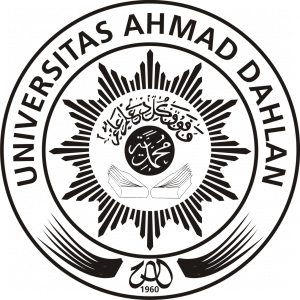
APLIKASI SISTEM INFORMASI KEUANGAN BERBASIS WEB MENGGUNAKAN KONSEP MVC (*MODEL, VIEW, CONTROL*) DENGAN FRAMEWORK LARAVEL UNTUK MENGELOLA KEUANGAN DI KELOMPOK PETANI SAYUR HIDROPONIK ASRI 12 KAUMAN YOGYAKARTA.



Disusun Oleh :

SAFRI ADAM (12018175)

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS AHMAD DAHLAN

YOGYAKARTA

2016

LEMBAR PERSETUJUAN

SKRIPSI

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.

Dipersiapkan dan disusun oleh :

SAFRI ADAM

12018175

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS AHMAD DAHLAN

YOGYAKARTA

Telah disetujui oleh:

Pembimbing

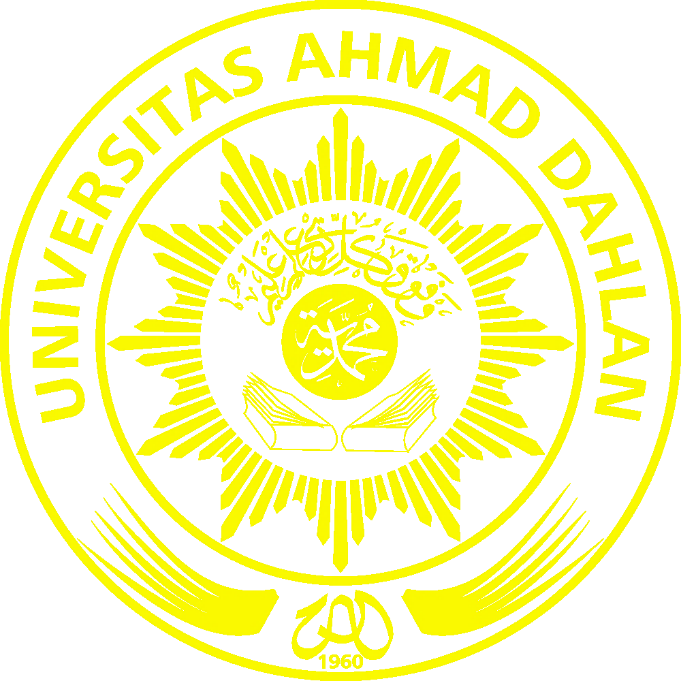
Arfiani Nur Khusna, S.T, M.Kom.

HALAMAN PENGESAHAN

SKRIPSI

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.

Yang dipersiapkan dan disusun oleh

Safri Adam

12018175

Telah dipertahankan di depan Dewan Penguji

Pada tanggal 29 Oktober 2016

Dan dinyatakan telah memenuhi syarat

Susunan Dewan Penguji :

Ketua : Arfiani Nur Khusna, S.T., M.Kom. ...................

Penguji I : Sri Handayaningsih, S.T., M.T. ...................

Penguji II : Drs.Wahyu Pujiono, M.Kom. ...................

Yogyakarta, 29 Oktober 2016

Dekan

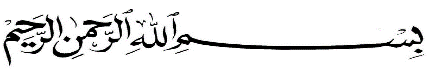
Fakultas Teknologi Industri

Universitas Ahmad Dahlan

Kartika Firdausy, S.T., M.T.

HALAMAN PERSEMBAHAN

KATA PENGANTAR



*Assalamu’alaikum Wr. Wb.*

Dengan mengucapkan *Alhamdulillahirobbil’alamin*, penulis memanjatkan puji dan syukur kehadirat Allah SWT atas rahmat, nikmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “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.” dengan baik. Skripsi ini disusun untuk memenuhi sebagian persyaratan mencapai derajat Sarjana Teknik di Teknik Informatika Fakultas Teknologi Industri di Universitas Ahmad Dahlan Yogyakarta.

Selama penulisan skripsi ini telah banyak pihak yang terlibat memberikan bantuannya, sehingga pada kesempatan kali ini penulis mengucapkan terima kasih yang sebesar besarnya kepada :

**DAFTAR ISI**

HALAMAN JUDULi

HALAMAN PERSETUJUANii

HALAMAN PENGESAHANiii

HALAMAN PERSEMBAHANiv

KATA PENGANTARv

DAFTAR ISIvi

DAFTAR GAMBAR DAN RUMUSviii

DAFTAR TABEL xi

DAFTAR LISTING xiii

HALAMAN PERNYATAAN xv

ABSTRAKxvi

BAB I PENDAHULUAN

1. Latar Belakang Masalah1
2. Identifikasi Masalah4
3. Batasan Masalah 5
4. Rumusan Masalah5
5. Tujuan Penelitian6
6. Manfaat Penelitian6

BAB II KAJIAN PUSTAKA

1. Kajian Penelitian Terdahulu7
2. Landasan Teori 14
3. Sistem Informasi14
4. Laporan Laba Rugi16
5. Harga Pokok17
6. Biaya Aset dan Biaya Produksi 18
7. Sistem Informasi Keuangan 19
8. PHP21
9. MySQL24
10. Framework Laravel25

BAB III METODOLOGI PENELITIAN

1. Subyek Penelitian27
2. Metode Pengumpulan Data 27
3. Metode Wawancara27
4. Survey Lokasi28
5. Alat dan Bahan28
6. Analisis Kebutuhan Sistem29
7. Desain Sistem (Pemodelan)29
8. Implementasi31
9. Model 31
10. View 31
11. Controller 32
12. Pengujian Sistem32

BAB IV HASIL DAN PEMBAHASAN

1. Analisis Kebutuhan Sistem34
2. Desain Sistem (Pemodelan)46
3. Implementasi80
4. Pengujian Sistem112

BAB V KESIMPULAN DAN SARAN

1. Kesimpulan117
2. Saran118

DAFTAR PUSTAKA119

LAMPIRAN120

DAFTAR GAMBAR DAN RUMUS

Gambar 2.0 gambaran umum sistem informasi14

Rumus (1) pembuatan laporan laba rugi16

Rumus (2) rumus laba 14

Rumus (3) rumus rugi 16

Rumus (4) harga pokok 17

Gambar 4.1 *use case diagram*35

Gambar 4.2 activity diagram login46

Gambar 4.3 *activity diagram* tambah pengguna47

Gambar 4.4 *Activity Diagram* tambah belanja modal 48

Gambar 4.5 *Activity Diagram*  edit belanja modal49

Gambar 4.6 *Activity Diagram* cetak anggaran50

Gambar 4.7 *Activity Diagram* tambah pengeluaran 51

Gambar 4.8 *Activity Diagram* tampil pengeluaran 52

Gambar 4.9 *Activity Diagram* cetak pengeluaran 53

Gambar 4.10 *Activity Diagram* tambah pemasukan 54

Gambar 4.11 *Activity Diagram* tampil pemasukan 55

Gambar 4.12 *Activity Diagram* cetak pemasukan 56

Gambar 4.13 *Activity Diagram* tampil laporan laba-rugi 57

Gambar 4.14 *Activity Diagram* tampil laporan keuangan bulanan 58

Gambar 4.15 *Activity Diagram* cetak laporan laba-rugi 59

Gambar 4.16 *Activity Diagram* cetak laporan keuangan bulanan 60

Gambar 4.17 *Activity Diagram* tampil harga pokok61

Gambar 4.18 *Activity Diagram* tampil anggaran 62

Gambar 4.19 *Activity Diagram* hapus anggaran 63

Gambar 4.20 *Activity Diagram* hapus pengguna 64

Gambar 4.21 *Activity Diagram* logout 65

Gambar 4.22 ERD (*Entity Relationship Diagram*)66

Gambar 4.23 *Mapping Table* 67

Gambar 4.24 struktur menu sistem informasi keuangan70

Gambar 4.25 desain halaman login71

Gambar 4.26 desain halaman *dashboard* 71

Gambar 4.27 desain halaman pengguna72

Gambar 4.28 desain halaman tambah pengguna 72

Gambar 4.29 desain halaman belanja modal 73

Gambar 4.30 desain halaman tambah belanja modal 74

Gambar 4.31 desain halaman edit belanja modal74

Gambar 4.32 desain halaman pemasukan 75

Gambar 4.33 desain halaman tambah pemasukan75

Gambar 4.34 desain halaman pengeluaran 76

Gambar 4.35 desain halaman tambah pengeluaran 77

Gambar 4.36 desain halaman laporan77

Gambar 4.37 desain halaman laporan laba-rugi 78

Gambar 4.38 desain halaman laporan keuangan bulanan78

Gambar 4.39 desain halaman harga pokok 79

Gambar 4.40 asritektur sistem aplikasi sistem informasi keuangan80

Gambar 4.41 tampilan halaman login81

Gambar 4.42 tampilan halaman *dashboard* 84

Gambar 4.43 tampilan halaman pengguna86

Gambar 4.44 tampilan halaman tambah pengguna 89

Gambar 4.45 tampilan halaman utama belanja modal 91

Gambar 4.46 tampilan halaman menu pemasukan96

Gambar 4.47 tampilan halaman tambah pemasukan99

Gambar 4.48 tampilan halaman pengeluaran 101

Gambar 4.49 tampilan halaman form tambah pengeluaran103

Gambar 4.50 tampilan halaman utama laporan 105

Gambar 4.51 tampilan halaman laporan laba-rugi 106

Gambar 4.52 tampilah halaman laporan keuangan bulanan108

Gambar 4.53 tampilan halaman harga pokok 110

DAFTAR TABEL

Tabel 2.1. tabel perbandingan kajian penelitian terdahulu11

Tabel 4.1. tabel SRS Fungsional34

Tabel 4.2 tabel *use case scenario* login 36

Tabel 4.3 tabel *use case scenario* input pengguna36

Tabel 4.4 tabel *use case scenario* input belanja modal37

Tabel 4.5 tabel *use case scenario* edit belanja modal 37

Tabel 4.6 tabel *use case scenario* cetak belanja modal 38

Tabel 4.7 tabel *use case scenario* tambah pengeluaran38

Tabel 4.8 tabel *use case scenario* tampil pengeluaran39

Tabel 4.9 tabel *use case scenario* cetak pengeluaran39

Tabel 4.10 tabel *use case scenario* tambah pemasukan 40

Tabel 4.11 tabel *use case scenario* tampil pemasukan40

Tabel 4.12 tabel *use case scenario* cetak pemasukan41

Tabel 4.13 tabel *use case scenario* tampil laporan laba-rugi41

Tabel 4.14 tabel *use case scenario* tampil laporan keuangan bulanan42

Tabel 4.15 tabel *use case scenario* cetak laporan laba-rugi42

Tabel 4.16 tabel *use case scenario* cetak laporan keuangan bulanan43

Tabel 4.17 tabel *use case scenario* tampil harga pokok43

Tabel 4.18 tabel *use case scenario* tampil belanja modal 44

Tabel 419 tabel *use case scenario* hapus belanja modal44

Tabel 4.20 tabel *use case scenario* hapus pengguna 45

Tabel 4.21 tabel *use case scenario* logout 45

Tabel 4.22 tabel users68

Tabel 4.23 tabel anggarans68

Tabel 4.24 tabel pemasukans68

Tabel 4.25 tabel pengeluarans69

Tabel 4.26 tabel transakasis69

Tabel 4.27 tabel labarugis69

Tabel 4.28 tabel hasil *black box test*113

Tabel 4.29 tabel daftar responden *alpha test* 114

Tabel 4.30 tabel hasil pengujian *alpha test*115

DAFTAR LISTING

Script 4.1 Model user.php81

Script 4.2 *view* login.blade.php82

Script 4.3 *contoller* dari *authController*.php83

Script 4.4 *route* halaman login83

Script 4.5 *model* home atau dashboard84

Script 4.6 *view* tampilan *home*.*blade*.*php85*

Script 4.7 *contoller* dari *homeController*86

Script 4.8 routes halaman dashboard86

Script 4.9 *view pengguna.blade.php*87

Script 4.10 *controller* menampilkan data pengguna87

Script 4.11 *routes* untuk halaman pengguna88

Script 4.12 *middleware* cekAdmin.php88

Script 4.13 view untuk tampilan tambah pengguna 89

Script 4.14 Controller user method create() dan store()91

Script 4.15 model belanja modal 92

Script 4.16 view belanja modal 92

Script 4.17 controller index() untuk belanja modal 94

Script 4.18 *routes* belanja modal94

Script 4.19 *view* dari crate.blade.php94

Script 4.20 *controller* tambah belanja modal95

Script 4.21 method update dan store96

Script 4.22 model pemasukan97

Script 4.23 view pemasukan98

Script 4.24 *Controller* pemasukan99

Script 4.25 *view* tambah pemasukan100

Script 4.26 *Controller create* dan *store*100

Script 4.27 model pengeluaran102

Script 4.28 *view* pengeluaran103

Script 4.29 *Controller* pengeluaran103

Script 4.30 *Controller* tambah pengeluaran104

Script 4.31 *View*  laporan105

Script 4.32 model laporan laba-rugi106

Script 4.33 *View* laporan laba-rugi107

Script 4.34 *Controller* laba-rugi108

Script 4.35 *view* laporan keuangan bulanan110

Script 4.36 *Controller* laporan keuangan bulanan110

Script 4.37 *View* laporan harga pokok111

SURAT PERNYATAAN

Yang bertanda tangan dibawah ini :

Nama : SAFRI ADAM

NIM : 12018175

Prodi : Teknik Informatika

Judul TA/ Skripsi : 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.

Dengan ini saya menyatakan bahwa Laporan Tugas Akhir in tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan disuatu Perguruan Tinggi. Dan sepanjang sepengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 20 Oktober 2016

Mengetahui, Yang menyatakan,

Dosen pembimbing Mahasiswa

(Arfiani Nur Khusna, S.T, M.Kom) (Safri Adam)

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 terlihat ketika melihat hasil penjualan yan selalu merugi. Hal ini diakibatkan tidak adanya sistem yang menangani masalah keuangan di kelompok petani hidroponik ini.

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 belanja modal bulanan, mendokumentasikan pengeluaran,dan pemasukan, serta dapat mencetak laporan keuangan bulanan dan laba-rugi setiap periode yang telah ditentukan.

Dari penelitian yang dilakukan menghasilkan sebuah perangkat aplikasi sistem informasi keuangan berbasis web menggunakan framework laravel menggunakan metode MVC (*Model, View, Control*). Hasil pengujian sistem menunjukan bahwa aplikasi layak digunakan sesuai kebutuhan pengguna

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 A. Nugroho, Beeh, dan Astuningdyas (2008) tentang Perancangan Aplikasi Rencana Anggaran Biaya (RAB). Mengatakan bahwa Penyusunan Rencana Anggaran Biaya (RAB)  
suatu proyek adalah kegiatan yang harus dilakukan sebelum proyek dilaksanakan. RAB adalah banyaknya biaya yang dibutuhkan baik upah maupun bahan dalam sebuah pekerjaan proyek konstruksi. Daftar ini berisi volume, harga satuan, serta total harga dari berbagai macam jenis material dan upah tenaga yang dibutuhkan untuk pelaksanaan proyek tersebut.

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

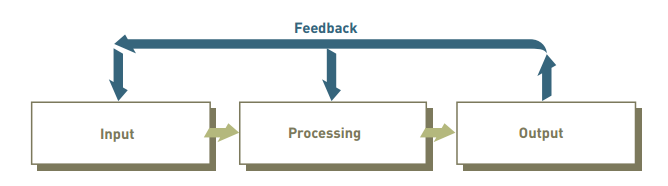
Tabel 2.1: Tabel Perbandingan Kajian Penelitian Terdahulu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Peneliti | Kajian 1 | Kajian 2 | Kajian 3 | Kajian 4 | Kajian 5 | Topik Yang Akan Dikembangkan |
| Judul | PERANCANGAN APLIKASI RENCANA ANGGARAN BIAYA (RAB) | 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 | Desktop Application menggunakan bahasa pemograman java | Desktop Application  (Borland Delphi 7.0.) | Desktop Application  (C#) | Web based application (PHP) | Computer Based System | Web based application (PHP) laravel framework |
| Database | mySQL | SQL server 7.0. | DBMS SQL Server 2008 R2 | MySQL |  | MySql |
| Fitur Umum | Manajeman pengguna, tahun anggaran, kategori material tenaga, manajemen satuan, material tenaga kerja, proyek, *time schedule* | 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 |  | Perencanaan anggraan, realisai anggaran, pencatatan pemasukan, laporan laba-rugi, laporan keuangan bulanan, laporan evaluasi berupa grafik. |
| Tempat Penelitian | DINAS PEKERJAAN UMUM KOTA SALATIGA | CV. X | SMK Marsudi Luhur I Yogyakarta |  | GMIM Nafiri Malalayang | Kelompok tani hidroponik RW 12 Kauman yogyakarta. |
| Hasil | perancangan aplikasi yang mampu menjabarkan RAB untuk pembangunan dan pemeliharaan/ rehabilitasi gedung menjadi daftar kebutuhan material dan tenaga dengan teliti, cepat, akurat, dan cermat. | 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 Pengguna**
5. **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. *Alpha Test*

Alpha test yaitu pengujian program dengan cara mengundang beberapa pengguna (ketua , anggota) untuk menjalankan aplikasi, kemudian masing-masing pengguna akan diberikan quisioner untuk memberikan penilaian terhadap aplikasi yang dijalankan baik ditinjau dari tampilan antar muka (*user interface*), kemudian dalam pemasukan data, menjalankan operasi yang ada di dalam aplikasi.

Pengujian sistem sangat diperlukan untuk menentukan keberhasilan suatu sistem dan melakukan perbaikan-perbaikan jika masih terjadi kesalahaan atau perubahan fitur sesuai dengan kebutuhan pengguna.

**Bab IV**

**Hasil Dan Pembahasan**

1. Analisis Kebutuhan Sistem

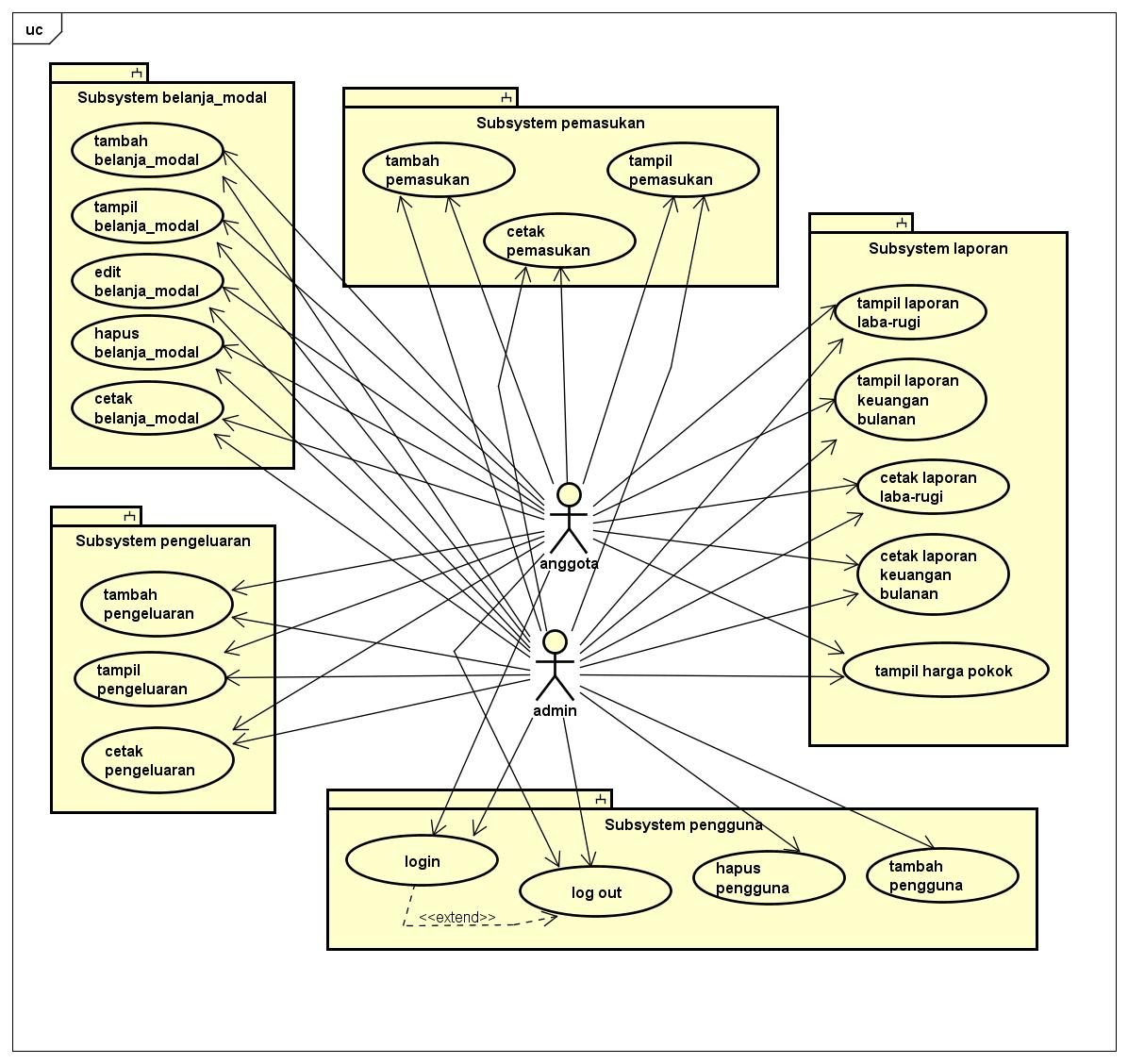
Dari hasil pengumpulan data yang dilakukan dengan metode wawancara, di dapatkan data-data yang yang diperlukan untuk membangun sistem informasi keuangan berbasis web. Data yang telah didapat kemudian di analisis dan menghasilkan spesifikasi kebutuhan sistem (*system requirement spesification*).

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 belanja modal 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 pengeluaran 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 belanja modal

Tabel 4.4 *use case scenario* input belanja modal

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

1. Edit belanja modal

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Edit belanja modal | |
| *Actor* | Admin ,Anggota | |
| *Description* | *Use case* edit belanja modal digunakan untuk menggambarkan proses mengedit data belanja modal yang sudah ada | |
| *Pre Contidion* | Data belanja modal yang di edit harus sudah ada sebelumnya | |
| *Trigger* | jika ingin mengubah data belanja modal . | |
| *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 belanja modal | |
| *Conclusion* | Data belanja modal berhasil di ubah | |
| *Post Condition* | Data belanja modal yang diubah sudah di perbaharui di tabel anggaran | |

Tabel 4.5 *use case scenario* edit belanja modal

1. Cetak belanja modal

Tabel 4.6 *use case scenario* cetak belanja modal

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

1. tambah pengeluaran

Tabel 4.7 *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 4.8 *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 4.8 *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 4.10 *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 4.11 *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 4.12 *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 4.13 *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 4.14 *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 4.15 *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 4.16 *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 4.17 *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 belanja modal

Tabel 4.18 *use case scenario* tampil belanja modal

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Tampil belanja modal | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* tampil anggaran menjelaskan proses menampilkan data belanja modal sesuai periode yang diinginkan | |
| *Pre Contidion* | Sudah ada data di tabel belanja modal | |
| *Trigger* | Saat actor ingin menampilkan belanja modal 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 belanja modal |
| *Alternate Course* | Jika actor tidak memilih tahun dan bulan periode anggaran, maka sistem akan menampilkan data belanja modal pada periode sekarang. | |
| *Conclusion* | Data belanja modal berhasil ditampilkan | |
| *Post Condition* | Data belanja modal berhasil ditampilkan sesuai periode. | |

1. Hapus belanja modal

Tabel 4.19 *use case scenario* hapus belanja modal

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Hapus belanja modal | |
| *Actor* | Admin, Anggota | |
| *Description* | *Use case* hapus belanja modal menjelaskan proses menghapus data anggaran | |
| *Pre Contidion* | Tampil data tabel belanja modal | |
| *Trigger* | Saat actor ingin menghapus salah satu data belanja modal | |
| *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* | Data belanja modal berhasil dihapus | |
| *Post Condition* | Data belanja modal berhasil dihapus | |

1. Hapus pengguna

Tabel 4.20 *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

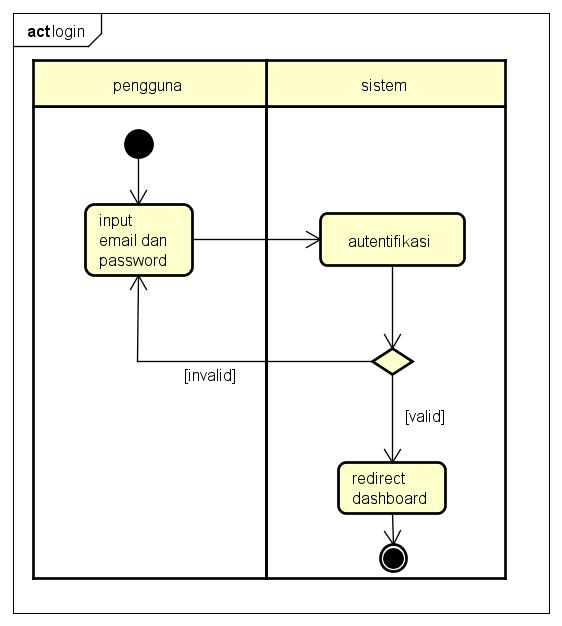
Tabel 4.21 *use case scenario* login

|  |  |  |
| --- | --- | --- |
| *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.

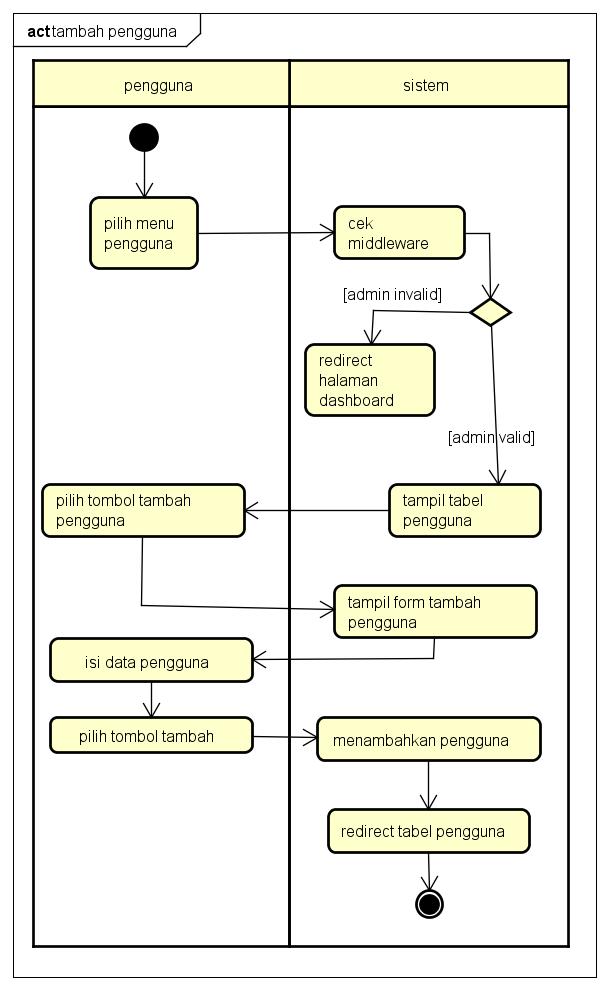
* 1. *Activity diagram* login



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.

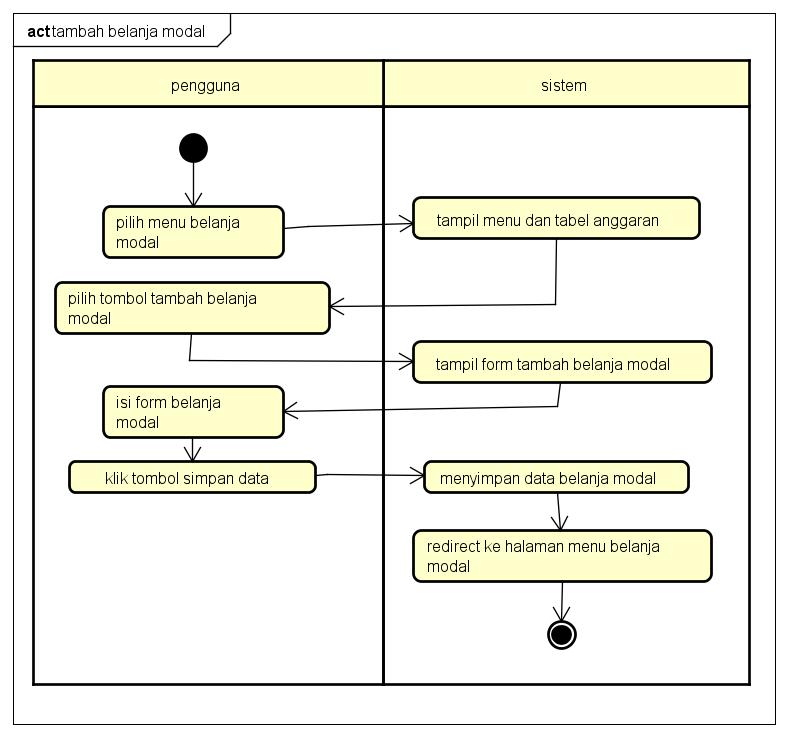
* 1. *Activity diagram* tambahpengguna

**

Gambar 4.3 *Activity diagram* tambahpengguna

Gambar 4.3 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.

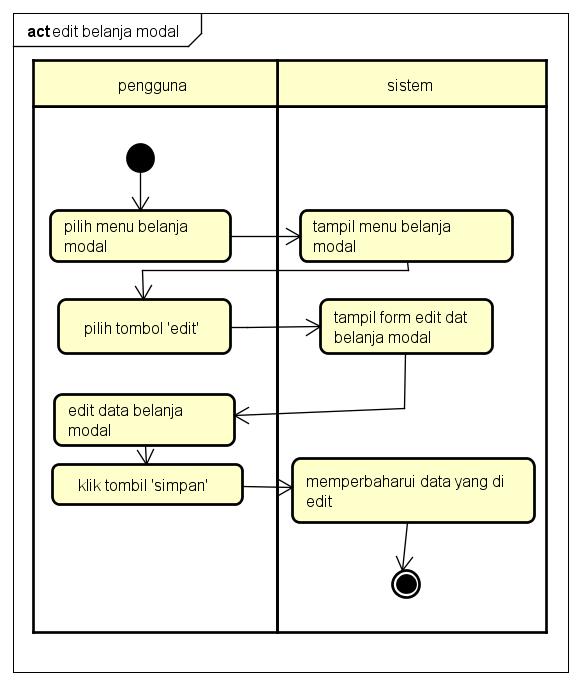
* 1. *Activity diagram* tambah belanja modal

**

Gambar 4.4 *Activity diagram* tambah belanja modal

Gambar 4.4 Aktifitas tambah belanja modal dimulai dengan memilih menu belanja modal kemudian pilih tombol tambah belanja modal. Kemudian actor mengisi data belanja modal yaitu periode, nama item belanja modal, harga satuan, QTY, anggaran. Selanjutnya klik tombol simpan data. Dan sistem akan meredirect ke halaman menu belanja modal.

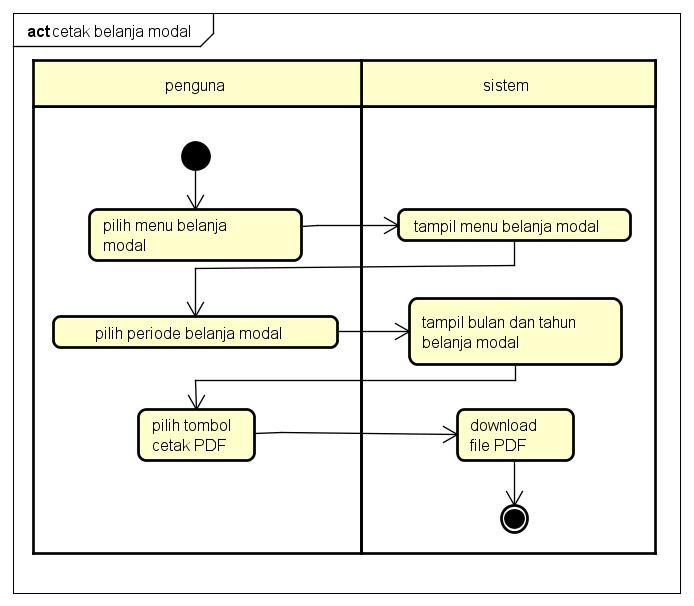
* 1. *Activity diagram* edit belanja modal



Gambar 4.5 *Activity diagram* edit belanja modal

Gambar 4.5 Aktifitas edit belanja modal dilakukan saat pengguna ingin mengubah data belanja modal. Dimulai dari memilih menu belanja modal, kemudian memilih data yang ingin di edit, klik tombol ‘edit’, maka sistem akan menampilkan halaman edit belanja modal 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 belanja modal.

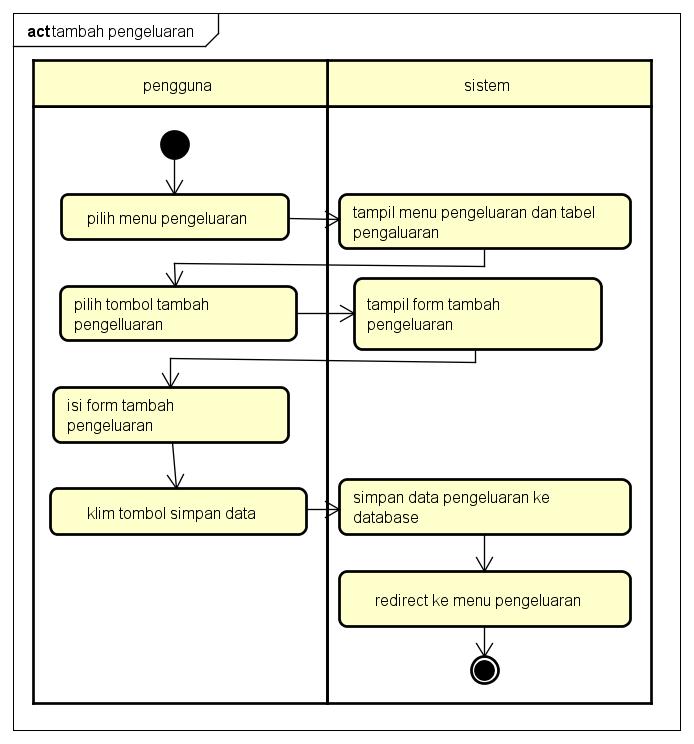
* 1. *Activity diagram* cetak belanja modal

**

Gambar 4.6 *Activity diagram* cetak anggaran

Sesuai gambar 4.6 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.

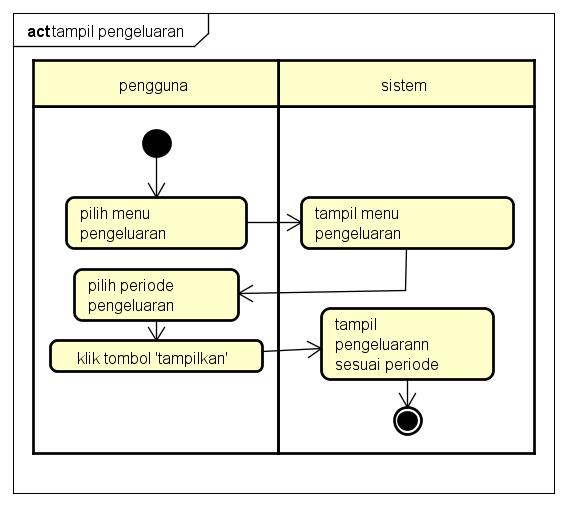
* 1. *Activity diagram* tambah pengeluaran



Gambar *4.7 Activity diagram* tambah pengeluaran

Gambar 4.7 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.

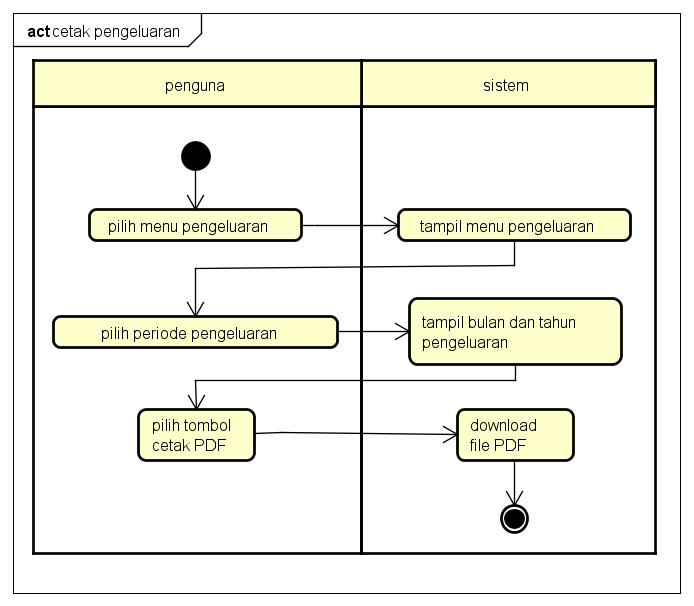
* 1. *Activity diagram* tampil pengeluaran



gambar 4.8 *Activity diagram* tampil pengeluaran

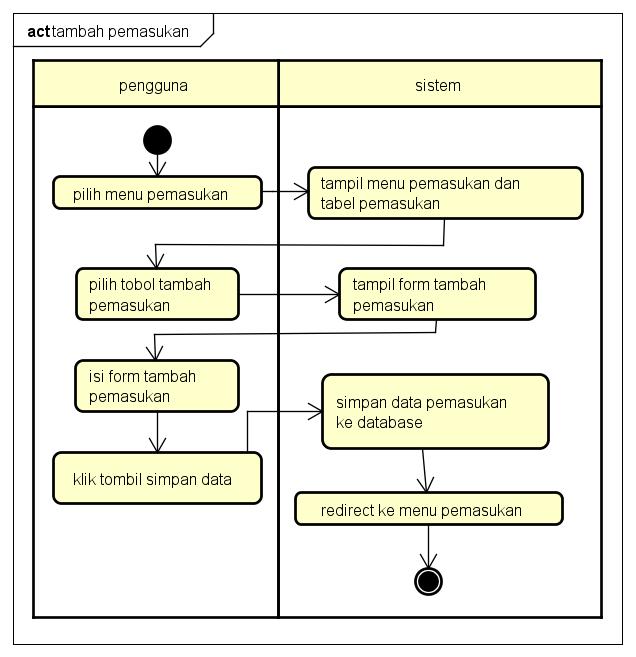
gambar 4.8 menjelaskan aktifitas tampil pengeluaran menjelaskan alur sistem dalam menampilkan data pengeluaran. dimulai dari pengguna memilih menu pengeluaran, kemudian sistem akan menampilkan menu pangeluaran. Pengguna memilih periode pengeluaran terdiri dari tahun dan bulan. Kemudian pilih tombol tampilkan. Maka sistem akan menampilkan data pengeluaran sesuai periode yang dipilih.

* 1. *Activity diagram* cetak pengeluaran

** Gambar 4.9 *Activity diagram* cetak pengeluaran

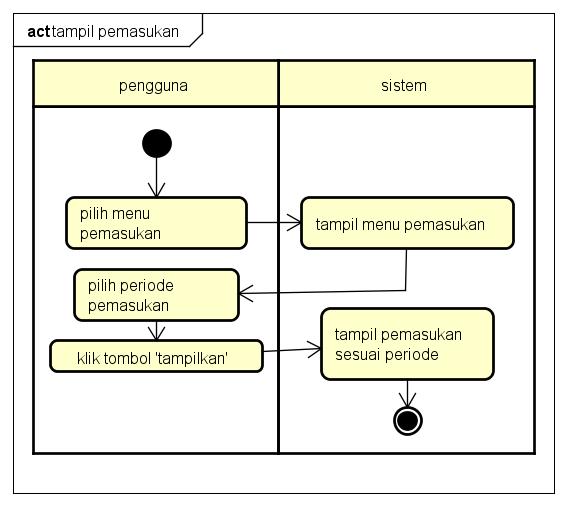
Gambar 5.9 aktifitas cetak pengeluaran menjelaskan alur saat sistem mencetak data pengeluaran dalam bentuk PDF. Dimulai dari pengguna memilih menu pengeluaran. Kemudian sistem akan menampilkan menu pengeluaran. Pengguna memilih periode pengeluaran yang akan dicetak terdiri dari tahun dan bulan. Kemudian pilih tombol ‘cetak PDF’. Maka sistem akan men­-*download* data pengeluaran dalam bentuk PDF.

* 1. *Activity diagram* tambah pemasukan

** Gambar 4.10 *Activity diagram* tambah pemasukan

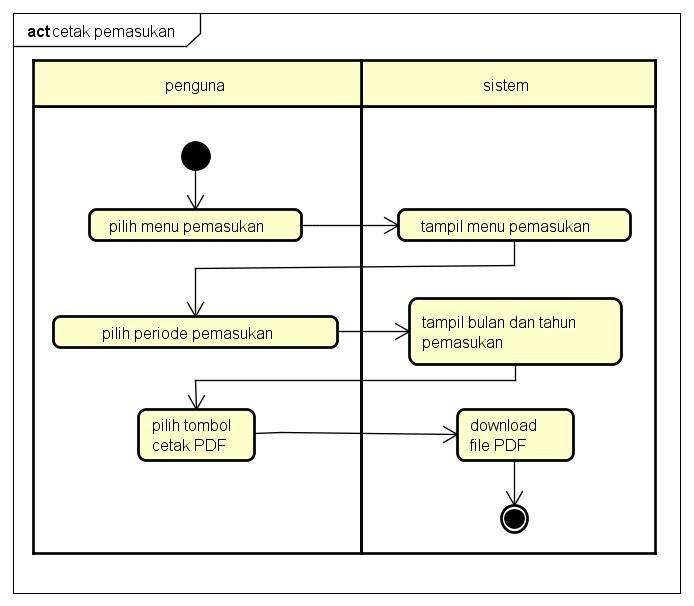
Sesuai gambar 4.10 aktifitas tambah pemasukan menjelaskan alur untuk menambah data pemasukan. Dimulai dari pengguna memilih menu pemasukan, kemudian sistem akan menampilkan menu pemasukan. Pengguna kemudian pilih tombol ‘tambah pemasukan’. Maka sistem akan menampilkan form untuk menambah pemasukan. Terdiri 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 4.11 *Activity diagram* tampil pemasukan

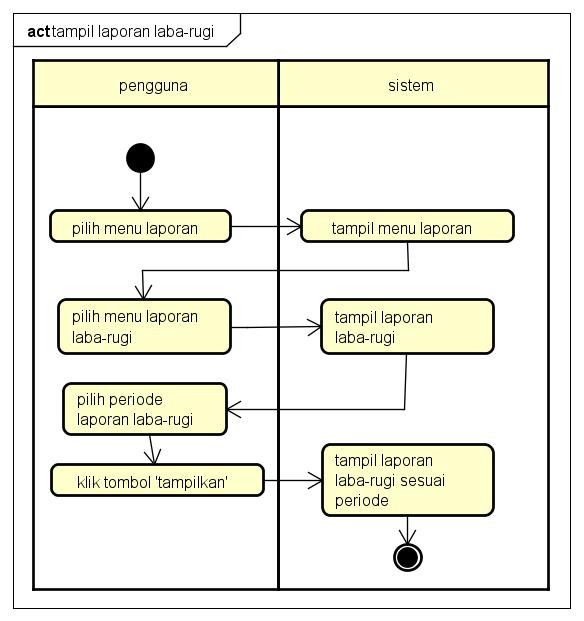
Sesuai gambar 4.11 aktifitas tampil pemasukan menjelaskan alur sistem dalam menampilkan data pemasukan. dimulai dari pengguna memilih menu pemasukan, kemudian sistem akan menampilkan menu pemasukan. Pengguna memilih periode pemasukan terdiri dari tahun dan bulan. Kemudian pilih tombol tampilkan. Maka sistem akan menampilkan data pemasukan sesuai periode yang dipilih.

* 1. *Activity diagram* cetak pemasukan

 Gambar 4.12 *Activity diagram* cetak pemasukan

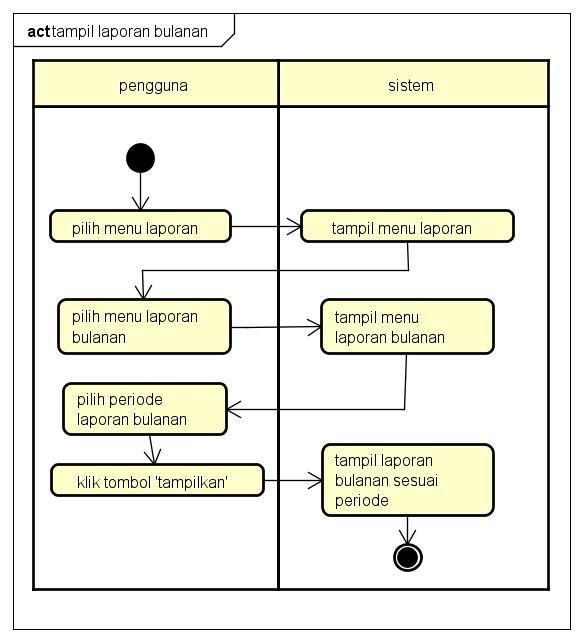
Gambar 4.12 yaitu aktifitas cetak pemasukan menjelaskan alur saat sistem mencetak data pemasukan dalam bentuk PDF. Dimulai dari pengguna memilih menu pemasukan. Kemudian sistem akan menampilkan menu pemasukan. Pengguna memilih periode pemasukan yang akan dicetak terdiri dari tahun dan bulan. Kemudian pilih tombol ‘cetak PDF’. Maka sistem akan men­-*download* data pemasukan

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

** Gambar 4.13 *Activity diagram* tampil laporan laba-rugi

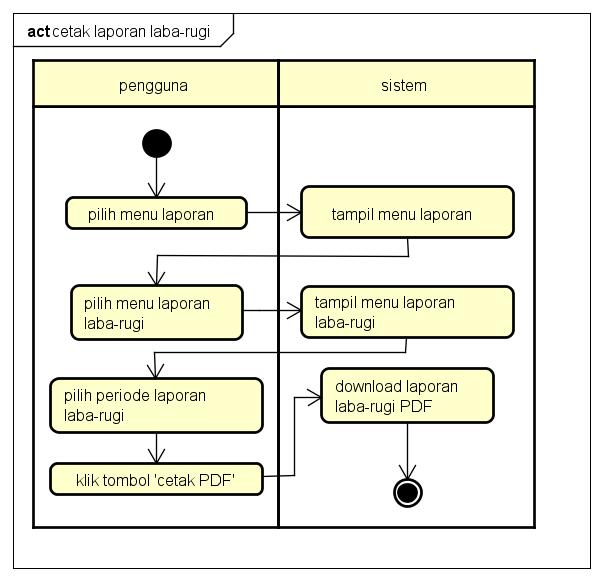
Gambar 4.13 aktifitas tampil laporan laba rugi menjelaskan alur saat sistem menampilkan laporan laba rugi. Dimulai dari pengguna memilih menu laporan, kemudian memilih laporan laba-rugi. Maka sistem akan menampilkan data laba-rugi. Pengguna memilih periode laporan terdiri dari tahun dan bulan. Kemudian pengguna pilih tombol ‘tampilkan’. Maka sistem akan menampilkan laporan laba-rugi sesuai periode.

* 1. *Activity diagram* tampil laporan keuangan bulanan

 Gambar 4.14 *Activity diagram* tampil laporan keuangan bulanan

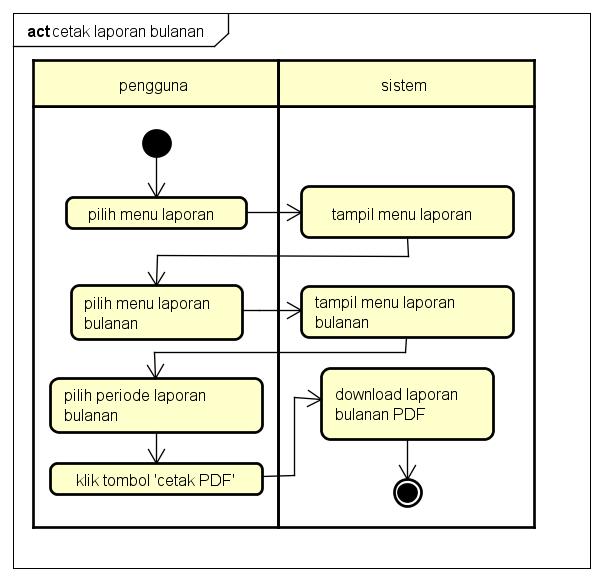
Gambar 4.14 menjelaskan alur saat sistem menampilkan laporan bulanan. Dimulai dari pengguna memilih menu laporan, kemudian memilih laporan bulanan. Maka sistem akan menampilkan data laporan bulanan. Pengguna memilih periode laporan terdiri dari tahun dan bulan. Kemudian pengguna pilih tombol ‘tampilkan’. Maka sistem akan menampilkan laporan bulanan sesuai periode yang dipilih.

* 1. *Activity diagram* cetaklaporan laba-rugi

 Gambar 4.15 *Activity diagram* cetak laporan laba-rugi

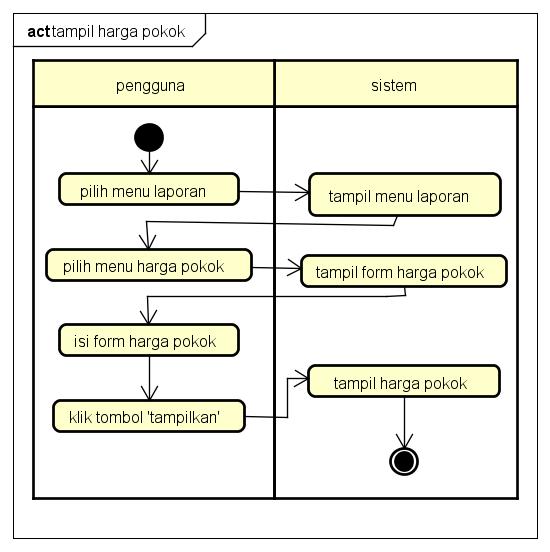
Gambar 4.15 menjelaskan alur saat sistem mencetak data laporan laba-rugi dalam bentuk PDF. Dimulai dari pengguna memilih menu laporan. Kemudian sistem akan menampilkan menu laporan laba-rugi . Pengguna memilih periode laba-rugi yang akan dicetak terdiri dari tahun dan bulan. Kemudian pilih tombol ‘cetak PDF’. Maka sistem akan men­-*download* data laporan laba-rugi dalam bentuk PDF.

* 1. *Activity diagram* cetak laporan keuangan bulanan

** Gambar 4.16 *Activity diagram* cetak laporan keuangan bulanan

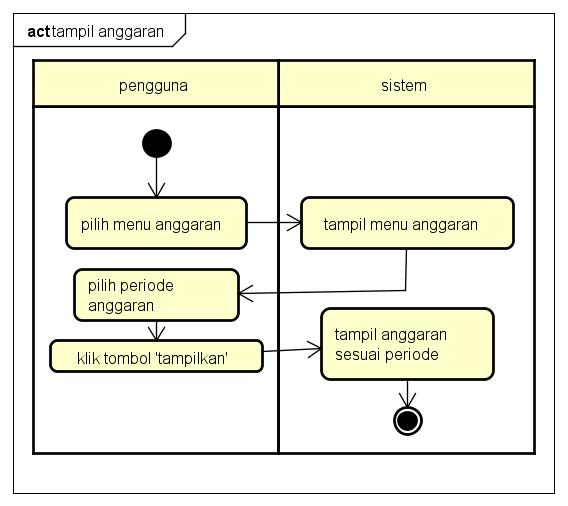
Gambar 4.16 menjelaskan alur saat sistem mencetak data laporan keuangan bulanan dalam bentuk PDF. Dimulai dari pengguna memilih menu laporan. Kemudian sistem akan menampilkan menu laporan keuangan bulanan. Pengguna memilih periode laporan keuangan bulanan yang akan dicetak terdiri dari tahun dan bulan. Kemudian pilih tombol ‘cetak PDF’. Maka sistem akan men­-*download* data laporan keuangan bulanan dalam bentuk PDF.

* 1. *Activity diagram* tampil harga pokok

** Gambar 4.17 *Activity diagram* tampil harga pokok

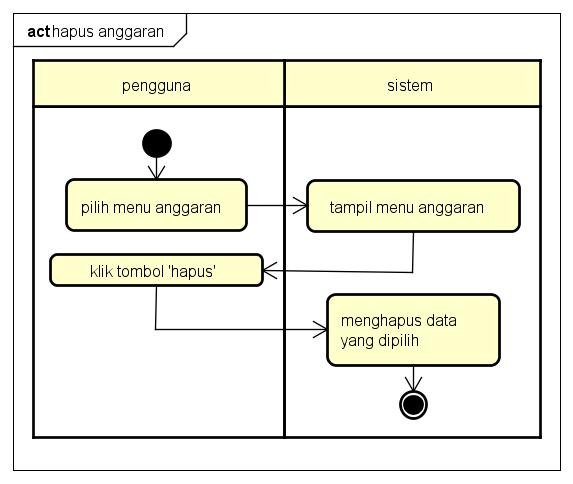
Gambar 4.17 menjelaskan alur sistem dalam menampilkan harga pokok penjualan. Dimulai dari pengguna memilih menu laporan, sistem akan menampilkan menu laporan. Kemudian pengguna memilih menu harga pokok. Sistem akan menampilkan form perhitungan harga pokok. Pengguna mengisi form yang terdiri dari jumlah 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 4.18 *Activity diagram* tampil anggaran

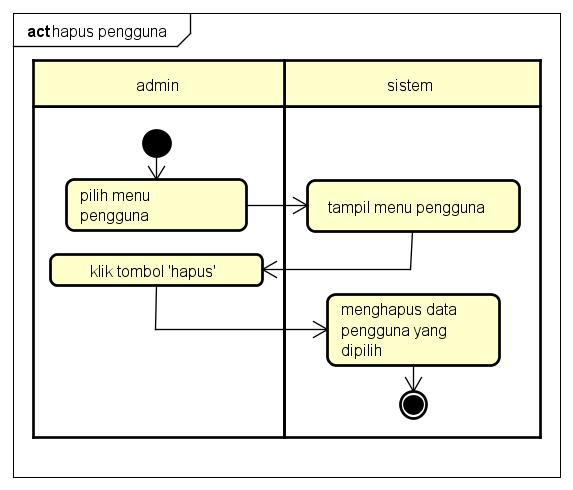
Gambar 4.18 menjelaskan alur sistem dalam menampilkan data anggaran. dimulai dari pengguna memilih menu anggaran, kemudian sistem akan menampilkan menu anggaran. Pengguna memilih periode anggaran terdiri dari tahun dan bulan. Kemudian pilih tombol tampilkan. Maka sistem akan menampilkan data anggaran sesuai periode yang dipilih.

* 1. *Activity diagram* hapus anggaran

 Gambar 4.19 *Activity diagram* hapus anggaran

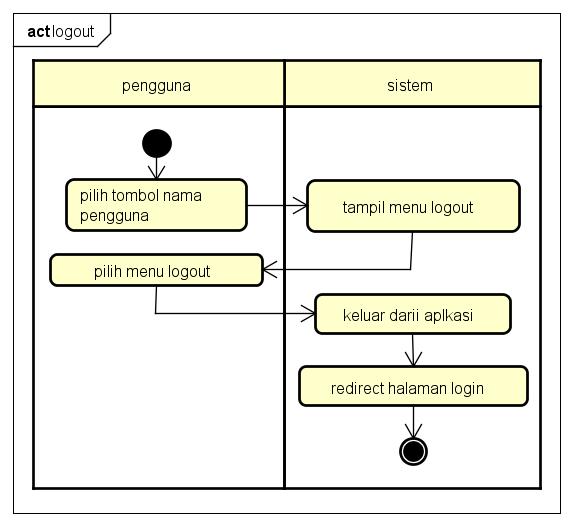
Gambar 4.19 menjelaskan alur saat pengguna akan menghapus salah satu data anggaran. Dimulai dari pengguna memilih menu anggaran. Kemudian memilih data yang dihapus dan pilih tombol ‘hapus’. Maka sistem akan menghapus data yang dipilih.

* 1. *Activity diagram* hapus pengguna

** Gambar 4.20 *Activity diagram* hapus pengguna

Gambar 4.20 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*

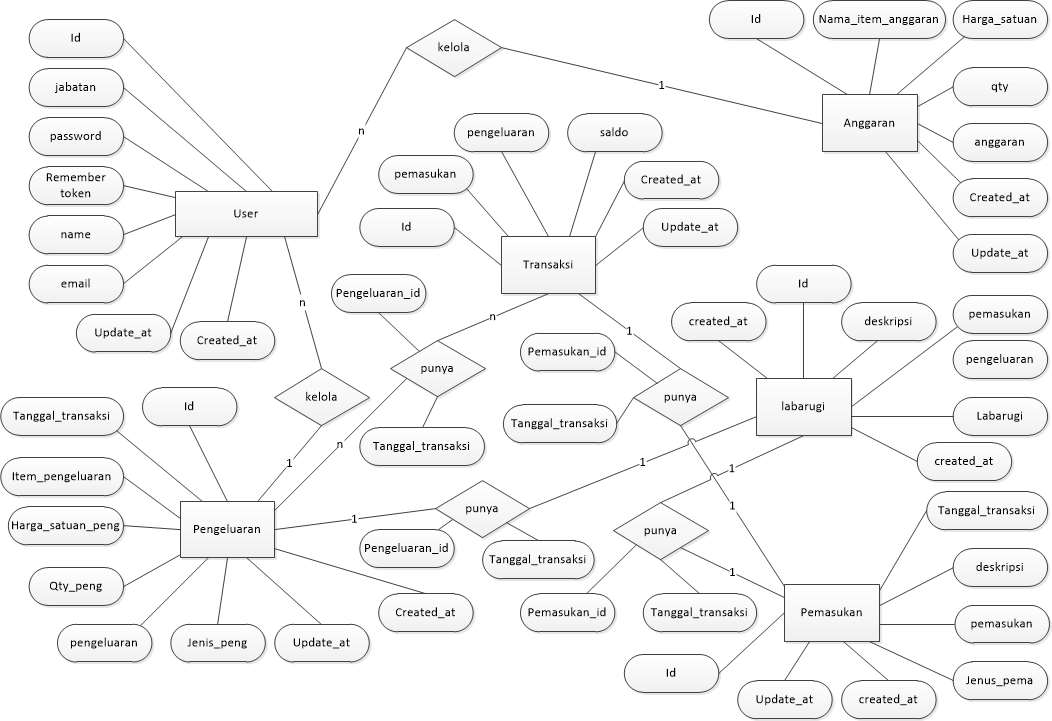
 Gambar 4.21 *Activity diagram logout*

Gambar 4.21 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

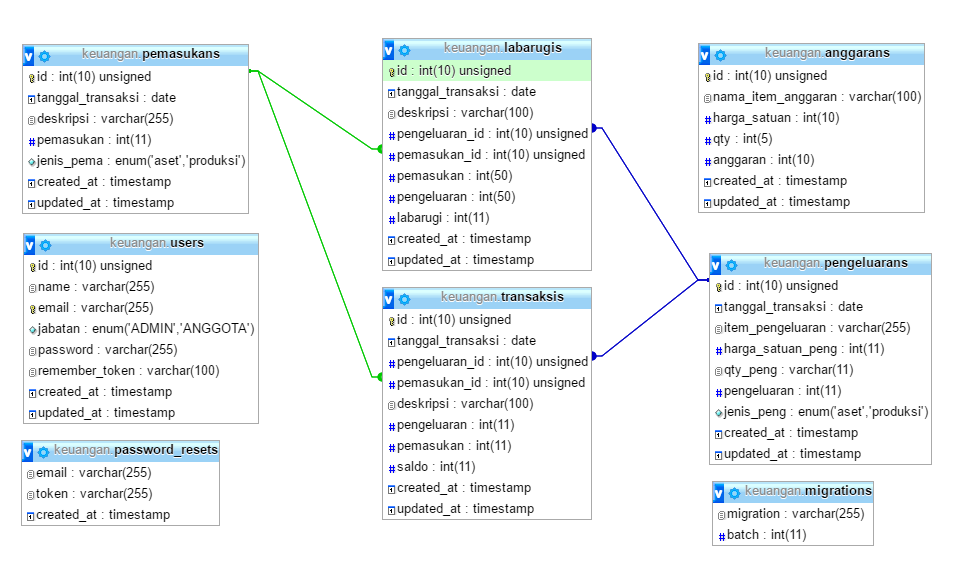
Desain data digunakan untuk manggambarkan pemodelan basis data yang ada pada aplikasi sistem informasi keuangan. *Entity Relationship Diagram (ERD)* digunakan dalam tahap ini untuk memodelkan struktur data dan hubungan antar tabel yang ada dalam sistem informasi keuangan ini. Erd dapat dilihat pada gambar 4.22.

* 1. ERD (*Entity Relationship Diagram*)

 Gambar 4.22. 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 4.23. Berikut adalah mapping table berdasarkan ERD yang telah dibuat.

 Gambar 4.23. *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 4.22 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 4.23 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 4.24 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 4.25 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 4.26 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

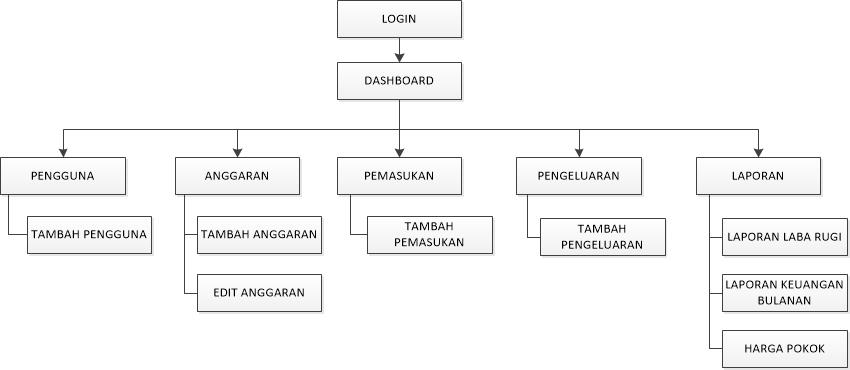
Tabel 4.27 tabel labarugis

|  |  |  |
| --- | --- | --- |
| 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 |

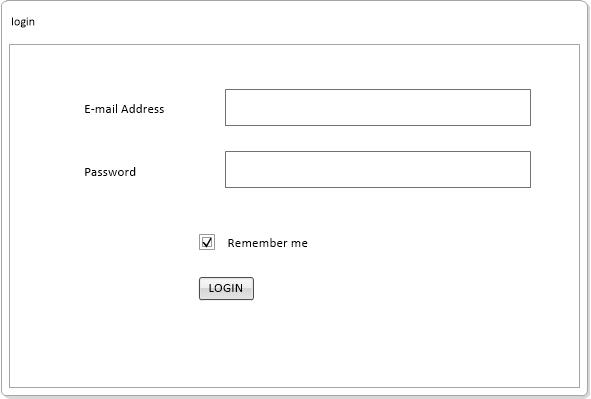
1. Desain Antarmuka

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 4.24 struktur menu sistem informasi keuangan

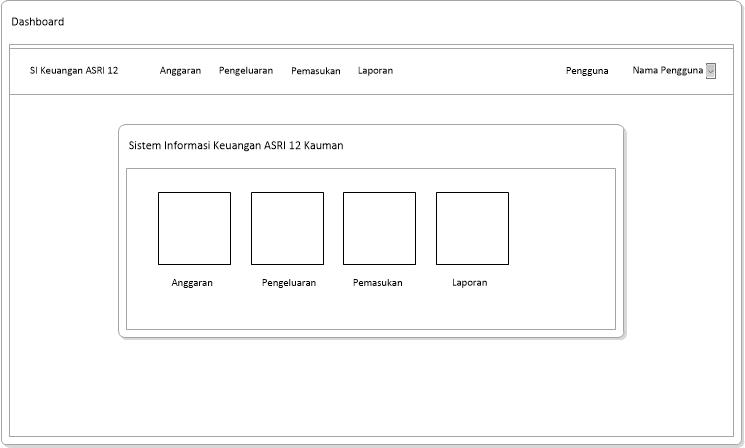
1. Login



Gambar 4.25 desain halaman login

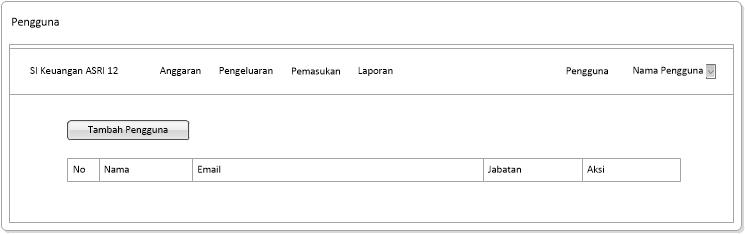
Gambar 4.25 halaman login digunakan untuk menampilkan form email dan password yang diperlukan untuk masuk ke dalam aplikasi.

1. Dashboard

 Gambar 4.26 desain halaman dashboard

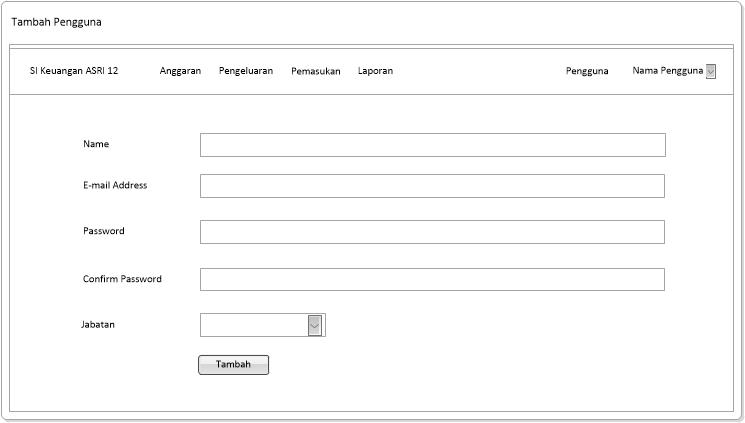
Halaman *dashboard* pada gambar Gambar 4.26 berisi menu utama dari aplikasi ini. Menampilkan menu utama yang bisa digunakan pengguna seperti belanja modal, pemasukan , pengeluaran dan laporan. Setiap menu utama menggunakan ikon yang gambarnya sesuai dengan nama menu utamanya.

1. Pengguna

 Gambar 4.27 desain halaman pengguna

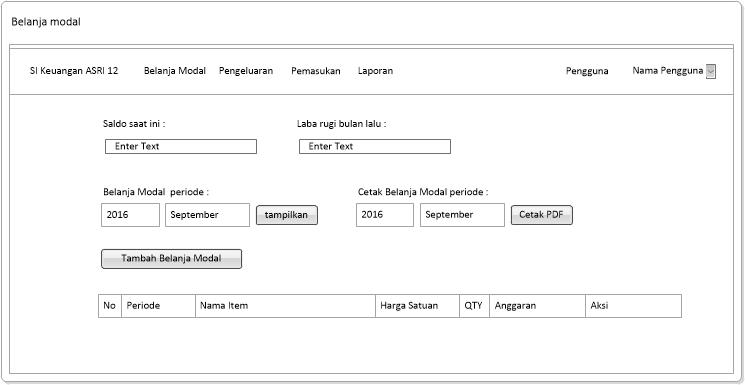
Pada halaman pengguna pada gambar 4.27 menampilkan tabel daftar pengguna yang dapat menggunakan aplikasi. Terdapat tombol tambah pengguna yang digunakan utntuk mengarah ke halaman form tambah pengguna.

1. Tambah pengguna

 Gambar 4.28 desain halaman tambah pengguna

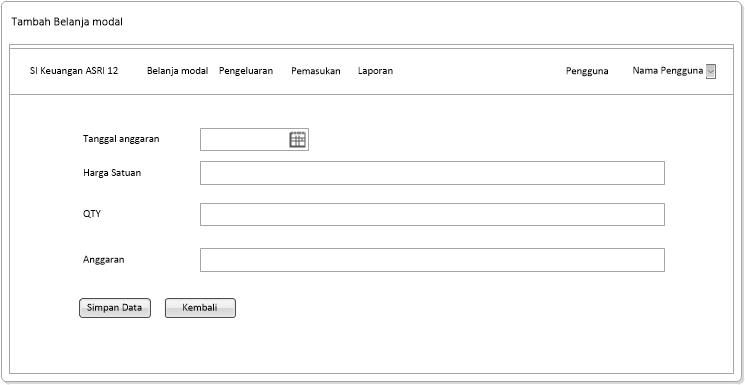
Halaman tambah pengguna pada gambar 4.28 berisi form-form yang digunakan untuk menyimpan data pengguna, seperti nama, email, password, confirm password, dan jabatan. Untuk form jabatan menggunakan menu drop down. Kemudian terdapat tombol tambah yang digunakan untuk menambahkan pengguna.

1. Belanja Modal

 Gambar 4.29 desain halaman belanja modal

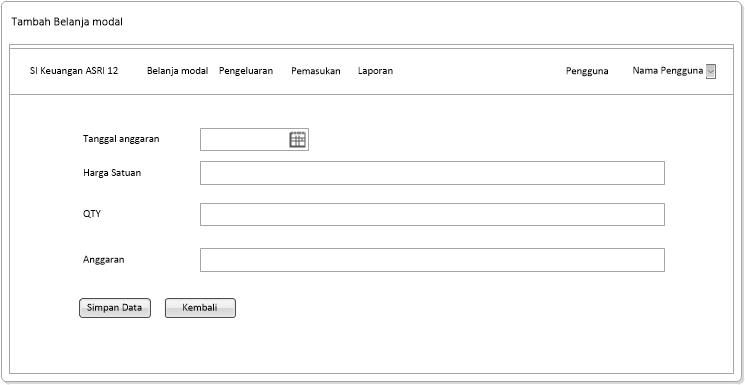
Halaman belanja modal pada gambar 4.29 berisi form dan tombol yang mendukung informasi daftar belanja modal. Form saldo saat ini digunakan untuk mengetahui saldo yang ada pada saat itu, sedangkan form labarugi bulan lalu digunakan sebagai patokan untuk membuat daftar belanja modal. Menu drop down untuk periode digunakan untuk memilih periode yang ingin di tampilkan dan dicetak. Tombol tambah belanja modal digunakan untuk mengarah ke halaman menambah data belanja modal.

1. Tambah belanja modal

 Gambar 4.30 desain halaman tambah belanja modal

Gambar 4.30 digunakan untuk menambah data belanja modal, terdapat form tanggal anggaran, harga satuan, qty dan form anggaran otomatis terkalkulasi.

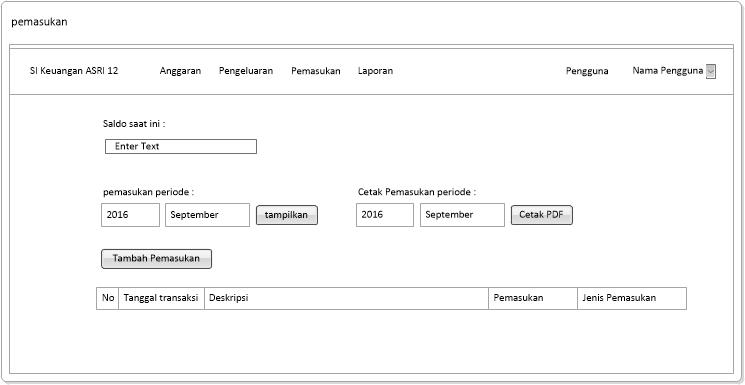
1. Edit belanja modal



Gambar 4.31 desain halaman edit belanja modal

Tampilan halaman edit belanja modal pada gambar 4.31 sama dengan tampilan halaman tambah belanja modal.

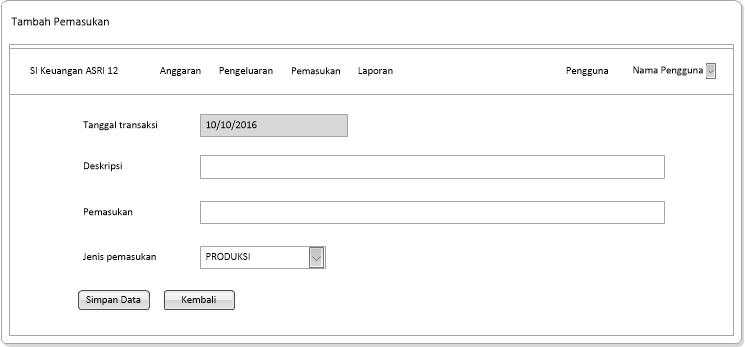
1. Pemasukan



Gambar 4.32 desain halaman pemasukan

Halaman utama pemasukan pada gambar 4.32 berisi form dan tombol yang mendukung informasi pemasukan. Form saldo saat ini digunakan untuk mengetahui saldo yang ada pada saat itu. Menu drop down untuk periode digunakan untuk memilih periode yang ingin di tampilkan dan dicetak. Tombol tambah pemasukan digunakan untuk mengarah ke halaman menambah data pemasukan.

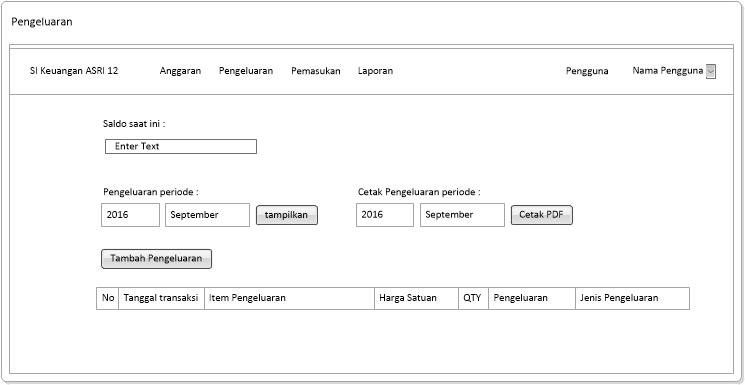
1. Tambah pemasukan



Gambar 4.33 desain halaman tambah pemasukan

Halaman tambah pemasukan pada gambar 4.33 digunakan untuk menambah data pemasukan yang terdiri dari form tanggal transaksi yang otomatis mengambil tanggal pada hari itu. Form deskripsi, dan pemasukan serta menu drop down pilihan jenis pemasukan

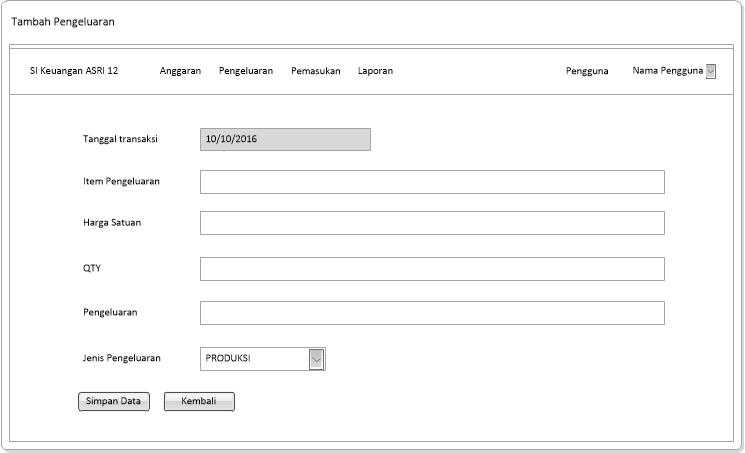
1. Pengeluaran



Gambar 4.34 desain halaman pengeluaran

Halaman utama pengeluaran pada gambar 4.34 berisi form dan tombol yang mendukung informasi pengeluaran. Form saldo saat ini digunakan untuk mengetahui saldo yang ada pada saat itu. Menu drop down untuk periode digunakan untuk memilih periode yang ingin di tampilkan dan dicetak. Tombol tambah pengeluaran digunakan untuk mengarah ke halaman menambah data pemasukan. Serta terdapat tabel yang menampilkan pengeluaran periode saat itu atau periode sesuai pilihan pengguna.

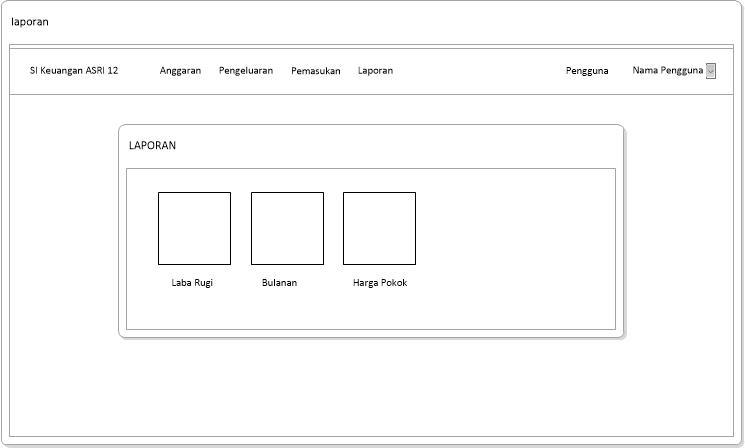
1. Tambah pengeluaran



Gambar 4.35 desain halaman tambah pengeluaran

Halaman tambah pengeluaran pada gambar 4.35 digunakan untuk menambah data pengeluaran yang terdiri dari form tanggal transaksi yang otomatis mengambil tanggal pada hari itu. Form deskripsi, harga satuan, qty,dan form pemasukan yan otomatis terkalkulasi serta menu drop down pilihan jenis pengeluaran

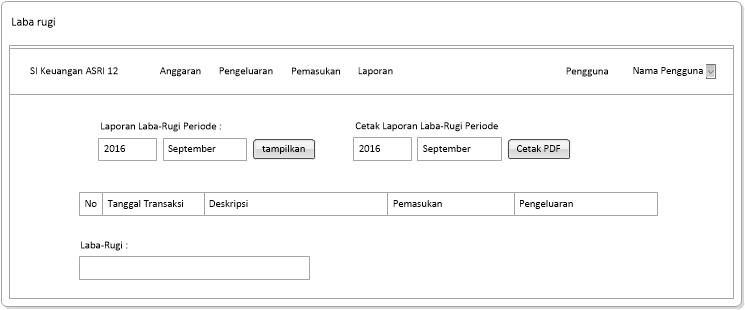
1. Laporan



Gambar 4.36 desain halaman laporan

Tampilan utama halaman laporan pada gambar 4.36 terdapat 3 menu utama laporan yaitu laba rugi, bulanan, dan harga pokok. Masing-masing menu utama di dukung ikon yang terkait dengan menunya.

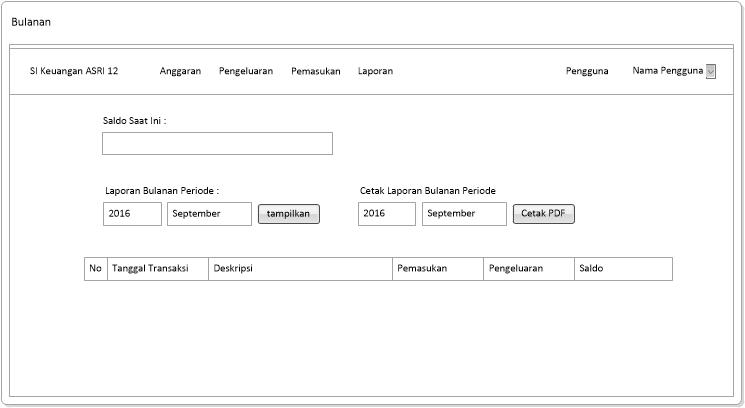
1. Laporan laba-rugi



Gambar 4.37 desain halaman laporan laba-rugi

Pada gambar 4.37 yaitu desain halaman laporan laba rugi berisi tombol dan form serta tabel yang mendukung informasi laporan laba rugi. Antara lain pilihan periode, tabel laba rugi dan form laba rugi pada periode itu.

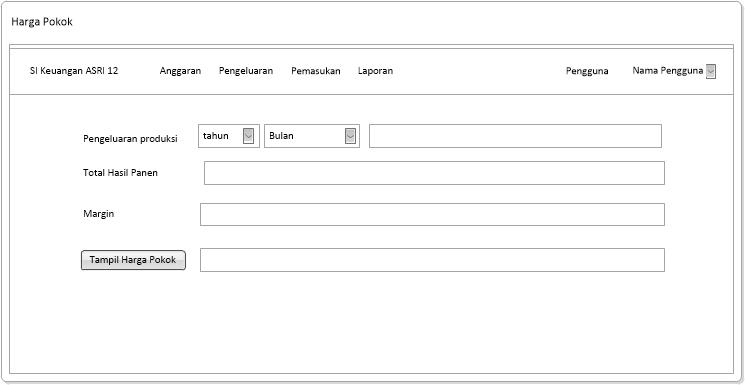
1. Laporan keuangan bulanan



Gambar 4.38 desain halaman laporan keuangan bulanan

Pada gambar 4.38 yaitu halaman laporan bulanan bersi menu yang mendukung informasi laporan keuangan bulanan. Terdapat form saldo saat ini, menu drop down untuk memilih periode, dan tabel pemasukan dan pengeluaran periode saat itu atau sesuai pilihan pengguna.

1. Harga pokok

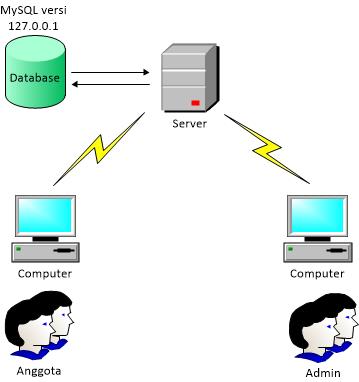


Gambar 4.39 desain halaman harga pokok

Halaman harga pokok pada gambar 4.39 digunakan untuk mengkalkulasi harga pokok penjualan produk yang akan dijual di pasaran. Terdapat form yang mendukung perhitungan 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. Arsitektur sistem dapat dilihat pada gambar 4.40.



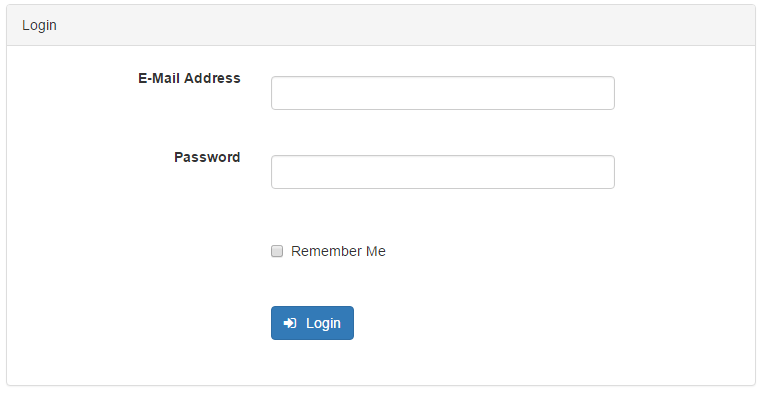
Gambar 4.40 arsitektur sistem aplikasi sistem informasi keuangan

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 4.41 Tampilan halaman login

Halaman login pada gambar 4.41 muncul saat pertama kali pengguna mengakses halaman [www.sikasri12kauman.com](http://www.sikasri12kauman.com) .

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;
4. class User extends Authenticatable
5. {        protected $fillable = [
6. 'name', 'email', 'password','jabatan',
7. ];
8. protected $hidden = [
9. 'password', 'remember\_token',
10. ];
11. }

Script 4.1: Model user.php

Model user berisi script yang mengizinkan aplikasi untuk mengakses database dan mengizinkan field apa saja yang bisa di akses.

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() }}
11. <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
12. <label for="email" class="col-md-4 control-label">E-Mail Address</label>
13. <div class="col-md-6">
14. <input id="email" type="email" class="form-control" name="email" value="{{ old('email') }}">
15. @if ($errors->has('email'))
16. <span class="help-block">
17. <strong>{{ $errors->first('email') }}</strong>
18. </span>
19. @endif
20. </div</div>
21. <div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">  <label for="password" class="col-md-4 control-label">Password</label>
22. <div class="col-md-6">
23. <input id="password" type="password" class="form-control" name="password">
24. @if ($errors->has('password'))
25. <span class="help-block">
26. <strong>{{ $errors->first('password') }}</strong</span>  @endif
27. </div>                          </div>
28. <div class="form-group">
29. <div class="col-md-6 col-md-offset-4">
30. <div class="checkbox">
31. <label>
32. <input type="checkbox" name="remember"> Remember Me  </label></div>
33. </div>
34. </div>
35. <div class="form-group">
36. <div class="col-md-6 col-md-offset-4">
37. <button type="submit" class="btn btn-primary">  <i class="fa fa-btn fa-sign-in"></i> Login   </button>  <a class="btn btn-link" href="{{ url('/password/reset') }}">Forgot Your Password?</a>
38. </div>
39. </div>
40. </form>
41. </div>
42. </div>
43. </div>
44. </div>
45. </div>
46. @endsection

Script 4.2: View dari login.blade.php

*Script* 4.2 berisi code untuk menampilkan form login ketika pertama kali di akses pengguna. Pada script 4.2 menggunakan integrasi bootsrap yang dapat mempercantik tampilan halaman login.

1. <?php
2. namespace App\Http\Controllers\Auth;
3. use Validator;
4. use App\Http\Controllers\Controller;
5. use Illuminate\Foundation\Auth\ThrottlesLogins;
6. use Illuminate\Foundation\Auth\AuthenticatesAndRegistersUsers;
7. class AuthController extends Controller
8. {
9. use AuthenticatesAndRegistersUsers, ThrottlesLogins;
10. protected $redirectPath = '/';
11. public function \_\_construct()
12. {
13. $this->middleware($this->guestMiddleware(), ['except' => 'logout']);
14. }
15. protected function validator(array $data)
16. {
17. return Validator::make($data, [
18. 'name' => 'required|max:255',
19. 'email' => 'required|email|max:255|unique:users',
20. 'password' => 'required|min:6|confirmed',
21. ]);
22. }
23. protected function create(array $data)
24. {
25. return User::create([
26. 'name' => $data['name'],
27. 'email' => $data['email'],
28. 'password' => bcrypt($data['password']),
29. 'jabatan'=>$data['jabatan'],
30. ]);
31. }
32. }

Script 4.3: Controller dari authController.php

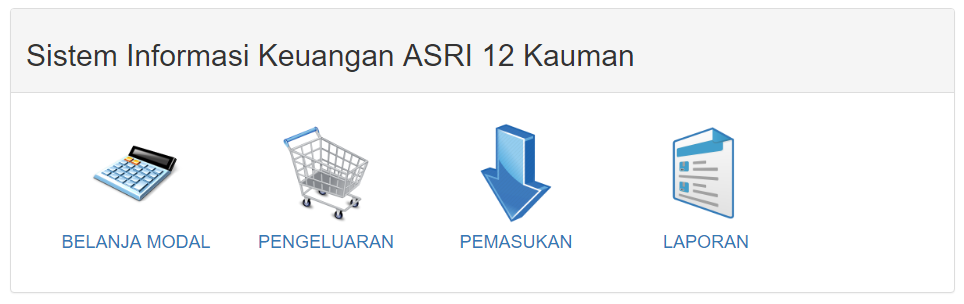
Controller auth digunakan untuk mengontrol autentifikasi pengguna.

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

Script 4.4: route untuk halaman login

Route halaman login didaftarkan di routes.php yang berfungsi untuk mengalamatkan sebuah halaman di url ketika diakses. Selain itu juga mendaftarkan *method* yang dipakai.

1. Dashboard atau *home*



Gambar 4.42 tampilan halaman [*home.blade.php*](http://home.blade.php)

Halaman *dashboard* pada gambar 4.42 muncul setelah pengguna login. Terdapat 4 menu utama lengkap dengan ikon yang terkait. Seperti belanja modal, pengeluaran, pemasukan, dan laporan.

1. <?php
2. namespace App\Http\Controllers;
3. use App\Http\Requests;
4. use Illuminate\Http\Request;
5. class HomeController extends Controller
6. {
7. public function \_\_construct()
8. {
9. $this->middleware('auth');
10. }
11. public function index()
12. {
13. return view('home');
14. }
15. }

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

Model *home* berisi middleware yang memastikan yang bisa mengakses halaman home adalah pengguna yang telah login.

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*](http://home.blade.php)

*View* dari home berisi kode yang menampilkan ikon dan mengandung link didalamnya. Setiap ikon akan diarahkan ke masing masing menu.

1. <?php
2. namespace App\Http\Controllers;
3. use App\Http\Requests;
4. use Illuminate\Http\Request;
5. class HomeController extends Controller
6. {
7. public function \_\_construct()
8. {
9. $this->middleware('auth');
10. }
11. public function index()
12. {
13. return view('home');
14. }
15. }

Script 4.7: *Controller* dari *homeController*

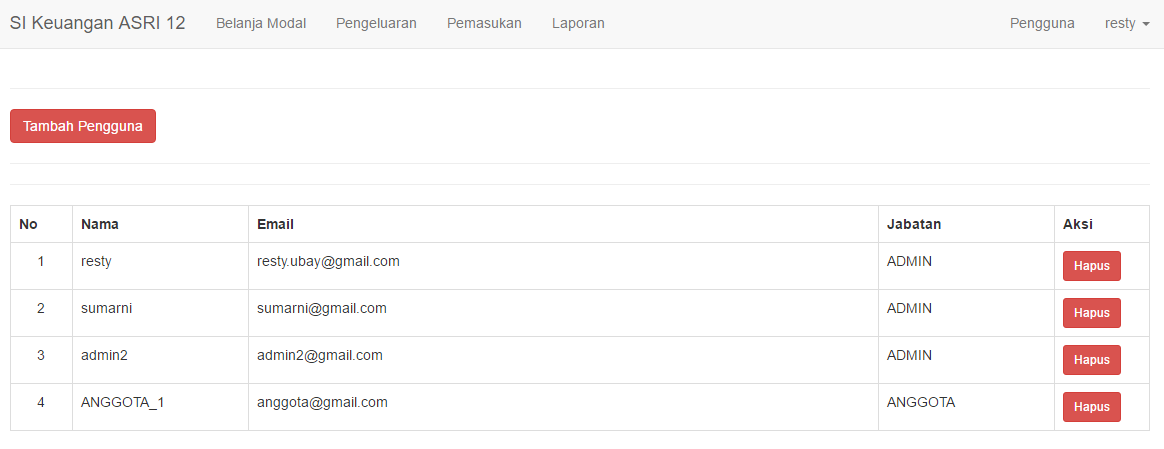
Controller dari home hanya berisi class index yang me-*redirect* ke halaman home.

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

Script 4.8: *Routes* dari halaman *dashboard*

*Route* halaman dashboard menggunakan url tanda slash (/) dan menggunakan method index() dalam controlernya.

1. Pengguna



Gambar 4.43 tampilan halaman pengguna

Pada gambar 4.43 yaitu halaman pengguna berfungsi menampilkan data pengguna yang terdaftar dan bisa login ke dalam aplikasi. Model users.php menggunakan model yang sama dengan model login.

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

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

Script 4.9 berisi code yang berfungsi menampilkan data pengguna yang bisa login ke dalam aplikasi.

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

Script 4.10 berisi controller untuk menampilkan data pengguna dengan method index(). Mengakses database user dan me*request*  semua data. Method index() mengembalikan nilai data berupa array 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. });

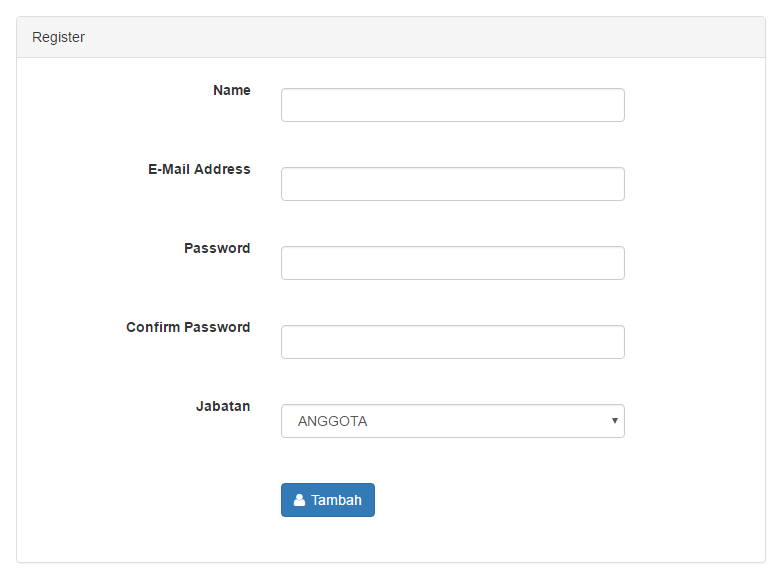
Script 4.11 *routes* untuk halaman pengguna

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 4.44 tampilan halaman tambah pengguna

Gambar 4.44 yaitu halaman untuk mendaftarkan pengguna yang bisa menggunakan aplikasi.

Model yang digunakan pada halaman tambah pengguna menggunakan model yang sama dengan model login. Halaman ini berfungsi untuk menambahkan penguna.

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">Register</div>
8. <div class="panel-body">
9. <form class="form-horizontal" role="form" method="POST" action="{{ url('/pengguna') }}">
10. {{ csrf\_field() }}
11. <div class="form-group{{ $errors->has('name') ? ' has-error' : '' }}">
12. <label for="name" class="col-md-4 control-label">Name</label>
13. <div class="col-md-6">
14. <input id="name" type="text" class="form-control" name="name" value="{{ old('name') }}">
15. @if ($errors->has('name'))
16. <span class="help-block">
17. <strong>{{ $errors->first('name') }}</strong>
18. </span>
19. @endif</div></div>
20. <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
21. <label for="email" class="col-md-4 control-label">E-Mail Address</label>
22. <div class="col-md-6">
23. <input id="email" type="email" class="form-control" name="email" value="{{ old('email') }}">
24. @if ($errors->has('email'))
25. <span class="help-block">
26. <strong>{{ $errors->first('email') }}</strong>
27. </span>
28. @endif</div></div>
29. <div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">
30. <label for="password" class="col-md-4 control-label">Password</label>
31. <div class="col-md-6">
32. <input id="password" type="password" class="form-control" name="password">
33. @if ($errors->has('password'))
34. <span class="help-block">
35. <strong>{{ $errors->first('password') }}</strong>
36. </span>
37. @endif</div></div>
38. <div class="form-group{{ $errors->has('password\_confirmation') ? ' has-error' : '' }}">
39. <label for="password-confirm" class="col-md-4 control-label">Confirm Password</label>
40. <div class="col-md-6">
41. <input id="password-confirm" type="password" class="form-control" name="password\_confirmation">
42. @if ($errors->has('password\_confirmation'))
43. span class="help-block">
44. <strong>{{ $errors->first('password\_confirmation') }}</strong>
45. </span>@endif</div></div>
46. <div class="form-group">
47. <label  class="col-md-4 control-label">Jabatan</label>
48. <div class="col-md-6"><select class="form-control" name='jabatan'>
49. <option value="ADMIN">ADMIN</option>
50. <option value="ANGGOTA" selected>ANGGOTA</option>
51. </select></div></div>
52. <div class="form-group">
53. <div class="col-md-6 col-md-offset-4">
54. <button type="submit" class="btn btn-primary">
55. <i class="fa fa-btn fa-user"></i>Tambah
56. </button></div></div></form><div></div></div></div></div>
57. @endsection

Script 4.13 *view* untuk tampilan tambah pengguna

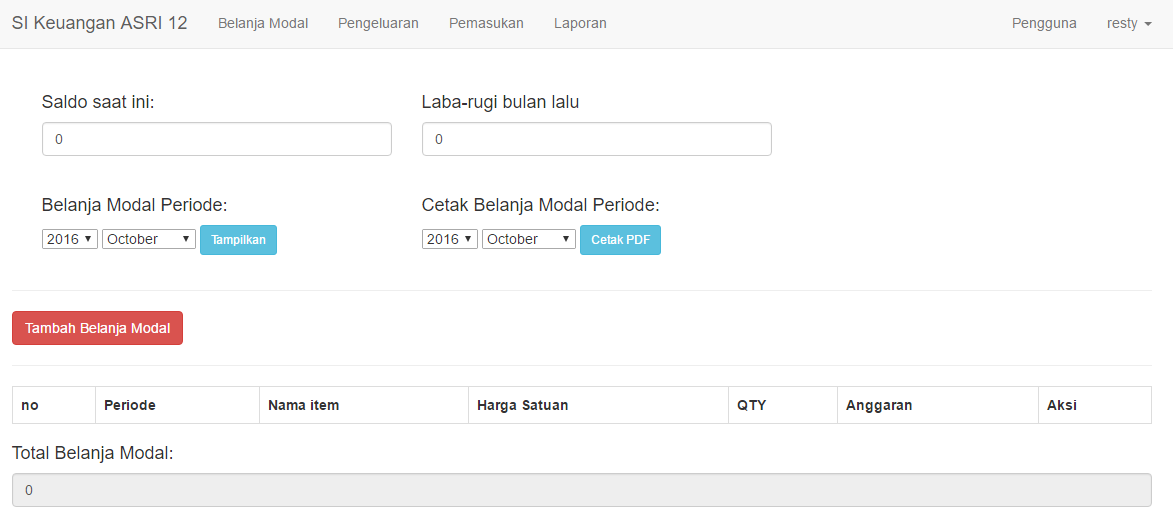
Script 4.13 berisi code yang berfungsi untuk menampilkan halaman untuk mendaftarkan pengguna.

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 user method create()* dan *store()*

Script 4.13 berisi controller create yang berfungsi menampilkan, mengambil data dari form , serta menyimpannya di database.

1. Belanja Modal
2. Tampil Belanja Modal



Gambar 4.45 tampilan utama belanja modal

Pada gambar 4.45 yaitu halaman belanja modal menampilkan informasi daftar belanja modal.

1. <?php
2. namespace App;
3. use Illuminate\Database\Eloquent\Model;
4. use Carbon\Carbon;
5. use DateTime;
6. class anggaran extends Model  {    protected $fillable = ['nama\_item\_anggaran','qty','harga\_satuan','anggaran','periode'];
7. public function getCreatedAtAttribute()
8. {
9. return Carbon::parse($this->attributes['periode']);
10. }
11. }

Script 4.15 model belanja modal

Script 4.15 berfungsi menyambungkan aplikasi dengan database anggaran yang berisi data belanja modal. Serta mengizinkan aplikasi untuk menyimpan data ke database.

1. @extends('layouts.app')
2. @section('content')
3. <div class="container">
4. <div class="container">
5. <div class="col-sm-4">
6. <h4>Saldo saat ini:</h4> <!-- untuk pilih periode  -->
7. <tr><td>{!! Form::text('saldo', number\_format($saldo->saldo) ,['class'=>'form-control']) !!}</td></tr>
8. </div>
9. <div class="col-sm-4">
10. <h4>Laba-rugi bulan lalu</h4> <!-- untuk pilih periode  -->
11. <tr><td>{!! Form::text('labarugi', number\_format($labarugi) ,['class'=>'form-control']) !!}</td></tr>
12. </div>
13. </div>
14. <div>
15. <div class="container">
16. <div class= "col-sm-4" >
17. <h4>Belanja Modal Periode: </h4> <!-- untuk pilih periode  -->
18. {!! Form::open(array('url'=>'anggaran/tahunBulan')) !!}
19. {!! Form::selectRange('year',2015, date('Y'), $y ,['class' => 'field']) !!}
20. {!! Form::selectMonth('month', $m ,['class' => 'field'] ) !!}
21. {!! form::submit('Tampilkan',['class'=>'btn btn-info btn-sm']) !!}
22. {!! form::close() !!}
23. </div>
24. <div class= "col-sm-4" >
25. <h4>Cetak Belanja Modal Periode: </h4> <!-- untuk pilih periode  -->
26. {!! Form::open(array('url'=>'anggaran/pdf')) !!}
27. {!! Form::selectRange('year',2015, date('Y'), $y ,['class' => 'field']) !!}
28. {!! Form::selectMonth('month', $m ,['class' => 'field'] ) !!}
29. {!! form::submit('Cetak PDF',['class'=>'btn btn-info btn-sm']) !!}
30. {!! form::close() !!}
31. </div>
32. </div>
33. </div>
34. <div>
35. <hr>
36. <div class="btn-group">
37. {!! link\_to('anggaran/create','Tambah Belanja Modal',['class'=>'btn btn-danger btn-md btn-space']) !!}
38. </div>
39. <hr>
40. <!-- <h4>pencarian</h4>
41. {!! Form::open(array('url'=>'anggaran/search')) !!}
42. {!! form::text('keyword','',['class'=>'form-control','placeholder'=>'masukan nama item'])!!}
43. <br>
44. {!! form::submit('Cari Item',['class'=>'btn btn-danger btn-sm']) !!}
45. {!! form::close() !!} -->
46. </div>
47. <table class="table table-bordered">
48. <tr><th>no</th><th>Periode</th><th>Nama item</th><th>Harga Satuan</th><th>QTY</th><th>Anggaran</th><th colspan="2">Aksi</th></tr>
49. <?php $no = 1; ?>
50. @foreach ( $anggaran as $n)
51. <tr>
52. <td width="50px" align="center">{{ $no++ }}</td>
53. <td width="140px">{{ $n->periode}}</td> <!-- liat model anggaran untuk date formatnya -->
54. <td width="500px">{{ $n->nama\_item\_anggaran }}</td>
55. <td width="140px">{{ number\_format($n->harga\_satuan) }}</td>
56. <td width="50px">{{ $n->qty }}</td>
57. <td>{{ number\_format($n->anggaran) }}</td>
58. <td width="50px">{!! link\_to('anggaran/'.$n->id.'/edit','Edit',['class'=>'btn btn-warning btn-sm']) !!}
59. <td>
60. {!! Form::open(array('method'=>'delete','url'=>'anggaran/'.$n->id))!!}
61. {!! Form::hidden('\_delete','DELETE')!!}
62. {!! Form::submit('Hapus',['class'=>'btn btn-danger btn-sm','id'=>'deletebtn', 'onclick' => 'return confirm("Apakah anda yakin ingin menghapus ?")'])!!}
63. {!! Form::close()!!}
64. </td>
65. </td> </tr>
67. @endforeach
68. </table>
69. {!! $anggaran->render() !!}
71. <h4>Total Belanja Modal:</h4>
72. {!! form::text('totalmod',number\_format($totalmod),['class'=>'form-control','placeholder'=>'','readonly'])!!}
73. <br>
74. </div>
75. @stop

Script 4.16 view dari belanja modal

Script 4.16 berisi listing untuk menampilkan halaman utama belanja modal.

1. public function index()
2. {
3. $anggaran = anggaran::whereYear('periode','=',date('Y'))
4. ->whereMonth('periode', '=', date('m'))
5. ->paginate(20);
6. $data['anggaran']   = $anggaran;
7. $data['y']          = date('Y');
8. $data['m']          = date('m');
9. $data['totalmod']   = anggaran::whereYear('periode','=',date('Y'))
10. ->whereMonth('periode', '=', date('m'))
11. ->sum('anggaran');
12. $data['saldo']      = transaksi::latest()->first();
13. $pema               = labarugi::whereYear('created\_at', '=', date('Y'))   ->whereMonth('created\_at', '=', date('m')-1)
14. ->sum('pemasukan');
15. $penge              = labarugi::whereYear('created\_at', '=', date('Y'))   ->whereMonth('created\_at', '=', date('m')-1)
16. ->sum('pengeluaran');
17. $data['labarugi']   = $pema - $penge;
18. return view('anggaran.index',$data);
19. }

Script 4.17 controller index() untuk belanja modal

Script 4.17 berisi code yang berfungsi menampilkan data belanja modal ke view. Dan beberapa form yang diluar belanja modal juga di berikan di method index() dan mengembalikan beberapa nilai ke view index.

1. Route::get('anggaran',function() {
2. return view('anggaran.index');  });

script 4.18 routes belanja modal

1. Tambah Belanja Modal
2. @extends('layouts.app')
3. @section('content')

6. <div class="container">
7. {!! Html::ul($errors->all()) !!}
9. {!! Form::open(array('url'=>'anggaran')) !!}
10. <table class="table table-bordered">
12. @include('anggaran.form') <!-- buat manggil form di view/anggaran/form.blade.php, jadi sekali panggil -->
13. <tr><td colspan="2">
14. {!! Form::submit('simpan data',['class'=>'btn btn-danger btn-sm']) !!}
15. {!! link\_to('anggaran','Kembali',['class'=>'btn btn-danger btn-sm']) !!}
16. </td></tr>
17. </table>
18. {!! Form::close() !!}
19. </div>
21. @stop

**Script 4.19 view dari create.blade.php**

**Script 4.19 berisi code untuk menampilkan halaman form menambahkan data belanja modal.**

1. public function create()
2. {
3. return view('anggaran.create');
4. }
5. public function store(createAnggaran $request)
6. {
7. $tgl  = $request['periode'];
8. $item = $request['nama\_item\_anggaran'];
9. $sat  = $request['harga\_satuan'];
10. $qty  = $request['qty'];
11. $anggaran = $sat \* $qty;
12. Anggaran::create(array('nama\_item\_anggaran'=> $item,'periode'=> $tgl,'harga\_satuan'=> $sat,'qty'=>$qty,'anggaran'=>$anggaran));
13. return redirect('anggaran');
14. }

Script 4.20 controller tambah belanja modal

Script 4.20 berisi method create() dan store(). Yang mana method create() berfungsi untuk menampilkan halaman form dan method store() untuk mengambil nilai dari form dan menyimpannya di database.

1. Edit belanja modal

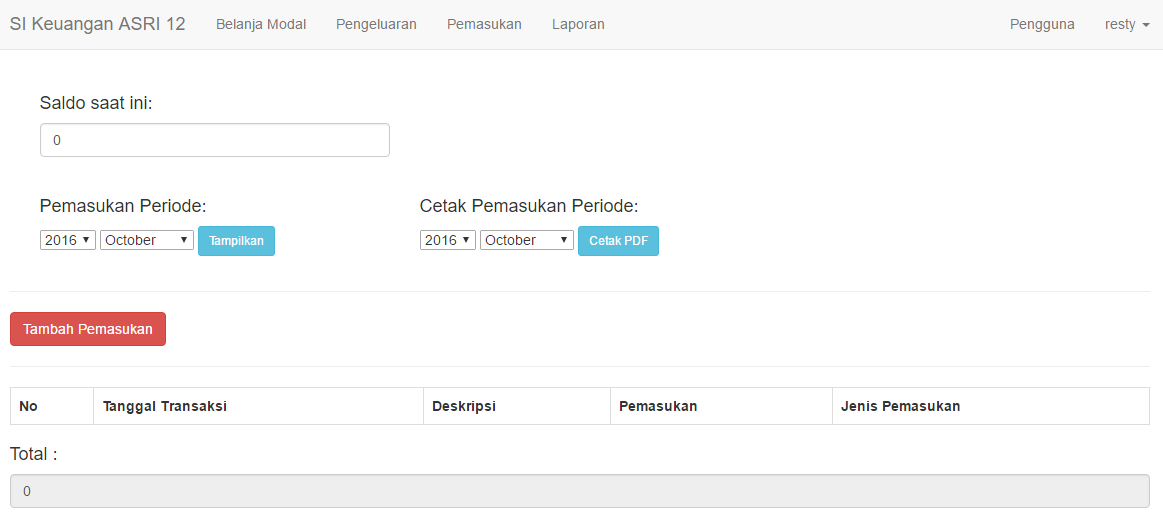
Script untuk view edit belanja modal sama persis dengan view tambah belanja modal karena me*request* halaman yang sama.

1. public function update($id, createAnggaran $request)
2. {  $data = $request->all();
3. $anggaran = anggaran::find($id);   $anggaran->update($data);
4. return redirect('anggaran');
5. }  public function edit($id)
6. {
7. $data['anggaran'] = anggaran::find($id);
8. return view('anggaran.edit',$data);
9. }

Script 4.21 method update dan store

Script 4.21 berisi method edit yang berfungsi untuk menampilkan halaman edit. Dan method update berfungsi untuk menyimpan data yang telah di edit.

1. Pemasukan
2. Tampil pemasukan



Gambar 4.46 tampilan halaman menu pemasukan

Halaman menu pemasukan pada gambar 4.46 berisi informasi tentang pemasukan. Terdapat beberapa form antara lain form saldo saat ini, menu drop down pilihan periode, tombol cetak pemasukan, tombol tambah pemasukan serta menampilkan tabel data pemasukan.

1. <?php
3. namespace App;
5. use Illuminate\Database\Eloquent\Model;
6. use App\transaksi;
7. use App\labarugi;
9. class pemasukan extends Model  {
10. protected $fillable = [ 'tanggal\_transaksi',
11. 'deskripsi',
12. 'pemasukan',
13. 'jenis\_pema',
14. 'id'];
15. public function transaksi()      {
16. return $this->hasOne('App\transaksi');
17. }
18. public function labarugi()
19. {
20. return $this->hasOne('App\labarugi');
21. }
23. }

Script 4.22 model pemasukan.

Script 4.22 model dari pemasukan yang berfungsi mengizinkan aplikasi dalam mengakses database. Dan mengizinkan aplikasi mengisi data di tabel pemasukan. Dan mendeklarasikan fungsi yang menyatakan relasi dari tabel pemasukan.

1. @extends('layouts.app')
2. @section('content')
3. <div class="container">
4. <div>
5. <div class="container">
6. <div class="col-sm-4">
7. <h4>Saldo saat ini:</h4> <!-- untuk pilih periode  -->
8. <tr><td>{!! Form::text('saldo', number\_format($saldo->saldo) ,['class'=>'form-control']) !!}</td></tr>
9. </div>
10. </div>
11. <div>
12. <div class="container">
13. <div class= "col-sm-4" >
14. <h4>Pemasukan Periode: </h4> <!-- untuk pilih periode  -->
15. {!! Form::open(array('url'=>'pemasukan/tahunBulan')) !!}                  {!! Form::selectRange('year',2015, date('Y'), $y ,['clas s' => 'field']) !!}
16. {!! Form::selectMonth('month', $m ,['class' => 'field'] ) !!}
17. {!! form::submit('Tampilkan',['class'=>'btn btn-info btn-sm']) !!}
18. {!! form::close() !!}
19. </div>
20. <div class= "col-sm-4" >
21. <h4>Cetak Pemasukan Periode: </h4> <!-- untuk pilih periode  -->
22. {!! Form::open(array('url'=>'pemasukan/pdf')) !!}
23. {!! Form::selectRange('year',2015, date('Y'), $y ,['class' => 'field']) !!}
24. {!! Form::selectMonth('month', $m ,['class' => 'field'] ) !!}
25. {!! form::submit('Cetak PDF',['class'=>'btn btn-info btn-sm']) !!}
26. {!! form::close() !!}
27. </div>
28. </div>
29. </div>
30. </div>
31. <div>
32. <hr>
33. {!! link\_to('pemasukan/create','Tambah Pemasukan',['class'=>'btn btn-danger btn-md']) !!}
34. <hr>
35. </div>
36. <table class="table table-bordered">
37. <tr><th>No</th><th>Tanggal Transaksi</th><th>Deskripsi</th><th>Pemasukan</th><th>Jenis Pemasukan</th><!-- <th colspan="2">Aksi</th> --></tr>
38. <?php $no = 1; ?>
39. @foreach ( $pemasukan as $n)
40. <tr>
41. <td width="50px" align="center">{{ $no++ }}</td>
42. <td width="140px">{{ $n->created\_at->format('d-m-Y') }}</td>
43. <td width="500px">{{ $n->deskripsi }}</td>
44. <td width="140px">{{ number\_format($n->pemasukan) }}</td>
45. <td width="150px">{{ $n->jenis\_pema }}</td>
46. </td> </tr>
47. @endforeach
48. </table>
49. {!! $pemasukan->render() !!}
50. <h4>Total :</h4>
51. {!! form::text('total',number\_format($total),['class'=>'form-control','placeholder'=>'','readonly'])!!}
52. <br>
53. </div>
54. @stop

Script 4.23 view dari pemasukan

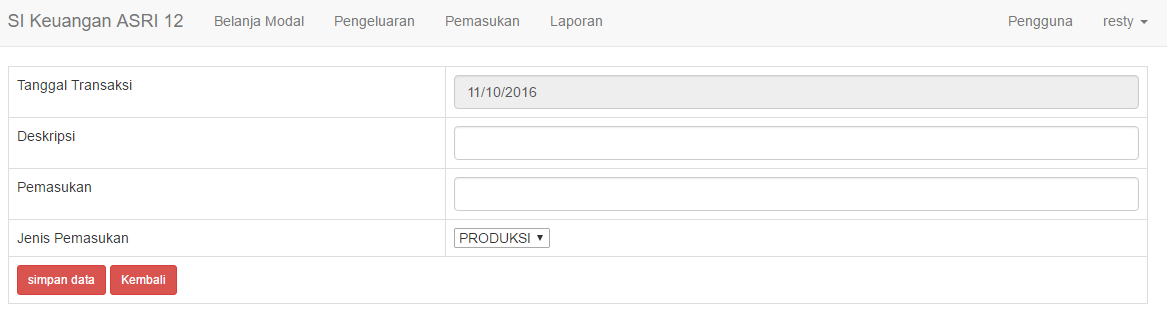
Script 2.23 berfungsi menampilkan halaman menu pemasukan.

1. public function index()
2. {
3. $pemasukan              = pemasukan::whereYear('created\_at', '=', date('Y'))
4. ->whereMonth('created\_at', '=',date('m'))
5. ->paginate(40);
6. $data['total'] = pemasukan::whereYear('created\_at','=',date('Y'))
7. ->whereMonth('created\_at', '=', date('m'))
8. ->sum('pemasukan');
9. $data['pemasukan']      = $pemasukan;
10. $data['y']              = date('Y');
11. $data['m']              = date('m');
12. $data['saldo']          = transaksi::latest()->first();
13. return view('pemasukan.index',$data);
14. }

Script 4.24 controller pemasukan

Script 4.24 berisi function index yang mengembalikan nilai hasil kueri database dan menampilkannya ke aplikasi.

1. Tambah pemasukan



Gambar 4.47 tampilan halaman tambah pemasukan

Gambar 4.47 yaitu halaman tambah pemasukan menampilkan form yang diperlukan untuk menambah data pemasukan .

1. @extends('layouts.app')
2. @section('content')
3. <div class="container">
4. {!! Html::ul($errors->all()) !!}
5. {!! Form::open(array('url'=>'pemasukan')) !!}
6. <table class="table table-bordered">
7. @include('pemasukan.form') <!-- buat manggil form di view/pemasukan/form.blade.php, jadi sekali panggil -->
8. <tr><td colspan="2">
9. {!! Form::submit('simpan data',['class'=>'btn btn-danger btn-sm']) !!}
10. {!! link\_to('pemasukan','Kembali',['class'=>'btn btn-danger btn-sm']) !!}
11. </td></tr>
12. </table>
13. {!! Form::close() !!}
14. </div>
15. @stop

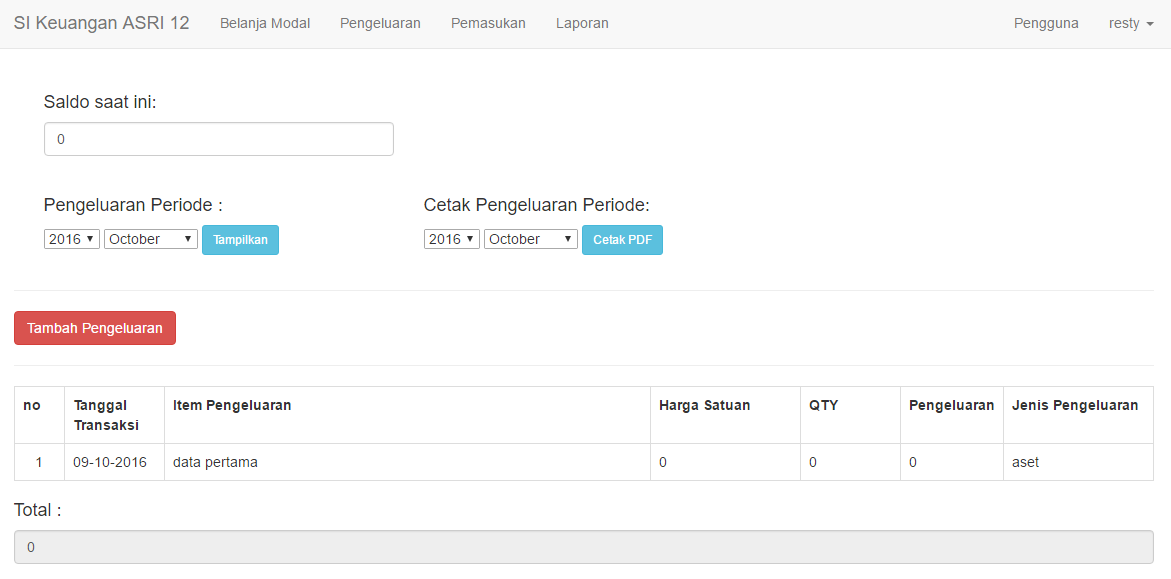
Script 4.25 view tambah pemasukan

1. public function create()
2. {
3. return view('pemasukan.create');
4. }
5. public function store(createPemasukan $request)
6. {   $pem  = $request['pemasukan'];
7. $des  = $request['deskripsi'];
8. $tgl  = $request['tanggal\_transaksi'];
9. $jen  = $request['jenis\_pema'];
10. $pemasukan = pemasukan::create(array('deskripsi'=> $des,
11. 'pemasukan'=>$pem,
12. 'jenis\_pema'=>$jen));
13. if($jen == 'produksi') // jika pemasukan berjenis produksi maka di masukan ke tabel labarugi
14. {
15. $labarugi = labarugi::latest()->first();
17. if($labarugi)
18. {
19. $labarugi = $labarugi->labarugi + $pem;
20. }
21. else
22. {
23. $labarugi = 0 + $pem;
24. }
25. $pemasukan->labarugi()->create(array('deskripsi'=> $des,'pemasukan'=> $pem,'labarugi'=>$labarugi));
26. }
27. else
28. {
29. // jika aset, maka lanjut
30. }
31. $saldo = transaksi::latest()->first(); //untuk mengambil data terakhir pada tabel transaksi kolom saldo
33. if ($saldo)
34. {
35. $saldo = $saldo->saldo + $pem;
36. }
37. else
38. {
39. $saldo = 0 + $pem;
40. }
41. $pemasukan->transaksi()->create(array('deskripsi'=> $des, 'pemasukan'=> $pem,'saldo'=>$saldo));
42. //masukan data ke tabel transaksi
43. return redirect('pemasukan');
44. }

Script 2.26 controller create dan store

Controller dengan function create berfungsi untuk menampilkan halaman from tambah pengguna, sedangkan function store untuk menyimpan data yang telah dimasukan.

1. Pengeluaran



Gambar 4.48 tampilan halaman pengeluaran

Halaman menu pengeluaran pada gambar 4.48 berisi informasi tentang transaksi pengeluaran. Terdapat beberapa form antara lain form saldo saat ini, menu drop down pilihan periode, tombol cetak pemasukan, tombol tambah pengeluaran serta menampilkan tabel data pengeluaran.

1. Tampil pengeluaran
2. class pengeluaran extends Model
3. { protected $fillable =
4. [ 'tanggal\_transaksi',
5. 'item\_pengeluaran',
6. 'harga\_satuan\_peng',
7. 'qty\_peng',
8. 'pengeluaran',
9. 'jenis\_peng',
10. 'id'];
11. public function transaksi()
12. {
13. return $this->hasOne('App\transaksi');   }
14. public function labarugi()
15. {
16. return $this->hasOne('App\labarugi');   } }

Script 2.27. model pengeluaran

Script 4.27 model dari pengeluaran yang berfungsi mengizinkan aplikasi dalam mengakses database. Dan mengizinkan aplikasi mengisi data di tabel pengeluaran. Dan mendeklarasikan fungsi yang menyatakan relasi dari tabel pengeluaran.

1. <div class="col-sm-4">
2. <h4>Saldo saat ini:</h4>
3. <tr><td>{!! Form::text('saldo', number\_format($saldo->saldo) ,['class'=>'form-control']) !!}</td></tr>
4. </div>
5. <div class= "col-sm-4" >
6. <h4>Pengeluaran Periode : </h4>
7. {!! Form::open(array('url'=>'pengeluaran/tahunBulan')) !!}
8. {!! Form::selectRange('year',2015, date('Y'), $y ,['class' => 'field']) !!}
9. {!! Form::selectMonth('month', $m ,['class' => 'field'] ) !!}
10. {!! form::submit('Tampilkan',['class'=>'btn btn-info btn-sm']) !!}
11. {!! form::close() !!}
12. </div>
13. <div class= "col-sm-4" >
14. <h4>Cetak Pengeluaran Periode: </h4> <!-- untuk pilih periode  -->
15. {!! Form::open(array('url'=>'pengeluaran/pdf')) !!}
16. {!! Form::selectRange('year',2015, date('Y'), $y ,['class' => 'field']) !!}
17. {!! Form::selectMonth('month', $m ,['class' => 'field'] ) !!}
18. {!! form::submit('Cetak PDF',['class'=>'btn btn-info btn-sm']) !!}
19. {!! form::close() !!}
20. </div><hr>
21. {!! link\_to('pengeluaran/create','Tambah Pengeluaran',['class'=>'btn btn-danger btn-md']) !!}
22. <hr>
23. <table class="table table-bordered">
24. <tr>
25. <th>no</th>
26. <th>Tanggal Transaksi</th><th>Item Pengeluaran</th><th>Harga Satuan</th>  <th>QTY</th><th>Pengeluaran</th><th>Jenis Pengeluaran</th>
27. <?php $no = 1; ?>
28. @foreach ( $pengeluaran as $n)
29. <tr>
30. <td width="50px" align="center">{{ $no++ }} </td>
31. <td width="100px">{{ $n->created\_at->format('d-m-Y')}}</td>
32. <td>{{ $n->item\_pengeluaran }}</td>
33. <td width="150px">{{ number\_format($n->harga\_satuan\_peng) }}</td>
34. <td width="100px">{{ $n->qty\_peng}}</td>
35. <td width="100px">{{ number\_format($n->pengeluaran) }}</td>
36. <td width="150px">{{ $n->jenis\_peng }}</td>
37. </td> </tr>
38. @endforeach
39. </table>
40. {!! $pengeluaran->render() !!}
41. <h4>Total :</h4>
42. {!! form::text('total',number\_format($total),['class'=>'form-control','placeholder'=>'','readonly'])!!}
43. <br>
44. </div>

Script 4.28 view pengeluaran

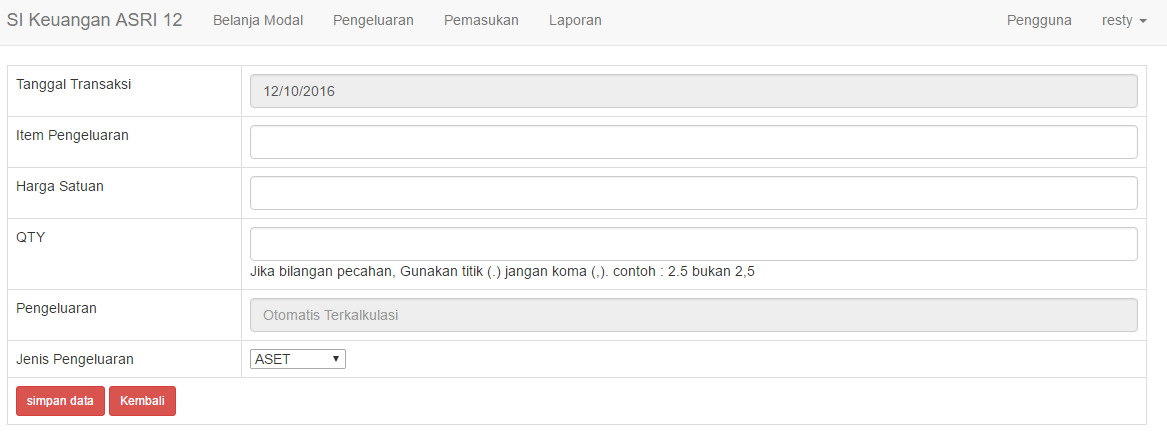
Script 4.28 berisi listing untuk menampilkan menu pengeluaran

1. public function index()
2. {
3. $pengeluaran            = pengeluaran::whereYear('created\_at', '=',date('Y'))
4. ->whereMonth('created\_at', '=', date('m'))
5. ->paginate(50);
6. $data['total']          = pengeluaran::whereYear('created\_at','=',date('Y'))
7. ->whereMonth('created\_at', '=', date('m'))
8. ->sum('pengeluaran');
9. $data['pengeluaran']    = $pengeluaran;
10. $data['y']              = date('Y');
11. $data['m']              = date('m');
12. $data['saldo']          = transaksi::latest()->first();
13. return view('pengeluaran.index',$data);
14. }

Script 4.29 controller pengeluaran

Controller untuk menampilkan halaman utama pengeluaran adalah dengan memanggil fungsi index().

1. Tambah pengeluaran



Gambar 4.49 tampilan form tambah pengeluaran

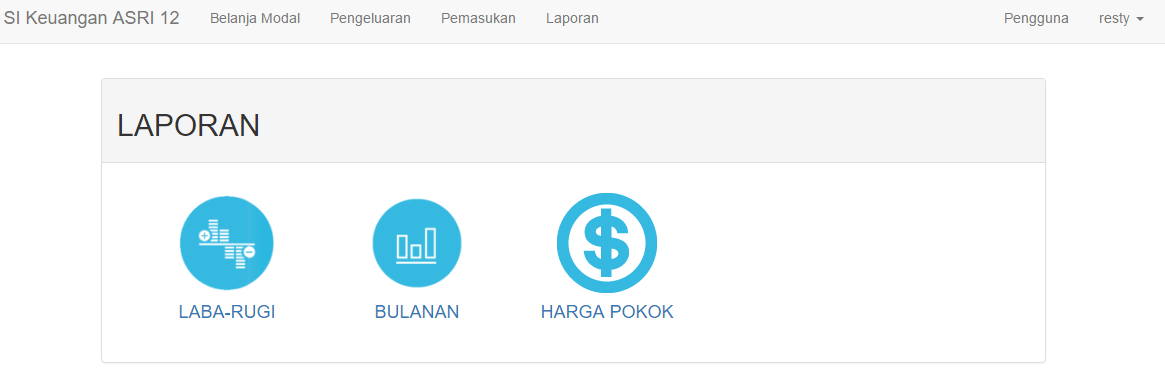
Form tambah pengeluaran pada gambar 4.49 tampil saat pengguna akan menambahkan data pengeluaran.

1. public function create()
2. {
3. return view('pengeluaran.create');
4. }
5. public function store(createPengeluaran $request)
6. {
7. $sat  = $request['harga\_satuan\_peng'];
8. $qty  = $request['qty\_peng'];
9. $peng = $sat \* $qty;
10. $item = $request['item\_pengeluaran'];
11. $tgl  = $request['tanggal\_transaksi'];
12. $jen  = $request['jenis\_peng'];
14. $pengeluaran = pengeluaran::create(array('item\_pengeluaran'=> $item,
15. 'harga\_satuan\_peng'=> $sat,
16. 'qty\_peng'=>$qty,
17. 'pengeluaran'=>$peng,
18. 'jenis\_peng'=>$jen));
19. if($jen == 'produksi') // jika pengeluaran berjenis produksi maka di masukan ke tabel labarugi
20. {
21. $labarugi = labarugi::latest()->first();
22. if($labarugi)
23. {
24. $labarugi = $labarugi->labarugi - $peng;
25. }
26. else
27. {
28. $labarugi = 0 - $peng;
29. }
30. $pengeluaran->labarugi()-> create (array ( 'pengeluaran'=> $peng,
31. 'deskripsi'=>$item,
32. 'labarugi'=>$labarugi));
33. }
34. else
35. {
36. // jika aset, maka lanjut
37. }
39. $saldo = transaksi::latest()->first();
40. if ($saldo)
41. {
42. $saldo = $saldo->saldo - $peng;
43. }
44. else
45. {
46. $saldo = 0 - $peng;
47. }
48. $pengeluaran->transaksi()->create(array('deskripsi'=> $item, 'pengeluaran'=> $peng,'saldo'=>$saldo));
49. return redirect('pengeluaran');
50. }

script 4.30 controller tambah pengeluaran

Fungsi create() di gunakan untuk menampilkan halaman from tambah pengeluaran. Sedangkan fungsi store() digunakan untuk menyimpan data pengeluaran yang telah di input.

1. Laporan



Gambar 4.50 tampilah halaman utama laporan

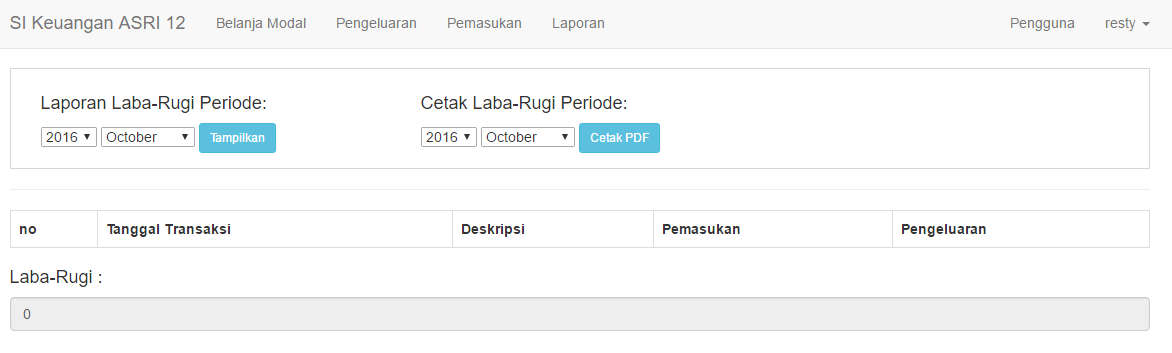
Menu utama laporan pada gambar 4.50 terdapat 3 menu yaitu laba-rugi, bulanan, dan harga pokok. Pengguna dapat memilih salah satu dari menu yang disediakan.

1. <div class="panel-heading"><h2>LAPORAN</h2></div>
2. <div class="panel-body">
3. <div class='container'>
4. <div class="col-sm-2 " align="center">
5. <a href="{{ url('laporan/labarugi') }}"><img src="img/labarugi.png"  width="100" height="100" class="img-responsive" alt="Generic placeholder thumbnail">
6. <h4 >LABA-RUGI</h4>
7. </div>
8. <div class="col-sm-2 " align="center">
9. <a href="{{ url('laporan/bulanan')}}"><img src="img/bulanan.png" width="100" height="100" class="img-responsive" alt="Generic placeholder thumbnail">
10. <h4>BULANAN</h4>
11. </div>
12. <div class="col-sm-2 " align="center">
13. <a href="{{ url('laporan/harga')}}"><img src="img/cost.png" width="100" height="100" class="img-responsive" alt="Generic placeholder thumbnail">
14. <h4>HARGA POKOK</h4></div>
15. </div>

Script 4.31 view laporan

Script 2.31 merisi listing untuk menampilakan menu utama laporan. Yang masing masing laporan diwakili dengan ikon yang terkait.

1. Laporan laba rugi



Gambar 4.51 tampilan halaman laporan laba-rugi

Menu laporan laba-rugi berisi pada gambar 4.51 laporan pengeluaran dan pemasukan yang terkait produksi.

1. class labarugi extends Model
2. {
3. protected $fillable = ['tanggal\_transaksi','deskripsi','pengeluaran\_id','pemasukan\_id','pemasukan','pengeluaran','labarugi'];
4. public function pemasukan()
5. {
6. return $this->belongsTo('App\pemasukan');
7. }
8. public function pengeluaran()
9. {
10. return $this->belongsTo('App\pengeluaran');
11. }
12. }

Script 4.32 model dari laporan laba-rugi

Script 4.32 berisi listing untuk mengizinkan aplikasi dalam mengakses database. Dan mendeklarasikan sebuah relasi terkait dengan tabel laba-rugi.

1. <div class="container">
2. <div style="border:1px solid #CFD7DF;">
3. <div class="container">
4. <div class= "col-sm-4" >
5. <h4>Laporan Laba-Rugi Periode: </h4> <!-- untuk pilih periode  -->
6. {!! Form::open(array('url'=>'laporan/labarugi/labarugiBulanan')) !!}
7. {!! Form::selectRange('year',2015, date('Y'), $y ,['class' => 'field']) !!}
8. {!! Form::selectMonth('month', $m ,['class' => 'field'] ) !!}
9. {!! form::submit('Tampilkan',['class'=>'btn btn-info btn-sm']) !!}
10. {!! form::close() !!}
11. </div>
12. <div class= "col-sm-5" >
13. <h4>Cetak Laba-Rugi Periode: </h4> <!-- untuk pilih periode  -->
14. {!! Form::open(array('url'=>'laporan/labarugi/labarugipdf')) !!}
15. {!! Form::selectRange('year',2015, date('Y'), $y ,['class' => 'field']) !!}
16. {!! Form::selectMonth('month', $m ,['class' => 'field'] ) !!}
17. {!! form::submit('Cetak PDF',['class'=>'btn btn-info btn-sm']) !!}
18. {!! form::close() !!}
19. </div>
20. </div>
21. </div>
22. <hr>
24. <table class="table table-bordered">
25. <tr>
26. <th>no</th>
27. <th>Tanggal Transaksi</th>
28. <th>Deskripsi</th>
29. <th>Pemasukan</th>
30. <th>Pengeluaran</th>
31. </tr>
32. <?php $no = 1; ?>
33. @foreach ( $laba\_rugi as $n)
34. <tr>
35. <td width="50px" align="center">{{ $no++ }} </td>
36. <td>{{ $n->created\_at->format('d-m-Y')}}</td>
37. <td>{{ $n->deskripsi }}</td>
38. <td>{{ number\_format($n->pemasukan) }}</td>
39. <td>{{ number\_format($n->pengeluaran) }}</td>
40. </tr>
41. @endforeach
42. </table>
43. <h4>Laba-Rugi :</h4>
44. {!! form::text('labarugi',number\_format($labarugi),['class'=>'form-control','placeholder'=>'','readonly'])!!}  <br>

Script 4.33 view dari laporan laba-rugi

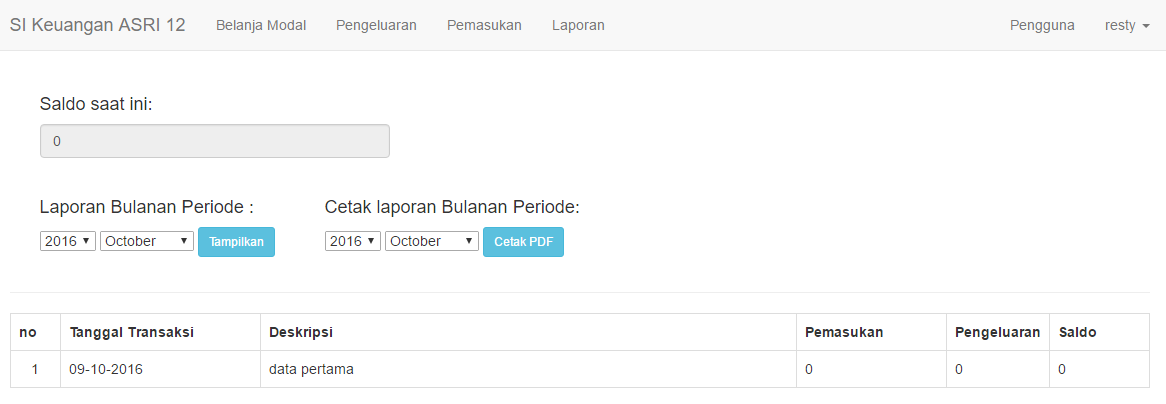
Script 4.33 berisi listing untuk menampilkan menu laporan laba-rugi yang menampilkan data laba-rugi usaha dari database.

1. public function labarugi()
2. {
3. $data['y']              = date('Y');
4. $data['m']              = date('m');
5. $labarugi               = labarugi::whereYear('created\_at', '=', date('Y'))
6. ->whereMonth('created\_at', '=', date('m'))
7. ->paginate(40);
8. $pema                   = labarugi::whereYear('created\_at', '=', date('Y'))
9. ->whereMonth('created\_at', '=', date('m'))
10. ->sum('pemasukan');
11. $penge                  = labarugi::whereYear('created\_at', '=', date('Y'))
12. ->whereMonth('created\_at', '=', date('m'))
13. ->sum('pengeluaran'
14. $data['labarugi']       = $pema - $penge;
15. $data['laba\_rugi']      = $labarugi;
16. return view('laporan.labarugi',$data);
17. }

Script 4.34 contoller laba-rugi

Script 4.34 berisi listing perintah kueri database labarugi untuk dikirim ke view labarugi.

1. Laporan keuangan bulanan



Gambar 4.52 tampilan halaman laporan keuangan bulanan

Tampilan halaman utama menu laporan keuangan bulanan yang bisa dilihat pada gambar 4.52 diakses dari menu utama laporan. Berisi informasi terkait laporan keuangan bulanan .

1. <div class="container">
2. <div class="col-sm-4">
3. <h4>Saldo saat ini:</h4> <!-- untuk pilih periode  -->
4. <tr><td>{!! Form::text('saldo', number\_format($saldo->saldo) ,['class'=>'form-control','readonly']) !!}</td></tr>
5. </div>
6. </div>
7. <div>
8. <div class="container">
9. <div class= "col-sm-3" >
10. <h4>Laporan Bulanan Periode : </h4> <!-- untuk pilih periode  -->
11. {!! Form::open(array('url'=>'laporan/bulanan/tahunBulan')) !!}
12. {!! Form::selectRange('year',2015, date('Y'), $y ,['class' => 'field']) !!}
13. {!! Form::selectMonth('month', $m ,['class' => 'field'] ) !!}
14. {!! form::submit('Tampilkan',['class'=>'btn btn-info btn-sm']) !!}
15. {!! form::close() !!}
16. </div>
17. <div class= "col-sm-5" >
18. <h4>Cetak laporan Bulanan Periode: </h4> <!-- untuk pilih periode  -->
19. {!! Form::open(array('url'=>'laporan/bulanan/bulananpdf')) !!}
20. {!! Form::selectRange('year',2015, date('Y'), $y ,['class' => 'field']) !!}
21. {!! Form::selectMonth('month', $m ,['class' => 'field'] ) !!}
22. {!! form::submit('Cetak PDF',['class'=>'btn btn-info btn-sm']) !!}
23. {!! form::close() !!}
24. </div>
25. </div>
26. </div>
27. <hr>
28. <table class="table table-bordered">
29. <tr>
30. <th>no</th>
31. <th>Tanggal Transaksi</th>
32. <th>Deskripsi</th>
33. <th>Pemasukan</th>
34. <th>Pengeluaran</th>
35. <th>Saldo</th>
36. </tr>
37. <?php $no = 1; ?>
38. @foreach ( $transaksi as $n)
39. <tr>
40. <td width="50px" align="center">{{ $no++ }} </td>
41. <td width="200px">{{ $n->created\_at->format('d-m-Y') }}</td>
42. <td>{{ $n->deskripsi }}</td>
43. <td width="150px">{{ number\_format($n->pemasukan) }}</td>
44. <td width="100px">{{ number\_format($n->pengeluaran) }}</td>
45. <td width="100px">{{ number\_format($n->saldo) }}</td>
46. </tr>
47. @endforeach
48. </table>
49. {!! $transaksi->render() !!}
50. </div>

Script 4.35 view laporan keuangan bulanan

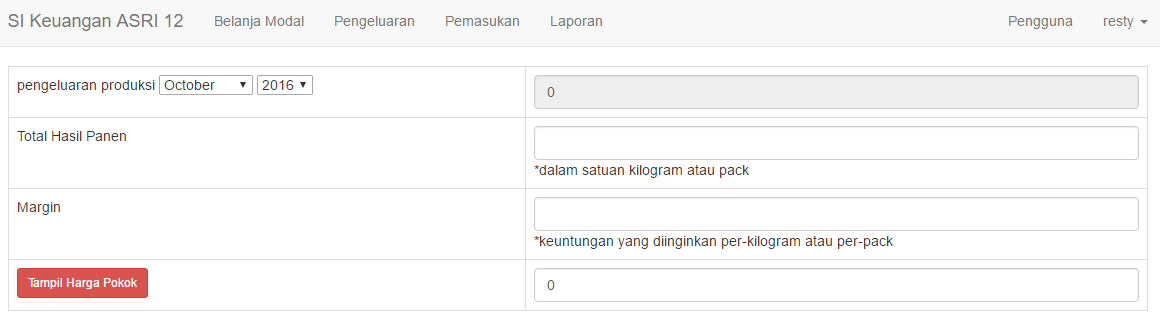
Script 4.35 adalah listing program untuk menampilkan data laporan kauangan bulanan yang diterima dari controller.

1. public function bulanan()
2. {
3. $data['y']              = date('Y');
4. $data['m']              = date('m');
5. $bulanan                = transaksi::whereYear('created\_at', '=’,date('Y'))
6. ->whereMonth('created\_at', '=', date('m'))
7. ->paginate(40);
8. $data['saldo']          = transaksi::latest()->first();
9. $data['transaksi']      = $bulanan;
10. return view('laporan.bulanan',$data);
11. }

Script 4.36 controller laporan keuangan bulanan

Script 4.36 berisi listing perintah kueri database tabel transaksi untuk dikirim ke view laporan bulanan.

1. Harga pokok



Gambar 4.53 tampilan halaman harga pokok

Menu laporan harga pokok pada gambar 4.53 menampilkan form kalkulasi yang digunakan untuk menghitung harga pokok penjualan produk bedasarkan hasil perhitungan pengeluaran produksi dibagi jumlah hasil panen dan di tambah margin atau keuntungan yang ingin didapatkan.

1. <div class="container">
2. {!! Html::ul($errors->all()) !!}
3. {!! Form::open(array('url'=>'laporan/harga/hasilHarga')) !!}  <table class="table table-bordered">
4. <tr><td>pengeluaran produksi  {!! Form::selectMonth('month', $m ,['class' => 'field'] ) !!}
5. {!! Form::selectRange('year',2015, date('Y'), $y ,['class' => 'field']) !!}
6. </td>
7. <td>{!! Form::text('pengeluaran\_prod',number\_format($pengeluaran\_prod),['class'=>'form-control','readonly']) !!}</td></tr>
8. <tr><td>Total Hasil Panen</td><td>{!! Form::text('hasil\_panen',$hasil\_panen,['class'=>'form-control']) !!} \*dalam satuan kilogram atau pack </td></tr<tr><td>Margin</td><td>{!! Form::text('margin',$margin,['class'=>'form-control']) !!}  \*keuntungan yang diinginkan per-kilogram atau per-pack</td></tr>
9. <tr>
10. <td>{!! Form::submit('Tampil Harga Pokok',['class'=>'btn btn-danger btn-sm']) !!}</td><td>{!! Form::text('hasil',number\_format($hasil),['class'=>'form-control']) !!}</td>

Script 4.37 view laporan harga pokok

Script 4.37 berisi listing program untuk menampilkan halaman menu harga pokok.

1. public function harga(request $request)
2. {
3. $data['y']                  = date('Y');
4. $data['m']                  = date('m');
5. $pengeluaran\_prod           = labarugi::whereYear('created\_at', '=', date('Y'))
6. ->whereMonth('created\_at', '=', date('m'))
7. ->sum('pengeluaran');
8. $data['pengeluaran\_prod']   = $pengeluaran\_prod;
9. $data['hasil\_panen']        = null;
10. $data['hasil']              = null;
11. $data['margin']             = null;
12. return view('laporan.harga',$data);
13. }
14. public function hasilHarga(createHarga $request)
15. {
16. $month                  = $request['month'];
17. $year                   = $request['year'];
18. $pengeluaran\_prod       = labarugi::whereYear('created\_at', '=', $year)
19. ->whereMonth('created\_at', '=', $month)
20. ->sum('pengeluaran');
21. $hasil\_panen            = $request['hasil\_panen'];
22. $margin                 = $request['margin'];
23. $data['y']              = $year;
24. $data['m']              = $month;
25. $hasil                  = ($pengeluaran\_prod / $hasil\_panen) + $margin;  //rumus penentuan harga pokok
26. $data['pengeluaran\_prod'] = $pengeluaran\_prod;
27. $data['hasil\_panen']    = $hasil\_panen;
28. $data['hasil']          = $hasil;
29. $data['margin']         = $margin;
30. return view('laporan.harga',$data);
31. }

Script 4.38 controller harga pokok

Script 4.38 mempunyai function harga() dan tampilHarga(). Untuk function harga() hanya untuk menampilkan halaman menu harga pokok. Sedangkan function hasilHarga() berfungsi menghitung hasil inputan dari pengguna yang hasilnya adalah harga pokok penjualan.

1. Pengujian Sistem

Setelah sistem selesai dibangun, pengguna diminta untuk melakukan test terhadap aplikasi. Dilakukan 2 test yaitu *Black Box test* dan *Aplha Test.*

* 1. *Black Box Test*

Pengujian dilakukan oleh ibu sumarni sebagai Ketua kelompok Petani hidroponik. Pengujian ini diberikan 9 pertanyaan dengan 2 alternatif jawaban yaitu ya atau tidak. Daftar pada pertanyaan pada pengujian ini dapat dilihat pada tabel 4.28.

Tabel 4.28 Hasil Black Box Test

|  |  |  |  |
| --- | --- | --- | --- |
| no | pertanyaan | Pilihan jawaban | |
| ya | tidak |
| 1 | Apakah sistem dapat menampilkan form login | √ |  |
| 2 | Apakah aplikasi mampu login | √ |  |
| 3 | Apakah aplikasi mampu membuat dan menghapus data pengguna yang hanya bisa di akses oleh admin. | √ |  |
| 4 | Apakah aplikasi mampu membuat, menampilkan, mengubah, menghapus dan mencetak data anggaran keuangan yang bisa di akses admin dan anggota. | √ |  |
| 5 | Apakah aplikasi mampu membuat, menampilkan, dan mencetak data pemasukan keuangan yang bisa di akses admin dan anggota. | √ |  |
| 6 | Apakah aplikasi mampu membuat, menampilkan, dan mencetak data pegeluaran keuangan yang bisa di akses admin dan anggota. | √ |  |
| 7 | Apakah aplikasi mampu menampilkan dan mencetak laporan laba rugi yang bisa di akses admin dan anggota. | √ |  |
| 8 | Apakah aplikasi mampu menampilkan dan mencetak laporan keuangan bulanan yang bisa di akses admin dan anggota. | √ |  |
| 9 | Apakah aplikasi mampu menampilkan harga pokok yang bisa di akses admin dan anggota. | √ |  |
| Jumlah | | 9 |  |

Dari hasil pengujian, diperoleh presentase penilaian jawaban Ya = 9/9x100% = 100 % dan jawaban Tidak = 0/9x100% = 0%. Dari hasil Pengujian tersebut dapat disimpulkan bahwa aplikasi berjalan dengan baik dan proses-proses yang terdapat dalam sistem aplikasi sesuai dengan kebutuhan.

* 1. *Alpha Test*

Pengujian dengan *alpha test* dilakukan dengan meminta salah satu anggota kelompok petani hidroponik untuk mencoba aplikasi kemudian pengguna diberikan daftar pertanyaan terkait program yang dijalankan. Pada pengujian ini disertakan 7 pertanyaan dan pilihan jawabannya menggunakan skala likert dimana terdapat 4 kriteria yaitu SS = Sangat Setuju, S = Setuju, TS = Tidak Setuju, STS = Sangat Tidak Setuju. Jumlah responden yang memilih salah satu skala terdapat pada kolom (N) dan presentase hasil pemilihan responden terdapat pada kolom (%) Dalam program ini pengujian dilakukan 9 orang responden. Daftar nama responden dapat dilihat pada tabel 4.29.

Tabel 4.29 daftar responden *alpha test*

|  |  |  |
| --- | --- | --- |
| no | Nama | Jabatan |
| 1 | Cakra dewi | Wakil ketua |
| 2 | Eny Winarsih | Anggota |
| 3 | Tutik Effendi | Anggota |
| 4 | Noviana Catur Wulandari | Anggota |
| 5 | Nur Setiati | Anggota |
| 6 | Muassomah | Anggota |
| 7 | Indah | Anggota |
| 8 | Nakisah Kurniawati | Anggota |
| 9 | Thoyyib Nugroho | Anggota |

Hasil pengujian dapat dilihat pada tabel 4.30.

Tabel 4.30 hasil pengujian *alpha test*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Pertanyaan | Jawaban | | | | | | | |
| SS | | S | | TS | | STS | |
|  |  | N | % | N | % | N | % | N | % |
| 1 | Tampilan aplikasi ini menarik dan mudah dimengerti | 2 | 22,2 % | 7 | 77,7 % | 0 | 0% | 0 | 0% |
| 2 | Fitur menu anggaran keuangan dapat membantu merencanakan anggaran belanja dengan baik | 1 | 11,1 % | 8 | 88,8 % | 0 | 0% | 0 | 0% |
| 3 | Fitur menu pemasukan dapat membantu mengelola pemasukan keuangan dengan baik | 2 | 22,2 % | 7 | 77,7 % | 0 | 0% | 0 | 0% |
| 4 | Fitur menu pengeluaran dapat membantu mengelola pengeluaran keuangan dengan baik | 1 | 11.1 % | 8 | 88,8 % | 0 | 0% | 0 | 0% |
| 5 | Fitur menu laporan laba-rugi dapat membantu mengetahui jumlah laba atau rugi dengan baik | 2 | 22,2 % | 7 | 77,7 % | 0 | 0% | 0 | 0% |
| 6 | Fitur menu laporan keuangan bulanan dapat membantu memantau arus dana kelompok dengan baik. | 1 | 11,1 % | 8 | 88,8 % | 0 | 0% | 0 | 0% |
| 7 | Fitur menu penentuan harga pokok jual dapat membantu kelompok dalam menentukan harga jual produk dengan baik | 4 | 44,4 % | 5 | 55,5 % | 0 | 0% | 0 | 0% |
|  | Total rata -rata | 20, 6 % | | 79,2 % | | 0% | | 0% | |

Berdasarkan hasil pengujian diperoleh presentase penilaian sebagai berikut :

STS (Sangat Tidak setuju) = 0 %

TS (KurangSetuju) = 0 %

S (Setuju) = 79,2 %

SS (Sangat Setuju) = 20,6 %

Dari hasil penilaian tersebut dapat disimpulkan bahwa aplikasi sistem informasi keuangan kelompok petani hidroponik dapat mempermudah anggota untuk melakukan pengelolaan keuangan di kelompok petani hidroponik ASRI 12 Kauman.

**Bab V**

**Kesimpulan dan saran**

1. Kesimpulan

Bedasarkan hasil penelitian dan pembahasan, didapat beberapa kesimpulan sebagai berikut :

* + 1. Telah dibangun aplikasi sistem informasi keuangan berbasis web menggunakan *framework* laravel untuk kelompok petani hidroponik ASRI 12 Kauman.
    2. Aplikasi sistem infromasi keuangan dapat memberikan informasi kepada anggota kelompok petani dalam memantau keadaan keuangan kelompok.
    3. Aplikasi sistem informasi keuangan dapat memberikan informasi perhitungan harga pokok penjualan produk sebagai acuan dalam menentukan harga penjualan produk sayur hidroponik.
    4. Hasil pengujian sistem menunjukkan presentase terbesar 79,2% setuju bahwa aplikasi telah berjalan dengan baik, memenuhi kebutuhan pengguna dan mempermudah anggota kelompok petani dalam mengelola keuangan kelompok.

1. Saran

Aplikasi yang dibangun tentunya masih jauh dari sempurna. Maka dari itu terdapat beberapa saran untuk *developer* aplikasi selanjutnya yang ingin mengembangkan sistem ini yaitu :

* 1. Tampilan antarmuka (*interface*) aplikasi ini masih sangat standar. Sehingga diperlukan tampilan aplikasi yang dikembangkan dengan integrasi *bootstrap, google material* dan *framework interface* lainya agar tampilan aplikasi lebih menarik dan interaktif.
  2. Aplikasi sistem infromasi keuangan ini berbasis web. Jika diakses di perangkat mobile maka perlu dibuka melalui *browser.* Maka perlu dikembangkan untuk versi *mobile* agar pengaksesan aplikasi lebih mudah dan cepat.

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, A., Beeh, Y. R., & Astuningdyas, H. (2008). Perancangan Aplikasi Rencana Anggaran Biaya ( Rab ). *JURNAL INFORMATIKA*, *10*, 10–18.

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.

LAMPIRAN