasil dihapus | |

1. Hapus pengguna

Tabel 4.21 *use case scenario* hapus pengguna

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Hapus pengguna | |
| *Actor* | Admin | |
| *Description* | *Use case* hapus pengguna menjelaskan proses menghapus pengguna aplikasi | |
| *Pre Contidion* | Tampil data tabel pengguna | |
| *Trigger* | Saat actor ingin menghapus salah satu data pengguna | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Klik ‘hapus’ pada pengguna yang ingin di hapus | * Meminta *id* pengguna yang di hapus * Menghapus data sesuai *id* yang di minta |
| *Alternate Course* |  | |
| *Conclusion* | Dat pengguna berhasil dihapus | |
| *Post Condition* | Data pengguna berhasil di hapus dan pengguna yang dihapus tidak bisa login | |

1. Logout

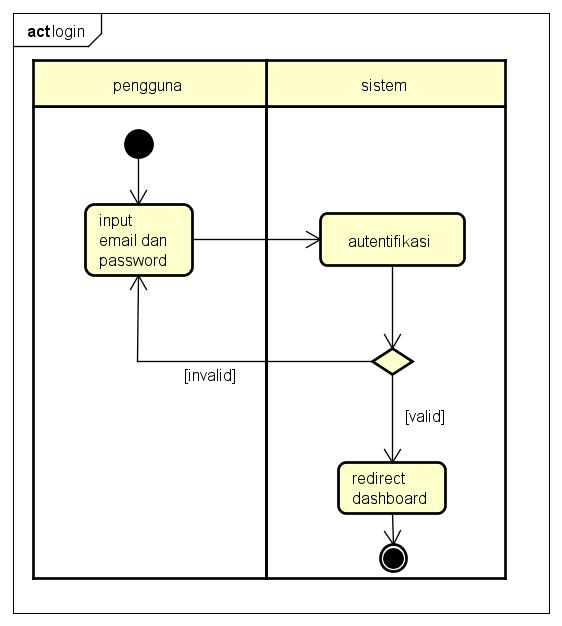
Tabel 4.22 *use case scenario* login

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Logout | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* logout digunakan untuk keluar dari aplikasi dan memastikan pengguna diluar hak akses tidak bisa mengakses aplikasi | |
| *Pre Contidion* | Actor dalam keadaan login | |
| *Trigger* | Saat pengguna ingin mengakses aplikasi sesuai dengan hak akses | |
| *Typical Course of Events* | *Actor action* | *System Response* |
|  | * Klik ‘logout’ | * Menutup akses aplikasi * Redirect ke halaman login |
| *Alternate Course* | - | |
| *Conclusion* | Actor keluar dari aplikasi | |
| *Post Condition* | Actor keluar dari aplikasi | |

1. Desain Sistem (Pemodelan)
2. Desain Aktifitas

Setelah mendefinisikan setiap *use case* dengan detail, tahap selanjutnya adalah menggambarkan *workflow* atau aliran kerja dengan *activity diagram*. Yaitu diagram yang menggambarkan alur aktifitas setiap *use case* yang dilakukan oleh sistem.

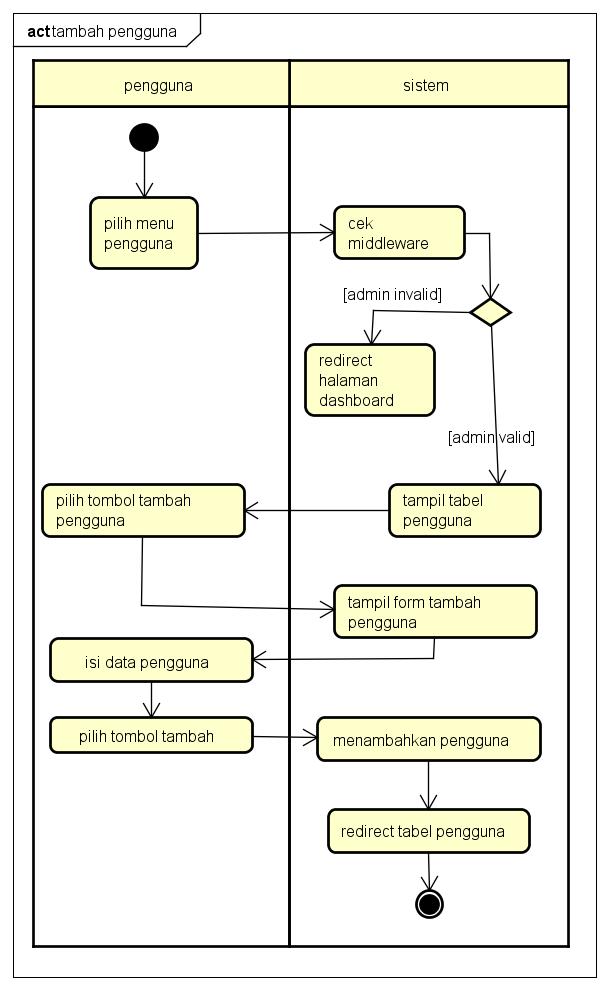
* 1. *Activity diagram* login



Gambar 4.2. *activity diagram login*

Aktivitas pada gambar 4.2 dilakukan oleh actor Admin, Anggota melakukan pengisian *email* dan *password* pada menu *login*. Kemudian sistem mengecek ketersediaan pengguna di tabel pengguna. Jika valid maka sistem mengarahkan ke halaman *dashboard,* tetapi jika tidak valid maka sistem mengarahkan ke halaman login kembali dan memunculkan pesan kesalahan.

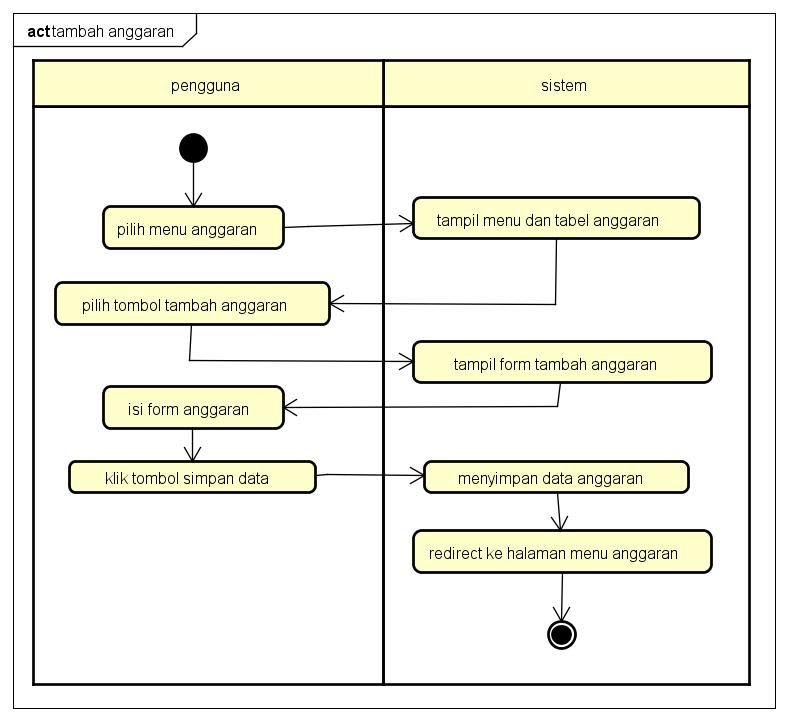
* 1. *Activity diagram* tambahpengguna

**

Gambar 4.3 *Activity diagram* tambahpengguna

Aktifitas tambah pengguna hanya bisa dilakukan oleh admin. Setelah login sebagai admin , pilih menu pengguna kemudian sistem akan mengecek *middeware* atau hak akses apakah login sebagai admin atau bukan. Jika bukan maka akan kembali ke halaman dashboard. Kemudian klik tombol tambah pengguna dan sistem akan menampilkan form tambah pengguna. Actor kemudian mengisi username, email, password, confirm password dan jabatan. Kemudian setelah selesai klik tombol tambah. Maka pengguna berhasil di tambahkan.

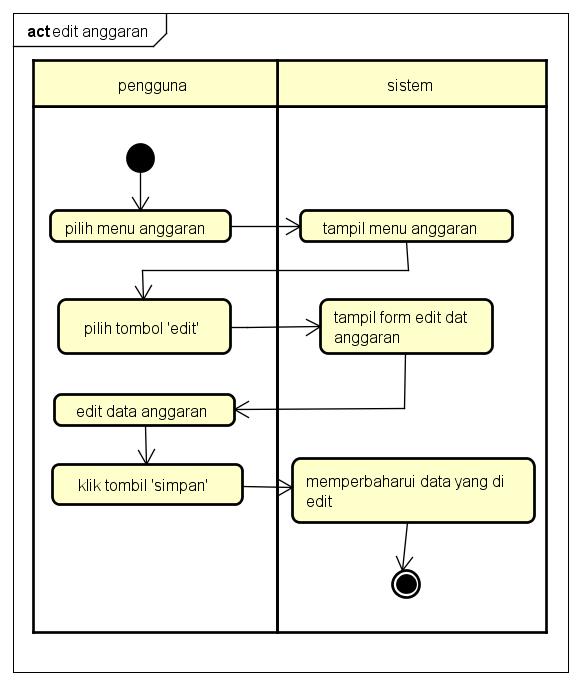
* 1. *Activity diagram* tambah anggaran

**

Gambar 4.4 *Activity diagram* tambah anggaran

Aktifitas tambah anggaran dimulai dengan memilih menu anggaran kemudian pilih tombol tambah anggaran. Kemudian actor mengisi data anggaran yaitu tanggal anggaran, nama item anggaran, harga satuan, QTY, anggaran. Selanjutnya klik tombol simpan data. Dan sistem akan meredirect ke halaman menu anggaran.

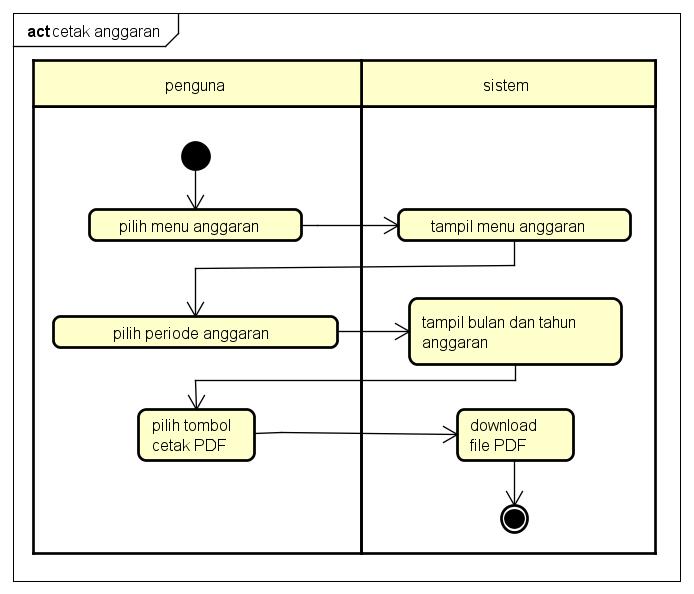
* 1. *Activity diagram* edit anggaran



Gambar 4.5 *Activity diagram* edit anggaran

Aktifitas edit anggaran dilakukan saat pengguna ingin mengubah data anggaran. Dimulai dari memilih menu anggaran , kemudian memilih data yang ingin di edit, klik tombol ‘edit’, maka sistem akan menampilkan halaman edit anggaran yang sama dengan halaman saat menambah halaman. Setelah selesai mengubah data, klik tombol ‘simpan’. Dan sistem akan memperbaharui data dan meredirect ke halaman utama menu anggaran.

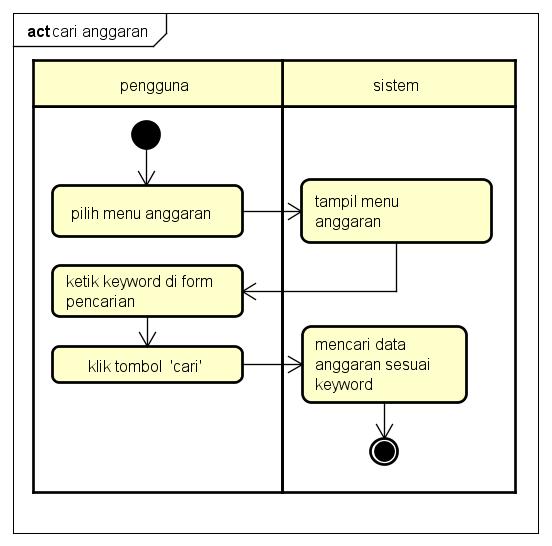
* 1. *Activity diagram* cetak anggaran

**

Gambar 4.6 *Activity diagram* cetak anggaran

Aktivitas cetak anggaran menjelaskan alur saat pengguna akan mencetak anggaran. Dimulai dari memilih menu angaran, kemudian memilih periode anggaran terdiri tahun dan bulan. Kemudian pilih tombol cetak PDF. Maka sistem akan mendownload data anggaran sesuai periode yang dipilih.

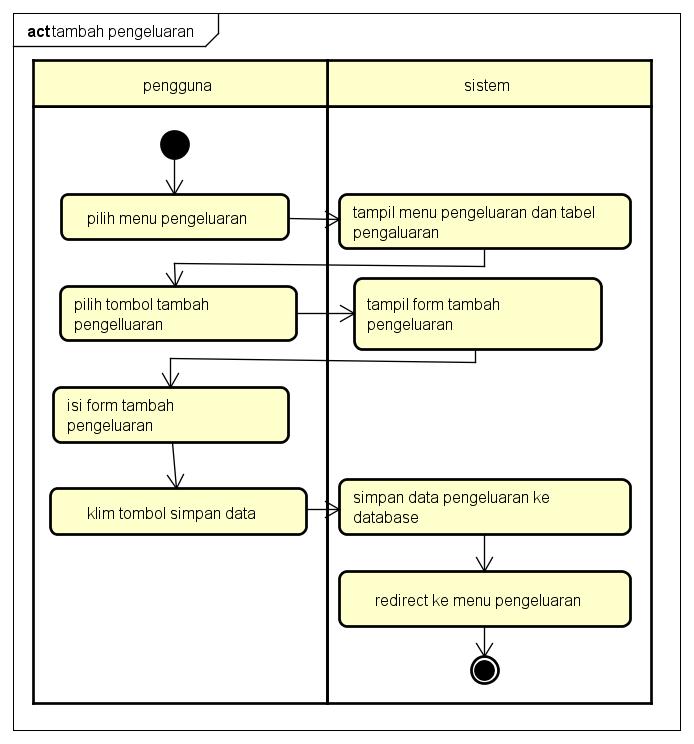
* 1. *Activity diagram* cari anggaran

**

Gambar 4.7 *Activity diagram* cari anggaran

Aktifitas cari anggaran menggambarkan alur kerja sistem saat pengguna akan mencari data anggaran. Di mulai dari memilih menu anggaran, kemudian mengetikan *keyword* di form pencarian. Kemudian klik tombol cari. Maka sistem akan menampilkan data item anggaran yang mirip dengan keyword.

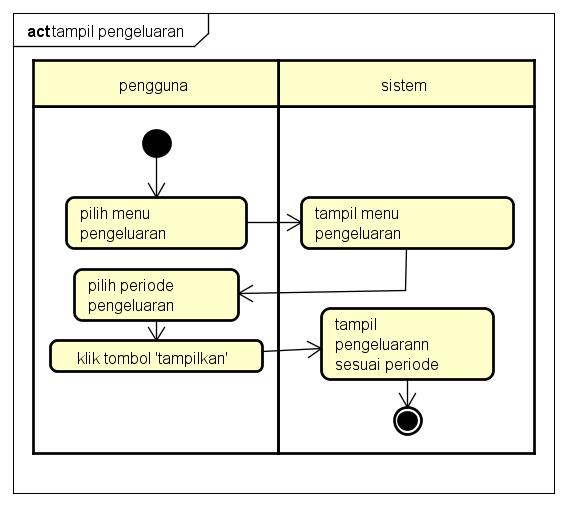
* 1. *Activity diagram* tambah pengeluaran



Gambar *4.8 Activity diagram* tambah pengeluaran

Gambar 4.8 menjelaskan aliran kerja aktifitas sistem saat pengguna akan menambahkan data pengeluaran. Dimulai dari memilih menu pengeluaran, maka sistem akan menampilkan menu anggara. Kemudian pengguna memilih tombol ‘Tambah Pengeluaran’. Dan sistem akan menampilkan form berisi detail pengeluaran yaitu, tanggal transaksi, harga satuan, QTY, deskripsii, dan jumlah pengeluaran. Kemudian pilih tombol ‘tambah’. Maka sistem akan menambahkan data pengeluaran.

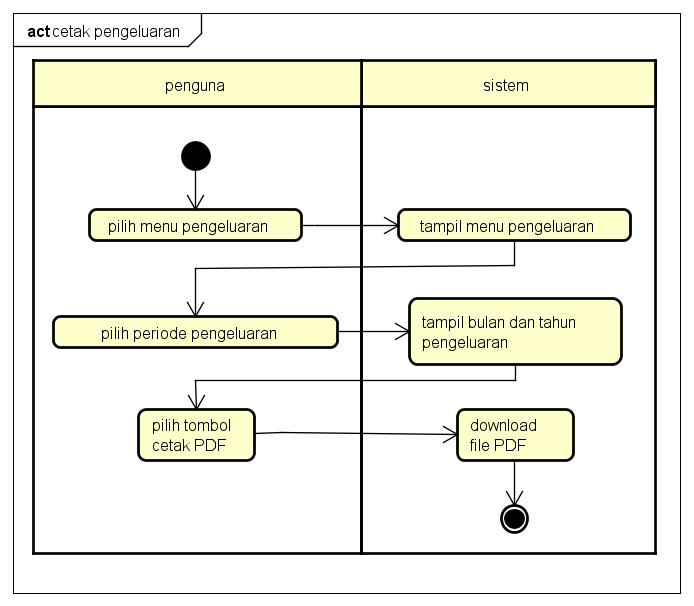
* 1. *Activity diagram* tampil pengeluaran



gambar 4.9 *Activity diagram* tampil pengeluaran

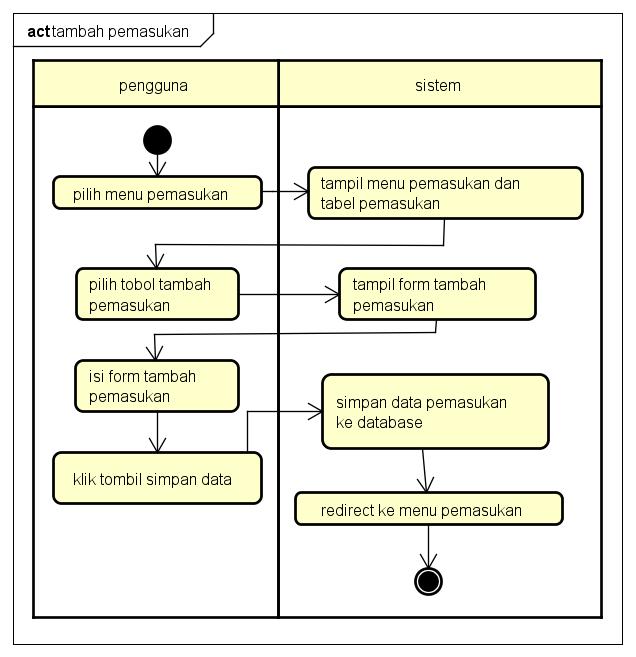
Aktifitas tampil pengeluaran menjelaskan alur sistem dalam menampilkan data pengeluaran. dimulai dari pengguna memilih menu pengeluaran, kemudian sistem akan menampilkan menu pangeluaran. Pengguna memilih periode pengeluaran terdiri dari tahun dan bulan. Kemudian pilih tombol tampilkan. Maka sistem akan menampilkan data pengeluaran sesuai periode yang dipilih.

* 1. *Activity diagram* cetak pengeluaran

** Gambar 4.10 *Activity diagram* cetak pengeluaran

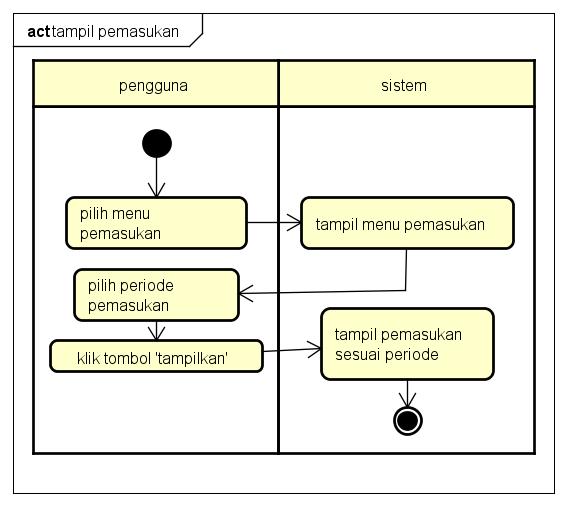
Aktifitas cetak pengeluaran menjelaskan alur saat sistem mencetak data pengeluaran dalam bentuk PDF. Dimulai dari pengguna memilih menu pengeluaran. Kemudian sistem akan menampilkan menu pengeluaran. Pengguna memilih periode pengeluaran yang akan dicetak terdiri dari tahun dan bulan. Kemudian pilih tombol ‘cetak PDF’. Maka sistem akan men­-*download* data pengeluaran dalam bentuk PDF.

* 1. *Activity diagram* tambah pemasukan

** Gambar 4.11 *Activity diagram* tambah pemasukan

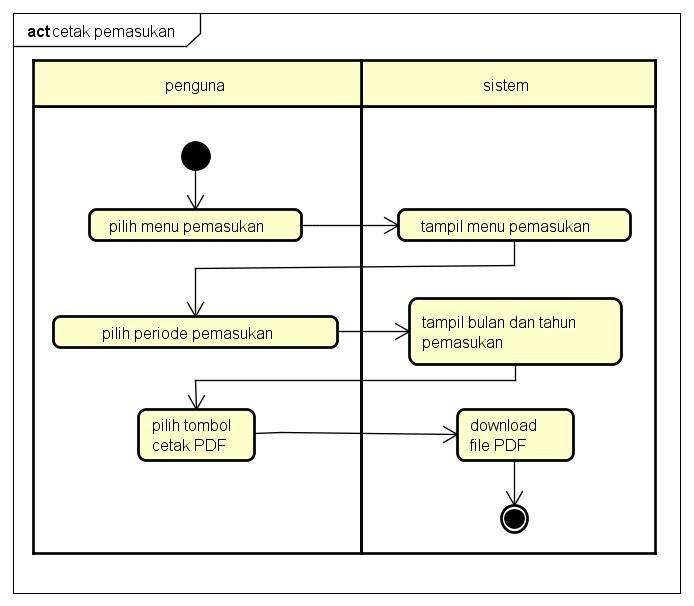
Aktifitas tambah pemasukan menjelaskan alur untuk menambah data pemasukan. Dimulai dari pengguna memilih menu pemasukan, kemudian sistem akan menampilkan menu pemasukan. Pengguna kemudian pilih tombol ‘tambah pemasukan’. Maka sistem akan menampilkan form untuk menambah pemasukan. Terdiri dari tanggal, deskripsi, jumlah pemasukan, dan jenis pemasukan. Kemudian pilih tombol ‘simpan data’. Maka sistem akan menyimpan data ke database dan me-*redirect* ke halaman menu pemasukan.

* 1. *Activity diagram* tampil pemasukan

** Gambar 4.12 *Activity diagram* tampil pemasukan

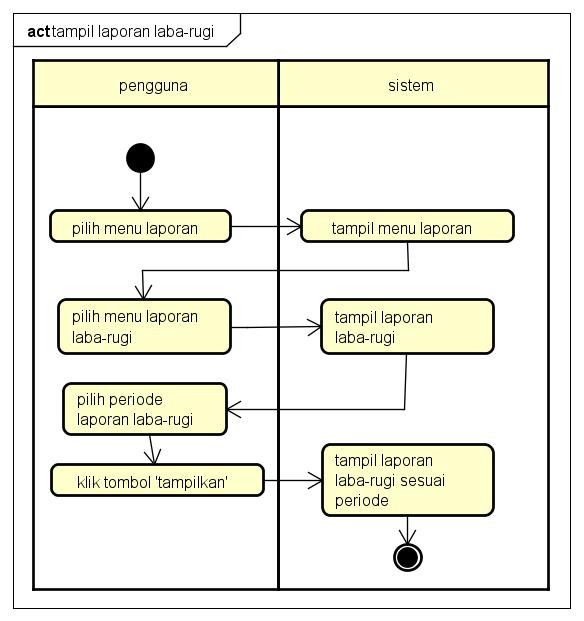
Aktifitas tampil pemasukan menjelaskan alur sistem dalam menampilkan data pemasukan. dimulai dari pengguna memilih menu pemasukan, kemudian sistem akan menampilkan menu pemasukan. Pengguna memilih periode pemasukan terdiri dari tahun dan bulan. Kemudian pilih tombol tampilkan. Maka sistem akan menampilkan data pemasukan sesuai periode yang dipilih.

* 1. *Activity diagram* cetak pemasukan

 Gambar 4.13 *Activity diagram* cetak pemasukan

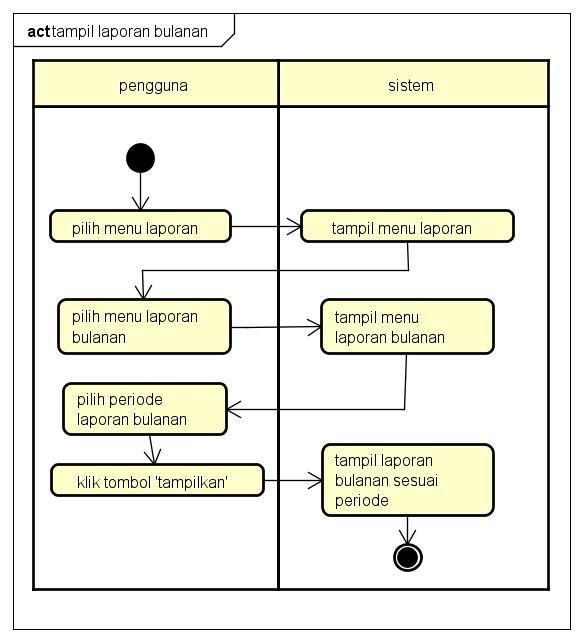
Aktifitas cetak pemasukan menjelaskan alur saat sistem mencetak data pemasukan dalam bentuk PDF. Dimulai dari pengguna memilih menu pemasukan. Kemudian sistem akan menampilkan menu pemasukan. Pengguna memilih periode pemasukan yang akan dicetak terdiri dari tahun dan bulan. Kemudian pilih tombol ‘cetak PDF’. Maka sistem akan men­-*download* data pemasukan

* 1. *Activity diagram* tampil laporan laba-rugi

** Gambar 4.14 *Activity diagram* tampil laporan laba-rugi

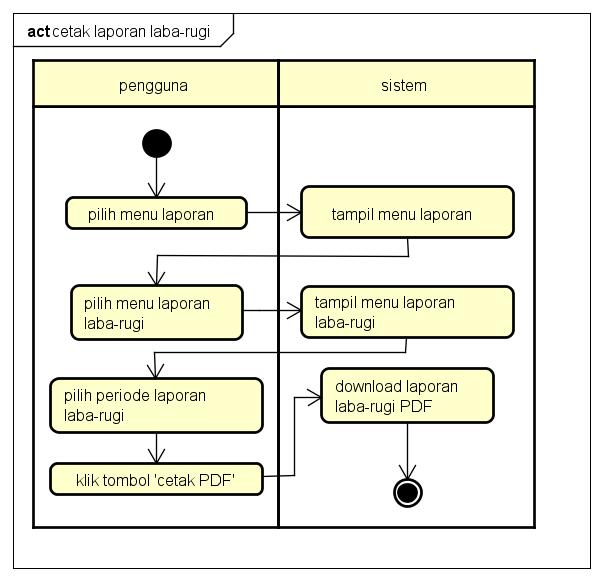
Aktifitas tampil laporan laba rugi menjelaskan alur saat sistem menampilkan laporan laba rugi. Dimulai dari pengguna memilih menu laporan, kemudian memilih laporan laba-rugi. Maka sistem akan menampilkan data laba-rugi. Pengguna memilih periode laporan terdiri dari tahun dan bulan. Kemudian pengguna pilih tombol ‘tampilkan’. Maka sistem akan menampilkan laporan laba-rugi sesuai periode.

* 1. *Activity diagram* tampil laporan keuangan bulanan

 Gambar 4.15 *Activity diagram* tampil laporan keuangan bulanan

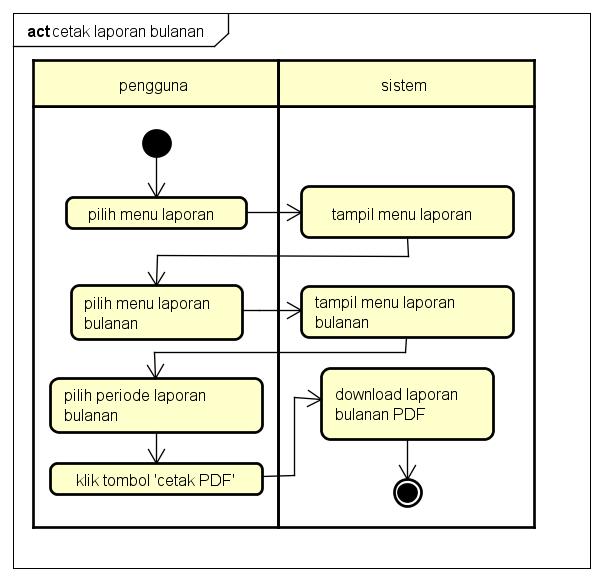
Aktifitas tampil laporan bulanan menjelaskan alur saat sistem menampilkan laporan bulanan. Dimulai dari pengguna memilih menu laporan, kemudian memilih laporan bulanan. Maka sistem akan menampilkan data laporan bulanan. Pengguna memilih periode laporan terdiri dari tahun dan bulan. Kemudian pengguna pilih tombol ‘tampilkan’. Maka sistem akan menampilkan laporan bulanan sesuai periode yang dipilih.

* 1. *Activity diagram* cetaklaporan laba-rugi

 Gambar 4.16 *Activity diagram* cetak laporan laba-rugi

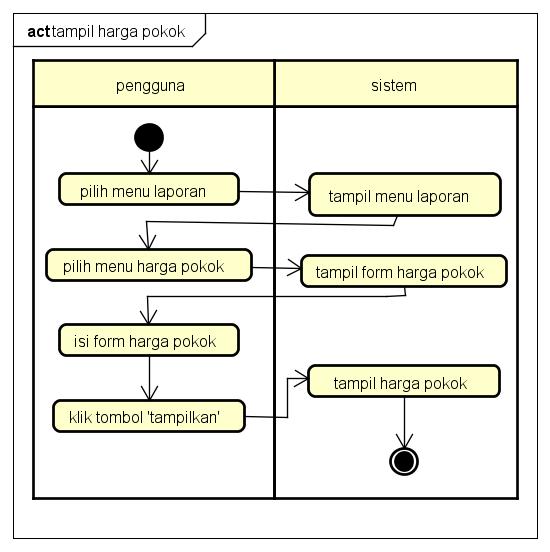
Aktifitas cetak laporan laba-rugi menjelaskan alur saat sistem mencetak data laporan laba-rugi dalam bentuk PDF. Dimulai dari pengguna memilih menu laporan. Kemudian sistem akan menampilkan menu laporan laba-rugi . Pengguna memilih periode laba-rugi yang akan dicetak terdiri dari tahun dan bulan. Kemudian pilih tombol ‘cetak PDF’. Maka sistem akan men­-*download* data laporan laba-rugi dalam bentuk PDF.

* 1. *Activity diagram* cetak laporan keuangan bulanan

** Gambar 4.17 *Activity diagram* cetak laporan keuangan bulanan

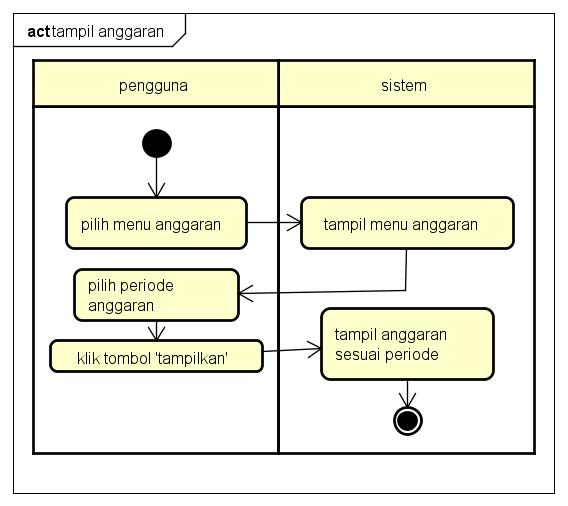
Aktifitas cetak laporan bulanan menjelaskan alur saat sistem mencetak data laporan keuangan bulanan dalam bentuk PDF. Dimulai dari pengguna memilih menu laporan. Kemudian sistem akan menampilkan menu laporan keuangan bulanan. Pengguna memilih periode laporan keuangan bulanan yang akan dicetak terdiri dari tahun dan bulan. Kemudian pilih tombol ‘cetak PDF’. Maka sistem akan men­-*download* data laporan keuangan bulanan dalam bentuk PDF.

* 1. *Activity diagram* tampil harga pokok

** Gambar 4.18 *Activity diagram* tampil harga pokok

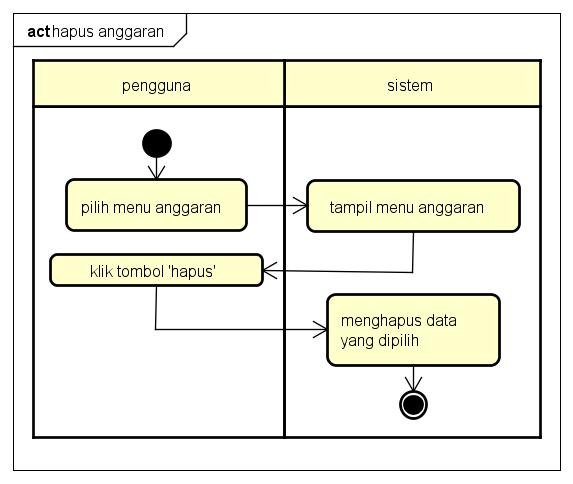
Aktifitas tampil harga pokok menjelaskan alur sistem dalam menampilkan harga pokok penjualan. Dimulai dari pengguna memilih menu laporan, sistem akan menampilkan menu laporan. Kemudian pengguna memilih menu harga pokok. Sistem akan menampilkan form perhitungan harga pokok. Pengguna mengisi form yang terdiri dari jumlah hasil panen dan margin atau keuntungan yang diinginkan. Pengguna lalu memilih tombol ‘tampil harga pokok’. Maka sistem akan menampilkan harga pokok di form harga pokok.

* 1. *Activity diagram* tampil anggaran

 Gambar 4.19 *Activity diagram* tampil anggaran

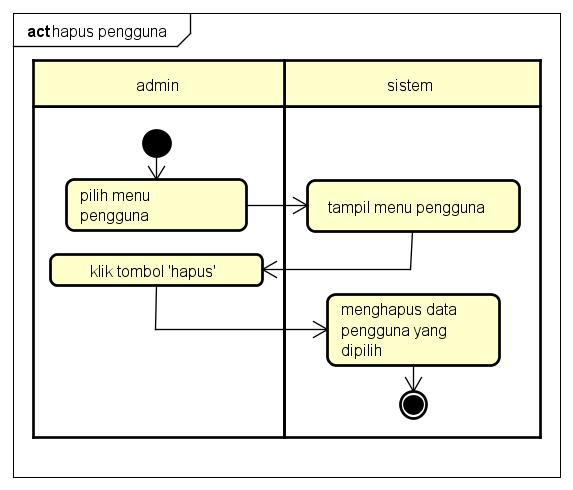
Aktifitas tampil anggaran menjelaskan alur sistem dalam menampilkan data anggaran. dimulai dari pengguna memilih menu anggaran, kemudian sistem akan menampilkan menu anggaran. Pengguna memilih periode anggaran terdiri dari tahun dan bulan. Kemudian pilih tombol tampilkan. Maka sistem akan menampilkan data anggaran sesuai periode yang dipilih.

* 1. *Activity diagram* hapus anggaran

 Gambar 4.20 *Activity diagram* hapus anggaran

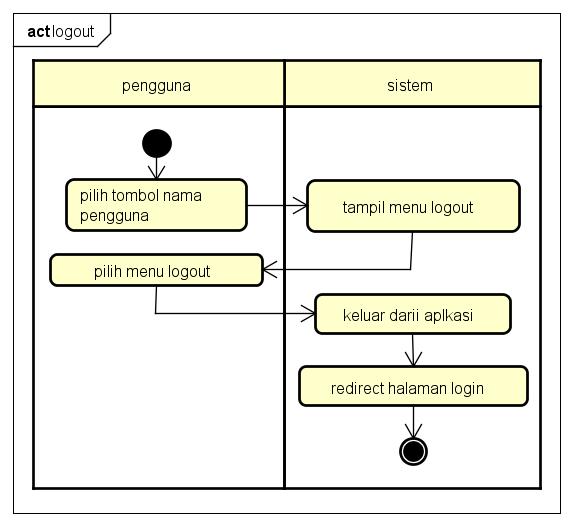
Aktifitas hapus anggaran menjelaskan alur saat pengguna akan menghapus salah satu data anggaran. Dimulai dari pengguna memilih menu anggaran. Kemudian memilih data yang dihapus dan pilih tombol ‘hapus’. Maka sistem akan menghapus data yang dipilih.

* 1. *Activity diagram* hapus pengguna

** Gambar 4.21 *Activity diagram* hapus pengguna

Aktifitas hapus pengguna menjelaskan alur saat admin akan menghapus salah satu pengguna. Dimulai dari admin memilih menu pengguna. Kemudian memilih data pengguna yang dihapus dan pilih tombol ‘hapus’. Maka sistem akan menghapus data pengguna yang dipilih.

* 1. *Activity diagram logout*

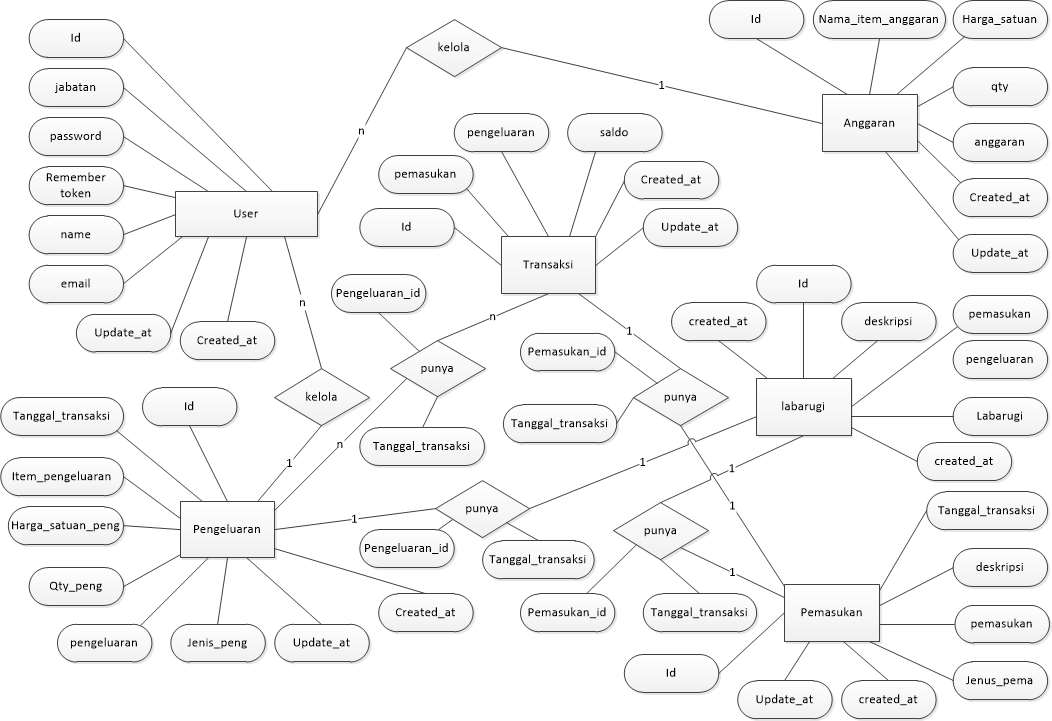
 Gambar 4.22 *Activity diagram logout*

Aktifitas logout menjelaskan alur saat pengguna ingin logout. Kondisi pengguna harus dalan keadaan login. Kemudian memilih menu nama pengguna yang sedang login, maka muncul menu *dropdown* untuk logout. Penguna kemudian pilih tombol logout dan sistem akan keluar dari aplikasi dan me-*redirect* ke halaman login.

1. Desain Data

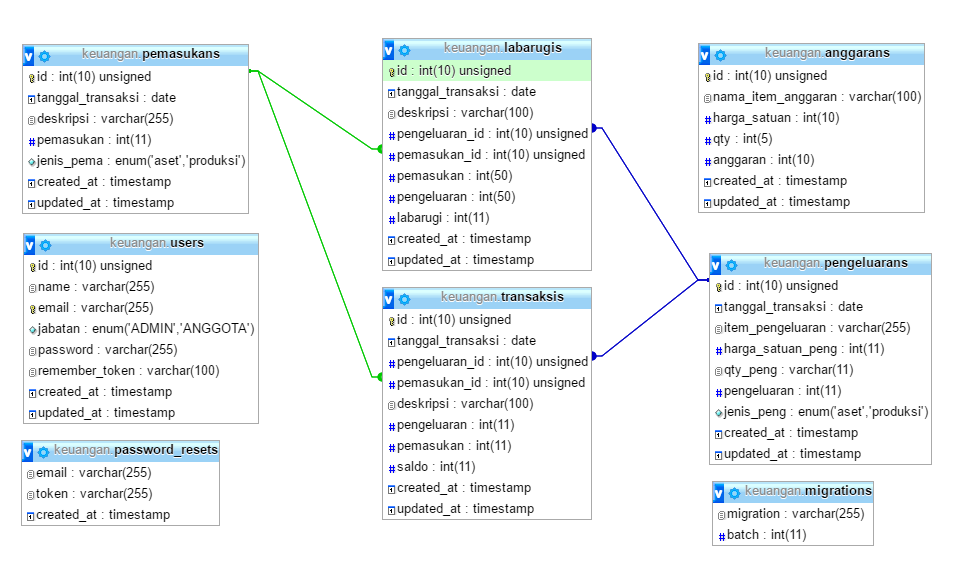
Desain data digunakan untuk manggambarkan pemodelan basis data yang ada pada aplikasi sistem informasi keuangan. *Entity Relationship Diagram (ERD)* digunakan dalam tahap ini untuk memodelkan struktur data dan hubungan antar tabel yang ada dalam sistem informasi keuangan ini.

* 1. ERD (*Entity Relationship Diagram*)

 Gambar 4.23. ERD (*Entity Relationship Diagram*)

* 1. *Mapping Table*

Setelah membuat ERD, tahap selanjutnya adalah memetakan entitas-entitas ataupun relasi antar tabelnya dalam bentuk *mapping table* seperti pada gambar 4.24. Berikut adalah mapping table berdasarkan ERD yang telah dibuat.

 Gambar 4.24. *Mapping Table*

* 1. Kamus Data

Aplikasi sistem informasi keuangan ASRI 12 menggunakan beberapa tabel untuk menyimpan data dari aplikasi. Tabel yang digunakan adalah sebagai berikut :

1. tabel users

tabel 4.23 tabel users

|  |  |  |
| --- | --- | --- |
| kolom | Tipe data | keterangan |
| Id | Int(10) | *Primary key* |
| Name | Varchar(225) |  |
| Email | Varchar(225) |  |
| Jabatan | Enum (“ADMIN”,”ANGGOTA”) |  |
| Password | Varchar(225) |  |
| Created\_at | Date | Timestamp |
| Update\_at | Date | timestamp |

1. tabel anggarans

tabel 4.24 tabel anggarans

|  |  |  |
| --- | --- | --- |
| kolom | Tipe data | keterangan |
| Id | Int(10) | *Primary key* |
| Nama\_item\_anggaran | Varchar(100) |  |
| Harga\_satuan | Int(10) |  |
| qty | Int(5) |  |
| anggaran | Int(5) |  |
| Created\_at | Date | Timestamp |
| Update\_at | Date | timestamp |

1. tabel pemasukans

tabel 4.25 tabel pemasukans

|  |  |  |
| --- | --- | --- |
| kolom | Tipe data | keterangan |
| Id | Int(10) | *Primary key* |
| Tanggal\_transaksi | Date |  |
| deskripsi | Varchar(225) |  |
| pemasukan | Int(10) |  |
| Jenis\_pema | Enum(“ASET”,”PRODUKSI”) |  |
| Created\_at | Date | Timestamp |
| Update\_at | Date | timestamp |

1. Tabel pengeluarans

Tabel 4.26 tabel pengeluarans

|  |  |  |
| --- | --- | --- |
| kolom | Tipe data | keterangan |
| Id | Int(10) | *Primary key* |
| Tanggal\_transaksi | Date |  |
| Item\_pengeluaran | Varchar(225) |  |
| Harga\_satuan\_peng | Int(11) |  |
| Qty\_peng | Int(5) |  |
| pengeluaran | Int(11) |  |
| Jenis\_peng | Enum(“ASET”,”PRODUKSI”) |  |
| Created\_at | Date | Timestamp |
| Update\_at | Date | timestamp |

1. Tabel transaksis

Tabel 4.27 tabel transaksis

|  |  |  |
| --- | --- | --- |
| kolom | Tipe data | keterangan |
| Id | Int(10) | *Primary key* |
| Tanggal\_transaksi | Date |  |
| Pemasukan\_id | Int(10) | *Foreign key* |
| Pengeluaran\_id | Int(10) | *Foreign key* |
| deskripsi | Varchar(100) |  |
| pengeluaran | Int(11) |  |
| pemasukan | Int(11) |  |
| saldo | Int(11) |  |
| Created\_at | Date | Timestamp |
| Update\_at | Date | timestamp |

1. Tabel labarugis

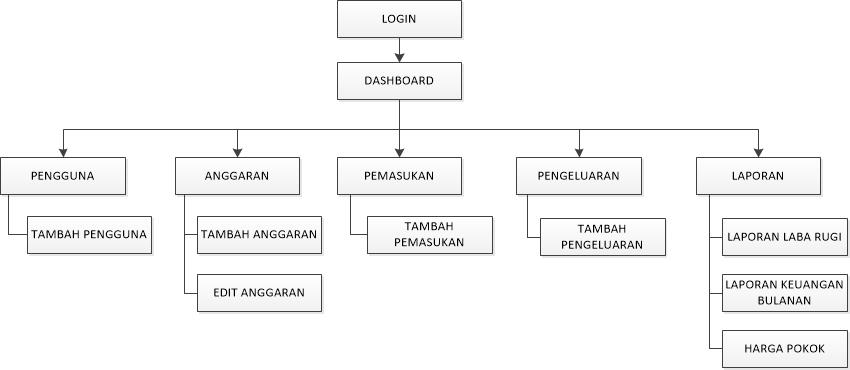
Tabel 4.28 tabel labarugis

|  |  |  |
| --- | --- | --- |
| kolom | Tipe data | keterangan |
| Id | Int(10) | *Primary key* |
| Tanggal\_transaksi | Date |  |
| Pemasukan\_id | Int(10) | *Foreign key* |
| Pengeluaran\_id | Int(10) | *Foreign key* |
| pengeluaran | Int(11) |  |
| pemasukan | Int(11) |  |
| labarugi | Int(11) |  |
| Created\_at | Date | Timestamp |
| Update\_at | Date | timestamp |

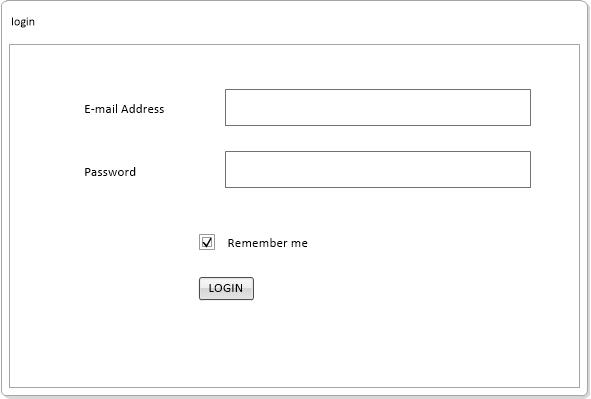
1. Desain Antarmuka

Perancangan antarmuka bertujuan untuk memberikan gambaran aplikasi sistem informasi keuangan kepada pengguna . perancangan antarmuka meliputi struktur menu, input dan output aplikasi, dan fungsi-fungsi yang ada didalam aplikasi. Hasil dari desain antarmuka ini adalah prototype aplikasi sistem informasi keuangan.

1. Struktur menu

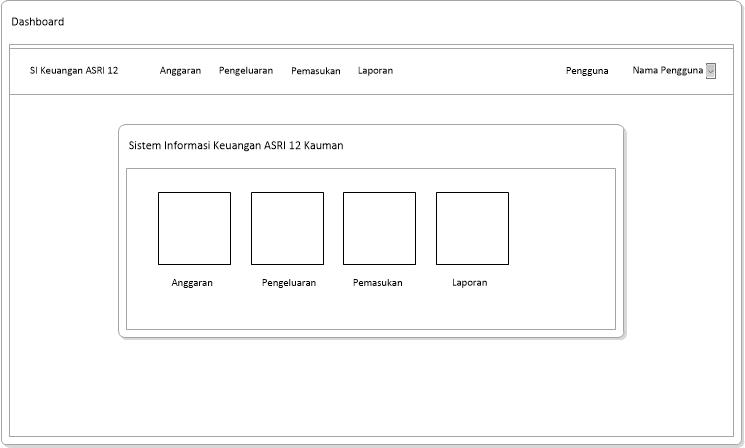
 Gambar 4.25 struktur menu sistem informasi keuangan

1. Login

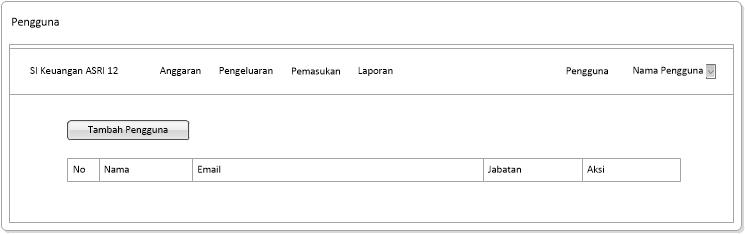


Gambar 4.26 desain halaman login

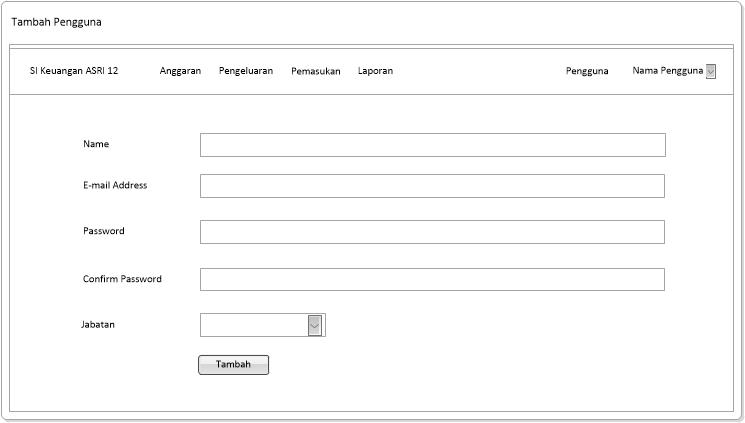
1. Dashboard

 Gambar 4.27 desain halaman dashboard

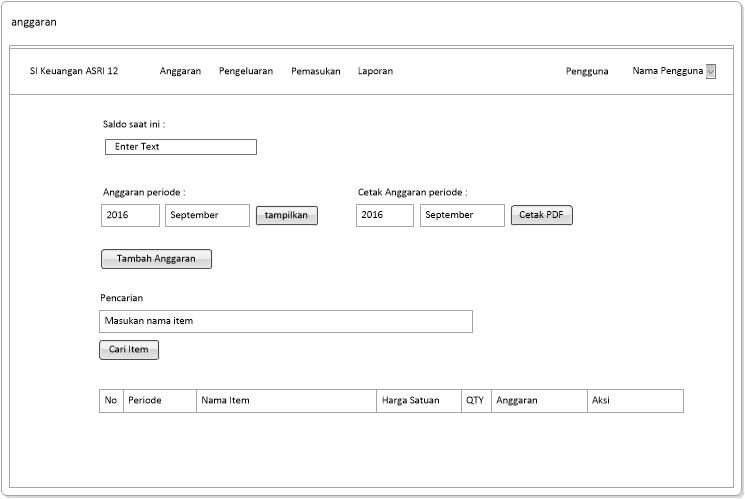
1. Pengguna

 Gambar 4.28 desain halaman pengguna

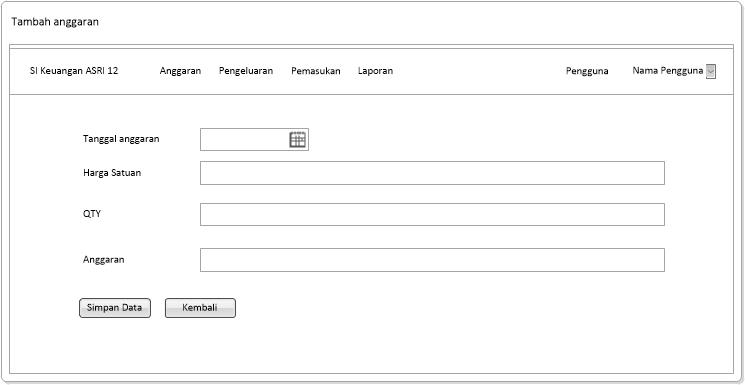
1. Tambah pengguna

 Gambar 4.29 desain halaman tambah pengguna

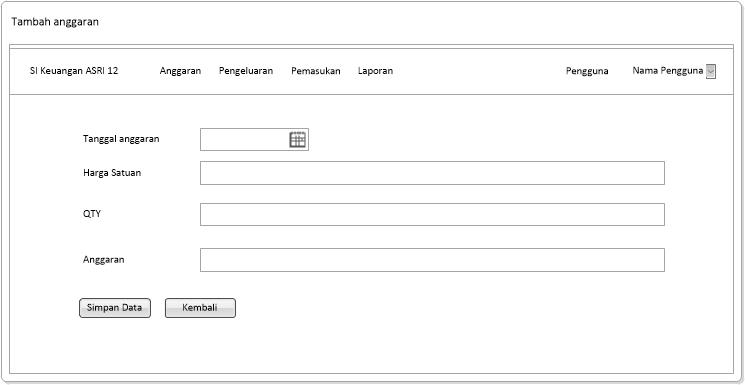
1. Anggaran

 Gambar 4.30 desain halaman anggaran

1. Tambah anggaran

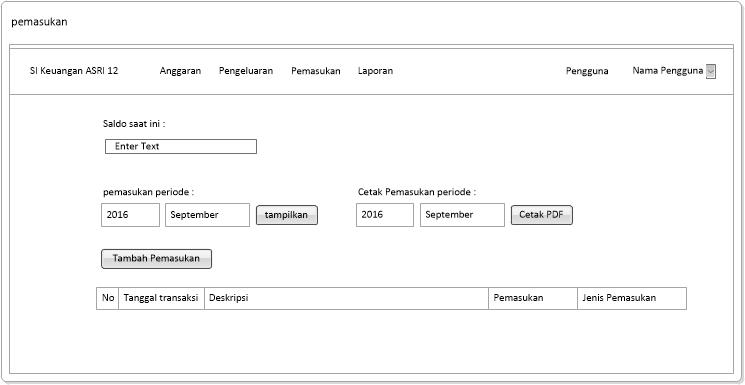
 Gambar 4.31 desain halaman tambah anggaran

1. Edit anggaran



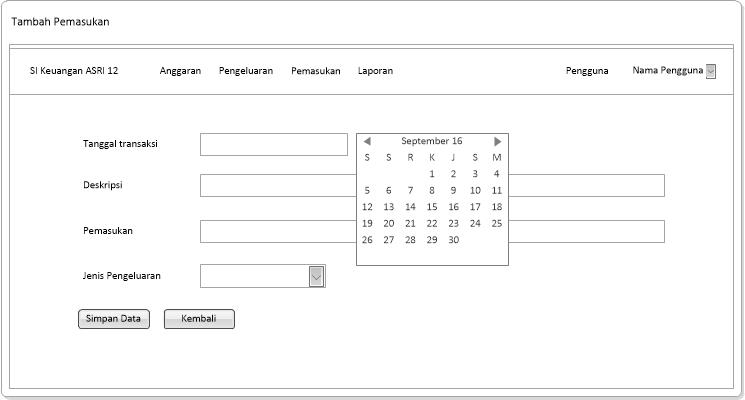
Gambar 4.32 desain halaman edit anggaran

1. Pemasukan



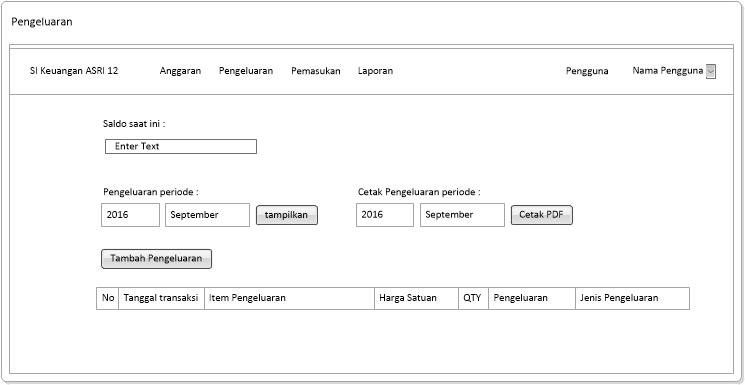
Gambar 4.33 desain halaman pemasukan

1. Tambah pemasukan



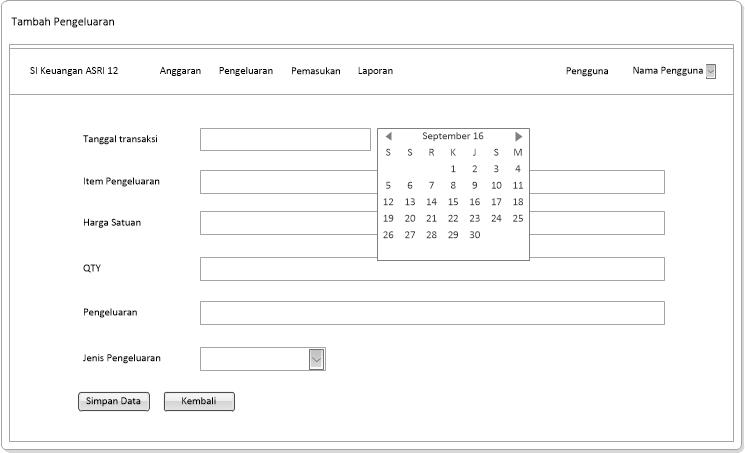
Gambar 4.34 desain halaman tambah pemasukan

1. Pengeluaran



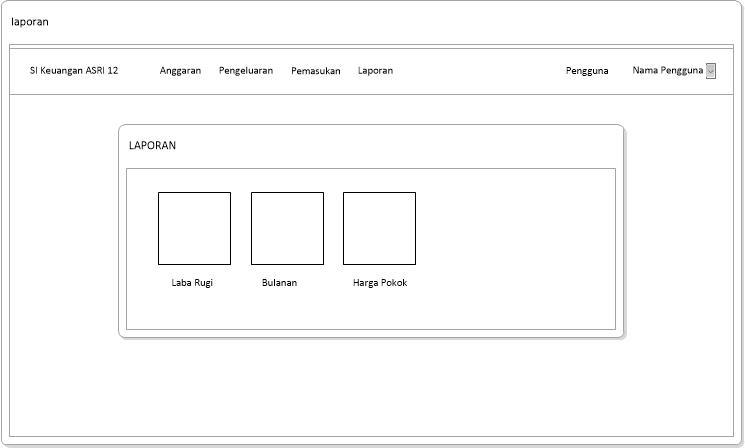
Gambar 4.35 desain halaman pengeluaran

1. Tambah pengeluaran



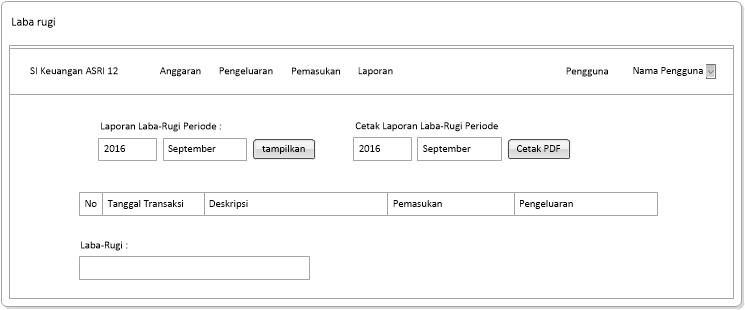
Gambar 4.36 desain halaman tambah pengeluaran

1. Laporan



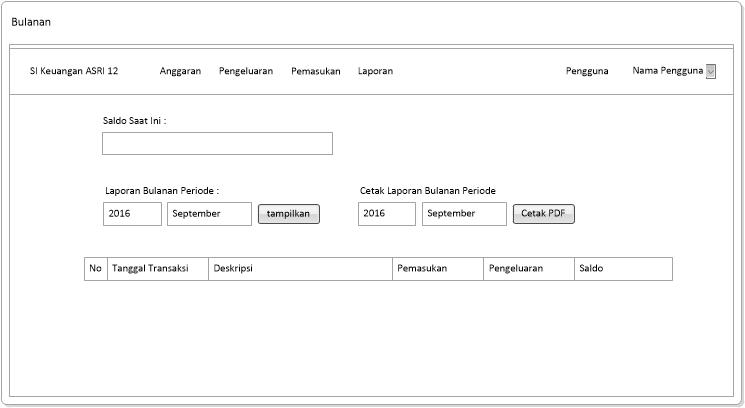
Gambar 4.37 desain halaman laporan

1. Laporan laba-rugi



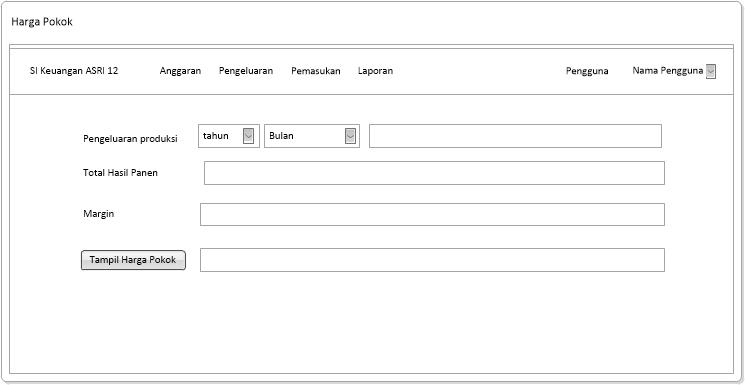
Gambar 4.38 desain halaman laporan laba-rugi

1. Laporan keuangan bulanan



Gambar 4.39 desain halaman laporan keuangan bulanan

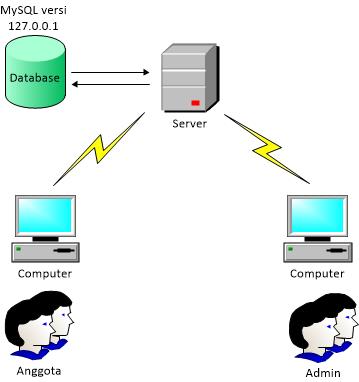
1. Harga pokok



Gambar 4.40 desain halaman harga pokok

1. Arsitektur Sistem

Arsitektur aplikasi sistem informasi keuangan ini memanfaatkan internet untuk menjalankan fungsinya seperti : mengakses data, menginput data serta melihat informasi hasil dari pengolahan sistem. Basis data yang disimpan di server digunakan untuk menyimpan seluruh data dan dapat diakses kapan saja selama terkoneksi internet.



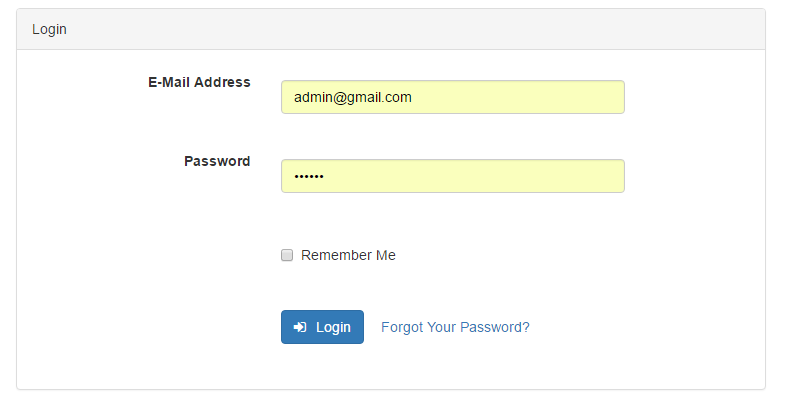
Gambar 4.41 arsitektur sistem aplikasi sistem informasi keuangan

1. Implementasi

Implementasi Aplikasi Sistem Informasi Keuangan ini dibangun menggunakan framework laravel 5.1 dengan bahasa pemograman HTML dan PHP yang berbasis web. Dengan beberapa bahasa pemograman yang menambah daya tarik tampilan web antara lain Bootstrap, Javascript. Data aplikasi ini di*upload* di server penyedia jasa hosting IDHostinger Indonesia. Dengan domain [www.siaksri12kauman.com](http://www.siaksri12kauman.com). Dan database yang dibuat menggunakan MySQL.

Pembangunan web menggunakan konsep MVC (*Model, View,Control*) dan kemudian akan dijelaskan setiap *view.*

1. Login

**

Gambar 4.42 Tampilan halaman login

Autentifikasi login bedasarkan ketersediaan pengguna di tabel *users.* berikut model dari user :

1. <?php
2. namespace App;
3. use Illuminate\Foundation\Auth\User as Authenticatable;
5. class User extends Authenticatable
6. {
8. protected $fillable = [
9. 'name', 'email', 'password','jabatan',
10. ];
11. protected $hidden = [
12. 'password', 'remember\_token',
13. ];
14. }

Script 4.1: Model user.php

1. @extends('layouts.app')
2. @section('content')
3. <div class="container">
4. <div class="row">
5. <div class="col-md-8 col-md-offset-2">
6. <div class="panel panel-default">
7. <div class="panel-heading">Login</div>
8. <div class="panel-body">
9. <form class="form-horizontal" role="form" method="POST" action="{{ url('auth/login') }}">
10. {{ csrf\_field() }}
12. <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
13. <label for="email" class="col-md-4 control-label">E-Mail Address</label>
15. <div class="col-md-6">
16. <input id="email" type="email" class="form-control" name="email" value="{{ old('email') }}">
18. @if ($errors->has('email'))
19. <span class="help-block">
20. <strong>{{ $errors->first('email') }}</strong>
21. </span>
22. @endif
23. </div>
24. </div>
25. <div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">
26. <label for="password" class="col-md-4 control-label">Password</label>
28. <div class="col-md-6">
29. <input id="password" type="password" class="form-control" name="password">
31. @if ($errors->has('password'))
32. <span class="help-block">
33. <strong>{{ $errors->first('password') }}</strong>
34. </span>
35. @endif
36. </div>
37. </div>
38. <div class="form-group">
39. <div class="col-md-6 col-md-offset-4">
40. <div class="checkbox">
41. <label>
42. <input type="checkbox" name="remember"> Remember Me
43. </label>
44. </div>
45. </div>
46. </div>
47. <div class="form-group">
48. <div class="col-md-6 col-md-offset-4">
49. <button type="submit" class="btn btn-primary">
50. <i class="fa fa-btn fa-sign-in"></i> Login
51. </button>
52. <a class="btn btn-link" href="{{ url('/password/reset') }}">Forgot Your Password?</a>
53. </div>
54. </div>
55. </form>
56. </div>
57. </div>
58. </div>
59. </div>
60. </div>
61. @endsection

Script 4.2: View dari login.blade.php

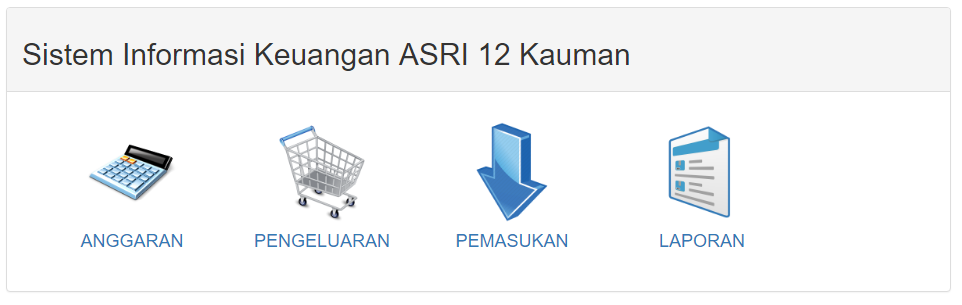
1. <?php
3. namespace App\Http\Controllers\Auth;
5. use Validator;
6. use App\Http\Controllers\Controller;
7. use Illuminate\Foundation\Auth\ThrottlesLogins;
8. use Illuminate\Foundation\Auth\AuthenticatesAndRegistersUsers;
10. class AuthController extends Controller
11. {
13. use AuthenticatesAndRegistersUsers, ThrottlesLogins;
14. protected $redirectPath = '/';
16. public function \_\_construct()
17. {
18. $this->middleware($this->guestMiddleware(), ['except' => 'logout']);
19. }
21. protected function validator(array $data)
22. {
23. return Validator::make($data, [
24. 'name' => 'required|max:255',
25. 'email' => 'required|email|max:255|unique:users',
26. 'password' => 'required|min:6|confirmed',
27. ]);
28. }
30. protected function create(array $data)
31. {
32. return User::create([
33. 'name' => $data['name'],
34. 'email' => $data['email'],
35. 'password' => bcrypt($data['password']),
36. 'jabatan'=>$data['jabatan'],
37. ]);
39. }
41. }

Script 4.3: Controller dari authController.php

1. Route::get('auth/login', 'Auth\AuthController@getLogin');
2. Route::post('auth/login', 'Auth\AuthController@postLogin');

Script 4.4: route untuk halaman login

1. Dashboard atau *home*

**

Gambar 4.43 tampilan halaman *home.blade.php*

1. <?php
3. namespace App\Http\Controllers;
4. use App\Http\Requests;
5. use Illuminate\Http\Request;
7. class HomeController extends Controller
8. {
9. public function \_\_construct()
10. {
11. $this->middleware('auth');
12. }
13. public function index()
14. {
15. return view('home');
16. }
17. }

Script 4.5: model untuk *home*  atau *dashboard*

1. @extends('layouts.app')
2. @section('content')
3. <div class="container">
4. <div class="row">
5. <div class="col-md-10 col-md-offset-1">
6. <div class="panel panel-default">
7. <div class="panel-heading"><h2>Sistem Informasi Keuangan ASRI 12 Kauman</h2></div>
9. <div class="panel-body">
10. <div class='container'>
11. <div class="col-sm-2 " align="center">
12. <a href="{{ url('anggaran')}}"><img src="img/anggaran.png"  width="100" height="100" class="img-responsive" alt="Generic placeholder thumbnail">
13. <h4 >ANGGARAN</h4>
14. </div>
15. <div class="col-sm-2 " align="center">
16. <a href="{{ url('pengeluaran')}}"><img src="img/realisasi.png" width="100" height="100" class="img-responsive" alt="Generic placeholder thumbnail">
17. <h4>PENGELUARAN</h4>
18. </div>
19. <div class="col-sm-2 placeholder" align="center">
20. <a href="{{ url('pemasukan')}}"><img src="img/pemasukan.png" width="100" height="100" class="img-responsive" alt="Generic placeholder thumbnail">
21. <h4>PEMASUKAN</h4>
22. </div>
23. <div class="col-sm-2 placeholder" align="center">
24. <a href="{{ url('laporan')}}"><img src="img/laporan.png" width="100" height="50" class="img-responsive" alt="Generic placeholder thumbnail">
25. <h4>LAPORAN</h4>
26. </div>
27. </div>
28. </div>
29. </div>
30. </div>
31. </div>
32. </div>
33. @endsection

Script 4.6 *View* dari tampilan *home.blade.php*

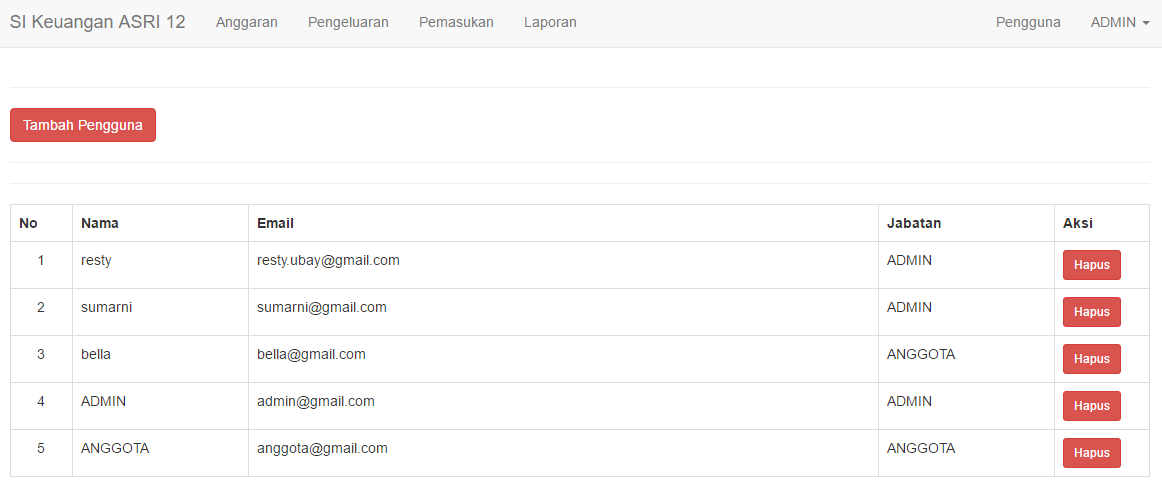
1. <?php
3. namespace App\Http\Controllers;
4. use App\Http\Requests;
5. use Illuminate\Http\Request;
7. class HomeController extends Controller
8. {
9. public function \_\_construct()
10. {
11. $this->middleware('auth');
12. }
14. public function index()
15. {
16. return view('home');
17. }
18. }

Script 4.7: *Controller* dari *homeController*

1. Route::auth();
2. Route::get('/', 'HomeController@index');

Script 4.8: *Routes* dari halaman *dashboard*

1. Pengguna



Gambar 4.44 tampilan halaman pengguna

Model users.php menggunakan model yang sama dengan model login

1. @extends('layouts.app')
2. @section('content')
3. <div class="container">
4. <div>
5. <hr>
6. {!! link\_to('pengguna/create','Tambah Pengguna',['class'=>'btn btn-danger btn-md']) !!}
7. <hr>
8. </div>
9. <hr>
10. <table class="table table-bordered">
11. <tr><th>No</th><th>Nama</th><th>Email</th><th>Jabatan</th><th colspan="2">Aksi</th></tr>
12. <?php $no = 1; ?>
14. @foreach ( $pengguna as $n)
15. <tr>
16. <td width="50px" align="center">{{ $no++ }}</td>
17. <td width="140px">{{ $n->name }}</td>
18. <td width="500px">{{ $n->email }}</td>
19. <td width="140px">{{ $n->jabatan }}</td>
20. <td width="50px">
22. {!! Form::open(array('method'=>'delete','url'=>'pengguna/'.$n->id))!!}
23. {!! Form::hidden('\_delete','DELETE')!!}
24. {!! Form::submit('Hapus',['class'=>'btn btn-danger btn-sm'])!!}
25. {!! Form::close()!!}
27. </td> </tr>
29. @endforeach
31. </table>
32. <!--  -->
33. </div>

36. @stop

Script 4.9: *View* dari *pengguna*.*index.blade.php*

1. public function index()
2. {
3. $pengguna = User::all();
4. $data['pengguna']   = $pengguna;
5. return view('pengguna.index',$data);
6. }

Script 4.10: *controller* untuk menampilkan data pengguna

1. Route::group(['middleware' => ['auth', 'periksaAdmin']], function ()
2. {
3. Route::get('pengguna',function()
4. {
5. return view('pengguna.index');
6. });
7. Route::get('pengguna/create',function()
8. {
9. return view('pengguna.create');
10. });
11. Route::resource('pengguna', 'PenggunaController');
12. });

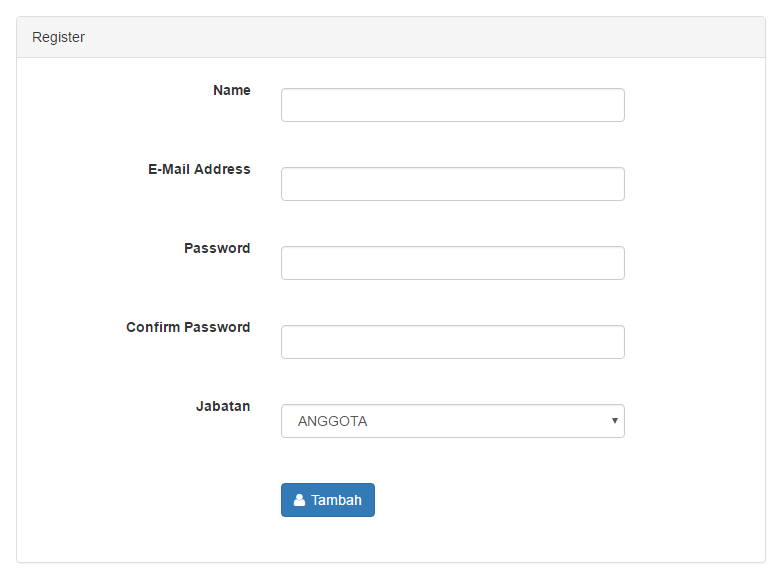
Script 4.11 *routes* untuk halaman pengguna

Fitur pengguna ini hanya bisa diakses oleh admin. Dalam laravel hak akses dikenal dengan *middleware*. Script diatas adalah routes *middleware* untuk admin. Untuk mengakses fitur ini diatur dalam script cekAdmin.php sebagai berikut:

1. <?php
3. class cekAdmin
4. {
5. public function handle($request, Closure $next)
6. {
7. if($request->user()->jabatan !='ADMIN')
8. {
9. return redirect('/');
10. }
11. return $next($request);
12. }
13. }

Script 4.12 *middleware* cekAdmin.php

1. Tambah pengguna

**

Gambar 4.45 tampilan halaman tambah pengguna

Model yang digunakan pada halaman tambah pengguna menggunakan model yang sama dengan model login.

1. @extends('layouts.app')
3. @section('content')
4. <div class="container">
5. <div class="row">
6. <div class="col-md-8 col-md-offset-2">
7. <div class="panel panel-default">
8. <div class="panel-heading">Register</div>
9. <div class="panel-body">
10. <form class="form-horizontal" role="form" method="POST" action="{{ url('/pengguna') }}">
11. {{ csrf\_field() }}
12. <div class="form-group{{ $errors->has('name') ? ' has-error' : '' }}">
13. <label for="name" class="col-md-4 control-label">Name</label>
14. <div class="col-md-6">
15. <input id="name" type="text" class="form-control" name="name" value="{{ old('name') }}">
16. @if ($errors->has('name'))
17. <span class="help-block">
18. <strong>{{ $errors->first('name') }}</strong>
19. </span>
20. @endif
21. </div>
22. </div>
23. <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
24. <label for="email" class="col-md-4 control-label">E-Mail Address</label>
25. <div class="col-md-6">
26. <input id="email" type="email" class="form-control" name="email" value="{{ old('email') }}">
27. @if ($errors->has('email'))
28. <span class="help-block">
29. <strong>{{ $errors->first('email') }}</strong>
30. </span>
31. @endif
32. </div>
33. </div>
34. <div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">
35. <label for="password" class="col-md-4 control-label">Password</label>
36. <div class="col-md-6">
37. <input id="password" type="password" class="form-control" name="password">
38. @if ($errors->has('password'))
39. <span class="help-block">
40. <strong>{{ $errors->first('password') }}</strong>
41. </span>
42. @endif
43. </div>
44. </div>
45. <div class="form-group{{ $errors->has('password\_confirmation') ? ' has-error' : '' }}">
46. <label for="password-confirm" class="col-md-4 control-label">Confirm Password</label>
47. <div class="col-md-6">
48. <input id="password-confirm" type="password" class="form-control" name="password\_confirmation">
50. @if ($errors->has('password\_confirmation'))
51. <span class="help-block">
52. <strong>{{ $errors->first('password\_confirmation') }}</strong>
53. </span>
54. @endif
55. </div>
56. </div>
57. <div class="form-group">
58. <label  class="col-md-4 control-label">Jabatan</label>
59. <div class="col-md-6">
60. <select class="form-control" name='jabatan'>
61. <option value="ADMIN">ADMIN</option>
62. <option value="ANGGOTA" selected>ANGGOTA</option>
63. </select>
64. </div>
65. </div>
66. <div class="form-group">
67. <div class="col-md-6 col-md-offset-4">
68. <button type="submit" class="btn btn-primary">
69. <i class="fa fa-btn fa-user"></i>Tambah
70. </button>
71. </div>
72. </div>
73. </form>
74. </div>
75. </div>
76. </div>
77. </div>
78. </div>
79. @endsection

Script 4.13 *view* untuk tampilan tambah pengguna

1. public function create()
2. {
3. return view('pengguna.create');
4. }
5. public function store(Request $request)
6. {
7. $name = $request['name'];
8. $email = $request['email'];
9. $password = bcrypt($request['password']);
10. $jabatan  = $request['jabatan'];
11. User::create(array('name'=> $name,'email'=>$email,'password'=>$password,'jabatan'=>$jabatan));
12. return redirect('pengguna');   }

Script 4.14 *Controller* untuk menampilkan form dan menyimpan data pengguna ke database

1. Anggaran
2. Tampil anggaran
3. Edit anggaran
4. Cetak anggaran
5. Cari anggaran
6. Pemasukan
7. Tampil pemasukan
8. Tambah pemasukan
9. Cetak pemasukan
10. Pengeluaran
11. Tampil pengeluaran
12. Tambah pengeluaran
13. Cetak pengeluaran
14. Laporan
15. Laporan laba rugi
16. Laporan keuangan bulanan
17. Harga pokok

DAFTAR PUSTAKA

Andreas Handojo, Go Ornella Aquaria, & Sri Maharsi. (2004). Pembuatan Sistem Informasi Akuntansi Terkomputerisasi Atas Siklus Pembelian Dan Penjualan Pada Cv. X. *Jurnal Informatika*, *5*(2), 86–94. Retrieved from http://puslit2.petra.ac.id/ejournal/index.php/inf/article/view/15843

Anggarani, A., Prasetyoning, W., & Safitri, V. I. (2015). Penguatan Sektor Umkm Sebagai Strategi Menghadapi Mea 2015, *5*.

Ariefin, M. R. S., & Ilhamsyah. (2015). MEMBANGUN WEBSITE SISTEM INFORMASI SEKOLAH DENGAN MENGGUNAKAN FRAMEWORK PHP LARAVEL PADA SMA NEGERI 4 PONTIANAK, 1–7.

Dangar, H. (2013). *Learning Laravel 4 Application Development*. BIRMINGHAM - MUMBAI: Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK.

Iswanto. (2006). *Membangun Aplikasi Berbasis PHP 5 dan Firebird 1.5*. Yogyakarta.

Jumingan. (2005). *Analisis Laporan Keuangan* (IV). Surakarta: Remaja Rosdakarya.

Kabuhung, M. (2013). Sistem Informasi Akuntansi Penerimaan Dan Pengeluaran Kas Untuk Perencanaan Dan Pengendalian Keuangan Pada Organisasi Nirlaba Keagamaan. *EMBA*, *1*(1988), 1–5. http://doi.org/10.1017/CBO9781107415324.004

Lasena, S. R. (2013). ANALISIS PENENTUAN HARGA POKOK PRODUKSI PADA PT. DIMEMBE NYIUR AGRIPRO. *Jurnal EMBA*, *1*(3), 585–592.

Nugroho, I. (2009). Peranan Teknologi Informasi Dalam Audit Sistem Informasi. *DINAMIKA INFORMATIKA*, *I*(2), 122–130.

Ralph M. Stair, G. W. R. (2010). *Principles of Information Systems A Managerial Approach* (9th ed.). florida.

Sarfi’ah, S. nur. (2013). Menghitung harga pokok penjualan, *39*(2), 80–90.

Satya, E. (2015). Pembangunan Sistem Informasi Keuangan Studi Kasus SMK Marsudi Luhur I Yogyakarta. *Universitas Atma Jaya Yogyakarta*, *Vol 6, No*.

Solichin, A. (2010). *MySQL Dari Pemula Hingga Mahir*. *Universitas Budi Luhur, Jakarta*.

Sugiantina. (2011). PENERAPAN SISTEM INFORMASI DALAM PRAKTEK LAPORAN KEUANGAN SEDERHANA. *Teknologi*, *Vol 1, No* .

Suwardjono. (2005). *Teori Akuntansi Perekayasaan Pelaporan Keuangan* (3rd ed.). Yogyakarta: BPFE-Yogyakarta.

Tata Sutabsi. (2003). *Sistem Informasi Akuntansi*. Jakarta: Penerbit ANDI.