**SISTEM INFORMASI EKSEKUTIF PENJUALAN KERAJINAN TAS ROTAN PADA BALINESE CLASSIC MENGGUNAKAN *FRAMEWORK* LARAVEL**

**TUGAS AKHIR**

****

**Oleh :**

**MADE YOGI NUGRAHA (160030638)**

**INSTITUT TEKNOLOGI DAN BISNIS**

**(ITB) STIKOM BALI**

**2019**

# PERSETUJUAN SEMINAR DAN SIDANG TUGAS AKHIR

**SISTEM INFORMASI EKSEKUTIF PENJUALAN KERAJINAN TAS ROTAN PADA BALINESE CLASSIC MENGGUNAKAN *FRAMEWORK* LARAVEL**

**Oleh:**

**MADE YOGI NUGRAHA (160030638)**

**Dosen Pembimbing Tanda Tangan Tanggal**

Dian Pramana, S.Kom., M.Kom

Lilis Yuningsih, SH., M.Kom

Denpasar, ...........................

Mengetahui,

Ketua Program Studi Sistem Informasi

Ricky Aurelius Nurtanto Diaz, S.Kom., M.T

***Halaman ini sengaja dikosongkan***

# PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

NIM : 160030638

Nama : Made Yogi Nugraha

Jenjang Studi : Strata Satu (S1)

Program Studi : Sistem Informasi

Tempat, Tgl. Lahir : Denpasar, 21 Agustus 1998

Alamat : Jl. Kebo Iwa Perum Swamandala XIII no. 3

Denpasar Barat, Bali.

NIK : 5171042108980001

Menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh Gelar **Sarjana Komputer (S.Kom)** di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya 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.

Denpasar, 6 Januari 2020

( Made Yogi Nugraha )

***Halaman ini sengaja dikosongkan***

**SISTEM INFORMASI EKSEKUTIF PENJUALAN KERAJINAN TAS ROTAN PADA BALINESE CLASSIC MENGGUNAKAN *FRAMEWORK* LARAVEL**

# ABSTRAK

Sistem Informasi Eksekutif Penjualan dapat membantu eksekutif sebuah perusahaan berskala besar hingga usaha menengah ke bawah dalam mengelola usaha atau bisnis, salah satunya adalah Balinese Classic. Balinese Classic merupakan perusahaan yang menjual berbagai macam kerajinan tas rotan khas dari bali. Selain menjual tas di dalam negeri, Balinese Classic juga melakukan *export* ke negara-negara Asia Tenggara seperti Singapore, Filipina dan Malaysia. Oleh sebab banyaknya proses operasional yang terjadi, tidak jarang terjadi kesalahan pencatatan maupun faktur- fatur yang telah lama disimpan rusak atau hilang sehingga lambat laun akan menimbulkan kendala untuk operasional pada Balinese Classic. Sistem Informasi Eksekutif Penjualan ini dibangun dengan harapan dapat membantu proses operasional pencatatan pelanggan, proses transaksi dan pembuatan laporan sehingga tidak terjadi lagi kesalahan yang nantinya dapat merugikan perusahaan. Pada pembuatan sistem tersebut, konsep perencanaan dan perancangan melalui tahap pengumpulan data, *Data Flow Diagram* (*DFD*), *Entity Relationship Diagram* (*ERD*), Konseptual *database* dan struktur table. Sistem ini dibangun dengan menggunakan *Framework* Laravel dan *Framework* Bootstrap. Hasil dari penelitian ini berupa Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan pada Balineses Classic Menggunakan *Framework* Laravel.

**Kata kunci:** Sistem Informasi Eksekutif, Penjualan, Tas Rotan, Balinese Classic, *Framework* Laravel.

***Halaman ini sengaja dikosongkan***

***EXECUTIVE INFORMATION SYSTEM OF SALES OF RATTAN BAG CRAFTS IN BALINESE CLASSIC USING FRAMEWORK LARAVEL***

# *ABSTRACT*

*Sales Executive Information System can help executives of large-scale companies to medium-sized businesses in managing a business or business, one of which is Bali Classic. Balinese Classic is a company that sells a variety of handicraft bags from Bali. In addition to selling bags domestically, Balinese Classic also exports to Southeast Asian countries such as Singapore, the Philippines and Malaysia. Because of the many operational processes that occur, it is not uncommon for recording errors and invoices to be stored for a long time to be damaged or lost, which in turn will cause operational constraints on the Balinese Classic. This Executive Sales Information System was built with the hope of helping the operational processes of customer records, transaction processing and report generation so that no errors occur that would later harm the company. In making the system, the concept of planning and design through the stages of data collection, Data Flow Diagrams (DFD), Entity Relationship Diagrams (ERD), conceptual databases and table structures. This system was built using the Laravel Framework and Bootstrap Framework. The results of this study are in the form of an Executive Information System for the Sale of Craft Rattan Bags in Balinese Classic Using the Laravel Framework.*

***Keywords:*** *Executive Information System, Sales, Rattan Bags, Balinese Classics, Laravel Framework.*

***Halaman ini sengaja dikosongkan***

# KATA PENGANTAR

Berkat Rahmat Tuhan Yang Maha Esa, penulis dapat menyelesaikan Tugas Akhir yang berjudul **“SISTEM INFORMASI EKSEKUTIF PENJUALAN KERAJINAN TAS ROTAN PADA BALINESE CLASSIC MENGGUNAKAN *FRAMEWORK* LARAVEL”** sesuai dengan waktu yang telah direncanakan. Selanjutnya penulis menyampaikan Terima Kasih kepada:

1. Rektor ITB STIKOM Bali Bapak Dr. Dadang Hermawan.
2. Bapak Dr. Muhammad Rusli., MT selaku Wakil Rektor I.
3. Ibu Ni Luh Putri Srinadi, SE., MM.Kom selaku Wakil Rektor II.
4. Bapak Dian Pramana, S.Kom., M.Kom selaku Dekan Fakultas Informatika dan Komputer ITB STIKOM Bali.
5. Bapak Ricky Aurelius Nurtanto Diaz, S.Kom., M.T selaku Ketua Program Studi Sistem Informasi ITB STIKOM Bali.
6. Bapak Dian Pramana, S.Kom., M.Kom selaku Dosen Pembimbing Utama yang telah membimbing penulis selama melaksanakan Tugas Akhir.
7. Ibu Lilis Yuningsih, SH., M.Kom selaku Dosen Pembimbing Pendamping yang juga turut membimbing penulis selama melaksanakan Tugas Akhir.
8. Semua teman dan berbagai pihak yang memberikan dukungan dan bantuan kepada penulis.

Semoga hasil penulisan Tugas Akhir ini bermanfaat bagi pihak yang berkepentingan.

Denpasar, Januari 2020

Penulis

***Halaman ini sengaja dikosongkan***

# DAFTAR ISI

[PERSETUJUAN SEMINAR DAN SIDANG TUGAS AKHIR i](#_Toc29287598)

[PERNYATAAN KEASLIAN iii](#_Toc29287599)

[ABSTRAK v](#_Toc29287600)

[*ABSTRACT* vii](#_Toc29287601)

[KATA PENGANTAR ix](#_Toc29287602)

[DAFTAR ISI xi](#_Toc29287603)

[DAFTAR TABEL xiii](#_Toc29287604)

[DAFTAR GAMBAR xv](#_Toc29287605)

[BAB I PENDAHULUAN 1](#_Toc29287606)

[1.1 Latar Belakang 1](#_Toc29287607)

[1.2 Rumusan Masalah 2](#_Toc29287608)

[1.3 Tujuan Penelitian 2](#_Toc29287609)

[1.4 Manfaat Penelitian 2](#_Toc29287610)

[1.5 Ruang lingkup Penelitian 3](#_Toc29287611)

[1.6 Sistematika Penulisan 4](#_Toc29287612)

[BAB II TINJAUAN PUSTAKA 7](#_Toc29287613)

[2.1 *State of The Art* 7](#_Toc29287614)

[2.2 Balinese *Class*ic 9](#_Toc29287615)

[2.3 Sistem Informasi Eksekutif (SIE) 12](#_Toc29287616)

[2.4 Tujuan Sistem Informasi Eksekutif 12](#_Toc29287617)

[2.5 *Website* 13](#_Toc29287618)

[2.6 *Data Flow Diagram (DFD)* 14](#_Toc29287619)

[2.7 *Entity Relationship Diagram (ERD)* 16](#_Toc29287620)

[2.8 *PHP* 17](#_Toc29287621)

[2.9 MariaDB 18](#_Toc29287622)

[2.10 Laravel 19](#_Toc29287623)

[2.11 Bootstrap 21](#_Toc29287624)

[2.12 *Blackbox Testing* 21](#_Toc29287625)

[2.13 Model *Waterfall* 22](#_Toc29287626)

[BAB III METODE PENELITIAN 25](#_Toc29287627)

[3.1 Waktu dan Lokasi Penelitian 25](#_Toc29287628)

[3.2 Alat dan bahan Penelitian 25](#_Toc29287629)

[3.3 Metode Pengumpulan Data 26](#_Toc29287630)

[3.4 Metode Perekayasaan 26](#_Toc29287631)

[3.5 Penulisan Laporan 27](#_Toc29287632)

[BAB IV HASIL DAN PEMBAHASAN 29](#_Toc29287633)

[4.1 Hasil Analisis 29](#_Toc29287634)

[4.2 Perancangan Sistem 33](#_Toc29287635)

[4.3 Perancangan Basis Data 44](#_Toc29287636)

[4.4 Perancangan Antarmuka 50](#_Toc29287637)

[4.4.1 Desain Antarmuka Admin 50](#_Toc29287638)

[4.4.2 Desain Antarmuka Kasir 62](#_Toc29287639)

[4.4.3 Desain Antarmuka Gudang 65](#_Toc29287640)

[4.5 Implementasi Sistem 68](#_Toc29287641)

[4.5.1 Tampilan Untuk Admin 68](#_Toc29287642)

[4.5.2 Tampilan Untuk Kasir 83](#_Toc29287643)

[4.5.3 Tampilan Untuk Gudang 86](#_Toc29287644)

[4.6 Pengujian Black Box 89](#_Toc29287645)

[4.6.1 Pengujian Halaman Admin 89](#_Toc29287646)

[4.6.2 Pengujian Halaman Kasir 104](#_Toc29287647)

[4.6.3 Pengujian Halama Gudang 105](#_Toc29287648)

[BAB V PENUTUP 107](#_Toc29287649)

[5.1 Kesimpulan 107](#_Toc29287650)

[5.2 Saran 107](#_Toc29287651)

[DAFTAR PUSTAKA 109](#_Toc29287652)

# DAFTAR TABEL

[Tabel 2.1 State of The Art 7](#_Toc29252124)

[Tabel 2.2 Simbol-simbol DFD 15](#_Toc29252125)

[Tabel 2.3 Simbol-simbol ERD 16](#_Toc29252126)

[Tabel 4.1 Analisa *User* 31](#_Toc29287953)

[Tabel 4.2 *Mapping* Hak Akses 31](#_Toc29287954)

[Tabel 4.3 Analisa Data 32](#_Toc29287955)

[Tabel 4.4 Analisa Proses 33](#_Toc29287956)

[Tabel 4.5 Struktur Tabel *Users* 46](#_Toc29287957)

[Tabel 4.6 Struktur Tabel *Customers* 46](#_Toc29287958)

[Tabel 4.7 Struktur Tabel *Suppliers* 47](#_Toc29287959)

[Tabel 4.8 Struktur Tabel *Categories* 47](#_Toc29287960)

[Tabel 4.9 Struktur Tabel *Product* 47](#_Toc29287961)

[Tabel 4.10 Struktur Tabel *Stock*s 48](#_Toc29287962)

[Tabel 4.11 Struktur Tabel Penjualan 48](#_Toc29287963)

[Tabel 4.12 Struktur Tabel Penjualan\_*products* 49](#_Toc29287964)

[Tabel 4.13 Struktur Tabel Pembelian 49](#_Toc29287965)

[Tabel 4.14 Struktur Tabel Pembelian\_*product* 50](#_Toc29287966)

[Tabel 4.15 Hasil Pengujian Halaman *Login* – Admin 89](#_Toc29287967)

[Tabel 4.16 Hasil Pengujian pada Navigasi Menu – Admin 90](#_Toc29287968)

[Tabel 4.17 Pengujian pada Halaman *Users* – Admin 91](#_Toc29287969)

[Tabel 4.18 Pengujian *Form* Tambah pada Halaman *Users* – Admin 91](#_Toc29287970)

[Tabel 4.19 Pengujian pada Halaman *Customers* – Admin 92](#_Toc29287971)

[Tabel 4.20 Pengujian *Form* Tambah pada Halaman *Customers* – Admin 93](#_Toc29287972)

[Tabel 4.21 Pengujian pada Halaman *Suppliers* – Admin 94](#_Toc29287973)

[Tabel 4.22 Pengujian *Form* Tambah pada Halaman *Suppliers* – Admin 94](#_Toc29287974)

[Tabel 4.23 Pengujian pada Halaman *Categories* – Admin 95](#_Toc29287975)

[Tabel 4.24 Pengujian *Form* Tambah pada Halaman *Categories* – Admin 96](#_Toc29287976)

[Tabel 4.25 Pengujian pada Halaman *Products* – Admin 97](#_Toc29287977)

[Tabel 4.26 Pengujian *Form* Tambah pada Halaman *Products* – Admin 97](#_Toc29287978)

[Tabel 4.27 Pengujian pada Halaman *Stock*s – Admin 98](#_Toc29287979)

[Tabel 4.28 Pengujian pada Halaman Penjualan – Admin 99](#_Toc29287980)

[Tabel 4.29 Pengujian pada Halaman Tambah Transaksi Jual – Admin 99](#_Toc29287981)

[Tabel 4.30 Pengujian pada Halaman Pembelian – Admin 100](#_Toc29287982)

[Tabel 4.31 Pengujian pada Halaman Tambah Transaksi Beli – Admin 101](#_Toc29287983)

[Tabel 4.32 Pengujian pada Halaman Laporan Penjualan – Admin 102](#_Toc29287984)

[Tabel 4.33 Pengujian pada Halaman Laporan Pembelian – Admin 103](#_Toc29287985)

[Tabel 4.34 Hasil Pengujian Halaman *Login* – Kasir 104](#_Toc29287986)

[Tabel 4.35 Hasil Pengujian Halaman *Login* – Gudang 105](#_Toc29287987)

# DAFTAR GAMBAR

[Gambar 2.1 Gudang 9](#_Toc29252162)

[Gambar 2.2 Akun *instagram* pertama 10](#_Toc29252163)

[Gambar 2.3 Akun *instagram* kedua 10](#_Toc29252164)

[Gambar 2.4 Grafik Penjualan 11](#_Toc29252165)

[Gambar 2.5 MariaDB 18](#_Toc29252166)

[Gambar 2.6 Laravel 20](#_Toc29252167)

[Gambar 2.7 Bootstrap 21](#_Toc29252168)

[Gambar 2.8 Model *Waterfall* 23](#_Toc29252169)

[Gambar 4.1 Diagram Konteks 34](#_Toc29287773)

[Gambar 4.2 Data *Flow* Diagram (DFD) Level 0 35](#_Toc29287774)

[Gambar 4.3 Data *Flow* Diagram (DFD) Level 1 Proses 1 *Login* 36](#_Toc29287775)

[Gambar 4.4 Data *Flow* Diagram (DFD) Level 1 Proses 2 *Users* 37](#_Toc29287776)

[Gambar 4. 5 Data *Flow* Diagram (DFD) Level 1 Proses 3 *Customers* 37](#_Toc29287777)

[Gambar 4.6 Data *Flow* Diagram (DFD) Level 1 Proses 4 *Suppliers* 38](#_Toc29287778)

[Gambar 4.7 Data *Flow* Diagram (DFD) Level 1 Proses 5 *Categories* 39](#_Toc29287779)

[Gambar 4.8 Data *Flow* Diagram (DFD) Level 1 Proses 6 *Product* 40](#_Toc29287780)

[Gambar 4.9 Data *Flow* Diagram (DFD) Level 1 Proses 7 *Stock* 41](#_Toc29287781)

[Gambar 4.10 Data *Flow* Diagram (DFD) Level 1 Proses 8 Transaksi Jual 42](#_Toc29287782)

[Gambar 4.11 Data *Flow* Diagram (DFD) Level 1 Proses 9 Transaksi Beli 43](#_Toc29287783)

[Gambar 4.12 Data *Flow* Diagram (DFD) Level 1 Proses 10 Laporan 43](#_Toc29287784)

[Gambar 4.13 *Entity Relationship* Diagram 44](#_Toc29287785)

[Gambar 4.14 Basis Data Konseptual 45](#_Toc29287786)

[Gambar 4.15 Antarmuka *Login* – Admin 51](#_Toc29287787)

[Gambar 4.16 Antarmuka *Dashboard* – Admin 51](#_Toc29287788)

[Gambar 4.17 Antarmuka Data *Users* – Admin 52](#_Toc29287789)

[Gambar 4.18 Antarmuka *Form* *Users* – Admin 52](#_Toc29287790)

[Gambar 4.19 Antarmuka Data *Customers* – Admin 53](#_Toc29287791)

[Gambar 4.20 Antarmuka *Form* *Customers* – Admin 53](#_Toc29287792)

[Gambar 4.21 Antarmuka Data *Suppliers* – Admin 54](#_Toc29287793)

[Gambar 4.22 Antarmuka *Form* *Suppliers* – Admin 54](#_Toc29287794)

[Gambar 4.23 Antarmuka Data *Categories* – Admin 55](#_Toc29287795)

[Gambar 4.24 Antarmuka *Form* *Categories* – Admin 55](#_Toc29287796)

[Gambar 4.25 Antarmuka Data *Products* – Admin 56](#_Toc29287797)

[Gambar 4.26 Antarmuka *Form* *Products* – Admin 56](#_Toc29287798)

[Gambar 4.27 Antarmuka Data *Stock* – Admin 57](#_Toc29287799)

[Gambar 4.28 Antarmuka *Form* *Stock* – Admin 57](#_Toc29287800)

[Gambar 4. 29 Antarmuka Data Penjualan – Admin 58](#_Toc29287801)

[Gambar 4.30 Antarmuka Transaksi Jual – Admin 58](#_Toc29287802)

[Gambar 4.31 Antarmuka Data Pembelian – Admin 59](#_Toc29287803)

[Gambar 4.32 Antarmuka Transaksi Beli – Admin 59](#_Toc29287804)

[Gambar 4.33 Antarmuka *Invoice* Penjualan – Admin 60](#_Toc29287805)

[Gambar 4.34 Antarmuka *Invoice* Pembelian – Admin 60](#_Toc29287806)

[Gambar 4.35 Antarmuka Laporan Penjualan – Admin 61](#_Toc29287807)

[Gambar 4.36 Antarmuka Laporan Pembelian – Admin 61](#_Toc29287808)

[Gambar 4.37 Antarmuka *Login* – Kasir 62](#_Toc29287809)

[Gambar 4.38 Antarmuka *Dashboard* – Kasir 62](#_Toc29287810)

[Gambar 4.39 Antarmuka *List* *Customer* – Kasir 63](#_Toc29287811)

[Gambar 4.40 Antarmuka *Form* *Customer* – Kasir 63](#_Toc29287812)

[Gambar 4.41 Antarmuka *List* Penjualan – Kasir 64](#_Toc29287813)

[Gambar 4.42 Antarmuka *Form* Penjualan – Kasir 64](#_Toc29287814)

[Gambar 4.43 Antarmuka *Login* – Gudang 65](#_Toc29287815)

[Gambar 4.44 Antarmuka *Dashboard* Kasir – Gudang 65](#_Toc29287816)

[Gambar 4.45 Antarmuka *List* *Supplier* – Gudang 66](#_Toc29287817)

[Gambar 4.46 Antarmuka *Form* *Supplier* – Gudang 66](#_Toc29287818)

[Gambar 4.47 Antarmuka *List* Pembelian – Gudang 67](#_Toc29287819)

[Gambar 4.48 Antarmuka *Form* Pembelian – Gudang 67](#_Toc29287820)

[Gambar 4.49 Tampilan Halaman *Login* – Admin 68](#_Toc29287821)

[Gambar 4.50 Tampilan Halaman *Dashboard* – Admin 69](#_Toc29287822)

[Gambar 4.51 Tampilan Halaman *List* *Users* – Admin 70](#_Toc29287823)

[Gambar 4.52 Tampilan *Form* *User* – Admin 70](#_Toc29287824)

[Gambar 4.53 Tampilan Halaman *List* *Customers* – Admin 71](#_Toc29287825)

[Gambar 4.54 Tampilan *Form* *Customer* – Admin 71](#_Toc29287826)

[Gambar 4.55 Tampilan Detail *Customer* – Admin 72](#_Toc29287827)

[Gambar 4.56 Tampilan Halaman *List* *Suppliers* – Admin 72](#_Toc29287828)

[Gambar 4.57 Tampilan *Form* *Supplier* – Admin 73](#_Toc29287829)

[Gambar 4.58 Tampilan Detail *Supplier* – Admin 73](#_Toc29287830)

[Gambar 4.59 Tampilan Halaman *List* *Categories* – Admin 74](#_Toc29287831)

[Gambar 4.60 Tampilan *Form* *Category* – Admin 74](#_Toc29287832)

[Gambar 4.61 Tampilan Halaman *List* *Products* – Admin 75](#_Toc29287833)

[Gambar 4.62 Tampilan *Form* *Product* – Admin 75](#_Toc29287834)

[Gambar 4.63 Tampilan Detail *Product* – Admin 76](#_Toc29287835)

[Gambar 4.64 Tampilan Halaman *List* *Stock*s – Admin 76](#_Toc29287836)

[Gambar 4.65 Tampilan *Form* *Stock* – Admin 77](#_Toc29287837)

[Gambar 4.66 Tampilan Halaman *List* Penjualan – Admin 77](#_Toc29287838)

[Gambar 4.67 Tampilan *Form* Penjualan – Admin 78](#_Toc29287839)

[Gambar 4.68 Tampilan Halaman *Invoice* Penjualan – Admin 78](#_Toc29287840)

[Gambar 4.69 Tampilan Halaman *List* Pembelian – Admin 79](#_Toc29287841)

[Gambar 4.70 Tampilan Halaman *Form* Pembelian – Admin 79](#_Toc29287842)

[Gambar 4.71 Tampilan Halaman *Invoice* Pembelian – Admin 80](#_Toc29287843)

[Gambar 4.72 Tampilan Halaman Laporan Penjualan – Admin 81](#_Toc29287844)

[Gambar 4.73 Tampilan Halaman Laporan Pembelian – Admin 82](#_Toc29287845)

[Gambar 4.74 Tampilan Halaman *Login* – Kasir 83](#_Toc29287846)

[Gambar 4.75 Tampilan Halaman *Dashboard* – Kasir 83](#_Toc29287847)

[Gambar 4.76 Tampilan Halaman *List* *Customers* – Kasir 84](#_Toc29287848)

[Gambar 4.77 Tampilan Halaman *List* Penjualan – Kasir 84](#_Toc29287849)

[Gambar 4.78 Tampilan *Form* Penjualan – Kasir 85](#_Toc29287850)

[Gambar 4.79 Tampilan Halaman *Invoice* Penjualan – Kasir 85](#_Toc29287851)

[Gambar 4.80 Tampilan Halaman *Login* – Gudang 86](#_Toc29287852)

[Gambar 4.81 Tampilan Halaman *Dashboard* – Gudang 86](#_Toc29287853)

[Gambar 4.82 Tampilan Halaman *List* *Suppliers* – Gudang 87](#_Toc29287854)

[Gambar 4.83 Tampilan *Form* *Supplier* – Gudang 87](#_Toc29287855)

[Gambar 4.84 Tampilan Halaman *List* Pembelian – Gudang 88](#_Toc29287856)

[Gambar 4.85 Tampilan Halaman *Form* Pembelian – Gudang 88](#_Toc29287857)

***Halaman ini sengaja dikosongkan***

# BAB I PENDAHULUAN

1. Latar Belakang

Balinese Classic merupakan sebuah usaha yang bergerak pada penjualan kerajinan bali khususnya tas rotan. Produk dipasarkan ke berbagai negara seperti Malaysia, Filipina, Singapore dan masih banyak lagi melalui *supplier* masing-masing negara. Perusahaan ini bertempat di Jalan Kebo Iwa Perumahan Swamandala XIII No.3 Denpasar, Bali. Perusahaan ini baru merintis usahanya sejak Juli 2018 lalu. Balinese Classic melakukan pembelian tas dari beberapa *supplier* di pasar tradisional Bali kemudian barang akan disimpan di gudang dan akan dipasarkan kembali ke berbagai belahan dunia.

Selama ini perusahaan dalam melakukan kegiatan pengolahan data laporan dilakukan secara manual dengan menggunakan buku catatan. Laporan tersebut berisi tentang informasi pembelian, penjualan, stok barang, jumlah barang yang dikirim ke masing-masing *supplier*. Laporan tersebut berasal dari hasil pencatatan dari nota-nota pembelian maupun penjualan setiap minggunya. Selain itu untuk menghitung laba juga masih menggunakan alat manual seperti kalkulator. Dari proses tersebut, maka lambat laun akan menimbulkan kendala untuk operasional pada Balinese Classic.

Sebelumnya sudah dilakukan penelitian terdahulu yang memiliki keterkaitan dengan sistem informasi penjualan. Penelitian pertama diteliti oleh Yazid Maulana dengan judul “Sistem Informasi Penjualan Kayu Ulin Pada UD Karya Perdana Berbasis *Web* Menggunakan *Framework* Codeigniter” [1]. Dalam penelitian tersebut dijabarkan bagaimana cara bertransaksi menggunakan sarana aplikasi berbasis *web* yang bertujuan memudahkan penjualan kayu ulin pada UD Karya Perdana. Penelitian kedua dengan judul ”Sistem Informasi Eksekutif Penjualan Jajanan Sagu pada Bagia Suli menggunakan *Framework* Laravel” yang diteliti oleh I Kadek Adi Adnyana [2]. Pada penelitian tersebut dihasilkan sebuah aplikasi eksekutif penjualan yang memiliki manfaat memudahkan penjualan jajanan sagu juga mempermudah dalam pembuatan laporan penjualan. Keterkaitan yang ada pada penelitian tersebut dengan penelitian yang akan dibuat ialah sama-sama sistem informasi eksekutif penjualan dan menggunakan *Framework* laravel.

Berdasarkan uraian sebelumnya dan referensi dari penelitian terdahulu maka pada penelitian ini penulis akan membuat Sistem Informasi Eksekutif penjualan kerajinan tas rotan pada Balinese Classic menggunakan *Framework* Laravel. Sistem Informasi Eksekutif (SIE) adalah salah satu jenis manajemen sistem informasi untuk memudahkan dan mendukung keterangan dan pembuatan keputusan yang dibutuhkan eksekutif senior dengan menyediakan kemudahan akses terhadap informasi baik dari dalam maupun luar. Sistem ini juga didukung dengan *Framework* Laravel yaitu suatu pengembangan *website* berbasis *Model* *View* *Controller* (MVC) yang ditulis dalam *PHP* yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan kode yang ekspresif, jelas dan menghemat waktu. Dengan adanya sistem informasi eksekutif ini dapat membantu mengelola proses transaksi seperti mencatat, mengolah serta menampilkan dalam bentuk grafik data transaksi. Keunggulan yang ditawarkan sistem ini adalah memudahkan kinerja perusahaan dalam proses perhitungan, pencatatan data-data perusahaan, serta arsip data perusahaan. Dalam hal ini sistem akan membantu pihak perusahaan dalam perencanaan stok barang dan perencanaan penjualan.

1. Rumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah bagaimana hasil rancang bangun Sistem Informasi Eksekutif penjualan kerajinan tas rotan pada Balinese *Class*ic menggunakan *Framework* Laravel?

1. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk merancang bangun Sistem Informasi Eksekutif penjualan kerajinan tas rotan pada Balinese Classic menggunakan *Framework* Laravel.

1. Manfaat Penelitian

Adapun Manfaat yang didapat dari penelitian ini adalah sebagai berikut:

1. Memudahkan pengecekan dan pencarian data *customer*, stok barang dan transaksi yang masuk.
2. Dapat melakukan *review* laporan penjualan per hari, per bulan, dan per tahun.
3. Efesiensi kerja, waktu, dan tenaga.
4. Menyajikan laporan penjualan dengan cepat dan tepat waktu dalam bentuk grafik atau diagram.
5. Mempermudah dalam memantau perkembangan perusahaan.
6. Menghindari kesalahan yang diakibatkan menggunakan sistem manual.
7. Ruang lingkup Penelitian

Adapun ruang lingkup pada penelitian ini adalah sebagai berikut:

1. *User* dari sistem informasi eksekutif penjualan ini adalah:
2. *Admin* adalah pengguna yang menjadi pengelola sistem informasi yang mendapat hak akses penuh terhadap sistem ini.
3. Kasir adalah pengguna yang memiliki tugas melakukan *input* penjualan.
4. Gudang adalah pengguna yang memiliki tugas melakukan *input* pembelian.
5. Sebelum masuk ke sistem, pengguna diharuskan melakukan *login* terlebih dahulu.
6. Untuk *login* Kasir dan Gudang, Admin harus mendaftarkan dulu pada halaman users.
7. Data yang digunakan pada sistem ini adalah data *user*, data pembelian, data barang, data kategori, data customer, data *supplier* serta data penjualan.
8. Sistem memiliki fitur *dashboard* di mana berisi menu – menu tersendiri sesuai dengan siapa yang *login*, apakah itu *Admin*, Kasir atau Gudang.
9. Admin dapat mengelola keseluruhan yang ada pada sistem serta mendapatkan laporan penjualan dan pembelian.
10. Kasir hanya dapat mengelola data customer, data kategori, data produk dan data transaksi penjualan.
11. Gudang hanya dapat mengelola data supplier, data kategori, data produk, data stock dan data transaksi pembelian.
12. Sistem ini dibangun dalam bentuk *web* menggunakan bahasa pemrograman *PHP* dan menggunakan *Framework* Laravel yang dikombinasikan dengan Javascript Ajax. *Database* yang digunakan dalam sistem ini menggunakan MySQL dengan XAMPP sebagai *software* untuk pengolahannya.
13. Perancangan sistem ini menggunakan *Data Flow Diagram* (*DFD*) dan untuk perancangan basis data digunakan *Entity Relationship Diagram* (*ERD*).
14. *Output* yang dihasilkan adalah Nota transaksi, Laporan penjualan dan Laporan pembelian.
15. Pengujian sistem yang dilakukan menggunakan metode pengujian *Blackbox testing.*
16. Sistematika Penulisan

Agar dalam penulisan laporan ini lebih tersusun rapi, maka sistematika penulisan laporan ini dibagi menjadi beberapa bab dengan masing – masing sub babnya yang saling berkaitan. Berikut adalah penulisan masing – masing bab yang terdapat pada laporan ini:

**BAB I PENDAHULUAN**

Bab I menjelaskan tentang informasi umum yaitu latar belakang penelitian, perumusan masalah, tujuan penelitian, manfaat penelitian, dan ruang lingkup penelitian.

**BAB II TINJAUAN PUSTAKA**

Bab II berisi tentang kajian-kajian teori yang menunjang penelitian selama penelitian berlangsung dengan topik yang akan digunakan yaitu sistem informasi, basis data, sistem informasi eksekutif, *PHP*, Laravel, Ajax, Bootstrap, MariaDB, *Flowchart*, *DFD*, *ERD*, Basis data konseptual.

**BAB III METODE PENELITIAN**

Bab III ini menjelaskan tentang lokasi dan waktu penelitian, alat dan bahan penelitian, metode pengumpulan data, dan metode perekayasaan pada sistem ini.

**BAB IV HASIL DAN PEMBAHASAN**

Bab IV menjelaskan tentang analisa dan perancangan sistem yang dimulai dari perancangan basis data, *data flow diagram*, *entity relantionship diagram*, basis data konseptual, struktur tabel, perancangan antarmuka, implementasi sistem sera pengujian sistem.

**BAB V PENUTUP**

Bab V sekaligus bab terakhir menjelaskan kesimpulan dan saran dari keseluruhan penelitian yang telah dilakukan, di mana kesimpulan dan saran ini dibuat sebagai tolak ukur apabila akan dilakukan pengembangan aplikasi selanjutnya untuk sistem ini.

***Halaman ini sengaja dikosongkan***

# BAB II TINJAUAN PUSTAKA

1. *State of The Art*

Penelitian tentang sistem informasi penjualan telah dilakukan oleh para peneliti sebelumnya. Penelitian-penelitian yang terkait dengan penelitian yang akan dilakukan oleh penulis terdapat pada Tabel 2.1.

Tabel 2.1 State of The Art

| **No** | **Nama** | **Judul** | **Deskripsi** |
| --- | --- | --- | --- |
| 1 | Yazid Maulana [1] | Sistem Informasi Penjualan Kayu Ulin Pada UD Karya Perdana Berbasis *Web* Menggunakan *Framework* Codeigniter (2018). | * Sistem informasi ini dibangun pada *platform* *website* dan bisa diakses secara *online* * Dari segi pembangunan sistem informasi ini menggunakan Bahasa pemrograman *web* yaitu *PHP* dengan menggunakan *Framework* Codeigniter dan Bootstrap. * Sistem informasi ini dapat melakukan *custom* order sesuai dengan permintaan *customer*. |
| 2 | I Kadek Adi Adnyana [2] | Sistem Informasi Eksekutif Penjualan Jajanan Sagu Pada Bagia Suli Menggunakan *Framework* Laravel (2018). | * Sebuah sistem informasi eksekutif yang dapat membantu dalam kegiatan penjualan. * Sebuah sistem informasi eksekutif yang dapat menghemat waktu dalam proses pengumpulan informasi laporan penjualan yang dilakukan oleh pegawai administrasi. * *Output* pada sistem ini berupa laporan penjualan per hari, per bulan, per tahun. |
| 3 | Ketut Oka Edi Artawan [3] | Sistem Informasi Penyewaan Busana Pada Denata Kebaya Menggunakan Framwork Laravel (2019). | * Dalam pengembangan sistem ini menggunakan metode *waterfall*. * Sistem ini dibangun atas *platform* *website* sehingga dalam operasional sistem ini dapat dijalankan dengan menggunakan internet dan browser. * Hasil dari penelitian telah menghasilkan sistem yang dapat mengelola data dan informasi penyewaan busana Denata Kebaya. |
| 4 | I Putu Kawiyana 2019 [4] | Sistem Informasi Eksekutif Pengadaan Barang Pada Saturday Cloth Menggunakan *Framework* Laravel (2019). | * Sistem ini menggunakan Bahasa pemrograman *PHP* dengan *Framework* Laravel dan basis data Mysql. * Dari segi perancangan sistem informasi penyewaan ini menggunakan *DFD* dan *ERD*. * Sistem ini memiliki beberapa fitur yaitu pengelolaan barang, laporan berbentuk grafik. |

Berdasarkan referensi penelitian terdahulu yang terdapat pada *Table* 2.1, dapat disimpulkan bahwa perekayasaan yang akan dibuat memiliki kemiripan dari segi metode pengembangan sistem yaitu dengan menggunakan metode *waterfall*, dari segi perancangan sistem menggunakan *ERD* dan *DFD*, dan dari segi *platform* yang digunakan untuk mengoperasikan sistem yaitu dengan menggunakan *platform* *website*. Berdasarkan beberapa penelitian tersebut dapat digunakan sebagai acuan dalam mengembangkan sistem pada perekayasaan ini. Selain melihat dari kemiripan adapun pengembangan dari segi teknologi yang akan digunakan pada perekayasaan sistem yaitu dibangun dengan menggunakan *Framework* Laravel.

1. Balinese *Class*ic

Balinese Classic merupakan sebuah usaha yang bergerak pada penjualan kerajinan bali khususnya tas rotan. Produk dipasarkan ke berbagai negara seperti Malaysia, Filipina, Singapore dan masih banyak lagi melalui *supplier* masing-masing negara. Perusahaan yang bertempat di Jalan Kebo Iwa Perumahan Swamandala XIII No.3 Denpasar, Bali. Perusahaan ini baru merintis usahanya sejak Juli 2018 lalu. Balinese *Class*ic melakukan pembelian tas dari beberapa *supplier* di pasar tradisional Bali kemudian barang akan disimpan di gudang dan akan dipasarkan kembali ke berbagai belahan dunia. Saat ini dalam menyimpan barang, untuk sementara balinese classic menggunakan kamar sebagai media penyimpanan dan juga untuk mencatat stok masih dilakukan secara manual melalui buku besar yang bisa dilihat pada Gambar 2.1

* *

Gambar 2.1 Gudang

1. Sejarah Usaha

Awal mulanya usaha penjualan tas ini didirikan secara tidak sengaja oleh luthvi dan yogi karena sedang ada *trend* tas rotan yang sedang maraknya dicari-cari oleh wisatawan asing. Langkah awal yang dilakukan luthvi dan yogi yaitu mencari pengerajin dari tas rotan ini, setelah berhasil mendapatkan beberapa pengerajin dan membeli beberapa *stock*, Kami mencoba memasarkannya melalui intagram dengan membuat 2 akun instagram, berikut merupakan halaman instagram dari usaha kecil ini yang bisa dilihat pada Gambar 2.1 dan 2.2



Gambar 2.2 Akun *instagram* pertama



Gambar 2.3 Akun *instagram* kedua

]

Akun di atas dikelola oleh luthvi dan yogi sendiri yang bertugas juga sebagai *admin*.

1. Visi dan Misi Balinese Classic

Adapun visi dan misi dari Balinese Classic meliputi :

**Visi**

1. Selalu menjadi perusahaan tas unik yang memiliki konsep kuat dan memiliki ciri khas tersendiri dibandingkan dengan tas lain.
2. Memiliki nilai jual tinggi dengan adanya pembaruan-pembaruan yang diterapkan pada produk sesuai tren yang berjalan.
3. Mampu menciptakan produk *original* agar perusahaan dapat menciptakan rasa bangga atas kepuasan dari konsumen Balinese Classic.

**Misi**

1. Membuat produk yang berkualitas dengan selalu memantau kualitas di setiap produksinya.
2. Membuat desain baru yang belum pernah dibuat orang lain.
3. Membuat ciri khas tersendiri yang menjadi kekuatan perusahaan Balinese Classic.
4. *Research* segala jenis *mode* yang berkembang dan dikembangkan ulang dengan konsep Balinese Classic.
5. Penjualan

Saat ini Balinese Classic dalam melakukan pengecekan laporan penjualan tidak dalam bentuk grafik melainkan masih dalam bentuk tabel yang dicatat pada buku besar, dari permasalahan tersebut maka penulis membuat contoh grafik penjualan yang bisa dilihat pada Gambar 2.4 berikut:

Gambar 2.4 Grafik Penjualan

Berikut penjelasan dari grafik penjualan pada Gambar 2.4 :

1. Pada bulan Januari, penjualan tas sebanyak 300*pcs* yang meliputi tas rotan sebanyak 200*pcs* dan tas ate sebanyak 100*pcs*.
2. Pada bulan Februari, penjualan tas sebanyak 550*pcs* yang meliputi tas rotan sebanyak 300*pcs* dan tas ate sebanyak 250*pcs*.
3. Pada bulan Januari, penjualan tas sebanyak 650*pcs* yang meliputi tas rotan sebanyak 450*pcs* dan tas ate sebanyak 200*pcs*.
4. Pada bulan Januari, penjualan tas sebanyak 650*pcs* yang meliputi tas rotan sebanyak 500*pcs* dan tas ate sebanyak 150*pcs*.
5. Pada bulan Januari, penjualan tas sebanyak 320*pcs* yang meliputi tas rotan sebanyak 200*pcs* dan tas ate sebanyak 120*pcs*.

Dengan adanya data penjualan berbentuk grafik seperti Gambar 2.4, dapat memudahkan dalam memantau perkembangan perusahaan, kapan penjualan menurun/meningkat bisa dilihat dengan jelas.

1. Sistem Informasi Eksekutif (SIE)

Sistem informasi eksekutif didefinisikan sebagai sebuah sistem terkomputerisasi yang menyediakan informasi internal dan eksternal bagi para eksekutif yang relevan terhadap faktor penentu keberhasilan dengan akses yang sangat mudah. Sistem informasi eksekutif dirancang untuk membantu eksekutif mencari informasi yang diperlukan pada saat mereka membutuhkannya dan dalam bentuk apa pun yang paling bermanfaat [5].

1. Tujuan Sistem Informasi Eksekutif

Tujuan dari perancangan sistem aplikasi eksekutif adalah untuk membantu para eksekutif dalam mempelajari tentang sebuah organisasi, bagaimana proses-proses bekerja, interaksi antara lingkungan luar dan lingkungan dalam satu organisasi untuk mendukung dalam membuat laporan berdasarkan data yang akurat. Secara global sistem informasi eksekutif merupakan sistem yang digunakan oleh *top level management* dalam mengetahui perkembangan yang terjadi pada perusahaan.

Sistem Informasi Eksekutif (SIE) dikembangkan untuk bermacam-macam alasan. Kajian-kajian memperlihatkan bahwa keuntungan-keuntungan yang paling umum dari SIE adalah perbaikan dalam kualitas dan kuantitas yang ada bagi para eksekutif.

1. Lebih banyak informasi yang tepat waktu.
2. Akses ke data operasional lebih besar.
3. Akses ke *database* perusahaan lebih besar.
4. Lebih banyak informasi yang bersaing.
5. Terdapat informasi baru atau tambahan.
6. Mengurangi biaya kertas.
7. Informasi tentang lingkungan luar lebih banyak [6].
8. *Website*

*WEB* (*Word Wide Web*) atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi, teks gambar diam atau bergerak, animasi suara dana atau gabungan dari semua itu. Baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling berkaitan di mana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Untuk menyediakan keberadaan sebuah *web*site. maka harus disediakan unsur-usur penunjang seperti:

1. Nama *Domain*

Nama *domain* atau *domain* *name* atau URL (*Uniform Resource Located*) adalah alamat unik di dunia internet yang digunakan untuk mengidentifikasi sebuah *website* atau digunakan untuk menemukan sebuah *website* pada dunia internet.

1. Tempat *Website* (*Web Hosting*)

*Web* *Hosting* dapat diartikan sebagai ruangan yang terdapat dalam *hardisk* tempat penyimpanan berbagai data, *file*, gambar. dan sebagainya yang akan ditampilkan di *website*.

1. Bahasa Program (*Script Program*)

Bahasa program merupakan bahasa yang digunakan untuk menerjemahkan setiap perintah dalam wesite pada saat diakses.

1. Desain *Website*

Setelah melakukan pembelian nama *domain* dan *web* *hosting* serta penguasaan program, maka harus memiliki desain *website*. Unsur *website* yang penting adalah desain, karena desain *website* sangat menentukan kualitas dan keindahan dari suatu *website*.

1. Publikasi *website*

Keberadaan sebuah *website* tidak ada gunanya dibangun apabila tidak dikunjungi atau dikenal oleh pengguna internet. Maka untuk mengenalkan situs *website* kepada masyarakat memerlukan adanya publikasi atau promosi.

1. Pemeliharaan *website*

Untuk mendukung kelanjutan dari situs *website* diperlukan pemeliharaan setiap waktu sesuai dengan keinginan seperti penambahan konten informasi, berita, *link*, artikel, gambar maupun informasi terbaru lainnya. Tanpa pemeliharaan yang baik situs akan terkesan membosankan dan monoton, sehingga akan mulai ditinggalkan oleh pengunjung internet [7].

1. *Data Flow Diagram (DFD)*

*Data Flow Diagram* (*DFD*) adalah suatu *network* yang menggambarkan suatu Sistem komputer, manual, atau gabungan dari keduanya yang penggambarannya disusun dalam bentuk kumpulan komp*one*n sistem yang saling berhubungan sesuai dengan aturan mainnya. Keuntungan dari *DFD* adalah memungkinkan untuk menggambarkan sistem dari *level* yang paling tinggi kemudian menguraikannya menjadi *level* yang lebih rendah. Sedangkan kekurangan *DFD* adalah tidak menunjukkan proses pengulangan, proses keputusan dan proses penghitungan.

Adapun langkah - langkah dalam membuat *Data* *Flow* *Diagram* dibagi menjadi 3 (tiga) tahap atau tingkat konstruksi *DFD*, yaitu:

* + 1. Diagram Konteks

Diagram ini dibuat untuk menggambarkan sumber serta tujuan data yang akan diproses atau dengan kata lain diagram tersebut digunakan untuk menggambarkan sistem secara umum/global dari keseluruhan sistem yang ada.

* + 1. Diagram *Level* Nol

Diagram ini dibuat untuk menggambarkan tahapan yang ada di dalam diagram konteks, yang penjabarannya lebih terperinci.

* + 1. Diagram Detail

Diagram ini dibuat untuk menggambarkan arus data lebih mendetail lagi dan tahapan proses yang ada di dalam diagram *level* nol [8].

Menurut Jogiyanto HM dalam buku yang berjudul Analisis & Desain Sistem Informasi simbol yang digunakan *Data Flow Diagram* (*DFD*) yaitu:

1. *External Entity* (kesatuan luar) atau *Boundary* (batas sistem)

setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang. organisasi atau sistem lainnya yang berada dilingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.

1. *Data Flow* (arus data)

*Data* *Flow* (arus data) di DFD diberi simbol sualu anak panah. Arus data ini mengalir di antara proses (*process*), simpanan data (*data store*) dan kesatuan luar (*external* *entity*). Arus data ini menunjukkan arus dari data yang dapat berupa masukkan untuk sistem atau hasil dari proses sistem.

1. *Process* (proses)

Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Untuk *physical* *data* *flow* *diagram* (PDFD), proses dapat dilakukan oleh orang, mesin atau komputer, sedangkan untuk *logical* *data* *flow* *diagram* (L*DFD*), suatu proses hanya menunjukkan proses dari komputer. Suatu proses dapat ditunjukkan dengan simbol lingkaran atau empat persegi panjang tegak dengan sudut-sudutnya tumpul.

1. *Data store* (simpanan data)

Simpanan data (*data store*) merupakan simpanan dari data yang dapat berupa:

* Suatu *file* atau *database* di sistem komputer
* Suatu arsip atau catatan manual
* Suatu kotak tempat data di meja seseorang
* Suatu tabel acuan manual
* Suatu agenda atau buku.

Dari uraian di atas dapat disimpulkan bahwa *DFD* adalah diagram yang menggunakan notasi-notasi ini untuk menggambarkan arus dari data sistem.

Adapun beberapa simbol yang digunakan dalam *Data Flow Diagram* yang dilukiskan pada *Table* 2.2 berikut:

Tabel 2.2 Simbol-simbol DFD

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1 |  | Entitas Luar | Suatu entitas eksternal adalah objek di luar sistem yang berkomunikasi dengan sistem sebagai sumber maupun pemakai informasi |
| 2 |  | Proses | Suatu proses dalam sistem yang mengolah data masukkan menjadi informasi |
| 3 |  | Penyimpanan Data | Merupakan tempat penyimpanan data yang perlu disimpan untuk digunakan oleh satu proses atau lebih |
| 4 |  | Garis Alir | Untuk menunjukkan arah dari aliran data pada suatu proses |

1. *Entity Relationship Diagram (ERD)*

*Entity Relationship Diagram* (*ERD*) adalah suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data relasional. *ERD* juga merupakan gambaran yang menghubungkan antara objek satu dengan objek yang lain dalam dunia nyata. Sebagai contoh, jika akan membuat *ERD* dari sistem penjualan suatu perusahaan, maka bahan sebagai objek *ERD* bisa berupa barang, pembeli, *supplier*, dan lain sebagainya. Pemodelan basis data dalam bentuk *Entity Relationship Diagram* hanyalah sebagai pendekatan dan perkiraan deskripsi data pada *database* yang akan dibangun, dibangun melalui evaluasi yang sangat subyektif dari informasi yang terkumpul selama menganalisis kebutuhan sistem. Dalam pembuatan *Entity Relationship* *Diagram* perlu diperhatikan penentuan sesuatu konsep apakah merupakan *entity*, *attribute*, atau *relationship* [9] .Beberapa elemen-elemen *ERD* adalah sebagai berikut:

Tabel 2.3 Simbol-simbol ERD

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1 |  | Entitas | Jenis entitas dapat berupa suatu elemen lingkungan, sumber daya atau transaksi yang *field-field*nya dipergunakan dalam aplikasi  program |
| 2 |  | Hubungan atau Relasi | Menunjukkan nama relasi antar satu entitas dengan entitas lainnya |
| 3 |  | Atribut | Atribut adalah karakteristik dari sebuah entitas |
| 4 | C:\Users\yogi\Desktop\ERD-Symbols-Relationships-Many-to-Many-5.png | *One to One* | Digunakan untuk menghubungkan antara dua tabel yang memungkinkan satu tabel berelasi dengan satu data saja pada tabel lainnya. |
| 5 | C:\Users\yogi\Desktop\ERD-Symbols-Relationships-Many-to-One-1.png | *One to many* | Digunakan menghubungkan antara dua tabel yang memungkinkan beberapa data tabel berelasi dengan beberapa data yang ada di tabel lain. |
| 6 | C:\Users\yogi\Desktop\ERD-Symbols-Relationships-Many-to-Many-3.png | *Many to many* | Digunakan menghubungkan antara dua tabel yang memungkinkan beberapa data tabel berelasi dengan beberapa tabel lainnya. |

1. *PHP*

*PHP* singkatan dari *PHP* *Hypertext Processor* yang digunakan sebagai bahasa *script* *server-side* dalam pengembangan *web* yang disisipkan pada dokumen HTML. Penggunaan *PHP* memungkinkan *web* dapat dibuat dinamis sehingga *maintenance* situs *web* tersebut menjadi lebih mudah dan efisien. *PHP* merupakan *software* open *source* yang disebarkan dan dilisensikan secara gratis serta dapat diunduh secara bebas dari situs resminya *http://www.php.net*. *PHP* ditulis dengan menggunakan bahasa C.

*PHP* diciptakan pertama kali oleh Rasmus Lerdrof pada tahun 1994. Awalnya, *PHP* digunakan untuk mencatat jumlah serta untuk mengetahui siapa saja pengunjung *homepage*-nya. Rasmus Lerdrof adalah seorang pendukung open *source*. Oleh karena itu, ia mengeluarkan Personal *Home* *Page* *Tool*s versi 1.0 secara gratis, kemudian menambah kemampuan *PHP* 1.0 dan meluncurkan *PHP* 2.0. Pada tahun 1996, telah banyak digunakan dalam *website* di dunia. Sebuah kelompok pengembang *software* yang terdiri dari Rasmus, Zeew Suraski, Andi Gutman, Stig Bakken Shane Caraveo, dan Jim Wistead bekerja sama untuk menyempurnakan *PHP* 2.0. Akhirnya, pada tahun 1998, *PHP* 3.0 diluncurkan. Penyempurnaan terus dilakukan sehingga pada tahun 2000 dikeluarkan *PHP* 4.0. Tidak sampai disitu, kemampuan *PHP* terus ditambah, dan saat ini versi terbaru yang telah dikeluarkan adalah *PHP* 7.2.x [10].

1. MariaDB

*Database* adalah sekumpulan data yang dapat diolah atau dapat dimanipulasi menggunakan suatu perangkat lunak. *Database* merupakan aspek yang sangat penting dalam sebuah *website* karena berfungsi sebagai gudang penyimpanan data yang akan diolah lebih lanjut karena dapat mengorganisasi data, menghindari duplikasi data, menghindari hubungan antar data yang tidak jelas dan juga melakukan *update* data yang rumit.

MariaDB merupakan versi pengembangan terbuka dan mandiri dari MySQL. Sejak diakuisisinya MySQL oleh Oracle pada September 2010, Monty Program sebagai penulis awal kode sumber MySQL memisahkan diri dari pengembangan dan membuat versi yang lebih mandiri yakni MariaDB [11].



Gambar 2.5 MariaDB

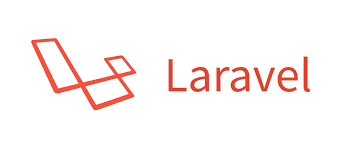
Semua kemampuan MySQL dimiliki pula oleh MariaDB yakni:

1. Portabilitas. MariaDB dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MariaDB didistribusikan sebagai perangkat lunak sumber terbuka, di bawah lisensi GPL sehingga dapat digunakan secara gratis.
3. *Multi-user*. MariaDB dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*, MariaDB memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MariaDB memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan fungsi. MariaDB memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan. MariaDB memiliki beberapa lapisan keamanan seperti *level* subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MariaDB mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. K*one*ktivitas. MariaDB dapat melakukan k*one*ksi dengan klien menggunakan protokol TCP/IP, *Unix soket* (UNIX), atau *Named Pipes* (NT).
10. Pelokalan bahasa. MariaDB dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Ind*one*sia belum termasuk di dalamnya.
11. Antarmuka. MariaDB memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. Klien dan peralatan. MariaDB dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
13. Struktur tabel. MariaDB memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam *PostgreSQL* ataupun *Oracle*.
14. Laravel

Laravel adalah sebuah *Framework* *PHP* yang dirilis di bawah lisensi MIT, dibangun dengan konsep MVC (*model* *view* *controller*). Laravel adalah pengembangan *website* berbasis MVP yang ditulis dalam *PHP* yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu [12].

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komp*one*n- komp*one*n aplikasi, seperti: manipulasi data, *controller*, dan *user interface*.

1. *Model*, *Model* mewakili struktur data. Biasanya *model* berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
2. *View*, *View* adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman *web*.
3. *Controller*, *Controller* merupakan bagian yang menjembatani *model* dan *view*.



Gambar 2.6 Laravel

Beberapa fitur yang terdapat di Laravel:

1. *Bundles*, yaitu sebuah fitur dengan sistem pengemasan modular dan tersedia beragam di aplikasi.
2. *Eloquent ORM*, merupakan penerapan *PHP* lanjutan menyediakan metode internal dari pola “*active record*” yang menagatasi masalah pada hubungan objek *database*.
3. *Application Logic*, merupakan bagian dari aplikasi, menggunakan *controller* atau bagian *Route*.
4. *Reverse Routing*, mendefinisikan relasi atau hubungan antara *Link* dan *Route*.
5. *Restful controllers*, memisahkan logika dalam melayani *HTTP* *GET* and *POST*.
6. *Class Auto Loading*, menyediakan *loading* otomatis untuk *class* *PHP*.
7. *View* *Composer*, adalah kode unit logikal yang dapat dieksekusi ketika *view* sedang *loading*.
8. *IoC Container*, memungkin obyek baru dihasilkan dengan pembalikan *controller*.
9. *Migration*, menyediakan sistem kontrol untuk skema *database*.
10. *Unit Testing*, banyak tes untuk mendeteksi dan mencegah regresi.
11. *Automatic Pagination*, menyederhanakan tugas dari penerapan halaman.
12. Bootstrap

Bootstrap adalah *open-source front-end toolkit* yang dikembangkan oleh Mark Otto dan Jacob Thornton dari perusahaan yang berkecimpung dalam bidang sosial media yaitu Twitter, dirilis pada Agustus 2011. Bootstrap dikembangkan dengan tujuan untuk membantu desainer dan pengembang dalam membangun *front-end* sebuah *website* secara cepat dan efisien. Twitter Bootstrap secara cepat menjadi proyek yang paling dilihat pada GitHub dengan lebih dari 33.000 pengguna GitHub [13].



Gambar 2.7 Bootstrap

*Framework* Bootstrap membagi ukuran resolusi layar ke dalam 4 kategori, yaitu:

* + 1. *Large Devices* (dengan ukuran resolusi layar 1200px).

1. *Medium Devices* (dengan ukuran resolusilayar 992px).
2. *SmallDevices* (dengan ukuran resolusilayar 768px).
3. *Extra Small Devices* (dengan ukuran resolusi layar < 768px)
4. *Blackbox Testing*

Pengujian disisi pengembang dilakukan oleh *programmer* dari aplikasi bersangkutan atau mereka yang mengerti dan terlibat di dalam pengembangan sistem tersebut. Pengujian disisi pengembang ini umum disebut dengan *Blackbox* *Testing*, penguji ini menekankan pada sejauh mana fungsionalitas sistem informasi berjalan dengan baik sesuai dengan yang diinginkan oleh pengguna dan

Pengembangan [14]. Terdapat empat buah jenis pengujian pada pengujian disisi pengembangan (*Blackbox*) ini antara lain:

1. Pengujian *Interface* (Antarmuka) Aplikasi

Pengujian *interface* (antarmuka) aplikasi sistem informasi bertujuan untuk mengetahui fungsionalitas dari setiap elemen *interface* yang ada di setiap halaman aplikasi sistem informasi di mana setiap elemen yang ditampilkan harus sesuai dengan fungsionalitas yang dimiliki.

1. Pengujian Fungsi Dasar Sistem

Pengujian fungsi dasar sistem bertujuan untuk mengetahui sejauh mana kinerja dari setiap fungsi dasar sistem yang ada di dalam aplikasi sistem informasi.

1. Pengujian *Form* *Handle* Sistem

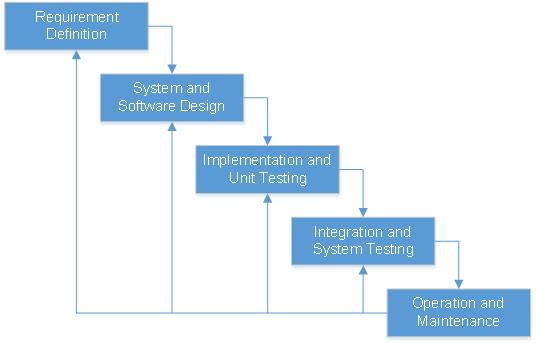
Pengujian *Form* *Handle* Sistem bertujuan untuk mengetahui seperti apa dan sejauh mana respons oleh sistem informasi terhadap masukan yang diberikan oleh pengguna di mana apabila terjadi kesalahan, sistem harus mampu memberikan respons agar pengguna dapat memperbaiki kesalahan yang dilakukan.

1. Pengujian Keamanan Sistem

Pengujian Keamanan Sistem bertujuan untuk mengetahui sejauh mana tingkat keamanan yang dimiliki oleh sistem informasi untuk dapat memberikan kenyamanan kepada para pengguna.

1. Model *Waterfall*

Prosedur pengembangan yang digunakan pada sistem informasi penjualan ini adalah menggunakan prosedur pengembangan model *Waterfall*. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, testing *I* *verification*, dan *maintenance*. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Sebagai contoh tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap *requirement*. Secara umum tahapan pada model *waterfall* dapat dilihat pada Gambar 2.5 berikut:



Gambar 2.8 Model *Waterfall*

Gambar di atas adalah tahapan umum dari model proses ini. Akan tetapi Pressman memecah model ini menjadi 6 tahapan meskipun secara garis besar sama dengan tahapan-tahapan model *waterfall* pada umumnya. Berikut adalah penjelasan dari tahap-tahap yang dilakukan di dalam model ini menurut Pressman:

1. *Requirements definition*

Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software*. Untuk mengetahui sifat dari program yang akan dibuat, maka para *software* *engineer* harus mengerti tentang domain informasi dari *software* misalnya fungsi yang dibutuhkan, *user* *interface*. Dari 2 aktivitas tersebut (pencarian kebutuhan sistem dan *software*) harus didokumentasikan dan ditunjukkan kepada pelanggan.

1. *Sistem and Software Design*

Proses ini digunakan untuk mengubah kebutuhankebutuhan diatas menjadi representasi ke dalam bentuk "*blueprint*" *software* sebelum *coding* dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti 2 aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*.

1. *Implementation and Unit Testing*

Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap *design* yang secara teknis nantinya dikerjakan oleh *programmer*.

1. *Integration and Sytem Testing*

Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan *software*. Semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari *error*, dan hasilnya harus benar - benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

1. *Operation and Maintenance*

Pemeliharaan suatu *software* diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada *error* kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada *software* tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya [15].

# BAB III METODE PENELITIAN

Metode penelitian yang digunakan dalam pembuatan “Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan pada Balinese Classic Menggunakan *Framework* Laravel” adalah sebagai berikut:

1. Waktu dan Lokasi Penelitian

Penelitian ini dilakukan pada bulan Mei 2019 sampai Agustus 2019. Lokasi penelitian di Jalan Kebo Iwa Perumahan Swamandala XIII No.3 Denpasar, Bali. Lokasi ini merupakan kantor dan gudang dari Balinese Classic.

1. Alat dan bahan Penelitian

Alat dan bahan penelitian yang digunakan dalam mengumpulkan informasi di Balinese Classic menggunakan alat penelitian berupa perangkat keras dan perangkat lunak yaitu:

1. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Laptop Asus A45a
2. Ram 4GB
3. *Processor* intel insidetm Core mi3
4. *Hardisk* 500GB
5. Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Windows 10
2. Visual Studio Code
3. Mozila Firefox
4. Google Chrome
5. Xampp
6. Ms Word
7. Visual Paradigm
8. Metode Pengumpulan Data

Pada perekayasaan ini digunakan teknik pengumpulan data sebagai berikut:

1. *Observasi*

Melakukan pengamatan langsung pada Balinese Classic dengan menggunakan alat penunjang yaitu buku, pulpen, dan lain-lain. Melihat langsung mengenai kegiatan operasional perusahaan seperti kegiatan transaksi, proses pencatatan hingga pembelian barang. Serta melihat studi kasus apa yang sering terjadi dalam pengambilan keputusan di perusahaan tersebut.

1. Studi Literatur

Pada tahap ini dilakukan pengumpulan data dan informasi dengan cara membaca buku-buku dan artikel ilmiah yang berhubungan dengan objek perekayasaan, seperti: *website*, *PHP*, *Framework* Laravel dan lainnya.

1. Wawancara

Wawancara merupakan teknik pengumpulan data yang dilakukan melalui tatap muka dan Tanya jawab langsung antara peneliti terhadap narasumber atau sumber data. Dalam tahap ini, penulis melakukan wawancara dengan anggota dari Balinese Classic mengenai apa saja kesulitan dan kebutuhan yang diperlukan sehingga dapat dijadikan pedoman untuk membuat sistem ini nantinya.

1. Metode Perekayasaan

Metode perekayasaan yang digunakan pada penelitian ini adalah sebagai berikut:

1. Analisis Sistem

Setelah informasi dikumpulkan maka dilakukan sebuah analisa kebutuhan terhadap sistem yang akan dibangun. Pada tahap ini akan dilakukan klasifikasi tipe pengguna dan batasannya, alur kerja sistem, kebutuhan penyimpanan dan desain yang disusun secara sistematis, selain itu juga akan dilakukan hipotesis untuk menentukan jawaban sementara terhadap rumusan masalah pada perekayasaan.

1. Perancangan Sistem

Tahap perancangan merupakan tahap untuk menciptakan sesuatu konsep

kerja terpadu antara manusia dan mesin sehingga dapat menghasilkan suatu informasi yang akurat. Perancangan sistem akan dibuat berdasarkan dari analisa kebutuhan sehingga dapat dirancang *DFD*, *ERD*, dan Konseptual *database* yang sesuai.

1. Pembuatan Program

Proses pembuatan program berbasis *web* ini dari sebuah desain ke dalam program menggunakan Bahasa pemrograman *PHP* menggunakan *Framework* Laravel. Perangkat Lunak *code editor* yang akan digunakan yaitu Visual Code Studio serta pengolahan *database* menggunakan *Php*MyAdmin.

1. Pengujian Sistem

Pengujian akan dilakukan dengan menggunakan metode pengujian *Black Box Testing*, Yaitu dengan pengujian hasil eksekusi dari semua fungsi-fungsi yang digunakan seperti tombol, *form*, dan lainnya. Jika nantinya hasil eksekusi fungsi-fungsi tersebut tidak sesuai dengan perencanaan awal, maka akan langsung diperbaharui sehingga memberikan hasil yang akurat.

1. Penulisan Laporan

Tahap penulisan laporan merupakan tahap terakhir dari proses penelitian terhadap pembuatan sebuah perangkat lunak. Pada tahap ini, hasil dari analisa, desain sistem, pembuatan program dan pengujian akan dirangkum ke dalam sebuah laporan berbentuk dokumen. Dalam dokumen ini nantinya akan tercatat semua aktivitas penelitian yang dilakukan untuk membangun aplikasi ini.

***Halaman ini sengaja dikosongkan***

# BAB IV HASIL DAN PEMBAHASAN

1. Hasil Analisis

Analisis sistem merupakan proses penganalisisan sebuah sistem yang bertujuan untuk menganalisa atau merancang suatu sistem. Tahapan ini merupakan tahapan yang terpenting karena jika terjadi kesalahan pada tahap ini, maka akan menyebabkan kesalahan pada tahapan berikutnya.

Adapun analisis kebutuhan sistem terdiri dari:

1. Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja yang dilakukan oleh sistem. Analisa kebutuhan fungsional terdiri dari:

1. Sistem dapat menyediakan *login* untuk admin
   1. Admin dapat *login* ke dalam halaman sistem.
   2. Admin dapat mengelola dan mengakses data dari aplikasi meliputi data *users*, data *customers*, data *suppliers*, data *categories*, data *products*, data *stock*, data penjualan, data pembelian, dan data laporan.
   3. Admin dapat menambah semua data ke dalam sistem, seperti tambah data *users*, tambah data *customers*, tambah data *suppliers*, tambah data *categories*, tambah data *products*, tambah data *stock*, tambah data penjualan, tambah data pembelian.
   4. Admin dapat meng*edit* semua data di dalam sistem, seperti *edit* data *users*, *edit* data *customers*, *edit* data *suppliers*, *edit* data *categories*, *edit* data *products*, *edit* data *stock*, *edit* data penjualan, *edit* data pembelian.
   5. Admin dapat melihat laporan dan mengekspor data di dalam sistem.
2. Sistem dapat menyediakan *login* untuk kasir
   1. Kasir dapat *login* ke dalam halaman sistem.
   2. Kasir dapat mengelola dan mengakses data dari aplikasi meliputi data *customers*, data *categories*, data *products*, data penjualan.
   3. Kasir dapat menambah data ke dalam sistem, seperti tambah data *customers*, tambah data *categories*, tambah data *products*, tambah data penjualan.
   4. Kasir dapat meng*edit* data di dalam sistem, seperti *edit* data *customers*, *edit* data *categories*, *edit* data *products*, *edit* data penjualan.
3. Sistem dapat menyediakan *login* untuk Gudang
   1. Gudang dapat *login* ke dalam halaman sistem.
   2. Gudang dapat mengelola dan mengakses data dari aplikasi meliputi data *suppliers*, data *categories*, data *products*, data *stock* dan data pembelian.
   3. Gudang dapat menambah data ke dalam sistem, seperti tambah data *suppliers*, data *categories*, data *products*, data *stock* dan data pembelian.
   4. Gudang dapat meng*edit* data di dalam sistem, seperti *edit* data *suppliers*, data *categories*, data *products*, data *stock* dan data pembelian.
4. Kebutuhan Non Fungsional

Kebutuhan non fungsional ini merupakan kebutuhan yang memenuhi dalam pembuatan sistem. Adapun kebutuhan non fungsional pada sistem yaitu:

1. Perangkat keras(*hardware*)

Untuk merancang dan membuat sistem informasi dibutuhkan perangkat keras agar sistem yang dibuat dapat berjalan dengan baik. Spesifikasi *hardware* yang digunakan adalah sebagai berikut:

1. Laptop : Asus X441U Series
2. *Processor* : Intel® Core™ i3-6006U CPU @ 2.00GHz 1.99 GHz
3. *Memory* : 4.00 GB
4. Sistem *Type* : 64-bit Operating Sistem
5. Perangkat Lunak(*Software*)

Untuk mendukung dan merancang pembuatan sistem dibutuhkan perangkat lunak agar sistem yang dibuat dapat berjalan dengan baik. Perangkat lunak yang digunakan adalah sebagai berikut:

1. Sistem operasi : Windows 10
2. Bahasa Pemrograman : PHP, Ajax
3. DBMS : MariaDB
4. *Framework* : Laravel, Bootstrap
5. Code *Edit*or : Visual Code Studio
6. Analisis *User*

Dari hasil analisa, terdapat 3 (tiga) tipe *user* yang dapat mengakses sistem. Berikut adalah penjabaran *user* pada *table* 4.1 analisa *users*:

Tabel 4.1 Analisa *User*

|  |  |  |
| --- | --- | --- |
| No | *Users* | Admin |
| 1 | Admin | Admin adalah tipe *user* yang memiliki otoritas paling tinggi, dapat mengelola data dan informasi yang terdapat pada sistem yaitu mengelola data *user*, data-data transaksi, data kategori, data produk, data stok, data *customer*, data *supplier* dan melihat laporan. |
| 2 | Kasir | Kasir adalah tipe *user* yang dapat mengelola data pada bagian kasir, seperti data transaksi jual, data kategori, data produk, dan data *customer*. |
| 3 | Gudang | Gudang merupakan tipe *user* yang dapat mengelola data pada bagian Gudang, seperti data transaksi beli, data kategori, data produk, data stok, dan data *supplier*. |

Berikut merupakan *mapping* dari aplikasi ini pada *table* 4.2:

Tabel 4.2 *Mapping* Hak Akses

|  |  |  |  |
| --- | --- | --- | --- |
| **Proses** | **Admin** | **Kasir** | **Gudang** |
| *Dashboard* | **V** | **X** | **X** |
| *User* | **V** | **X** | **X** |
| Penjualan | **V** | **V** | **X** |
| Pembelian | **V** | **X** | **V** |
| Kategori | **V** | **V** | **V** |
| *Stock* | **V** | **X** | **V** |
| Produk | **V** | **V** | **V** |
| *Customer* | **V** | **V** | **X** |
| *Supplier* | **V** | **X** | **V** |
| Laporan | **V** | **X** | **X** |

1. Analisa Data

Pada sistem yang akan dibangun dapat diidentifikasi beberapa data yang diperlukan agar sistem dapat dijalankan sesuai dengan yang direncanakan. Hasil Analisa data ini selanjutnya akan dijadikan acuan dalam pembuatan data *store* pada perancangan sistem dan *table* pada basis data. Adapun data yang dimaksud dapat dilihat pada *table* 4.3 di bawah ini:

Tabel 4.3 Analisa Data

| No | Data | Keterangan |
| --- | --- | --- |
| 1 | *User* | Adalah data yang berisikan informasi mengenai data pengguna dan data yang digunakan untuk *login* ke sistem. |
| 2 | Penjualan | Adalah data yang berisikan informasi mengenai transaksi penjualan |
| 3 | Penjualan Produk | Adalah data yang berisikan informasi produk dari transaksi penjualan |
| 4 | Pembelian | Adalah data yang berisikan informasi mengenai transaksi pembelian |
| 5 | Pembelian Produk | Adalah data yang berisikan informasi produk dari transaksi pembelian |
| 6 | Kategori | Adalah data yang berisikan informasi kategori dari produk |
| 7 | Produk | Adalah data yang berisikan informasi mengenai produk |
| 8 | Stok | Adalah data yang berisikan jumlah stok dari produk |
| 9 | *Customer* | Adalah data yang berisikan informasi pelanggan |
| 10 | *Supplier* | Adalah data yang berisikan informasi pemasok |

1. Analisis Proses

Terdapat 4 proses dalam sistem ini. Proses ini merupakan gambaran umum dari fungsi-fungsi yang dimiliki sistem dalam pengelolaan terhadap suatu data yang ada. Pada Tabel 4.4 berikut akan dijelaskan proses yang ada:

Tabel 4.4 Analisa Proses

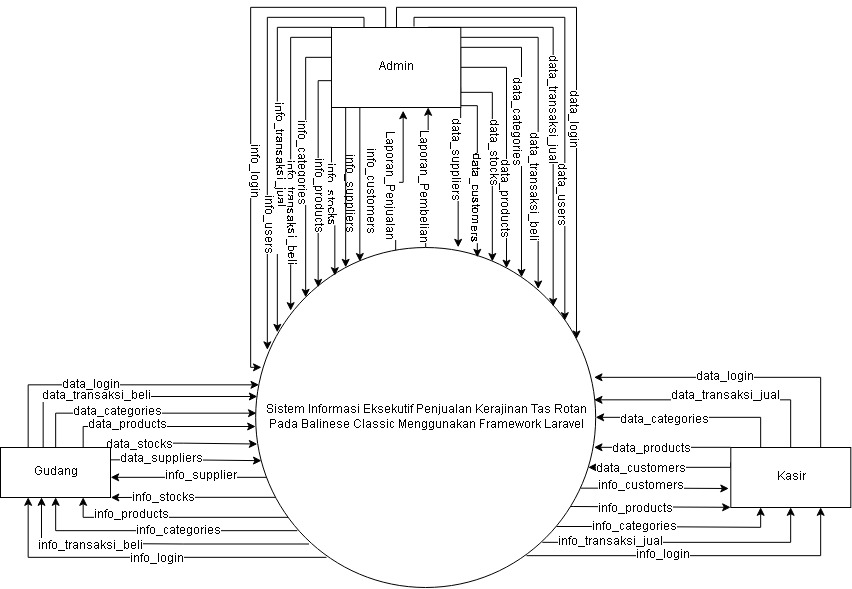
|  |  |  |
| --- | --- | --- |
| No | Proses | Keterangan |
| 1 | *Login* | Merupakan proses awal yang harus dilakukan oleh semua *Users* agar dapat mengoperasikan sistem. *Login* *user* dibedakan berdasarkan level guna memberikan hak akses yang berbeda kepada setiap pengguna aplikasi ini |
| 2 | *Users* | Merupakan proses pengolahan data yang dilakukan pada data *user*, data *customer*, data *supplier* |
| 3 | Transaksi | Merupakan proses pengolahan data utama dalam sistem ini. Proses yang dilakukan adalah mengelola data transaksi penjualan dan transaksi pembelian |
| 4 | Produk | Merupakan proses pengolahan data-data produk seperti data kategori, data produk dan data stok produk |
| 5 | Laporan | Merupakan proses mencetak laporan – laporan data transaksi |

1. Perancangan Sistem

Pada tahapan perancangan sistem ini digunakan Data *Flow* Diagram (DFD), untuk menggambarkan aliran data yang terjadi dalam proses sistem. Perancangan sistem ini memiliki beberapa tahap, dimulai dari diagram konteks, DFD level 0 dan DFD level 1 yang lebih detail. Berikut ini adalah hasil perancangan sistem menggunakan DFD:

1. Diagram Konteks

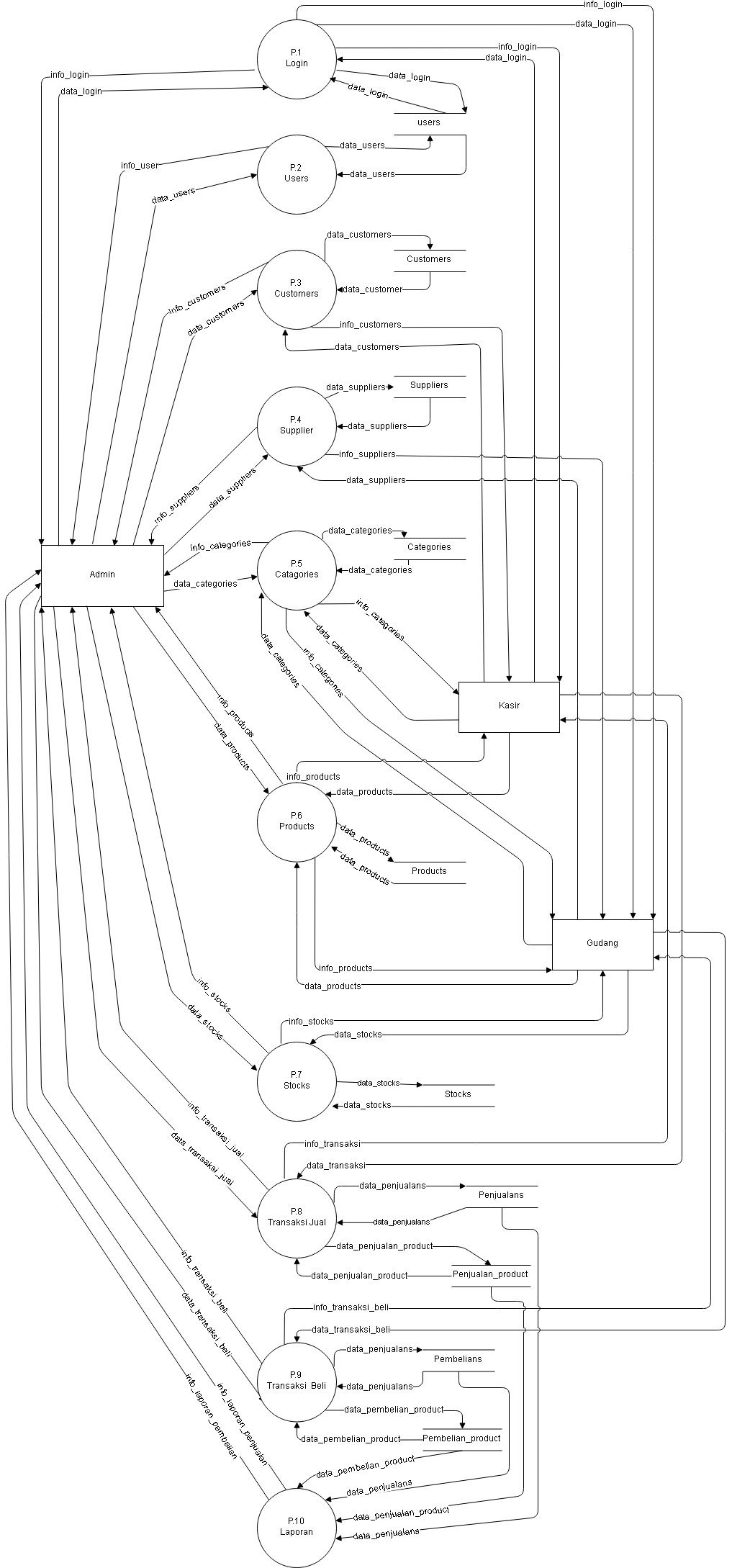
Diagram Konteks merupakan gambaran secara umum bagaimana sistem ini bekerja. Gambar 4.1 di bawah ini merupakan diagram konteks dari Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel yang merupakan dasar yang digunakan dalam penyusunan sistem ke level berikutnya. Terdapat 3 *user* yang dapat mengakses sistem ini yaitu, Admin, Kasir dan Gudang.



Gambar 4.1 Diagram Konteks

1. Data *Flow* Diagram (DFD) Level 0

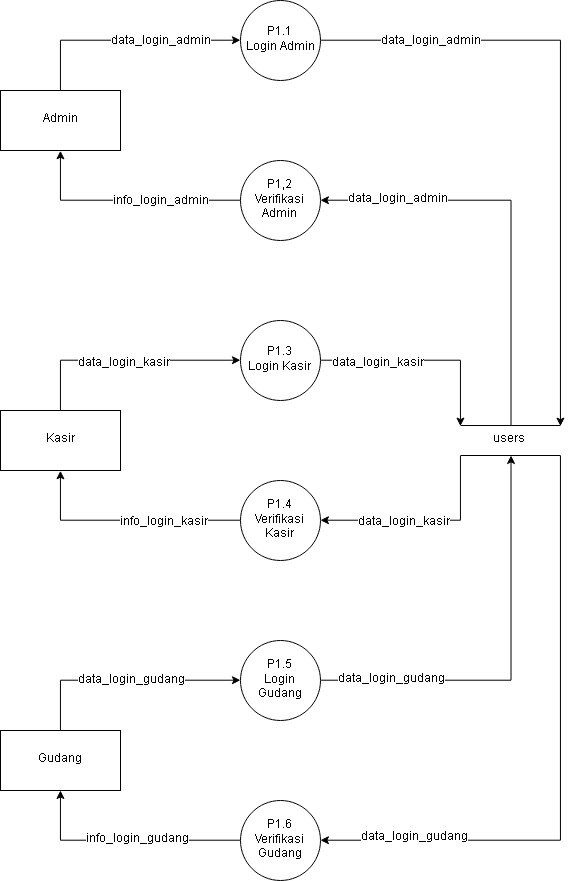
Data *Flow* Diagram Level 0 menggambarkan arus data yang lebih jelas dan detail yang dikelompokkan ke dalam proses\_proses yang lebih kecil beserta dengan data *store* yang terlibat. Beikut ini adalah DFD Level 0, dapat dilihat pada Gambar 4.2:



Gambar 4.2 Data *Flow* Diagram (DFD) Level 0

1. Data *Flow* Diagram (DFD) Level 1 Proses 1 *Login*

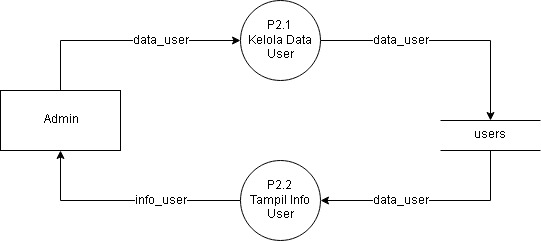
DFD Level 1 *Login* menggambarkan arus data yang lebih jelas dan detail di dalam sistem pada proses *login*. Proses 1.1(*Login* Admin) merupakan proses data *login* admin yang diverifikasi ke *database* *users*. Proses 1.3(*Login* Kasir) merupakan proses data *login* pegawai kasir yang diverifikasi ke *database* *users*. Proses 1.5(*Login* Gudang) merupakan proses data *login* pegawai Gudang yang diverifikasi ke *database* *users*. Berikut adalah gambar 4. Data *Flow* Diagram (DFD) Level 1 Proses *Login*:



Gambar 4.3 Data *Flow* Diagram (DFD) Level 1 Proses 1 *Login*

1. Data *Flow* Diagram (DFD) Level 1 Proses 2 *Users*

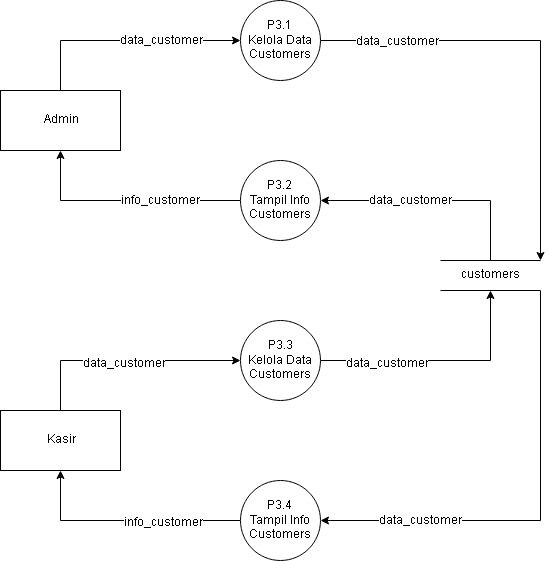
Data *Flow* Diagram (DFD) Level 1 Proses *Users* menggambarkan arus data pada proses kelola data *users*. Proses kelola data *users* ini dilakukan oleh admin saja. Berikut ini adalah DFD Level 1 Proses *Users* dari sistem ini:



Gambar 4.4 Data *Flow* Diagram (DFD) Level 1 Proses 2 *Users*

1. Data *Flow* Diagram (DFD) Level 1 Proses 3 *Customers*

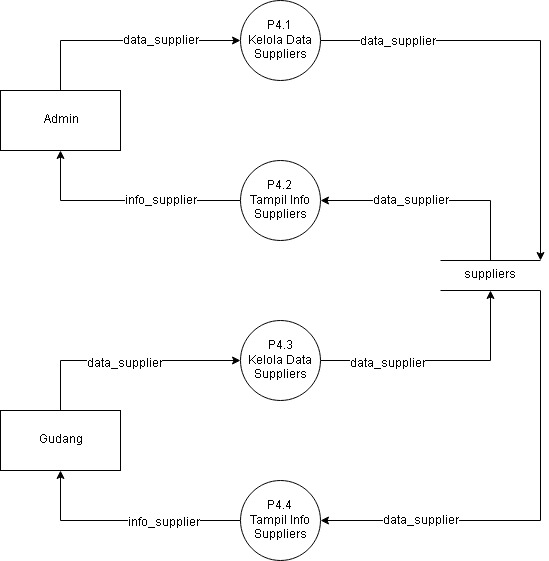
Data *Flow* Diagram (DFD) Level 1 Proses 3 *Customers* menggambarkan arus data pada proses kelola data *customers*. Proses kelola data *customers* ini dilakukan oleh admin dan kasir. Berikut ini adalah DFD Level 1 Proses *Customers* dari sistem ini:



Gambar 4. 5 Data *Flow* Diagram (DFD) Level 1 Proses 3 *Customers*

1. Data *Flow* Diagram (DFD) Level 1 Proses 4 *Suppliers*

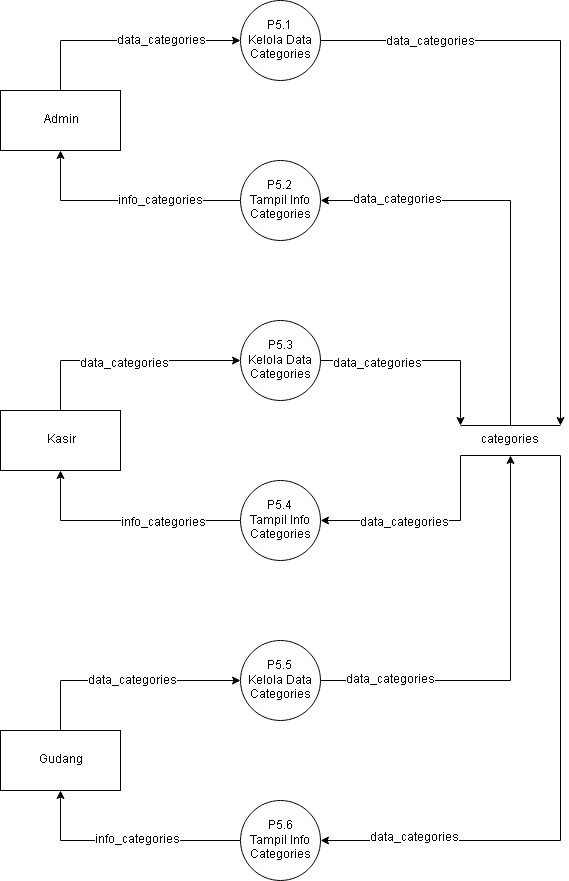
Data *Flow* Diagram (DFD) Level 1 Proses 4 *Suppliers* menggambarkan arus data pada proses kelola data *customers*. Proses kelola data *suppliers* ini dilakukan oleh admin dan gudang. Berikut ini adalah DFD Level 1 Proses *Suppliers* dari sistem ini:



Gambar 4.6 Data *Flow* Diagram (DFD) Level 1 Proses 4 *Suppliers*

1. Data *Flow* Diagram (DFD) Level 1 Proses 5 *Categories*

Data *Flow* Diagram (DFD) Level 1 Proses 5 *Categories* menggambarkan arus data pada proses kelola data *categories*. Proses kelola data *categories* ini dilakukan oleh admin, kasir dan gudang. Berikut ini adalah DFD Level 1 Proses 5 *Categories* dari sistem ini:



Gambar 4.7 Data *Flow* Diagram (DFD) Level 1 Proses 5 *Categories*

1. Data *Flow* Diagram (DFD) Level 1 Proses 6 *Products*

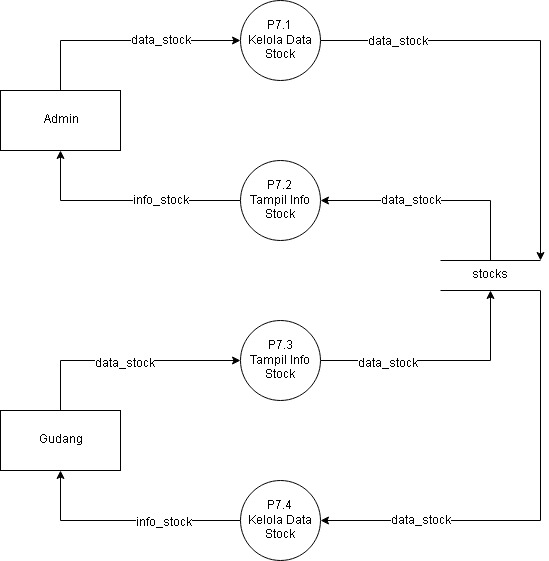
Data *Flow* Diagram (DFD) Level 1 Proses 6 *Products* menggambarkan arus data pada proses kelola data *product*. Proses kelola data *products* ini dilakukan oleh admin, kasir dan gudang. Berikut ini adalah DFD Level 1 Proses 6 *Products* dari sistem ini:



Gambar 4.8 Data *Flow* Diagram (DFD) Level 1 Proses 6 *Product*

1. Data *Flow* Diagram (DFD) Level 1 Proses 7 *Stock*s

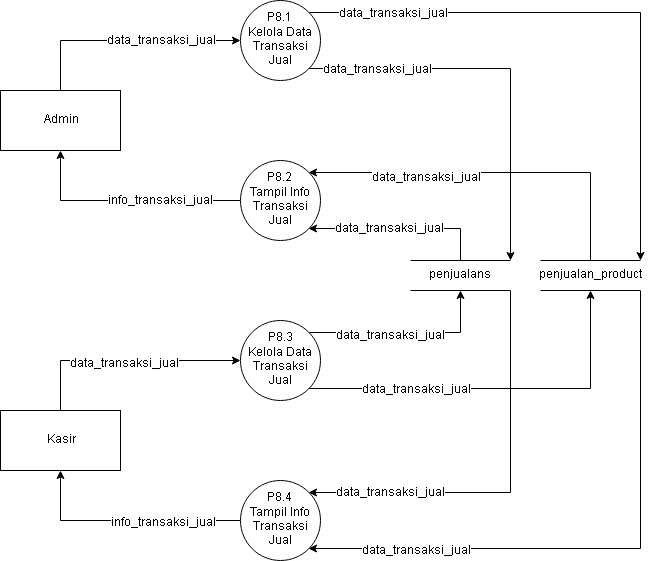
Data *Flow* Diagram (DFD) Level 1 Proses 7 *Stock*s menggambarkan arus data pada proses kelola data *stock*s. Proses kelola data *stock* ini dilakukan oleh admin dan gudang. Berikut ini adalah DFD Level 1 Proses 7 *Stock*s dari sistem ini:



Gambar 4.9 Data *Flow* Diagram (DFD) Level 1 Proses 7 *Stock*

1. Data *Flow* Diagram (DFD) Level 1 Proses 8 Transaksi Jual

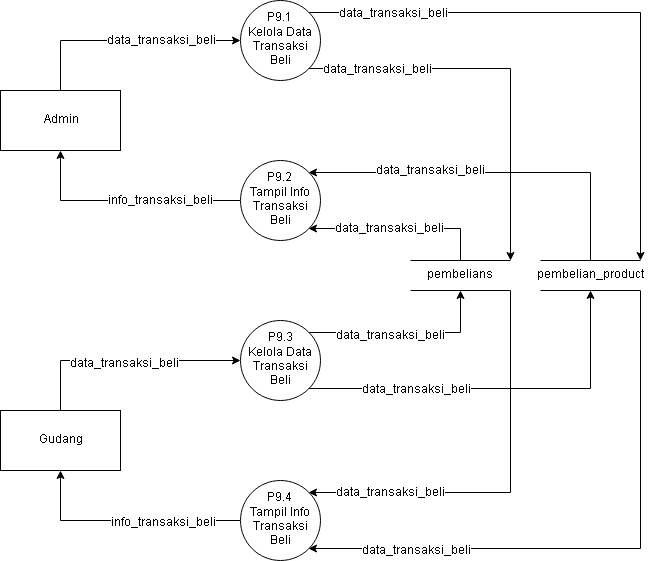
Data *Flow* Diagram (DFD) Level 1 Proses 8 Transaksi Jual menggambarkan arus data pada proses kelola data transaksi jual. Proses kelola data transaksi jual ini dilakukan oleh admin dan kasir. Berikut ini adalah DFD Level 1 Proses 8 Transaksi Jual dari sistem ini:



Gambar 4.10 Data *Flow* Diagram (DFD) Level 1 Proses 8 Transaksi Jual

1. Data *Flow* Diagram (DFD) Level 1 Proses 9 Transaksi Beli

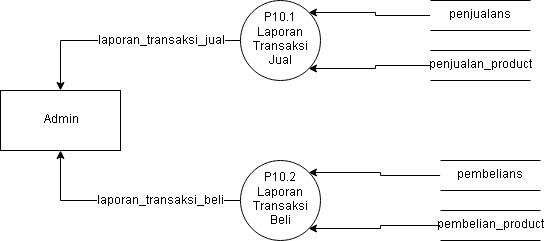
Data *Flow* Diagram (DFD) Level 1 Proses 9 Transaksi Beli menggambarkan arus data pada proses kelola data transaksi beli. Proses kelola data transaksi beli ini dilakukan oleh admin dan gudang. Berikut ini adalah DFD Level 1 Proses 9 Transaksi Beli dari sistem ini:



Gambar 4.11 Data *Flow* Diagram (DFD) Level 1 Proses 9 Transaksi Beli

1. Data *Flow* Diagram (DFD) Level 1 Proses 10 Laporan

Data *Flow* Diagram (DFD) Level 1 Proses 10. Laporan. Proses 10.1 (laporan transaksi jual) dan Proses 10.2 (laporan transaksi beli) merupakan gambaran data-data yang sudah masuk dari transaksi yang telah dilakukan oleh sistem ini. Berikut ini adalah gambar 4. DFD Level 1 Proses 10 Laporan:



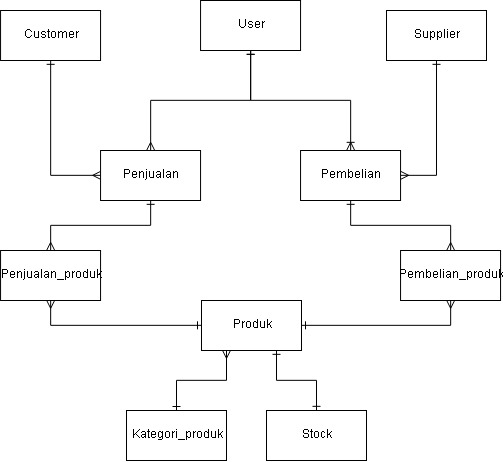
Gambar 4.12 Data *Flow* Diagram (DFD) Level 1 Proses 10 Laporan

1. Perancangan Basis Data

Pada tahapan ini dilakukan perancangan basis data dari Sistem Informasi Eksekutif Penjualan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel yang terdiri dari *Entity Relationship* Diagram (ERD), Basis Data Konseptual dan Struktur Tabel, di antaranya sebagai berikut:

1. *Entity Relationship* Diagram

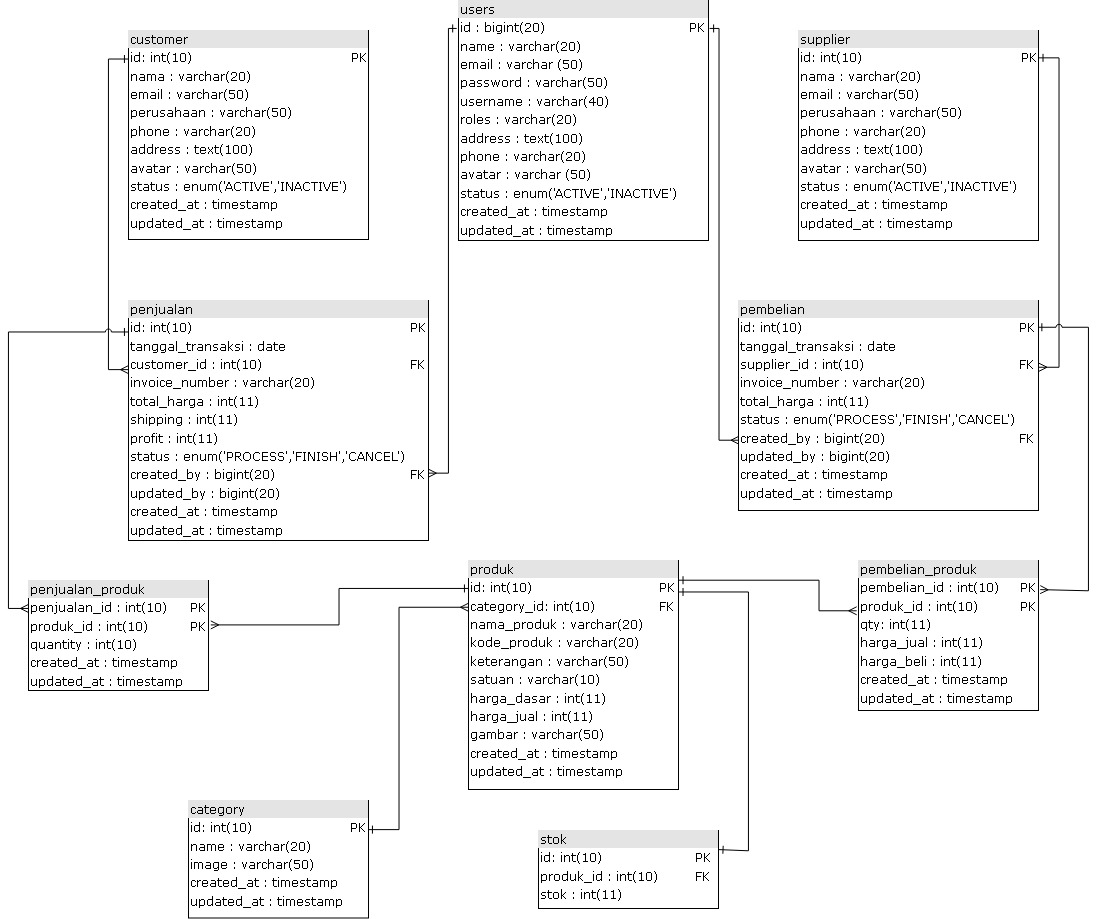
*Entity Relationship* Diagram adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan pada diagram DFD. Pada perancangan ini terdapat ERD (*Entity Relationship* Diagram) yang berisikan 10 buah *table*. *User* memiliki relasi *one* *to many* dengan penjualan dan pembelian, *customer* memiliki relasi *one* *to many* dengan penjualan, *supplier* memiliki relasi *one* *to many* dengan pembelian, penjualan memiliki relasi *one* *to many* dengan penjualan\_produk, pembelian memiliki relasi *one* *to many* dengan pembelian\_produk, penjualan\_produk memiliki relasi *many to* *one* dengan produk, pembelian\_produk memiliki relasi *one* *to many* dengan produk, produk memiliki relasi *many to* *one* dengan kategori\_produk, produk memiliki relasi *one* to *one* dengan *stock*. Berikut gambar 4. Merupakan ERD dari sistem yang dibangun:



Gambar 4.13 *Entity Relationship* Diagram

1. Basis Data Konseptual

Basis Data Konseptual pada Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel ini terdiri dari 7 tabel seperti pada Gambar 4.14 Berikut:



Gambar 4.14 Basis Data Konseptual

1. Struktur Tabel

Pada penyimpanan basis data diperlukan kumpulan *tabel* - tabel yang digunakan untuk menyimpan data. Data yang dikelola akan disimpan pada *field*-*field* terkait sesuai dengan *table* yang digunakan. Berikut ini adalah struktut *table* dari sistem yang dibuat.

1. Struktur Tabel *User*

Digunakan untuk menyimpan seluruh data *user* pada Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese *Class*ic Menggunakan *Framework* Laravel. Berikut struktur *table* *user* pada Tabel 4.5.

Tabel 4.5 Struktur Tabel *Users*

| **No** | **Nama Atribut** | **Tipe Data** | **Ukuran** | ***Key Constraint*** |
| --- | --- | --- | --- | --- |
| 1 | *id* | *bigint* | 20 | *Primary Key* |
| 2 | *name* | *varchar* | 20 |  |
| 4 | *password* | *varchar* | 50 |  |
| 5 | *username* | *varchar* | 40 |  |
| 6 | *roles* | *varchar* | 20 |  |
| 7 | *address* | *text* | 100 |  |
| 8 | *phone* | *varchar* | 20 |  |
| 9 | *avatar* | *varchar* | 50 |  |
| 10 | status | *enum* | ACTIVE,INACTIVE |  |
| 11 | *created\_at* | timestamp |  |  |
| 12 | *updated\_at* | timestamp |  |  |

1. Struktur Tabel *Customer*

Digunakan untuk menyimpan seluruh data *customer* pada Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel. Berikut struktur *table* *customer* pada Tabel 4.6.

Tabel 4.6 Struktur Tabel *Customers*

| **No** | **Nama Atribut** | **Tipe Data** | **Ukuran** | ***Key Constraint*** |
| --- | --- | --- | --- | --- |
| 1 | *id* | *int* | 10 | *Primary Key* |
| 2 | nama | *varchar* | 20 |  |
| 3 | email | *varchar* | 50 |  |
| 4 | perusahaan | *varchar* | 50 |  |
| 5 | *phone* | *varchar* | 20 |  |
| 6 | *address* | *text* | 100 |  |
| 7 | *avatar* | *varchar* | 50 |  |
| 8 | status | *enum* | ACTIVE,INACTIVE |  |
| 9 | *created\_at* | timestamp |  |  |
| 10 | *updated\_at* | timestamp |  |  |

1. Struktur Tabel *Supplier*

Digunakan untuk menyimpan seluruh data *supplier* pada Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel. Berikut struktur *table* *supplier* pada Tabel 4.7.

Tabel 4.7 Struktur Tabel *Suppliers*

| **No** | **Nama Atribut** | **Tipe Data** | **Ukuran** | ***Key Constraint*** |
| --- | --- | --- | --- | --- |
| 1 | *id* | *int* | 10 | *Primary Key* |
| 2 | nama | *varchar* | 20 |  |
| 3 | email | *varchar* | 50 |  |
| 4 | perusahaan | *varchar* | 50 |  |
| 5 | *phone* | *varchar* | 20 |  |
| 6 | *address* | *text* | 100 |  |
| 7 | *avatar* | *varchar* | 50 |  |
| 8 | status | *enum* | ACTIVE,INACTIVE |  |
| 9 | *created\_at* | timestamp |  |  |
| 10 | *updated\_at* | timestamp |  |  |

1. Struktur Tabel *Category*

Digunakan untuk menyimpan seluruh data kategori pada Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel. Berikut struktur *table* kategori pada Tabel 4.8:

Tabel 4.8 Struktur Tabel *Categories*

| **No** | **Nama Atribut** | **Tipe Data** | **Ukuran** | ***Key Constraint*** |
| --- | --- | --- | --- | --- |
| 1 | *id* | *int* | 10 | *Primary Key* |
| 2 | *name* | *varchar* | 20 |  |
| 3 | image | *varchar* | 50 |  |
| 4 | *created\_at* | timestamp |  |  |
| 5 | *updated\_at* | timestamp |  |  |

1. Struktur Tabel *Product*

Digunakan untuk menyimpan seluruh data *product* pada Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel. Berikut struktur *table* *product* pada Tabel 4.9:

Tabel 4.9 Struktur Tabel *Product*

| **No** | **Nama Atribut** | **Tipe Data** | **Ukuran** | ***Key Constraint*** |
| --- | --- | --- | --- | --- |
| 1 | *id* | *int* | 10 | *Primary Key* |
| 2 | *category*\_id | *int* | 10 | *Foreign Key* |
| 3 | nama\_produk | *varchar* | 20 |  |
| 4 | kode\_produk | *varchar* | 20 |  |
| 5 | keterangan | *text* | 50 |  |
| 6 | satuan | *varchar* | 10 |  |
| 7 | harga\_dasar | *int* | 11 |  |
| 8 | harga\_jual | *int* | 11 |  |
| 9 | gambar | *varchar* | 50 |  |
| 10 | *created\_at* | timestamp |  |  |
| 11 | *updated\_at* | timestamp |  |  |

1. Struktur Tabel *Stock*

Digunakan untuk menyimpan seluruh data stok pada Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel. Berikut struktur *table* stok pada Tabel 4.10:

Tabel 4.10 Struktur Tabel *Stock*s

| **No** | **Nama Atribut** | **Tipe Data** | **Ukuran** | ***Key Constraint*** |
| --- | --- | --- | --- | --- |
| 1 | *id* | *int* | 10 | *Primary Key* |
| 2 | *product*\_id | *int* | 10 | *Foreign Key* |
| 3 | stok | *int* | 11 |  |

1. Struktur Tabel Penjualan

Digunakan untuk menyimpan seluruh data penjualan pada Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel. Berikut struktur *table* penjualan pada Tabel 4.11:

Tabel 4.11 Struktur Tabel Penjualan

| **No** | **Nama Atribut** | **Tipe Data** | **Ukuran** | ***Key Constraint*** |
| --- | --- | --- | --- | --- |
| 1 | *id* | *int* | 10 | *Primary Key* |
| 2 | tanggal\_transaksi | date |  |  |
| 3 | *customer*\_id | *int* | 10 | *Foreign Key* |
| 4 | *invoice*\_number | *varchar* | 20 |  |
| 5 | total\_harga | *int* | 11 |  |
| 6 | *shipping* | *varchar* | 11 |  |
| 7 | profit | *int* | 11 |  |
| 8 | status | *enum* |  |  |
| 9 | created\_by | *bigint* | 20 | *Foreign Key* |
| 10 | *Update*d\_by | *bigint* | 20 |  |
| 11 | *created\_at* | timestamp |  |  |
| 12 | *updated\_at* | timestamp |  |  |

1. Struktur Tabel Penjualan\_*product*

Digunakan untuk menyimpan seluruh data detail penjualan pada Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel. Berikut struktur *table* penjualan\_*product* pada Tabel 4.12:

Tabel 4.12 Struktur Tabel Penjualan\_*products*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Atribut** | **Tipe Data** | **Ukuran** | ***Key Constraint*** |
| 1 | penjualan\_id | *int* | 10 | *Primary Key* |
| 2 | *product*\_id | *int* | 10 | *Primary Key* |
| 3 | qty | *int* | 11 |  |
| 4 | *created\_at* | timestamp |  |  |
| 5 | *updated\_at* | timestamp |  |  |

1. Struktur Tabel Pembelian

Digunakan untuk menyimpan seluruh data pembelian pada Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel. Berikut struktur *table* *user* pada Tabel 4.13:

Tabel 4.13 Struktur Tabel Pembelian

| **No** | **Nama Atribut** | **Tipe Data** | **Ukuran** | ***Key Constraint*** |
| --- | --- | --- | --- | --- |
| 1 | *id* | *int* | 10 | PK |
| 2 | tanggal\_transaksi | date |  |  |
| 3 | *supplier*\_id | *int* | 10 | *Foreign Key* |
| 4 | *invoice*\_number | *varchar* | 20 |  |
| 5 | total\_harga | *int* | 11 |  |
| 6 | status | *enum* |  |  |
| 7 | created\_by | *bigint* | 20 | *Foreign Key* |
| 8 | *Update*d\_by | *bigint* | 20 |  |
| 9 | *created\_at* | timestamp |  |  |
| 10 | *updated\_at* | timestamp |  |  |

1. Struktur Tabel Pembelian\_*product*

Digunakan untuk menyimpan seluruh data detail pembelian pada Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel. Berikut struktur *table* pembelian\_*product* pada Tabel 4.14:

Tabel 4.14 Struktur Tabel Pembelian\_*product*

| **No** | **Nama Atribut** | **Tipe Data** | **Ukuran** | ***Key Constraint*** |
| --- | --- | --- | --- | --- |
| 1 | pembelian\_id | *int* | 10 | PK |
| 2 | *product*\_id | *int* | 10 | PK |
| 3 | qty | *int* | 11 |  |
| 4 | *created\_at* | timestamp |  |  |
| 5 | *updated\_at* | timestamp |  |  |

1. Perancangan Antarmuka

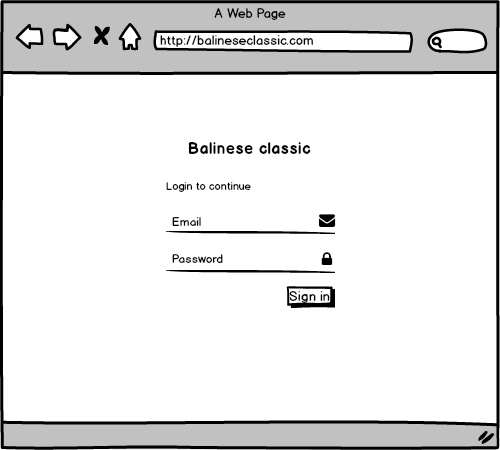
Perancangan antarmuka akan menjelaskan rancangan antarmuka dari sistem yang dirancang. Perancangan antarmuka ini akan memberikan gambaran tentang bagaimana bentuk dari tampilan sistem yang selanjutnya akan diimplementasikan menjadi Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel.

1. Desain Antarmuka Admin

Bagian ini akan menjelaskan tentang rancangan antarmuka untuk *roles* admin. Rancangan antarmuka tersebut antaranya sebagai berikut:

* + 1. Desain Antarmuka *Login* Admin

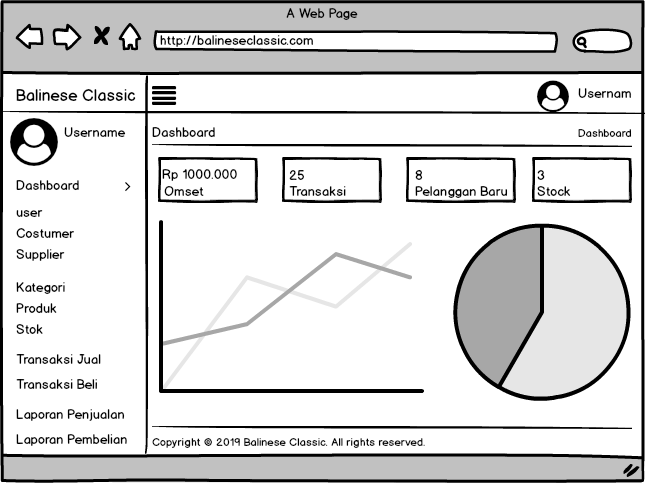
Halaman *login* adalah halaman pertama di mana admin diminta untuk memasukkan email dan *password* agar bisa masuk ke dalam sistem. Pada tahap ini sistem akan menentukan tampilan selanjutnya pada halaman *dashboard* berdasarkan *roles* admin tersebut. Desain antarmuka *login* dapat dilihat pada Gambar 4.15 berikut:



Gambar 4.15 Antarmuka *Login* – Admin

* + 1. Desain Antarmuka *Dashboard* Admin

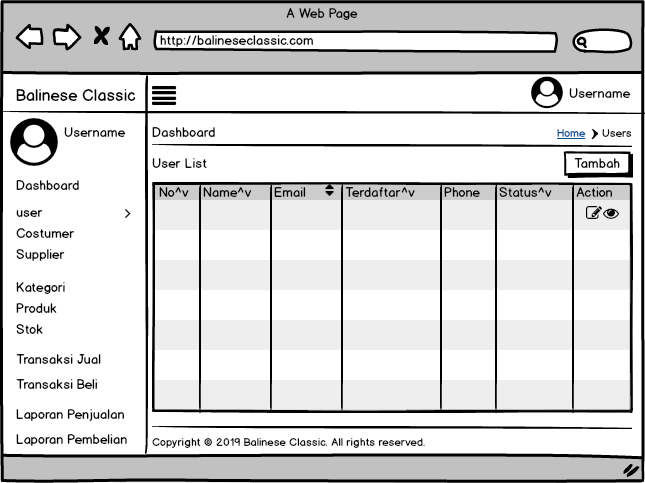
Halaman *Dashboard* Admin adalah halaman setelah admin memasukkan data *login* yang memiliki *roles* admin. Halaman ini menampilkan jumlah omset, jumlah transaksi, jumlah pelanggan baru, dan jumlah *stock* serta terdapat menu *sidebar* di sebelah kiri seperti, menu *dashboard*, menu *users*, menu *customer*, menu *supplier*, menu kategori, menu produk, menu *stock*, menu transaksi jual, menu transaksi beli, menu laporan penjualan, menu laporan pembelian. Desain antarmuka halaman *dashboard* admin dapat dilihat pada Gambar 4.16 berikut:



Gambar 4.16 Antarmuka *Dashboard* – Admin

* + 1. Desain Antarmuka Kelola Data *Users*

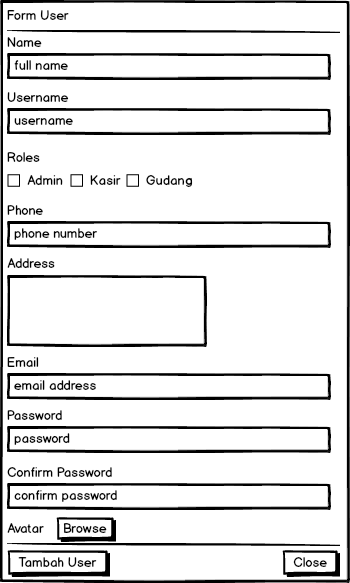
Halaman Kelola Data *Users* menampilkan *sidebar* menu dan daftar data *users* yang telah disimpan di dalam sistem.



Gambar 4.17 Antarmuka Data *Users* – Admin

* + 1. Desain *Form* Data *Users*

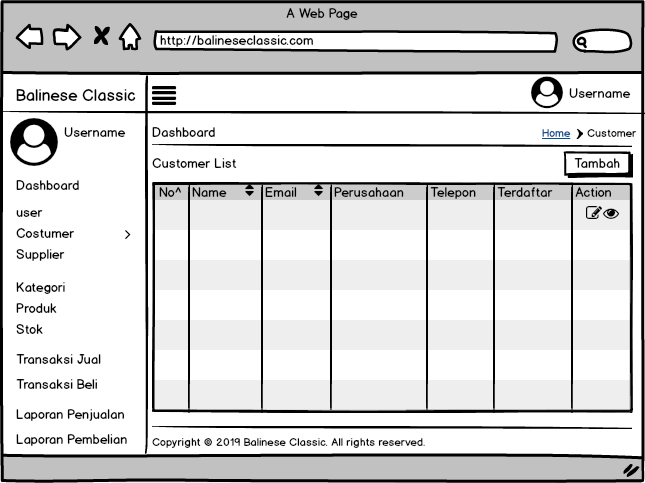
*Form* kelola data *users* adalah *form* yang digunakan untuk memasukkan data *users* dan meng*edit* data *user*



Gambar 4.18 Antarmuka *Form* *Users* – Admin

* + 1. Desain Antarmuka Kelola Data *Customers*

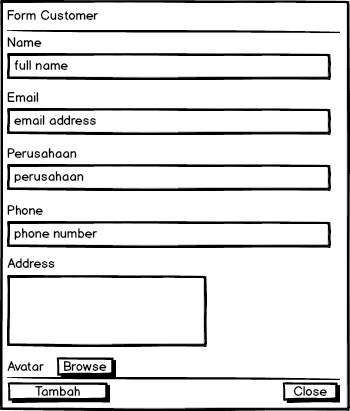
Halaman Kelola data *customer* menampilkan *sidebar* menu dan daftar data *customer* yang telah disimpan di dalam sistem.



Gambar 4.19 Antarmuka Data *Customers* – Admin

* + 1. Desain *Form* Data *Customers*

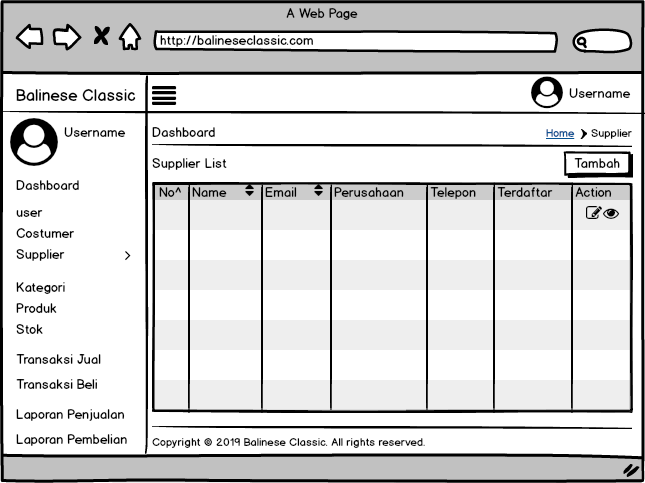
*Form* kelola data *customer* adalah *form* yang digunakan untuk memasukkan data *customer* dan meng*edit* data *customer*.



Gambar 4.20 Antarmuka *Form* *Customers* – Admin

* + 1. Desain Antarmuka Kelola Data *Suppliers*

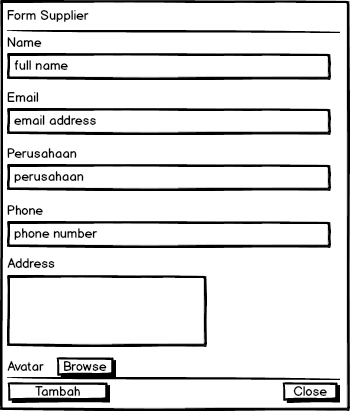
Halaman Kelola data *supplier* menampilkan *sidebar* menu dan daftar data *supplier* yang telah disimpan di dalam sistem.



Gambar 4.21 Antarmuka Data *Suppliers* – Admin

* + 1. Desain *Form* Data *Suppliers*

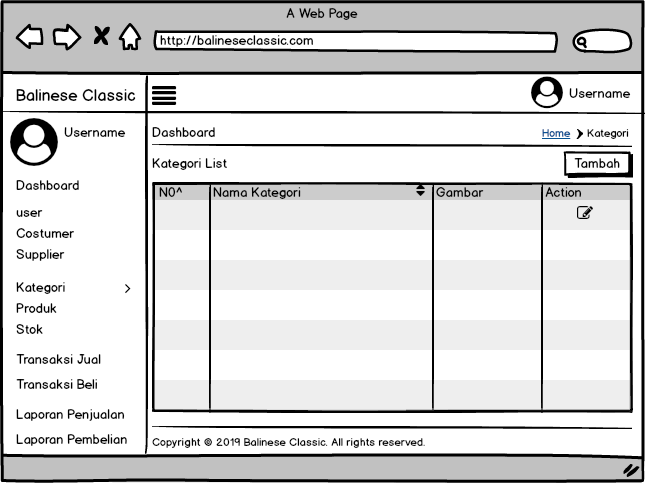
*Form* kelola data *supplier* adalah *form* yang digunakan untuk memasukkan data *supplier* dan meng*edit* data *supplier*.



Gambar 4.22 Antarmuka *Form* *Suppliers* – Admin

* + 1. Desain Antarmuka Kelola Data *Categories*

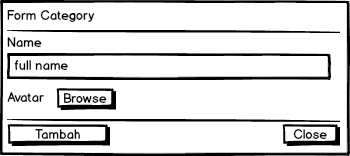
Halaman Kelola data *category* menampilkan *sidebar* menu dan daftar data *category* yang telah disimpan di dalam sistem.



Gambar 4.23 Antarmuka Data *Categories* – Admin

* + 1. Desain *Form* Data *Categories*

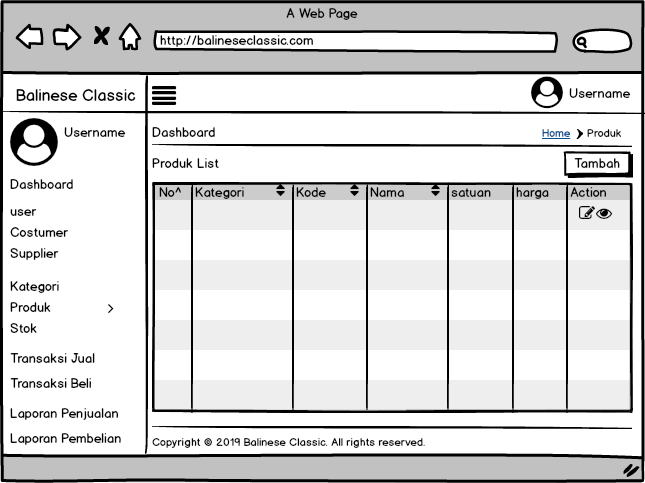
*Form* kelola data *category* adalah *form* yang digunakan untuk memasukkan data *category* dan meng*edit* data *category*



Gambar 4.24 Antarmuka *Form* *Categories* – Admin

* + 1. Desain Antarmuka Kelola Data *Products*

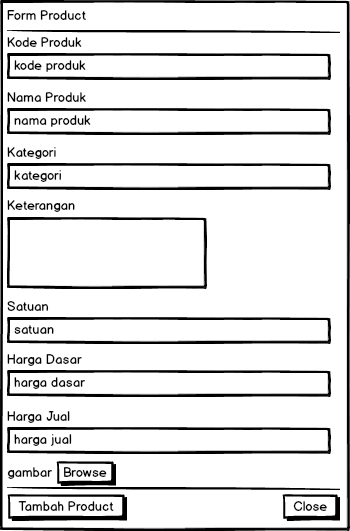
Halaman Kelola data *product* menampilkan *sidebar* menu dan daftar data *product* yang telah disimpan di dalam sistem.



Gambar 4.25 Antarmuka Data *Products* – Admin

* + 1. Desain *Form* Data *Products*

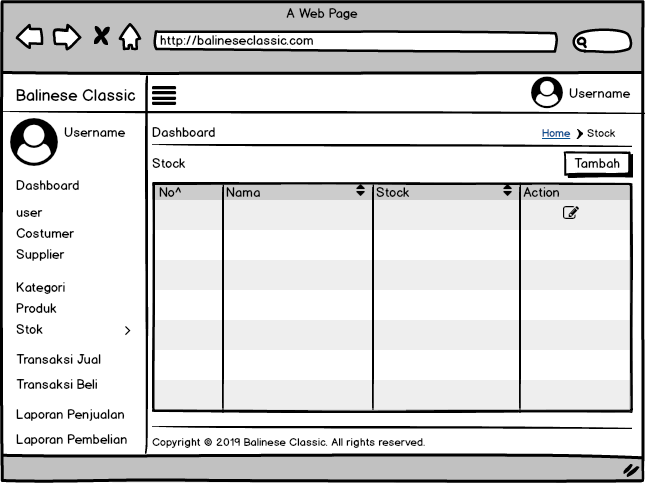
*Form* kelola data *products* adalah *form* yang digunakan untuk memasukkan data *products* dan meng*edit* data *products*.



Gambar 4.26 Antarmuka *Form* *Products* – Admin

* + 1. Desain Antarmuka Kelola Data *Stock*

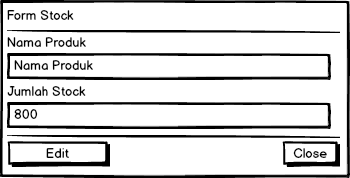
Halaman Kelola data *stock* menampilkan *sidebar* menu dan daftar data *stock* yang telah disimpan di dalam sistem.



Gambar 4.27 Antarmuka Data *Stock* – Admin

* + 1. Desain *Form* Data *Stock*

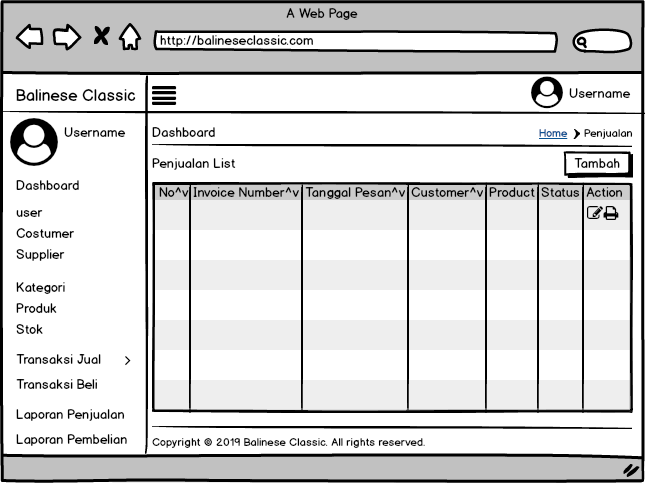
*Form* kelola data *stock* adalah *form* yang digunakan untuk memasukkan data *stock* dan meng*edit* data *stock*.



Gambar 4.28 Antarmuka *Form* *Stock* – Admin

* + 1. Desain Antarmuka Kelola Data Penjualan

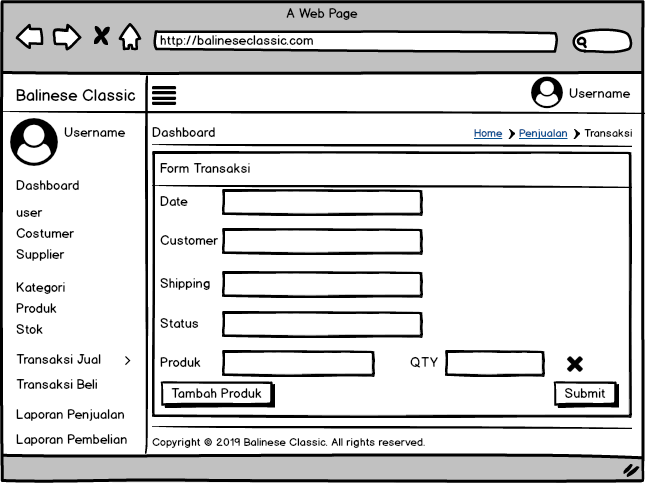
Halaman Kelola data penjualan menampilkan *sidebar* menu dan daftar data penjualan yang telah disimpan di dalam sistem



Gambar 4. 29 Antarmuka Data Penjualan – Admin

* + 1. Desain Antarmuka Transaksi Penjualan

Pada halaman transaksi penjualan ini berisi *form* yang berfungsi untuk menampung data transaksi penjualan.



Gambar 4.30 Antarmuka Transaksi Jual – Admin

* + 1. Desain Antarmuka Kelola Data Pembelian

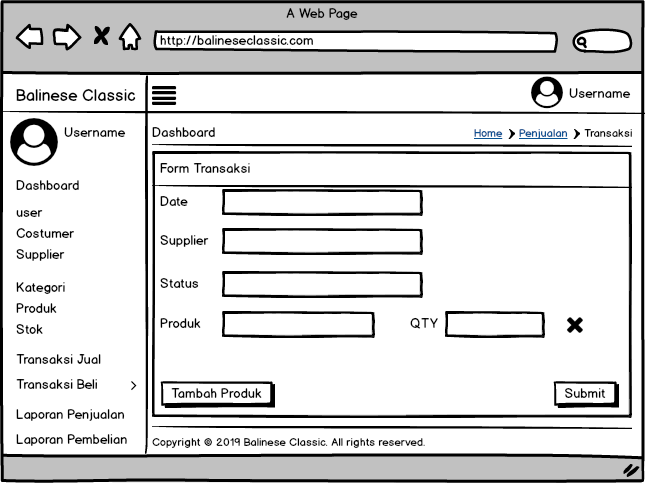
Halaman Kelola data pembelian menampilkan *sidebar* menu dan daftar data pembelian yang telah disimpan di dalam sistem.



Gambar 4.31 Antarmuka Data Pembelian – Admin

* + 1. Desain Antarmuka Transaksi Pembelian

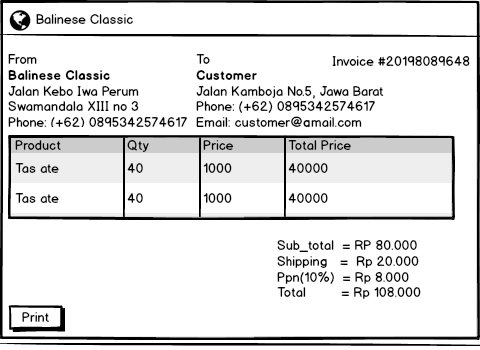
Pada halaman transaksi pembelian ini berisi *form* yang berfungsi untuk menampung data transaksi pembelian.



Gambar 4.32 Antarmuka Transaksi Beli – Admin

* + 1. Desain Antarmuka *Invoice* Penjualan

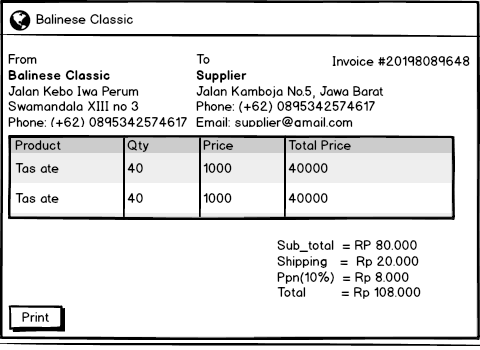
Halaman ini menampilkan nota dari transaksi penjualan yang sudah ada di dalam sistem.



Gambar 4.33 Antarmuka *Invoice* Penjualan – Admin

* + 1. Desain Antarmuka *Invoice* Pembelian

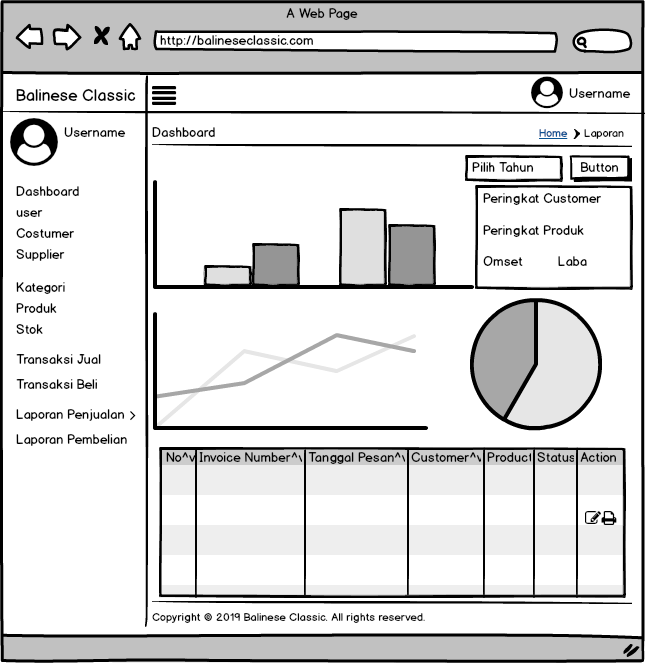
Halaman ini menampilkan nota dari transaksi pembelian yang sudah ada di dalam sistem.



Gambar 4.34 Antarmuka *Invoice* Pembelian – Admin

* + 1. Desain Antarmuka Laporan Penjualan

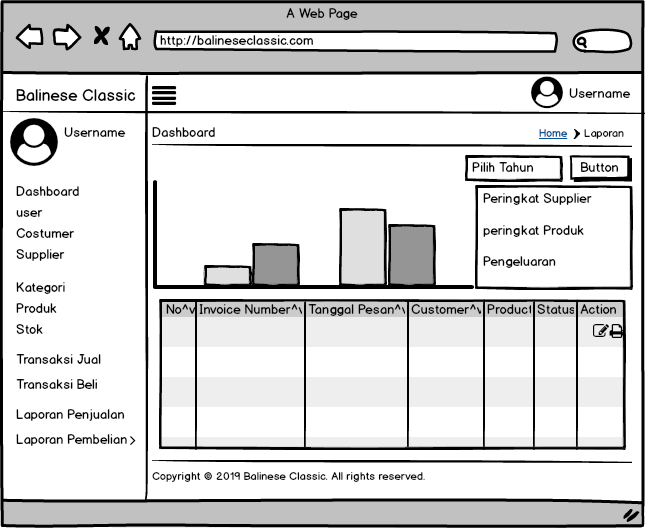
Halaman ini menampilkan data-data penjualan berupa grafik dan tabel di mana *user* diminta untuk memasukkan tahun untuk memfilter data tersebut.



Gambar 4.35 Antarmuka Laporan Penjualan – Admin

* + 1. Desain Antarmuka Laporan Pembelian

Halaman ini menampilkan data-data pembelian berupa grafik dan tabel di mana *user* diminta untuk memasukkan tahun untuk memfilter data tersebut.



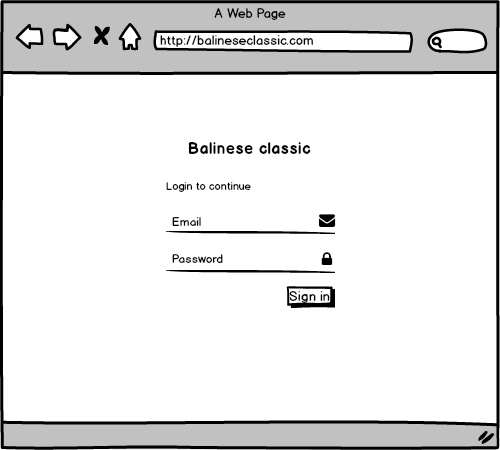
Gambar 4.36 Antarmuka Laporan Pembelian – Admin

1. Desain Antarmuka Kasir

Bagian ini akan menjelaskan tentang rancangan antarmuka untuk *roles* Kasir. Rancangan antarmuka tersebut antaranya sebagai berikut:

1. Desain Antarmuka *Login* Kasir

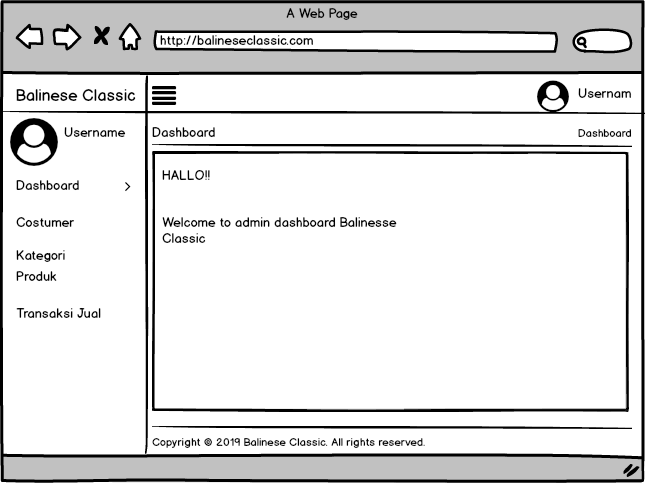
Halaman *login* adalah halaman pertama di mana kasir diminta untuk memasukkan email dan *password* agar bisa masuk ke dalam sistem.



Gambar 4.37 Antarmuka *Login* – Kasir

1. Desain Antarmuka *Dashboard* Kasir

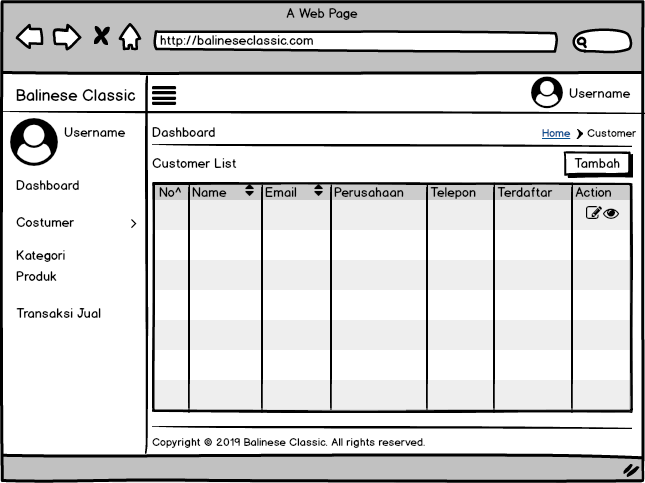
Halaman *Dashboard* Kasir merupakan halaman di mana kasir setelah berhasil melakukan *login*.



Gambar 4.38 Antarmuka *Dashboard* – Kasir

1. Desain Antarmuka *List* *Customer*

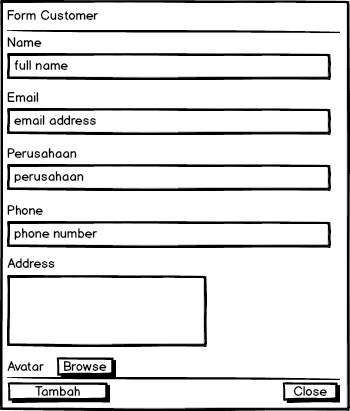
Halaman ini menampilkan data-data *customer* yang ada di *database*, juga terdapat tombol untuk menuju *form* *customer* dan detail *customer*.



Gambar 4.39 Antarmuka *List* *Customer* – Kasir

1. Desain *Form* *Customer*

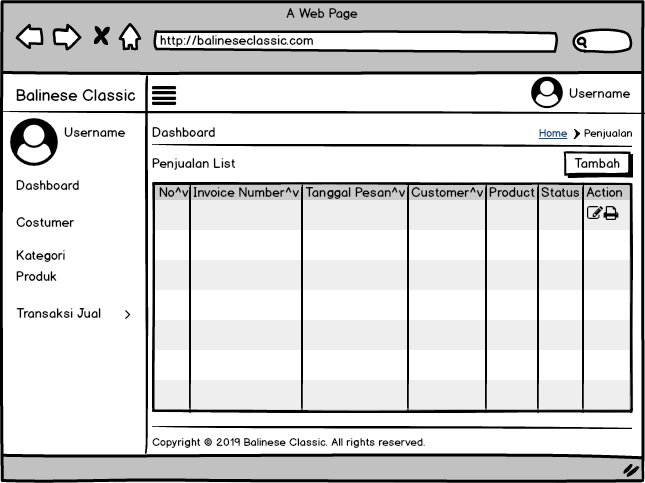
*Form* kelola data *customer* adalah *form* yang digunakan untuk memasukkan data *customer* dan meng*edit* data *customer*.



Gambar 4.40 Antarmuka *Form* *Customer* – Kasir

1. Desain Antarmuka *List* Penjualan

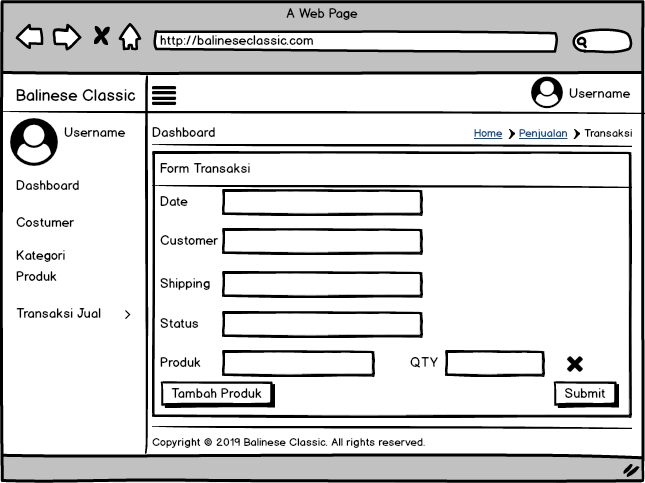
Halaman ini menampilkan data-data penjualan yang ada di *database*, juga terdapat tombol untuk menuju *form* penjualan dan *invoice* penjualan.



Gambar 4.41 Antarmuka *List* Penjualan – Kasir

1. Desain Transaksi Penjualan

Pada halaman transaksi penjualan ini berisi *form* yang berfungsi untuk menampung data transaksi penjualan.



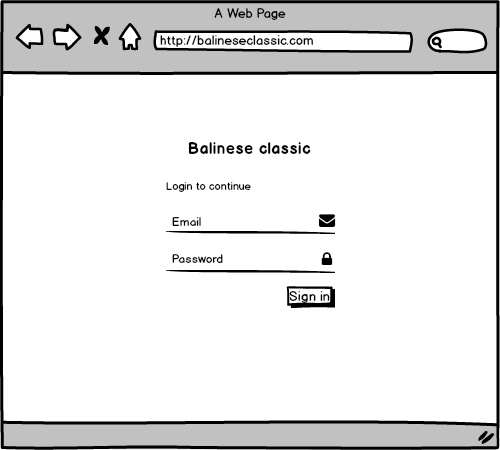
Gambar 4.42 Antarmuka *Form* Penjualan – Kasir

1. Desain Antarmuka Gudang

Bagian ini akan menjelaskan tentang rancangan antarmuka untuk *roles* Gudang. Rancangan antarmuka tersebut antaranya sebagai berikut:

1. Desain Antarmuka *Login* Gudang

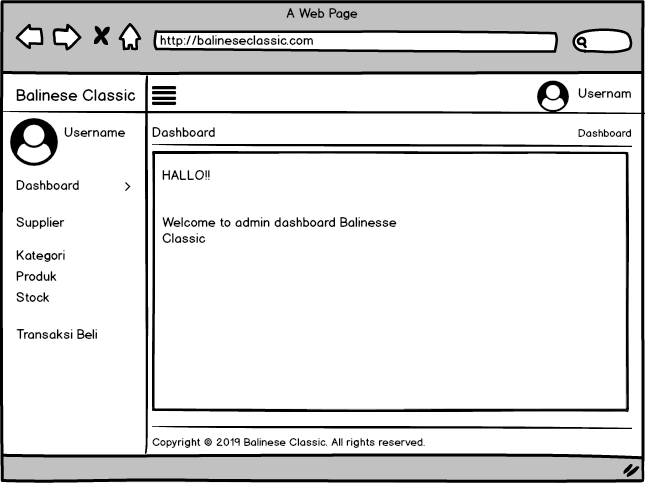
Halaman *login* adalah halaman pertama di mana kasir diminta untuk memasukkan email dan *password* agar bisa masuk ke dalam sistem.



Gambar 4.43 Antarmuka *Login* – Gudang

1. Desain Antarmuka *Dashboard* Gudang

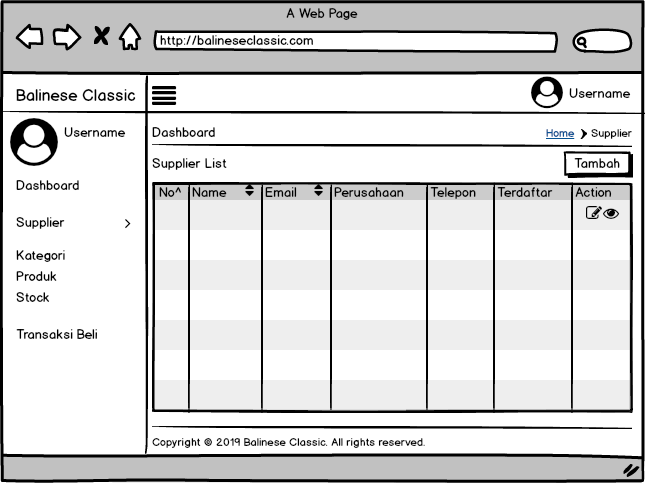
Halaman *Dashboard* Gudang merupakan halaman di mana gudang setelah berhasil melakukan *login*.



Gambar 4.44 Antarmuka *Dashboard* Kasir – Gudang

1. Desain Antarmuka *List* *Supplier*

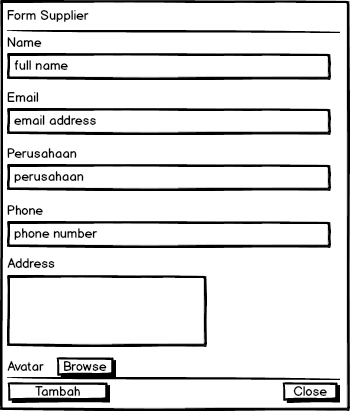
Halaman ini menampilkan data-data *supplier* yang ada di *database*, juga terdapat tombol untuk menuju *form* *supplier* dan detail *supplier*.



Gambar 4.45 Antarmuka *List* *Supplier* – Gudang

1. Desain *Form* *Supplier*

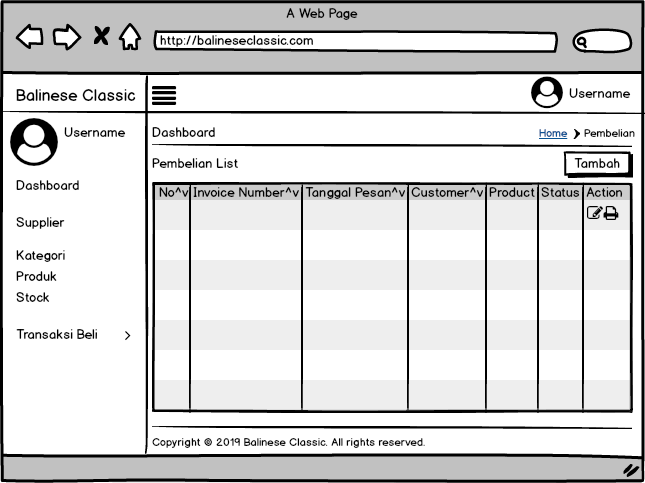
*Form* kelola data *supplier* adalah *form* yang digunakan untuk memasukkan data *supplier* dan meng*edit* data *supplier*.



Gambar 4.46 Antarmuka *Form* *Supplier* – Gudang

1. Desain Antarmuka *List* Pembelian

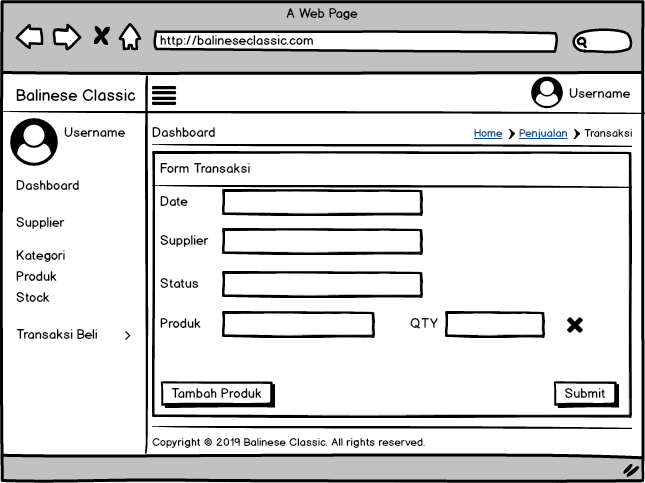
Halaman ini menampilkan data-data pembelian yang ada di *database*, juga terdapat tombol untuk menuju *form* pembelian.



Gambar 4.47 Antarmuka *List* Pembelian – Gudang

1. Desain Transaksi Pembelian

Pada halaman transaksi pembelian ini berisi *form* yang berfungsi untuk menampung data transaksi pembelian.



Gambar 4.48 Antarmuka *Form* Pembelian – Gudang

1. Implementasi Sistem

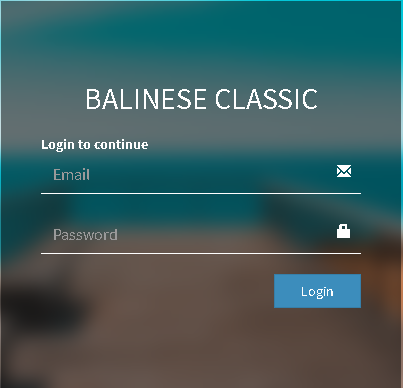
Pada implementasi sistem akan dijelaskan mengenai tampilan pada sistem. Berikut tampilan pada Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese *Class*ic Menggunakan *Framework* Laravel

1. Tampilan Untuk Admin

Tampilan untuk Admin, ditujukan untuk *user* yang memiliki *roles* admin, di mana terdapat tampilan sesuai dengan kebutuhannya. Adapun tampilannya sebagai berikut:

1. Tampilan Halaman *Login* – Admin

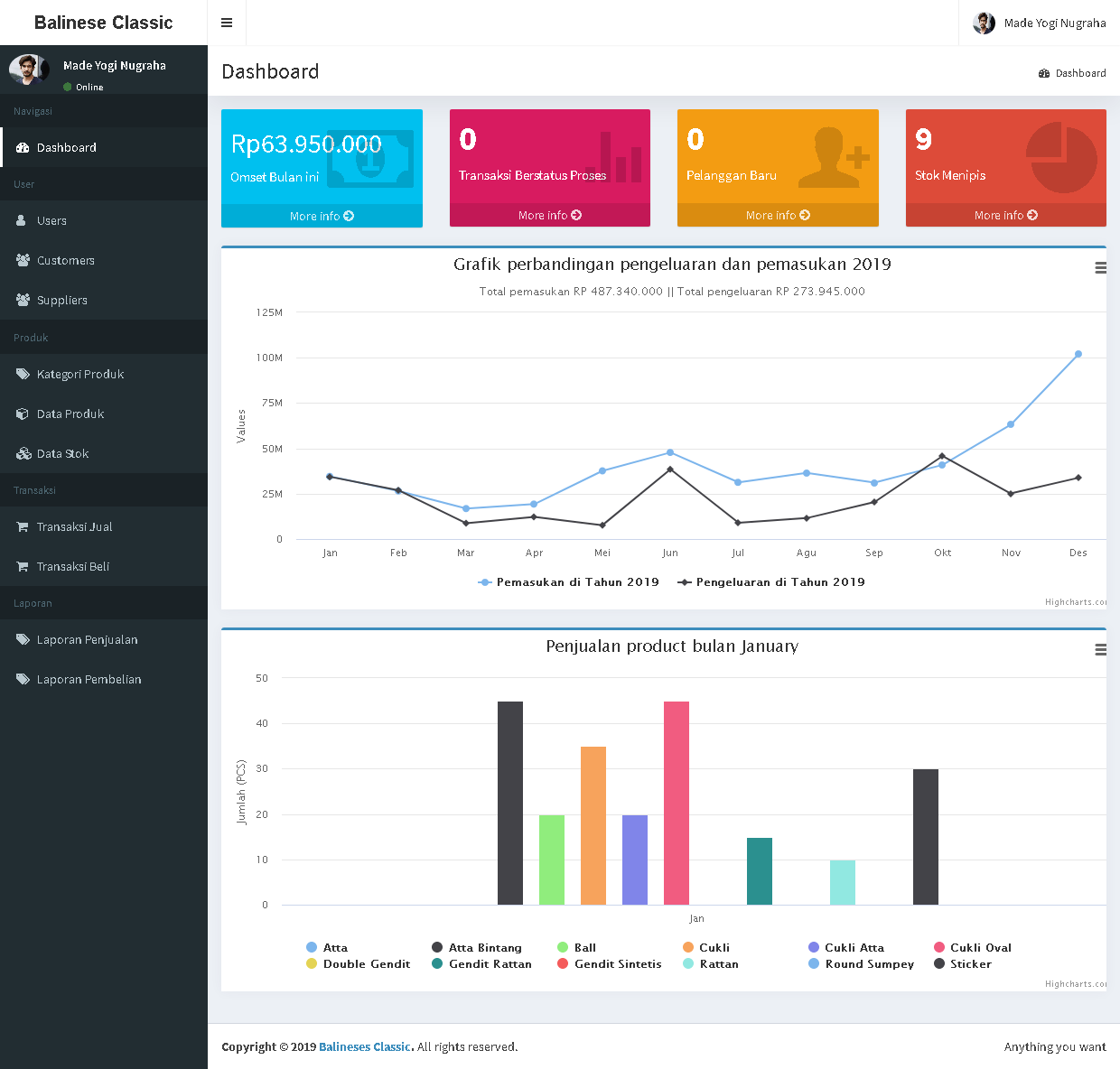
Halaman *login* adalah halaman pertama di mana admin diminta untuk memasukkan email dan *password* agar bisa masuk ke dalam sistem. Pada tahap ini sistem akan menentukan tampilan selanjutnya pada halaman *dashboard* berdasarkan *roles* admin tersebut.



Gambar 4.49 Tampilan Halaman *Login* – Admin

1. Tampilan Halaman *Dashboard* – Admin

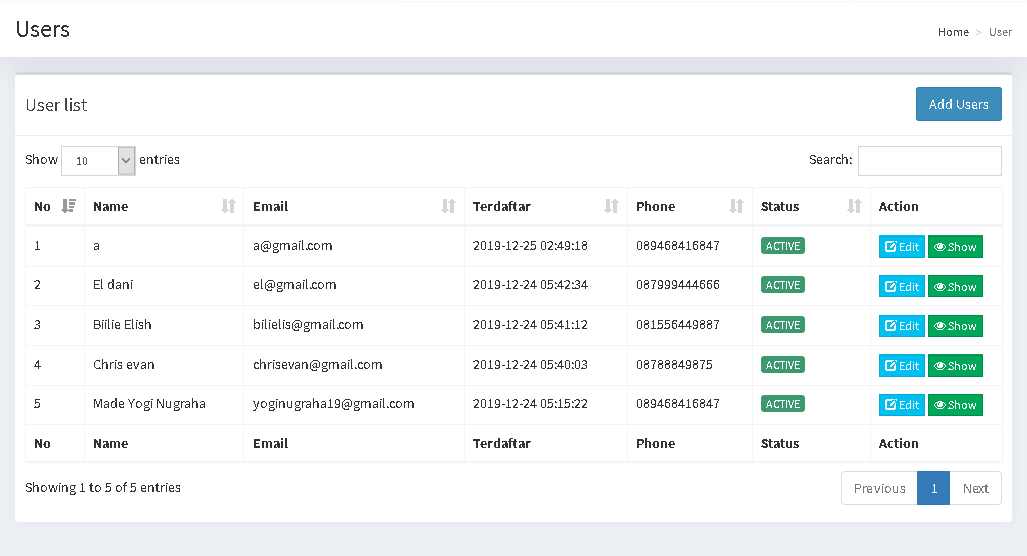
Halaman *Dashboard* merupakan tampilan setelah berhasil melakukan *login*.



Gambar 4.50 Tampilan Halaman *Dashboard* – Admin

1. Tampilan Halaman *List* *Users* – Admin

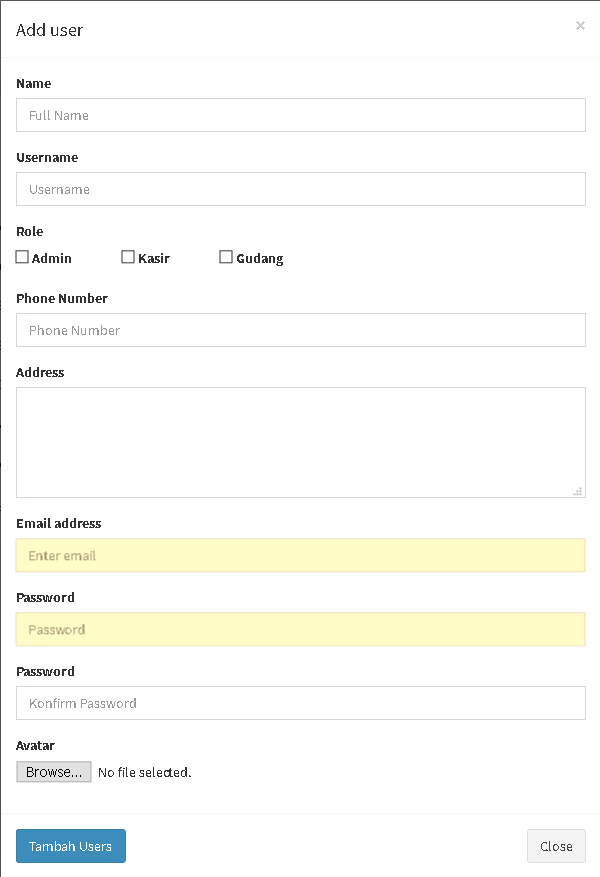
Halaman *List* *Users* merupakan halaman setelah klik menu *Users*. Pada halaman ini terdapat data-data *user* yang tersimpan di sistem.



Gambar 4.51 Tampilan Halaman *List* *Users* – Admin

1. Tampilan *Form* *User* – Admin

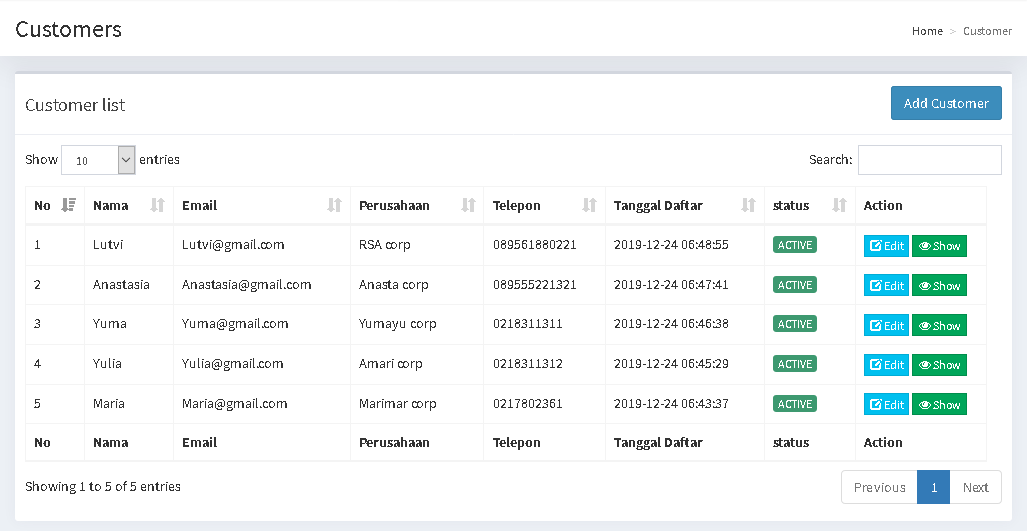
*Form* *User* merupakan *form* yang muncul setelah klik tombol *add* *user* pada halaman *list* *users*.



Gambar 4.52 Tampilan *Form* *User* – Admin

1. Tampilan Halaman *List* *Customers* – Admin

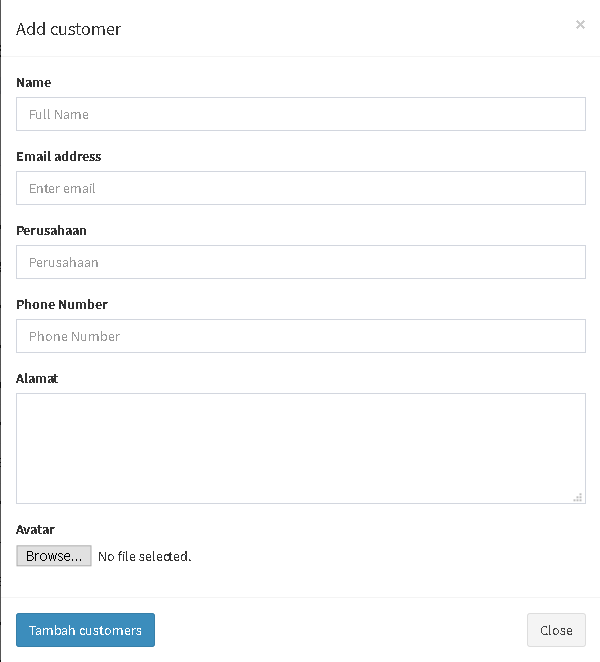
Halaman *List* *Customers* merupakan halaman setelah klik menu *Customers*. Pada halaman ini terdapat data-data *Customers* yang tersimpan di sistem.



Gambar 4.53 Tampilan Halaman *List* *Customers* – Admin

1. Tampilan *Form* *Customer* – Admin

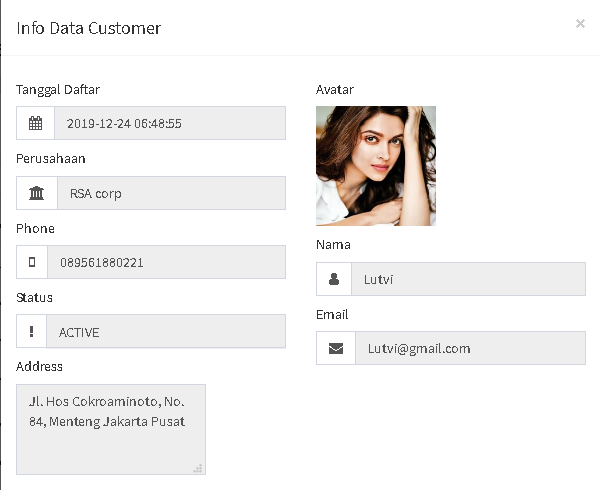
*Form* *Customer* merupakan *form* yang muncul setelah klik tombol *add* *Customer* pada halaman *list* *Customers*.



Gambar 4.54 Tampilan *Form* *Customer* – Admin

1. Tampilan Detail *Customer* – Admin

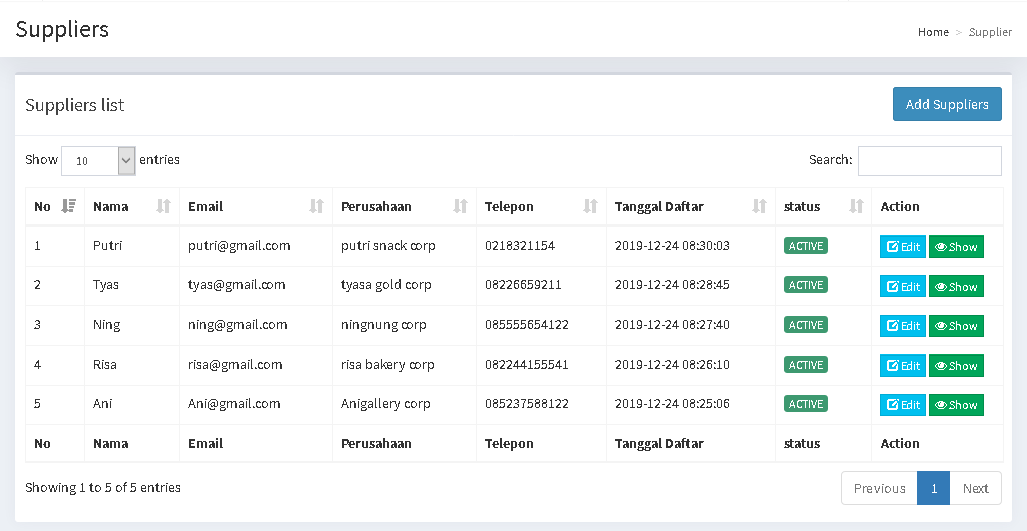
Detail *Customer* merupakan tampilan yang muncul setelah klik tombol *show* pada halaman *list* *Customers*.



Gambar 4.55 Tampilan Detail *Customer* – Admin

1. Tampilan Halaman *List* *Suppliers* – Admin

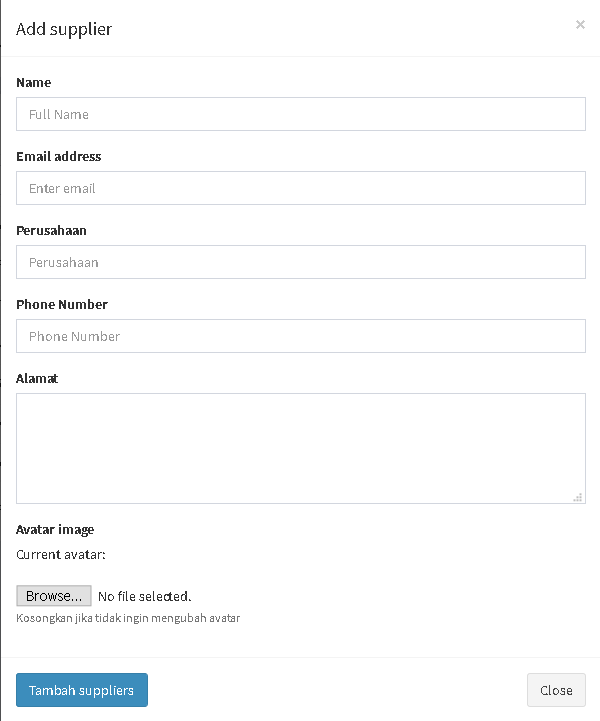
Halaman *List* *Suppliers* merupakan halaman setelah klik menu *Suppliers*. Pada halaman ini terdapat data-data *Suppliers* yang tersimpan di sistem.



Gambar 4.56 Tampilan Halaman *List* *Suppliers* – Admin

1. Tampilan *Form* *Supplier* – Admin

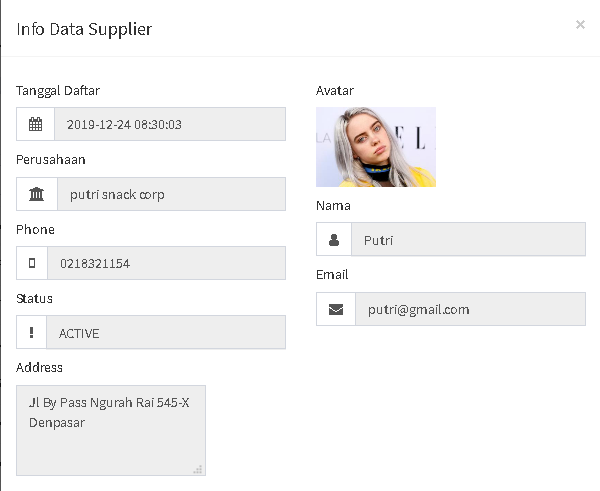
*Form* *Supplier* merupakan *form* yang muncul setelah klik tombol *add* *Supplier* pada halaman *list* *Suppliers*.



Gambar 4.57 Tampilan *Form* *Supplier* – Admin

1. Tampilan Detail *Supplier* – Admin

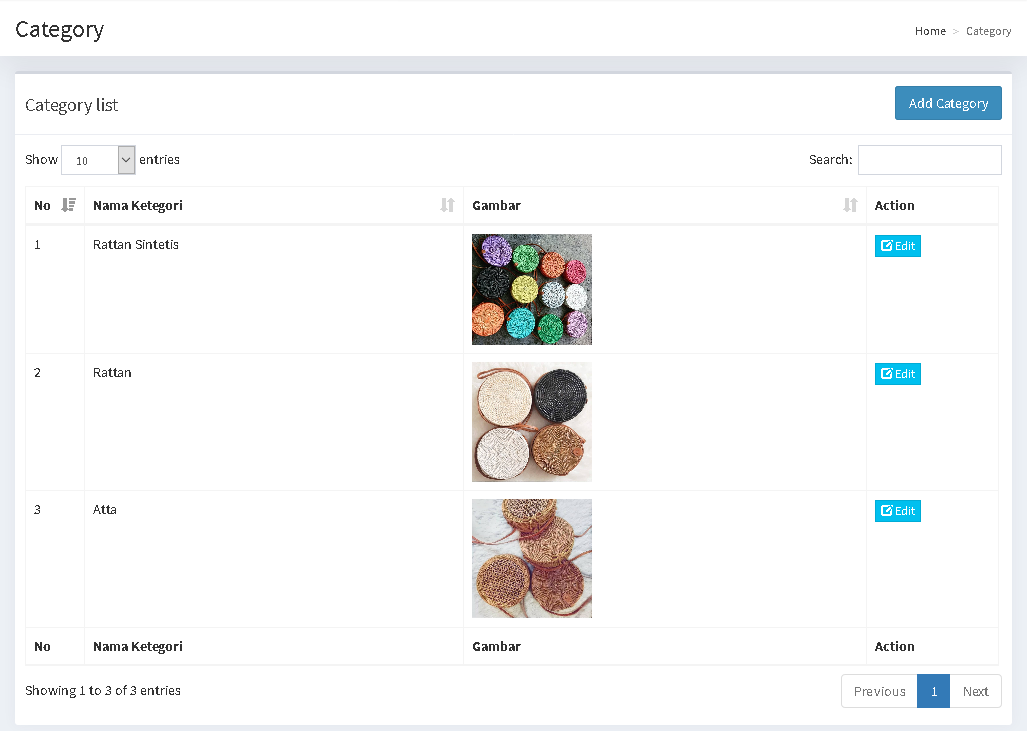
Detail *Supplier* merupakan tampilan yang muncul setelah klik tombol *show* pada halaman *list* *Suppliers*.



Gambar 4.58 Tampilan Detail *Supplier* – Admin

1. Tampilan Halaman *List* *Categories* – Admin

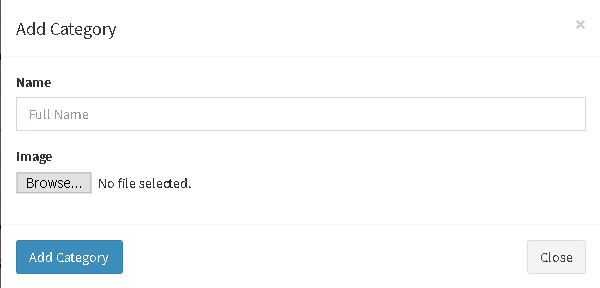
Halaman *List* *Categories* merupakan halaman setelah klik menu *Categories*. Pada halaman ini terdapat data-data *Categories* yang tersimpan di sistem.



Gambar 4.59 Tampilan Halaman *List* *Categories* – Admin

1. Tampilan *Form* *Category* – Admin

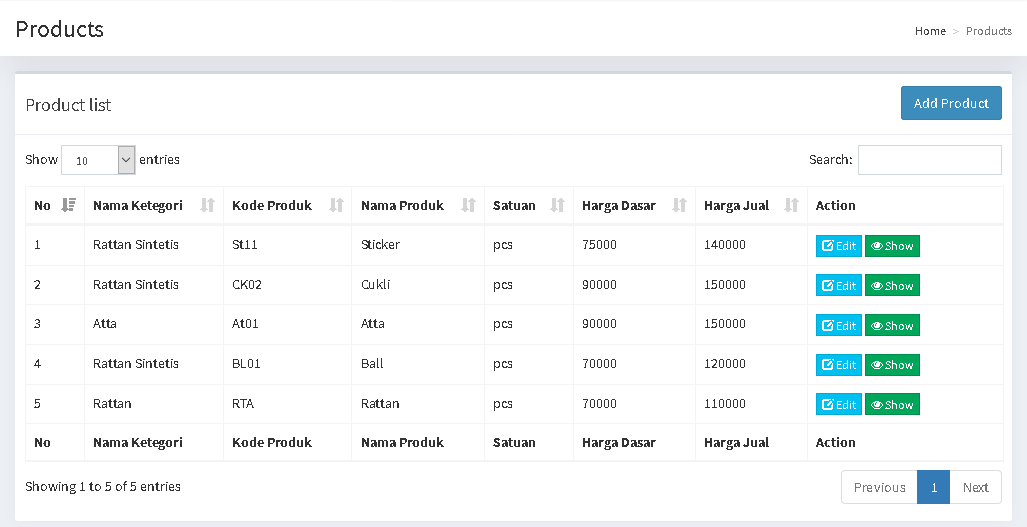
*Form* *Category* merupakan *form* yang muncul setelah klik tombol *add* *Category* pada halaman *list* *Categories*.



Gambar 4.60 Tampilan *Form* *Category* – Admin

1. Tampilan Halaman *List* *Products* – Admin

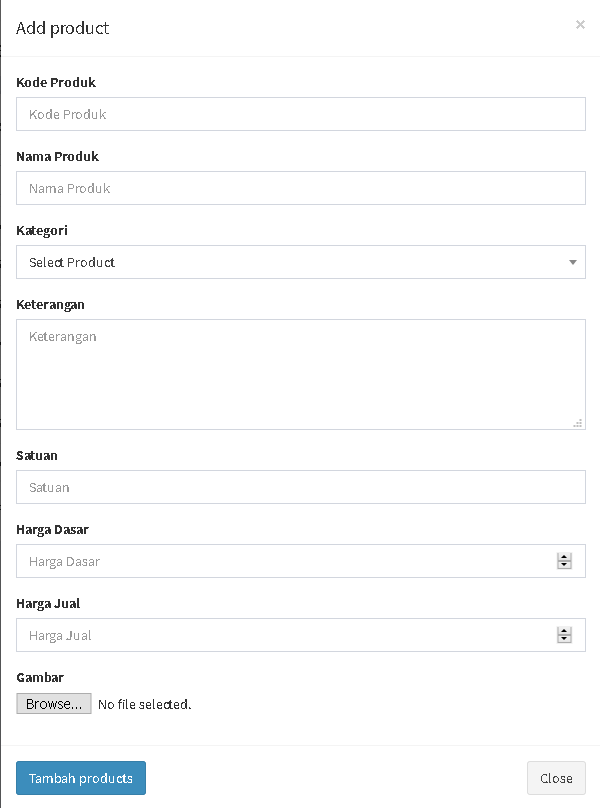
Detail *Category* merupakan tampilan yang muncul setelah klik tombol *show* pada halaman *list* *Categories*.



Gambar 4.61 Tampilan Halaman *List* *Products* – Admin

1. Tampilan *Form* *Product* – Admin

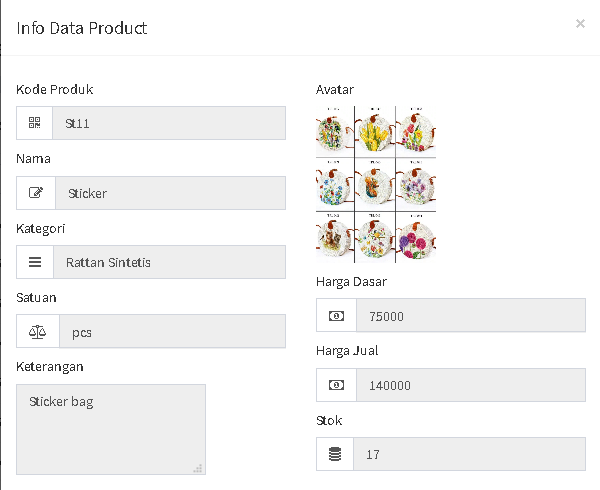
Halaman *List* *Products* merupakan halaman setelah klik menu *Products*. Pada halaman ini terdapat data-data *Products* yang tersimpan di sistem.



Gambar 4.62 Tampilan *Form* *Product* – Admin

1. Tampilan Detail *Product* – Admin

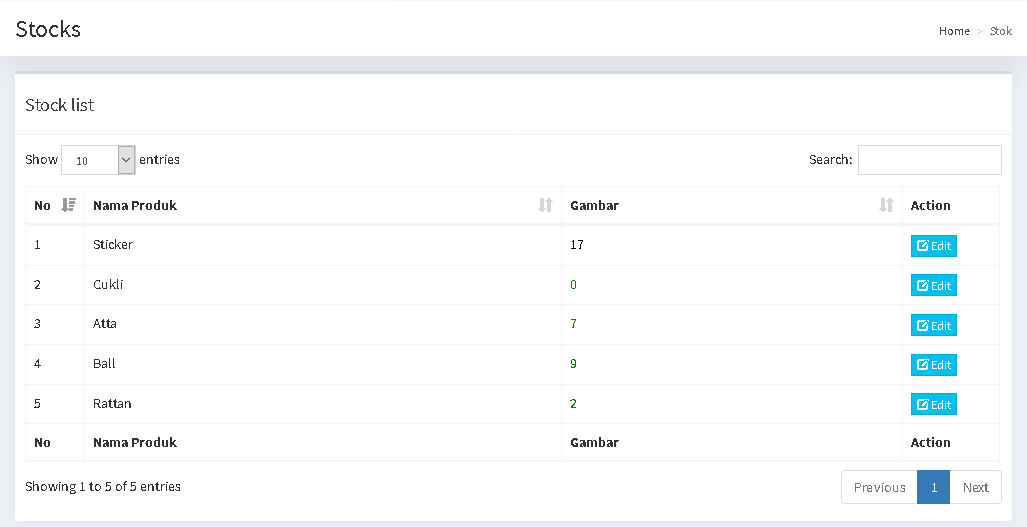
Detail *Product* merupakan tampilan yang muncul setelah klik tombol *show* pada halaman *list* *Products*.



Gambar 4.63 Tampilan Detail *Product* – Admin

1. Tampilan Halaman *List* *Stock*s – Admin

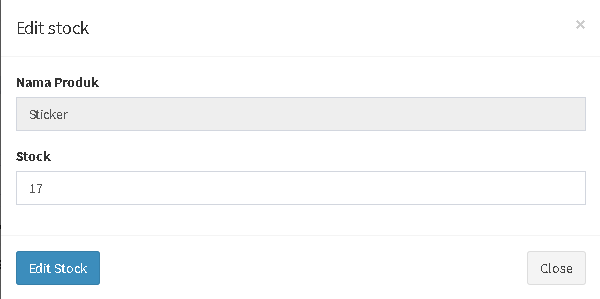
Halaman *List* *Stock*s merupakan halaman setelah klik menu *Stock*s. Pada halaman ini terdapat data-data *Stock*s yang tersimpan di sistem.



Gambar 4.64 Tampilan Halaman *List* *Stock*s – Admin

1. Tampilan *Form* *Stock* – Admin

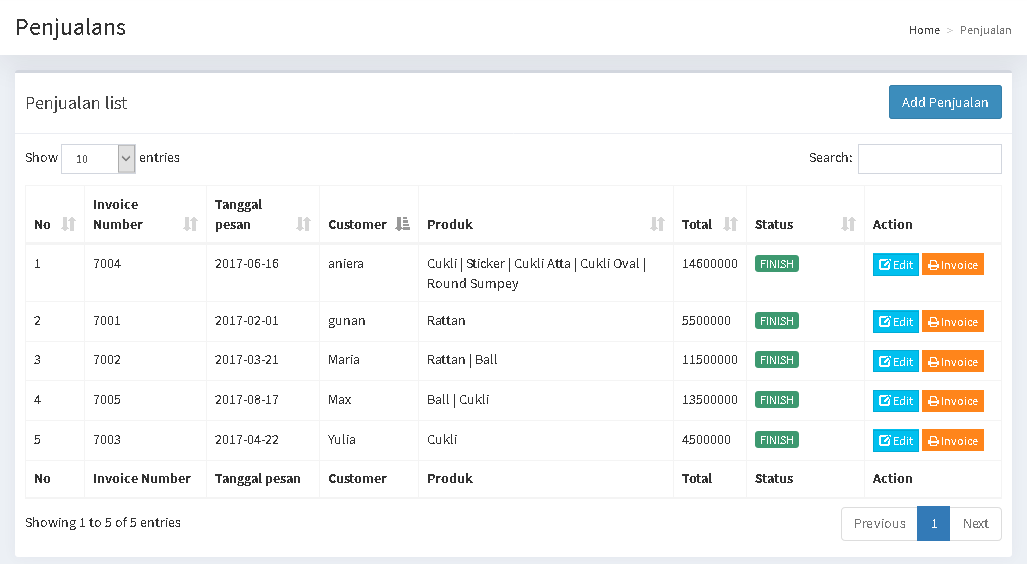
*Form* *Stock* merupakan *form* yang muncul setelah klik tombol *add* *Stock* pada halaman *list* *Stock*s.



Gambar 4.65 Tampilan *Form* *Stock* – Admin

1. Tampilan Halaman *List* Penjualan – Admin

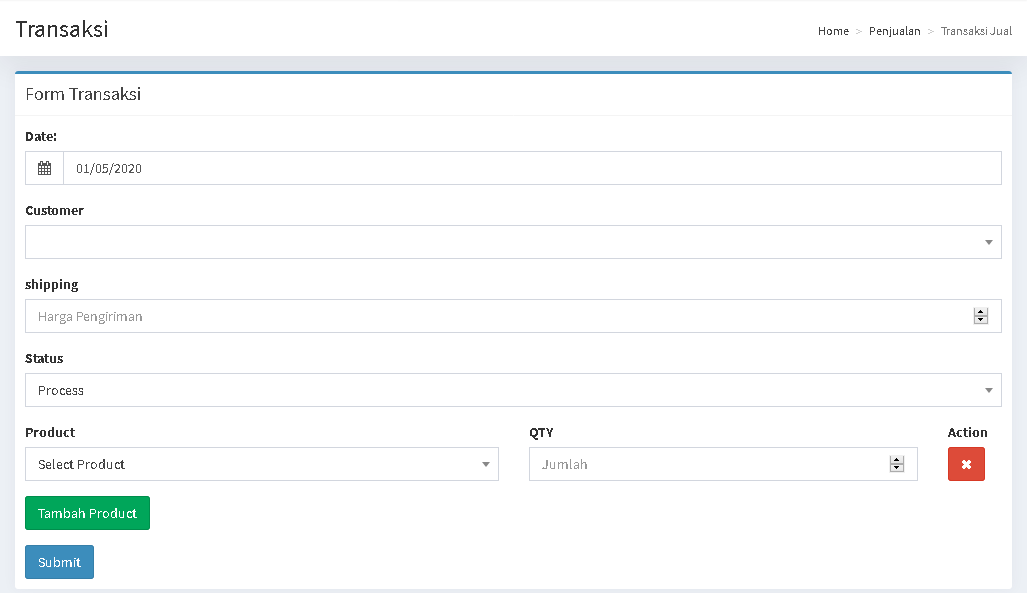
Halaman *List* Penjualan merupakan halaman setelah klik menu transaksi jual. Pada halaman ini terdapat data-data transaksi jual yang tersimpan di sistem.



Gambar 4.66 Tampilan Halaman *List* Penjualan – Admin

1. Tampilan *Form* Penjualan – Admin

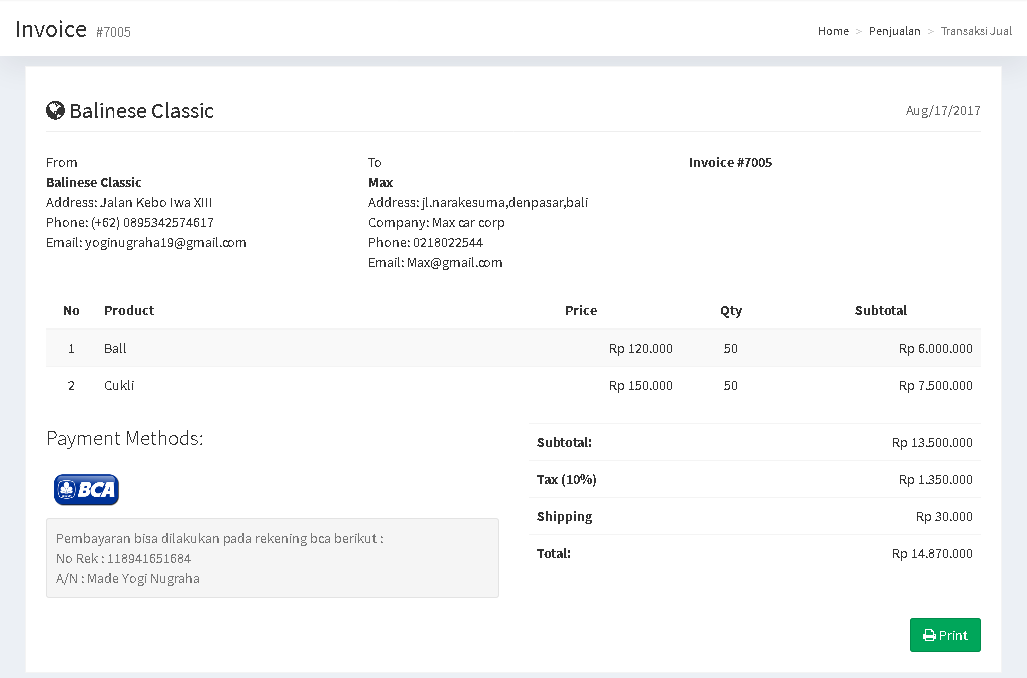
Pada halaman ini terdapat *form* yang wajib diisi saat melakukan transaksi. Tampilan *form* penjualan dapat dilihat pada gambar berikut:



Gambar 4.67 Tampilan *Form* Penjualan – Admin

1. Tampilan Halaman *Invoice* Penjualan – Admin

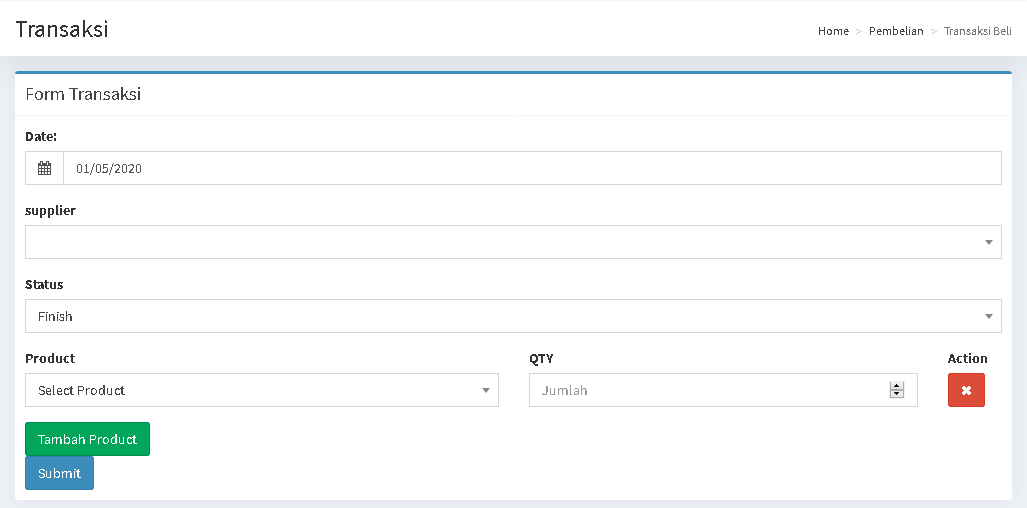
Tampilan Halaman *Invoice* Penjualan merupakan tampilan setelah klik tombol *invoice* pada halaman *list* penjualan.



Gambar 4.68 Tampilan Halaman *Invoice* Penjualan – Admin

1. Tampilan Halaman *List* Pembelian – Admin

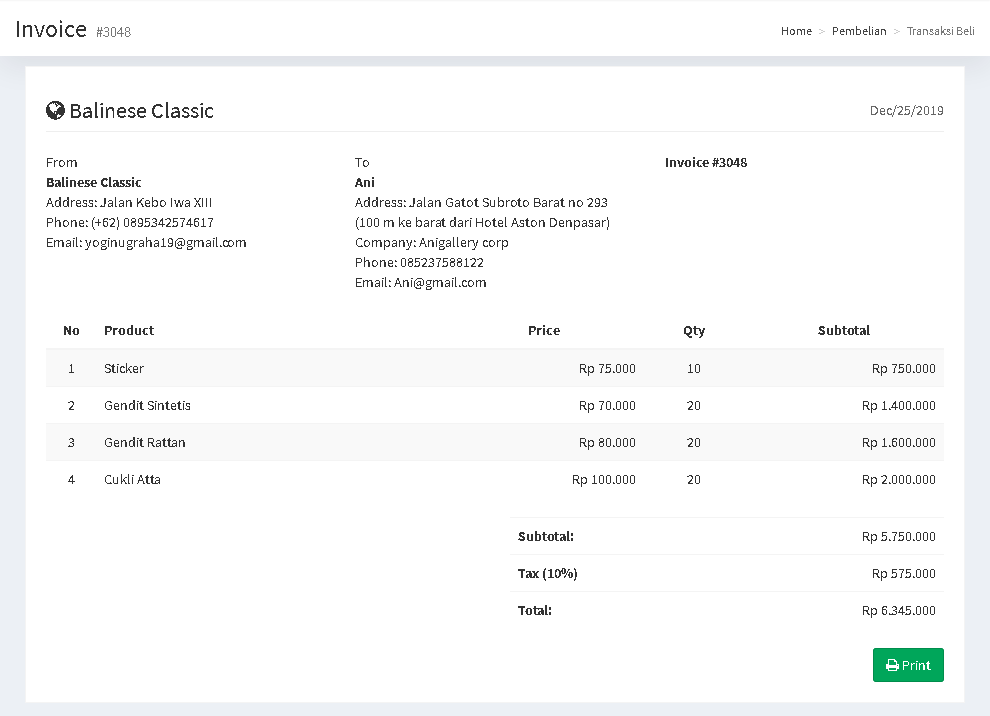
Halaman *List* Pembelian merupakan halaman setelah klik menu transaksi beli. Pada halaman ini terdapat data-data transaksi beli yang tersimpan di sistem.



Gambar 4.69 Tampilan Halaman *List* Pembelian – Admin

1. Tampilan Halaman *Form* Pembelian – Admin

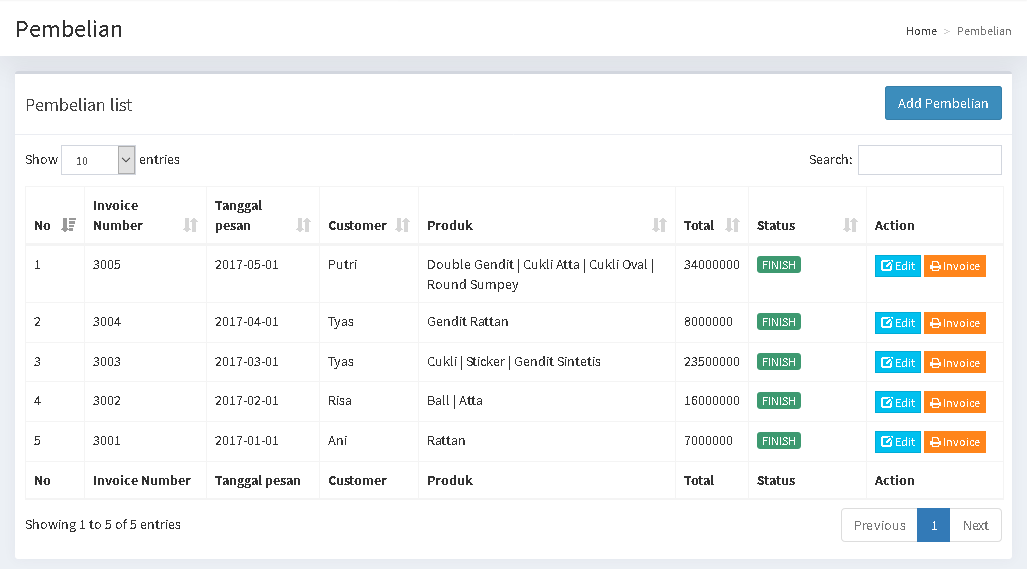
Pada halaman ini terdapat *form* yang wajib diisi saat melakukan transaksi. Tampilan *form* pembelian dapat dilihat pada gambar berikut:



Gambar 4.70 Tampilan Halaman *Form* Pembelian – Admin

1. Tampilan Halaman *Invoice* Pembelian – Admin

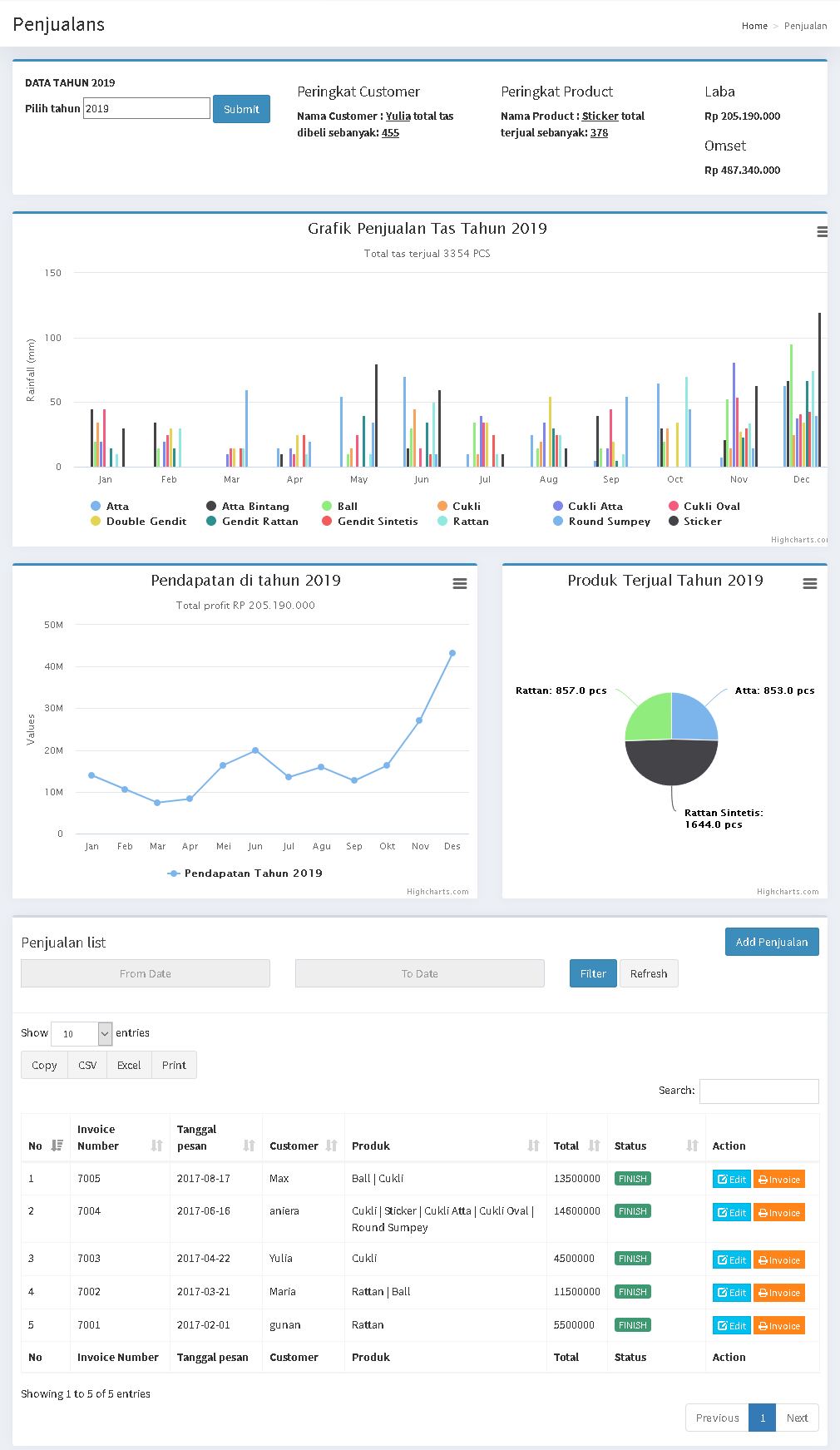
Tampilan Halaman *Invoice* Pembelian merupakan tampilan setelah klik tombol *invoice* pada halaman *list* pembelian.



Gambar 4.71 Tampilan Halaman *Invoice* Pembelian – Admin

1. Tampilan Halaman Laporan Penjualan – Admin

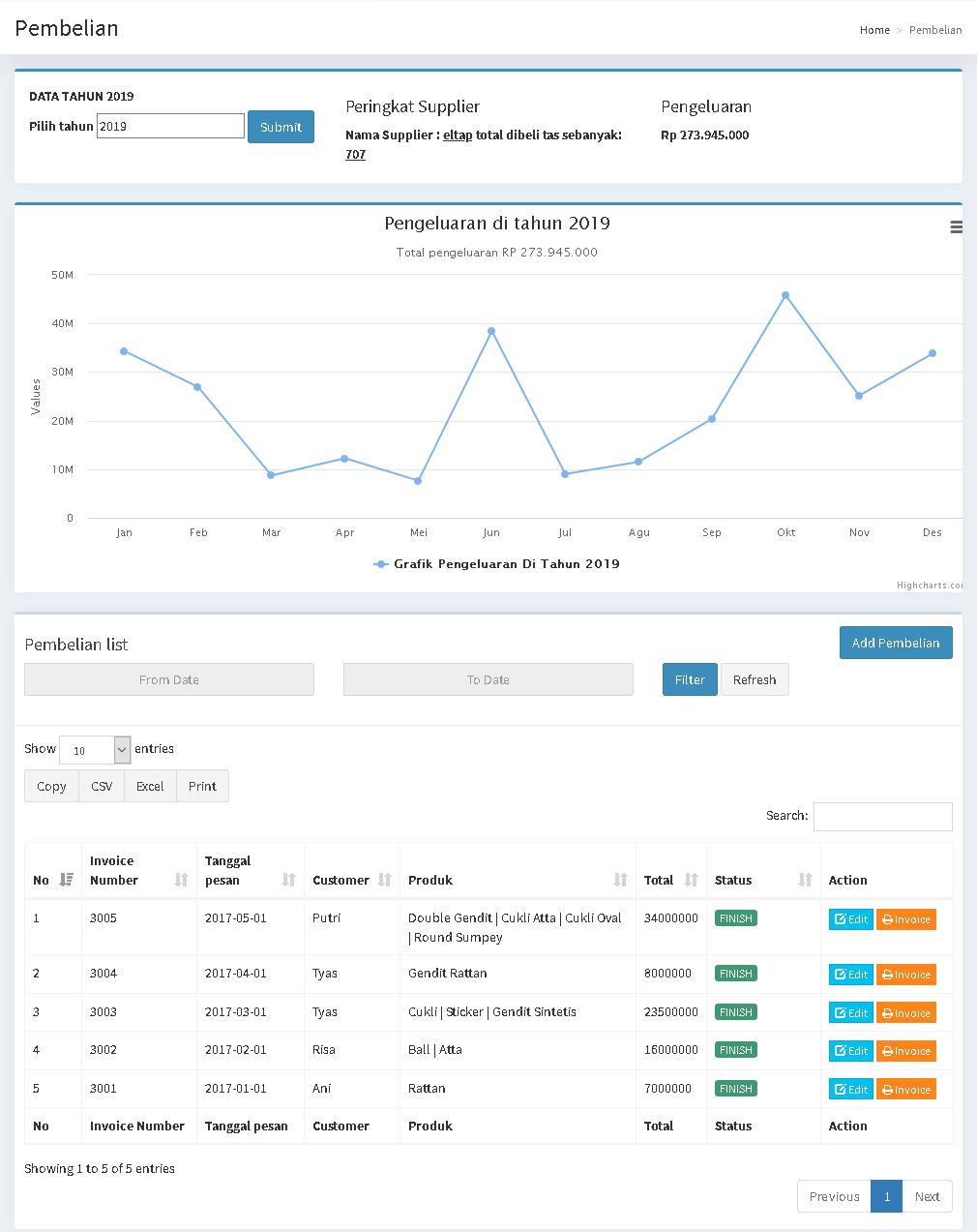
Pada halaman ini terdapat laporan penjualan dalam bentuk grafik dan data tabel.



Gambar 4.72 Tampilan Halaman Laporan Penjualan – Admin

1. Tampilan Halaman Laporan Pembelian – Admin

Tampilan Halaman Laporan Pembelian merupakan tampilan setelah klik menu Laporan Pembelian. Pada halaman ini terdapat laporan dalam bentuk grafik dan data tabel.



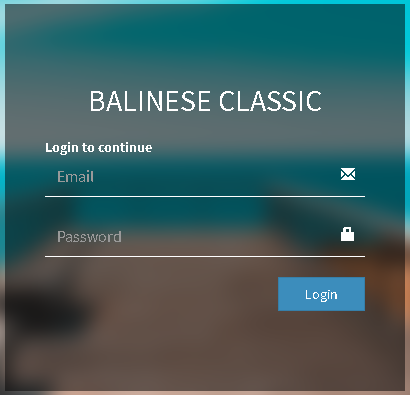
Gambar 4.73 Tampilan Halaman Laporan Pembelian – Admin

1. Tampilan Untuk Kasir

Tampilan untuk Kasir, ditujukan untuk *user* yang memiliki *roles* Kasir, di mana terdapat tampilan sesuai dengan kebutuhannya. Adapun tampilannya sebagai berikut:

1. Tampilan Halaman *Login* – Kasir

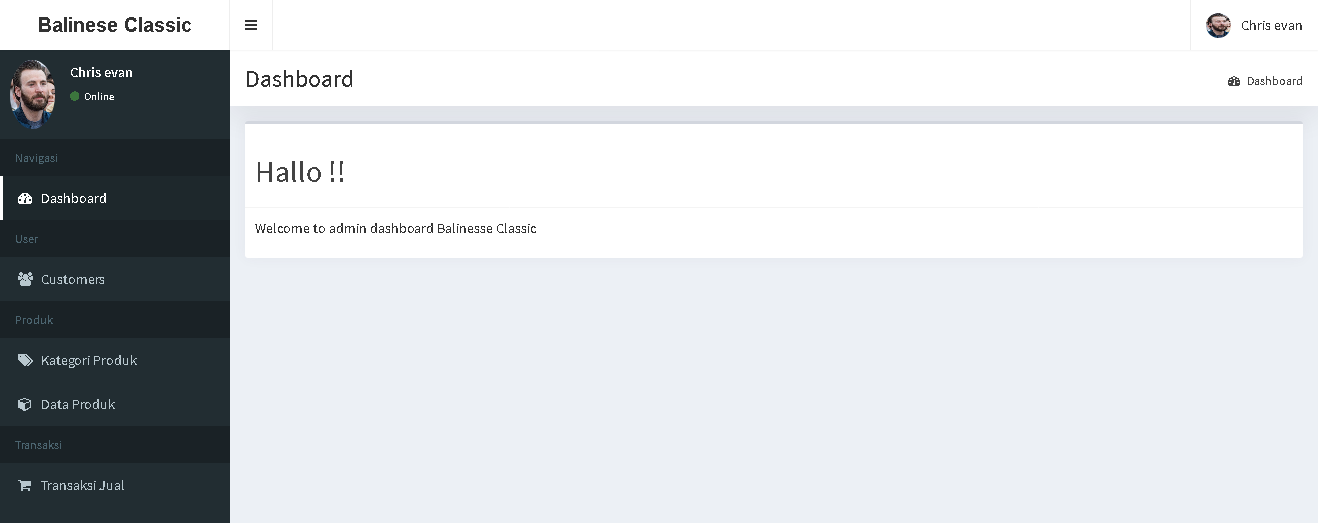
Pada halaman *login* sistem akan melakukan validasi agar Kasir dapat masuk ke halaman sistem.



Gambar 4.74 Tampilan Halaman *Login* – Kasir

1. Tampilan Halaman *Dashboard* – Kasir

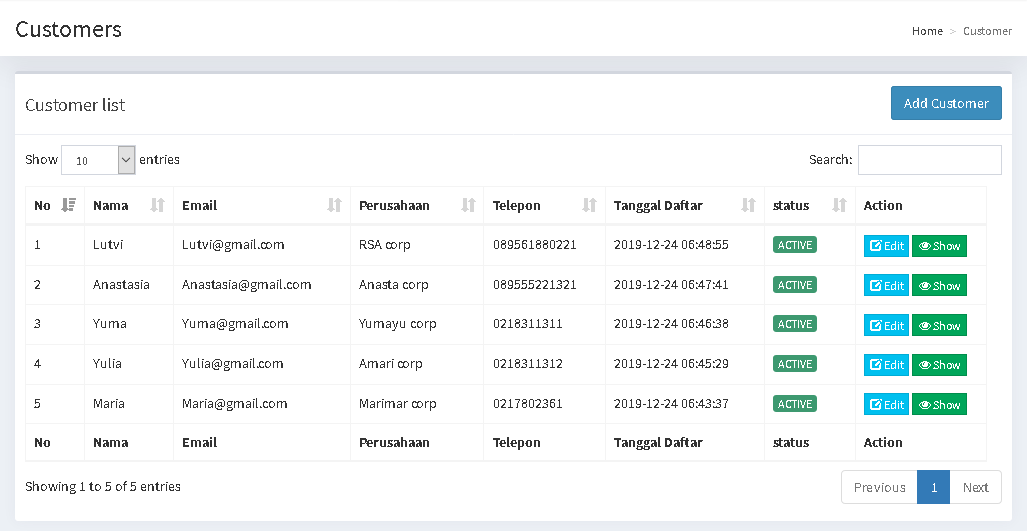
Halaman *Dashboard* merupakan tampilan setelah berhasil melakukan *login*.



Gambar 4.75 Tampilan Halaman *Dashboard* – Kasir

1. Tampilan Halaman *List* *Customers* – Kasir

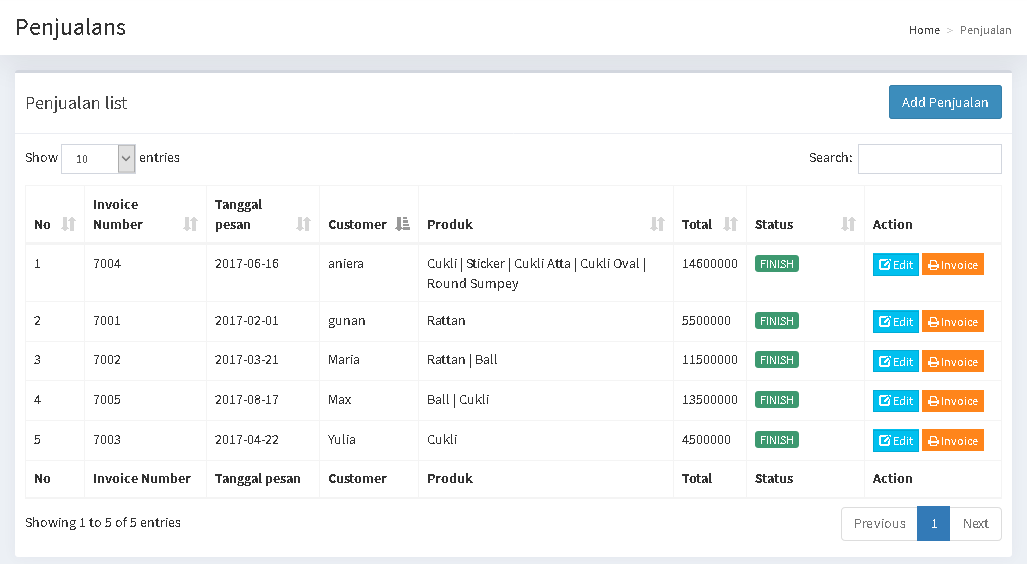
Halaman *List* *Customers* merupakan halaman setelah klik menu *Customers*. Pada halaman ini terdapat data-data *Customers* yang tersimpan di sistem.



Gambar 4.76 Tampilan Halaman *List* *Customers* – Kasir

1. Tampilan Halaman *List* Penjualan – Kasir

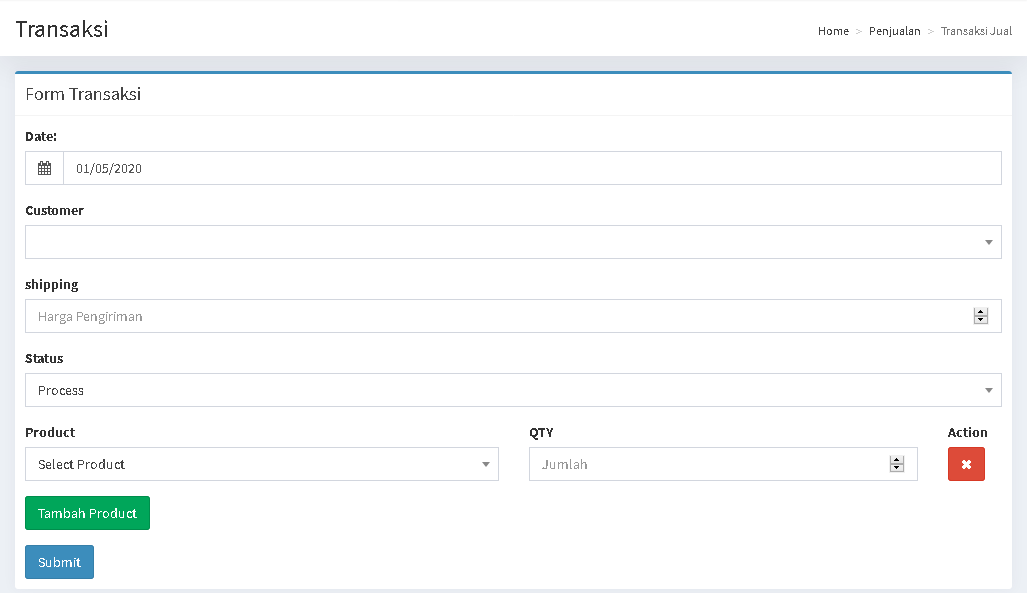
Halaman *List* Penjualan merupakan halaman setelah klik menu transaksi jual. Pada halaman ini terdapat data-data transaksi jual yang tersimpan di sistem.



Gambar 4.77 Tampilan Halaman *List* Penjualan – Kasir

1. Tampilan *Form* Penjualan – Kasir

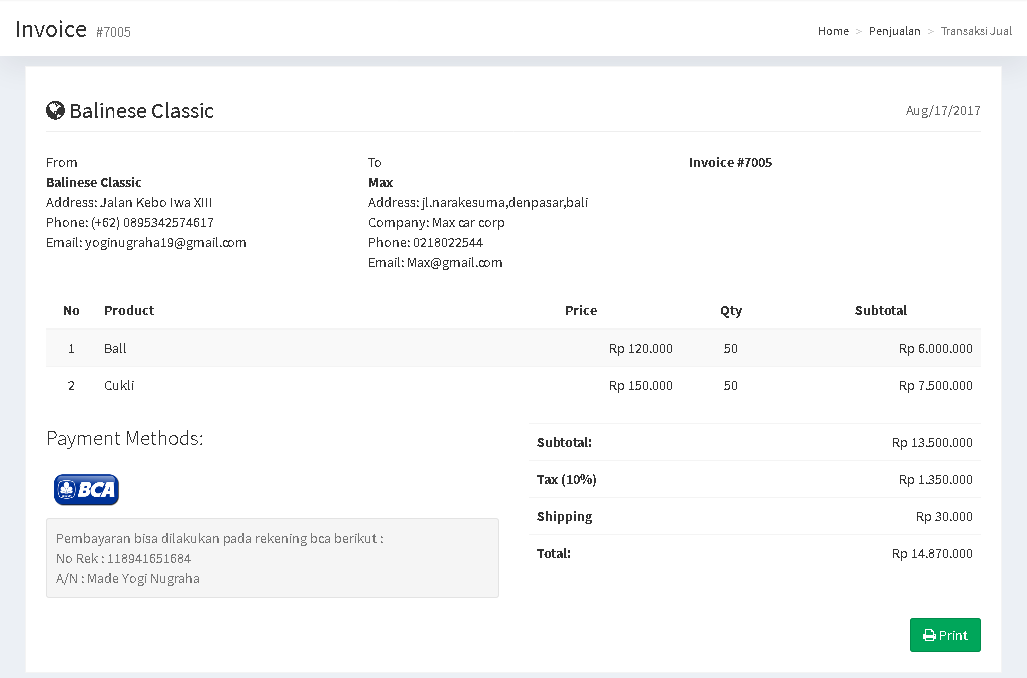
Pada halaman ini terdapat *form* yang wajib diisi saat melakukan transaksi. Tampilan *form* penjualan dapat dilihat pada gambar berikut:



Gambar 4.78 Tampilan *Form* Penjualan – Kasir

1. Tampilan Halaman *Invoice* Penjualan – Kasir

Tampilan Halaman *Invoice* Penjualan merupakan tampilan setelah klik tombol *invoice* pada halaman *list* penjualan.



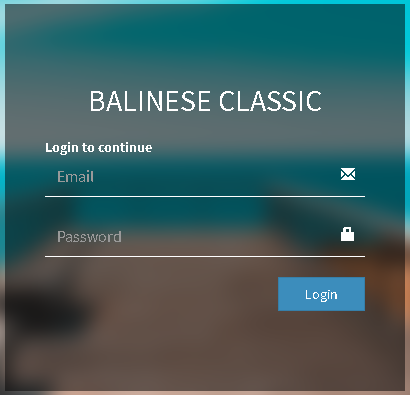
Gambar 4.79 Tampilan Halaman *Invoice* Penjualan – Kasir

1. Tampilan Untuk Gudang

Tampilan untuk Gudang, ditujukan untuk *user* yang memiliki *roles* Gudang, di mana terdapat tampilan sesuai dengan kebutuhannya. Adapun tampilannya sebagai berikut:

1. Tampilan Halaman *Login* – Gudang

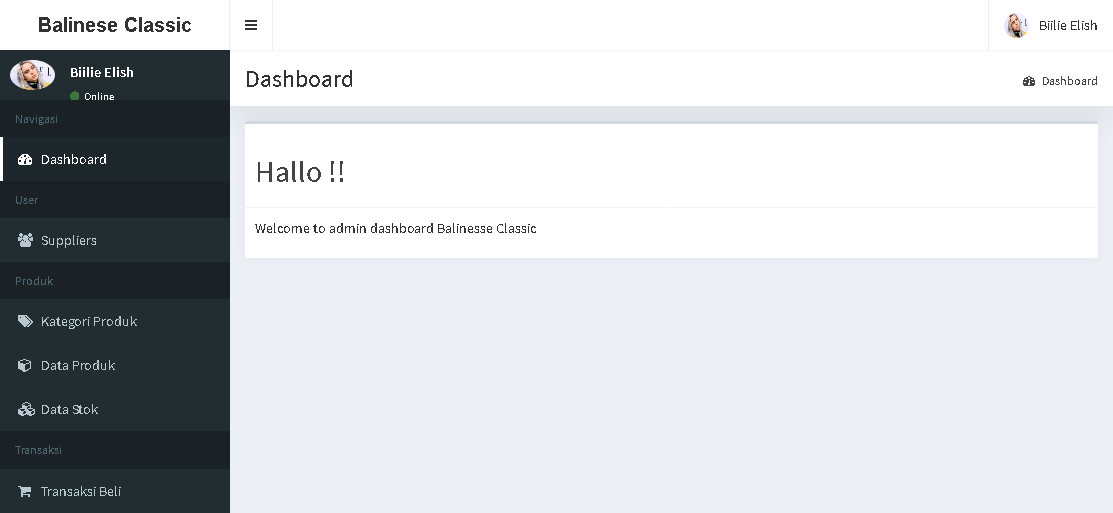
Pada halaman *login* sistem akan melakukan validasi agar Gudang dapat masuk ke halaman sistem.



Gambar 4.80 Tampilan Halaman *Login* – Gudang

1. Tampilan Halaman *Dashboard* – Gudang

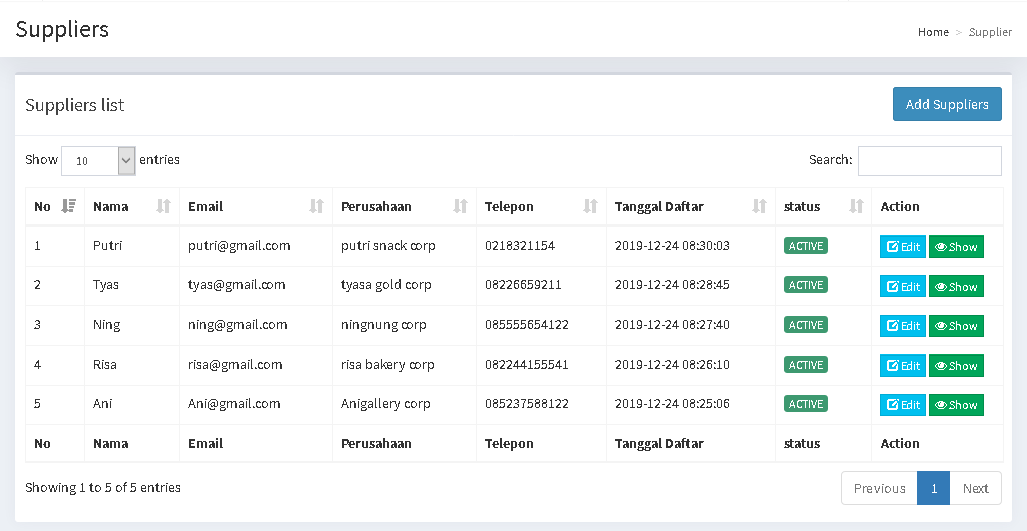
Halaman *Dashboard* merupakan tampilan setelah berhasil melakukan *login*.



Gambar 4.81 Tampilan Halaman *Dashboard* – Gudang

1. Tampilan Halaman *List* *Suppliers* – Gudang

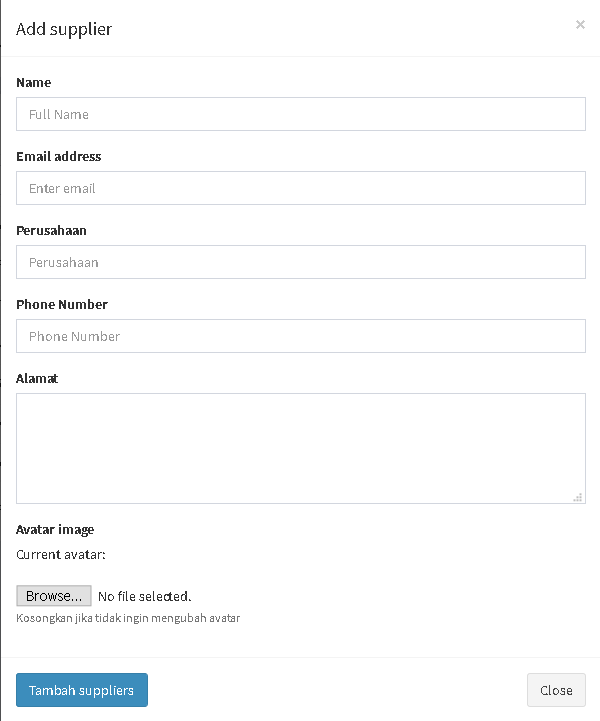
Halaman *List* *Suppliers* merupakan halaman setelah klik menu *Suppliers*. Pada halaman ini terdapat data-data *Suppliers* yang tersimpan di sistem.



Gambar 4.82 Tampilan Halaman *List* *Suppliers* – Gudang

1. Tampilan *Form* *Supplier* – Gudang

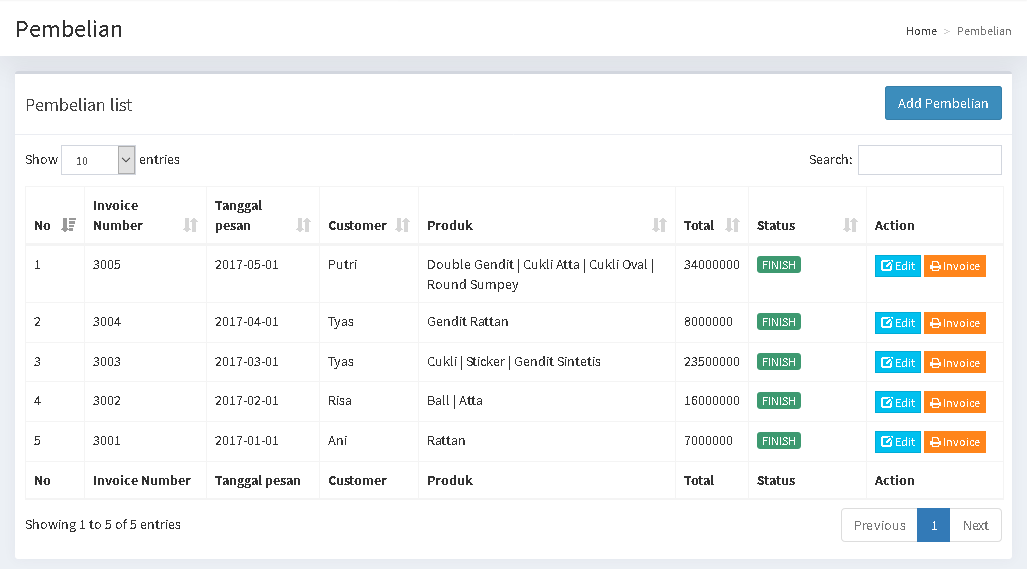
*Form* *Supplier* merupakan *form* yang muncul setelah klik tombol *add* *Supplier* pada halaman *list* *Suppliers*.



Gambar 4.83 Tampilan *Form* *Supplier* – Gudang

1. Tampilan Halaman *List* Pembelian – Gudang

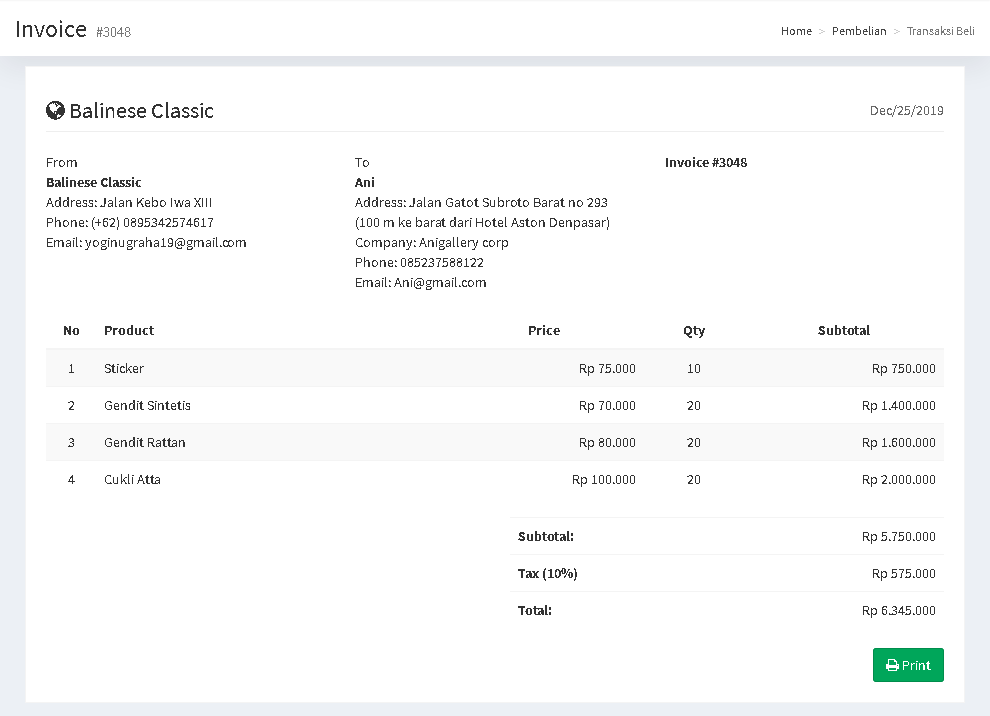
Halaman *List* Pembelian merupakan halaman setelah klik menu transaksi beli. Pada halaman ini terdapat data-data transaksi beli yang tersimpan di sistem.



Gambar 4.84 Tampilan Halaman *List* Pembelian – Gudang

1. Tampilan Halaman *Form* Pembelian – Gudang

Pada halaman ini terdapat *form* yang wajib diisi saat melakukan transaksi. Tampilan *form* pembelian dapat dilihat pada gambar berikut:



Gambar 4.85 Tampilan Halaman *Form* Pembelian – Gudang

1. Pengujian Black Box

Pada pengujian Black Box ini ditunjukkan untuk menemukan *error* dari sistem yang telah dibangun. Adapun penjelasannya sebagai berikut:

1. Pengujian Halaman Admin

Berikut adalah hasil dari pengujian *Blackbox* yang telah dilakukan sesuai dengan rencana pengujian:

1. Pengujian pada Halaman *Login* – Admin

Pengujian dilakukan pada tiga sampel data *input* untuk menguji fungsionalitas tombol *Login*. Hasil pengujian halaman *login* dapat dilihat pada Tabel 4.15.

Tabel 4.15 Hasil Pengujian Halaman *Login* – Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data *Input*** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| 1 | Admin memasukkan email dan *password* dengan benar kemudian tekan tombol *Login*. | Proses autentikasi *user* berhasil dan diarahkan ke halaman *Dashboard* | Admin berhasil *login* dan diarahkan ke halaman *Dashboard* | Sesuai |
| 2 | Admin memasukkan email dan *password* yang salah kemudian tekan tombol *Login*. | Muncul pesan pemberitahuan *login* gagal | Admin tidak berhasil *login* dan muncul pesan pemberitahuan bahwa *login* gagal | Sesuai |
| 3 | Admin tidak memasukkan email dan *password* kemudian tekan tombol *Login*. | Muncul pemberitahuan untuk mengisi *field* yang masih kosong. | Admin tidak berhasil *login* dan muncul pemberitahuan untuk melengkapi *field* yang masih kosong. | Sesuai |

1. Pengujian pada Navigasi Menu – Admin

Pengujian dilakukan pada sebelas menu untuk menguji fungsionalitasnya. Hasil pengujian navigasi menu dapat dilihat pada Tabel 4.16.

Tabel 4.16 Hasil Pengujian pada Navigasi Menu – Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| 1 | Klik menu *Dashboard*. | Menampilkan halaman *Dashboard*. | Admin berhasil diarahkan ke halaman *Dashboard*. | Sesuai |
| 2 | Klik menu *Users*. | Menampilkan halaman *Users*. | Admin berhasil diarahkan ke halaman *Users*. | Sesuai |
| 3 | Klik menu *Customers* | Menampilkan halaman *Customers*. | Admin berhasil diarahkan ke halaman *Customers*. | Sesuai |
| 4 | Klik menu *Suppliers*. | Menampilkan halaman *Suppliers*. | Admin berhasil diarahkan ke halaman *Suppliers*. | Sesuai |
| 5 | Klik menu *Categories*. | Menampilkan halaman *Categories*. | Admin berhasil diarahkan ke halaman *Categories*. | Sesuai |
| 6 | Klik menu *Products*. | Menampilkan halaman *Products*. | Admin berhasil diarahkan ke halaman *Products*. | Sesuai |
| 7 | Klik menu Penjualan. | Menampilkan halaman Penjualan. | Admin berhasil diarahkan ke halaman Penjualan. | Sesuai |
| 8 | Klik menu Laporan Penjualan. | Menampilkan halaman Laporan Penjualan. | Admin berhasil diarahkan ke halaman Laporan Penjualan. | Sesuai |

1. Pengujian pada Halaman *Users* – Admin

Pengujian dilakukan pada beberapa fungsi yang terdapat pada halaman *Users*. Hasil pengujian pada halaman *users* dapat dilihat pada Tabel 4.17.

Tabel 4.17 Pengujian pada Halaman *Users* – Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| 1 | Klik tombol *Add* *User* | Menampilkan modal *form* tambah *User*. | Berhasil membuka *form* untuk tambah data *User*. | Sesuai |
| 2 | Klik tombol *Edit* | Menampilkan modal *form* *edit* *user*. | Berhasil membuka *form* untuk *edit* data *User*. | Sesuai |
| 3 | Klik tombol *Show* | Menampilkan modal detail data *user*. | Berhasil membuka detail data dari *User*. | Sesuai |
| 4 | Cari data pada *field* *Search* | Menampilkan data sesuai isi *field* *Search*. | Berhasil menampilkan data *Users* sesuai dengan isi *field* *Search*. | Sesuai |

1. Pengujian *Form* Tambah pada Halaman *Users* – Admin

Pengujian dilakukan pada tiga sampel data *input* untuk menguji fungsionalitas tombol Tambah *User*. Hasil pengujian *form* tambah pada halaman *users* dapat dilihat pada Tabel 4.18.

Tabel 4.18 Pengujian *Form* Tambah pada Halaman *Users* – Admin

| **No** | **Data *Input*** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| --- | --- | --- | --- | --- |
| 1 | Admin memasukkan data dengan benar kemudian tekan tombol Tambah. | Proses *insert* data berhasil dan *pop up* *form* tambah menghilang. | Admin berhasil memasukkan data *users* dan *pop up* *form* tambah menghilang. | Sesuai |
| 2 | Admin memasukkan data yang salah kemudian tekan tombol Tambah. | Muncul peringatan bahwa masih ada *field* yang salah. | Admin tidak berhasil memasukkan data *user* dan muncul peringatan bahwa masih ada *field* yang salah. | Sesuai |
| 3 | Admin tidak memasukkan data dengan lengkap pada *field* kemudian tekan tombol Tambah. | Muncul peringatan bahwa masih ada *field* yang kosong. | Admin tidak berhasil memasukkan data *user* dan muncul peringatan bahwa masih ada *field* yang kosong. | Sesuai |

1. Pengujian pada Halaman *Customers* – Admin

Pengujian dilakukan pada beberapa fungsi yang terdapat pada halaman *Customers*. Hasil pengujian pada halaman *customers* dapat dilihat pada Tabel 4.19.

Tabel 4.19 Pengujian pada Halaman *Customers* – Admin

| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| --- | --- | --- | --- | --- |
| 1 | Klik tombol *Add* *Customer* | Menampilkan modal *form* tambah *Customer*. | Berhasil membuka *form* untuk tambah data *Customer*. | Sesuai |
| 2 | Klik tombol *Edit* | Menampilkan modal *form* *edit* *Customer*. | Berhasil membuka *form* untuk *edit* data *Customer*. | Sesuai |
| 3 | Klik tombol *Show* | Menampilkan modal detail data *Customer*. | Berhasil membuka detail data dari *Customer*. | Sesuai |
| 4 | Cari data pada *field* *Search* | Menampilkan data sesuai isi *field* *Search*. | Berhasil menampilkan data *Customer* sesuai dengan isi *field* *Search*. | Sesuai |

1. Pengujian *Form* Tambah pada Halaman *Customers* – Admin

Pengujian dilakukan pada tiga sampel data *input* untuk menguji fungsionalitas tombol Tambah *Customer*. Hasil pengujian *form* tambah pada halaman *Customers* dapat dilihat pada Tabel 4.20.

Tabel 4.20 Pengujian *Form* Tambah pada Halaman *Customers* – Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data *Input*** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| 1 | Admin memasukkan data dengan benar kemudian tekan tombol Tambah. | Proses *insert* data berhasil dan *pop up* *form* tambah menghilang. | Admin berhasil memasukkan data *customer* dan *pop up* *form* tambah menghilang. | Sesuai |
| 2 | Admin memasukkan data yang salah kemudian tekan tombol Tambah. | Muncul peringatan bahwa masih ada *field* yang salah. | Admin tidak berhasil memasukkan data *customer* dan muncul peringatan bahwa masih ada *field* yang salah. | Sesuai |
| 3 | Admin tidak memasukkan data dengan lengkap pada *field* kemudian tekan tombol Tambah. | Muncul peringatan bahwa masih ada *field* yang kosong. | Admin tidak berhasil memasukkan data *customer* dan muncul peringatan bahwa masih ada *field* yang kosong. | Sesuai |

1. Pengujian pada Halaman *Suppliers* – Admin

Pengujian dilakukan pada beberapa fungsi yang terdapat pada halaman *Suppliers*. Hasil pengujian pada halaman *suppliers* dapat dilihat pada Tabel 4.21.

Tabel 4.21 Pengujian pada Halaman *Suppliers* – Admin

| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| --- | --- | --- | --- | --- |
| 1 | Klik tombol *Add* *Supplier* | Menampilkan modal *form* tambah *Supplier*. | Berhasil membuka *form* untuk tambah data *Supplier*. | Sesuai |
| 2 | Klik tombol *Edit* | Menampilkan modal *form* *edit* *Supplier*. | Berhasil membuka *form* untuk *edit* data *Supplier*. | Sesuai |
| 3 | Klik tombol *Show* | Menampilkan modal detail data *Supplier*. | Berhasil membuka detail data dari *Supplier*. | Sesuai |
| 4 | Cari data pada *field* *Search* | Menampilkan data sesuai isi *field* *Search*. | Berhasil menampilkan data *Supplier* sesuai dengan isi *field* *Search*. | Sesuai |

1. Pengujian *Form* Tambah pada Halaman *Suppliers* – Admin

Pengujian dilakukan pada tiga sampel data *input* untuk menguji fungsionalitas tombol Tambah *Supplier*. Hasil pengujian *form* tambah pada halaman *Suppliers* dapat dilihat pada Tabel 4.22.

Tabel 4.22 Pengujian *Form* Tambah pada Halaman *Suppliers* – Admin

| **No** | **Data *Input*** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| --- | --- | --- | --- | --- |
| 1 | Admin memasukkan data dengan benar kemudian tekan tombol Tambah. | Proses *insert* data berhasil dan *pop up* *form* tambah menghilang. | Admin berhasil memasukkan data *supplier* dan *pop up* *form* tambah menghilang. | Sesuai |
| 2 | Admin memasukkan data yang salah kemudian tekan tombol Tambah. | Muncul peringatan bahwa masih ada *field* yang salah. | Admin tidak berhasil memasukkan data *supplier* dan muncul peringatan bahwa masih ada *field* yang salah. | Sesuai |
| 3 | Admin tidak memasukkan data dengan lengkap pada *field* kemudian tekan tombol Tambah. | Muncul peringatan bahwa masih ada *field* yang kosong. | Admin tidak berhasil memasukkan data *supplier* dan muncul peringatan bahwa masih ada *field* yang kosong. | Sesuai |

1. Pengujian pada Halaman *Categories* – Admin

Pengujian dilakukan pada beberapa fungsi yang terdapat pada halaman *Categories*. Hasil pengujian pada halaman *categories* dapat dilihat pada Tabel 4.23.

Tabel 4.23 Pengujian pada Halaman *Categories* – Admin

| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| --- | --- | --- | --- | --- |
| 1 | Klik tombol *Add* *Category* | Menampilkan modal *form* tambah *Category*. | Berhasil membuka *form* untuk tambah data *Category*. | Sesuai |
| 2 | Klik tombol *Edit* | Menampilkan modal *form* *edit* *Category*. | Berhasil membuka *form* untuk *edit* data *Category*. | Sesuai |
| 3 | Cari data pada *field* *Search* | Menampilkan data sesuai isi *field* *Search*. | Berhasil menampilkan data *Category* sesuai dengan isi *field* *Search*. | Sesuai |

1. Pengujian *Form* Tambah pada Halaman *Categories* – Admin

Pengujian dilakukan pada tiga sampel data *input* untuk menguji fungsionalitas tombol Tambah *Category*. Hasil pengujian *form* tambah pada halaman *Categories* dapat dilihat pada Tabel 4.24.

Tabel 4.24 Pengujian *Form* Tambah pada Halaman *Categories* – Admin

| **No** | **Data *Input*** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| --- | --- | --- | --- | --- |
| 1 | Admin memasukkan data dengan benar kemudian tekan tombol Tambah. | Proses *insert* data berhasil dan *pop up* *form* tambah menghilang. | Admin berhasil memasukkan data *category* dan *pop up* *form* tambah menghilang. | Sesuai |
| 2 | Admin memasukkan data yang salah kemudian tekan tombol Tambah. | Muncul peringatan bahwa masih ada *field* yang salah. | Admin tidak berhasil memasukkan data *category* dan muncul peringatan bahwa masih ada *field* yang salah. | Sesuai |
| 3 | Admin tidak memasukkan data dengan lengkap pada *field* kemudian tekan tombol Tambah. | Muncul peringatan bahwa masih ada *field* yang kosong. | Admin tidak berhasil memasukkan data *category* dan muncul peringatan bahwa masih ada *field* yang kosong. | Sesuai |

1. Pengujian pada Halaman *Products* – Admin

Pengujian dilakukan pada beberapa fungsi yang terdapat pada halaman *Products*. Hasil pengujian pada halaman *products* dapat dilihat pada Tabel 4.25.

Tabel 4.25 Pengujian pada Halaman *Products* – Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| 1 | Klik tombol *Add* *Product* | Menampilkan modal *form* tambah *Product*. | Berhasil membuka *form* untuk tambah data *Product*. | Sesuai |
| 2 | Klik tombol *Edit* | Menampilkan modal *form* *edit* *Product*. | Berhasil membuka *form* untuk *edit* data *Product*. | Sesuai |
| 3 | Cari data pada *field* *Search* | Menampilkan data sesuai isi *field* *Search*. | Berhasil menampilkan data *Product* sesuai dengan isi *field* *Search*. | Sesuai |

1. Pengujian *Form* Tambah pada Halaman *Products* – Admin

Pengujian dilakukan pada tiga sampel data *input* untuk menguji fungsionalitas tombol Tambah *Product*. Hasil pengujian *form* tambah pada halaman *Products* dapat dilihat pada Tabel 4.26.

Tabel 4.26 Pengujian *Form* Tambah pada Halaman *Products* – Admin

| **No** | **Data *Input*** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| --- | --- | --- | --- | --- |
| 1 | Admin memasukkan data dengan benar, tekan tombol Tambah. | Proses *insert* data berhasil dan *pop up* *form* tambah menghilang. | Admin berhasil memasukkan data *product* dan *pop up* *form* tambah menghilang. | Sesuai |
| 2 | Admin memasukkan data yang salah kemudian tekan tombol Tambah. | Muncul peringatan bahwa masih ada *field* yang salah. | Admin tidak berhasil memasukkan data *product* dan muncul peringatan bahwa masih ada *field* yang salah. | Sesuai |
| 3 | Admin tidak memasukkan data dengan lengkap pada *field* kemudian tekan tombol Tambah. | Muncul peringatan bahwa masih ada *field* yang kosong. | Admin tidak berhasil memasukkan data *product* dan muncul peringatan bahwa masih ada *field* yang kosong. | Sesuai |

1. Pengujian pada Halaman *Stock*s – Admin

Pengujian dilakukan pada beberapa fungsi yang terdapat pada halaman *Stock*s. Hasil pengujian pada halaman *stock* dapat dilihat pada Tabel 4.27.

Tabel 4.27 Pengujian pada Halaman *Stock*s – Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| 1 | Klik tombol *Edit* | Menampilkan modal *form* *edit* *Stock*. | Berhasil membuka *form* untuk *edit* data *Stock*. | Sesuai |
| 2 | Pilih *select* *option* Entries | Menampilkan *list* data *Stock* sesuai pilihan. | Berhasil menampilkan banyaknya *list* data *Stock* sesuai pilihan. | Sesuai |
| 3 | Cari data pada *field* *Search* | Menampilkan data sesuai isi *field* *Search*. | Berhasil menampilkan data *Stock* sesuai dengan isi *field* *Search*. | Sesuai |
| 4 | Klik tombol *Pagination* | Menampilkan data selanjutnya. | Berhasil menampilkan data sesuai dengan halaman *Pagination*. | Sesuai |

1. Pengujian pada Halaman Penjualan – Admin

Pengujian dilakukan pada beberapa fungsi yang terdapat pada halaman Penjualan. Hasil pengujian pada halaman Penjualan dapat dilihat pada Tabel 4.28.

Tabel 4.28 Pengujian pada Halaman Penjualan – Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| 1 | Klik tombol *Add* Penjualan | Menampilkan Halaman tambah transaksi jual | Berhasil menampilkan Halaman tambah transaksi jual | Sesuai |
| 2 | Klik tombol *Edit* | Menampilkan Halaman *edit* transaksi jual | Berhasil menampilkan Halaman *edit* transaksi jual | Sesuai |
| 3 | Klik tombol *Invoice* | Menampilkan Halaman *invoice* transaksi jual | Berhasil Menampilkan Halaman *invoice* transaksi jual | Sesuai |
| 4 | Cari data pada *field* *Search* | Menampilkan data sesuai isi *field* *Search*. | Berhasil menampilkan data Penjualan sesuai dengan isi *field* *Search*. | Sesuai |

1. Pengujian pada Halaman Tambah Transaksi Jual – Admin

Pengujian dilakukan pada tiga sampel data *input* untuk menguji fungsionalitas tombol *Submit* Hasil pengujian pada halaman Transaksi Jual dapat dilihat pada Tabel 4.29.

Tabel 4.29 Pengujian pada Halaman Tambah Transaksi Jual – Admin

| **No** | **Data *Input*** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| --- | --- | --- | --- | --- |
| 1 | Admin memasukkan data dengan benar kemudian tekan tombol *Submit*. | Proses *insert* data berhasil dan isi *field* di*reset*. | Admin berhasil memasukkan data penjualan dan isi *field* di*reset*. | Sesuai |
| 2 | Admin memasukkan data yang salah kemudian tekan tombol *Submit*. | Muncul peringatan bahwa masih ada *field* yang salah. | Admin tidak berhasil memasukkan data penjualan dan muncul peringatan bahwa masih ada *field* yang salah. | Sesuai |
| 3 | Admin tidak memasukkan data dengan lengkap pada *field* kemudian tekan tombol *Submit*. | Muncul peringatan bahwa masih ada *field* yang kosong. | Admin tidak berhasil memasukkan data penjualan dan muncul peringatan bahwa masih ada *field* yang kosong. | Sesuai |

1. Pengujian pada Halaman Pembelian – Admin

Pengujian dilakukan pada beberapa fungsi yang terdapat pada halaman Pembelian. Hasil pengujian pada halaman Pembelian dapat dilihat pada Tabel 4.30.

Tabel 4.30 Pengujian pada Halaman Pembelian – Admin

| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| --- | --- | --- | --- | --- |
| 1 | Klik tombol *Add* Pembelian | Menampilkan Halaman tambah transaksi beli | Berhasil menampilkan Halaman tambah transaksi beli | Sesuai |
| 2 | Klik tombol *Edit* | Menampilkan Halaman *edit* transaksi beli | Berhasil menampilkan Halaman *edit* transaksi beli | Sesuai |
| 3 | Klik tombol *Invoice* | Menampilkan Halaman *invoice* transaksi beli | Berhasil Menampilkan Halaman *invoice* transaksi beli | Sesuai |
| 4 | Pilih *select* *option* Entries | Menampilkan *list* data Penjualan sesuai pilihan. | Berhasil menampilkan banyaknya *list* data Pembelian sesuai pilihan. | Sesuai |
| 5 | Cari data pada *field* *Search* | Menampilkan data sesuai isi *field* *Search*. | Berhasil menampilkan data Pembelian sesuai dengan isi *field* *Search*. | Sesuai |
| 6 | Klik tombol *Pagination* | Menampilkan data selanjutnya. | Berhasil menampilkan data sesuai dengan halaman *Pagination*. | Sesuai |

1. Pengujian pada Halaman Tambah Transaksi Beli – Admin

Pengujian dilakukan pada tiga sampel data *input* untuk menguji fungsionalitas tombol *Submit* Hasil pengujian pada halaman Tambah Transaksi Beli dapat dilihat pada Tabel 4.31.

Tabel 4.31 Pengujian pada Halaman Tambah Transaksi Beli – Admin

| **No** | **Data *Input*** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| --- | --- | --- | --- | --- |
| 1 | Admin memasukkan data dengan benar kemudian tekan tombol *Submit*. | Proses *insert* data berhasil dan isi *field* di*reset*. | Admin berhasil memasukkan data pembelian dan isi *field* di*reset*. | Sesuai |
| 2 | Admin memasukkan data yang salah kemudian tekan tombol *Submit*. | Muncul peringatan bahwa masih ada *field* yang salah. | Admin tidak berhasil memasukkan data pembelian dan muncul peringatan bahwa masih ada *field* yang salah. | Sesuai |
| 3 | Admin tidak memasukkan data dengan lengkap pada *field* kemudian tekan tombol *Submit*. | Muncul peringatan bahwa masih ada *field* yang kosong. | Admin tidak berhasil memasukkan data pembelian dan muncul peringatan bahwa masih ada *field* yang kosong. | Sesuai |

1. Pengujian pada Halaman Laporan Penjualan – Admin

Pengujian dilakukan pada beberapa fungsi yang terdapat pada halaman Laporan Penjualan. Hasil pengujian pada halaman Laporan Penjualan dapat dilihat pada Tabel 4.32.

Tabel 4.32 Pengujian pada Halaman Laporan Penjualan – Admin

| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| --- | --- | --- | --- | --- |
| 1 | Pilih tahun dan klik tombol Submit. | Menampilkan data grafik sesuai dengan tahun yang dipilih. | Berhasil Menampilkan data grafik sesuai dengan tahun yang dipilih. | Sesuai |
| 2 | Pilih tanggal dan klik tombol Filter. | Menampilkan tabel sesuai dengan tanggal yang dipilih | Berhasil menampilkan data tabel sesuai dengan tanggal yang dipilih | Sesuai |
| 3 | Klik tombol excel | *Download* data tabel bertipe xlsx | Berhasil *Download* data tabel bertipe xlsx | Sesuai |
| 4 | Klik tombol print | Print data tabel |  | Sesuai |
| 5 | Klik tombol *Add* Penjualan | Menampilkan Halaman tambah transaksi jual | Berhasil menampilkan Halaman tambah transaksi jual | Sesuai |
| 6 | Cari data pada *field* *Search* | Menampilkan data sesuai isi *field* *Search*. | Berhasil menampilkan data Penjualan sesuai dengan isi *field* *Search*. | Sesuai |

1. Pengujian pada Halaman Laporan Pembelian – Admin

Pengujian dilakukan pada beberapa fungsi yang terdapat pada halaman Laporan Pembelian. Hasil pengujian pada halaman Laporan Pembelian dapat dilihat pada Tabel 4.33.

Tabel 4.33 Pengujian pada Halaman Laporan Pembelian – Admin

| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| --- | --- | --- | --- | --- |
| 1 | Pilih tahun dan klik tombol Submit. | Menampilkan data grafik sesuai dengan tahun yang dipilih. | Berhasil Menampilkan data grafik sesuai dengan tahun yang dipilih. | Sesuai |
| 2 | Pilih tanggal dan klik tombol Filter. | Menampilkan tabel sesuai dengan tanggal yang dipilih | Berhasil menampilkan data tabel sesuai dengan tanggal yang dipilih | Sesuai |
| 3 | Klik tombol excel | *Download* data tabel bertipe xlsx | Berhasil *Download* data tabel bertipe xlsx | Sesuai |
| 4 | Klik tombol print | Print tabel |  | Sesuai |
| 5 | Klik tombol *Add* Pembelian | Menampilkan Halaman tambah transaksi beli | Berhasil menampilkan Halaman tambah transaksi beli | Sesuai |

1. Pengujian Halaman Kasir

Berikut adalah hasil dari pengujian *Blackbox* yang telah dilakukan sesuai dengan rencana pengujian:

1. Pengujian pada Halaman *Login* – Kasir

Pengujian dilakukan pada tiga sampel data *input* untuk menguji fungsionalitas tombol *Login*. Hasil pengujian halaman *login* dapat dilihat pada Tabel 4.34.

Tabel 4.34 Hasil Pengujian Halaman *Login* – Kasir

| **No** | **Data *Input*** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| --- | --- | --- | --- | --- |
| 1 | Kasir memasukkan email dan *password* dengan benar kemudian tekan tombol *Login*. | Proses autentikasi *user* berhasil dan diarahkan ke halaman *Dashboard* | Kasir berhasil *login* dan diarahkan ke halaman *Dashboard* | Sesuai |
| 2 | Kasir memasukkan email dan *password* yang salah kemudian tekan tombol *Login*. | Muncul pesan pemberitahuan *login* gagal | Kasir tidak berhasil *login* dan muncul pesan pemberitahuan bahwa *login* gagal | Sesuai |
| 3 | Kasir tidak memasukkan email dan *password* kemudian tekan tombol *Login*. | Muncul pemberitahuan untuk mengisi *field* yang masih kosong. | Kasir tidak berhasil *login* dan muncul pemberitahuan untuk melengkapi *field* yang masih kosong. | Sesuai |

1. Pengujian Halama Gudang

Berikut adalah hasil dari pengujian *Blackbox* yang telah dilakukan sesuai dengan rencana pengujian:

1. Pengujian pada Halaman *Login* – Gudang

Pengujian dilakukan pada tiga sampel data *input* untuk menguji fungsionalitas tombol *Login*. Hasil pengujian halaman *login* dapat dilihat pada Tabel 4.35.

Tabel 4.35 Hasil Pengujian Halaman *Login* – Gudang

| **No** | **Data *Input*** | **Hasil yang Diharapkan** | **Hasil Pengamatan** | **Kesimpulan** |
| --- | --- | --- | --- | --- |
| 1 | Gudang memasukkan email dan *password* dengan benar kemudian tekan tombol *Login*. | Proses autentikasi *user* berhasil dan diarahkan ke halaman *Dashboard* | Gudang berhasil *login* dan diarahkan ke halaman *Dashboard* | Sesuai |
| 2 | Gudang memasukkan email dan *password* yang salah kemudian tekan tombol *Login*. | Muncul pesan pemberitahuan *login* gagal | Gudang tidak berhasil *login* dan muncul pesan pemberitahuan bahwa *login* gagal | Sesuai |
| 3 | Gudang tidak memasukkan email dan *password* kemudian tekan tombol *Login*. | Muncul pemberitahuan untuk mengisi *field* yang masih kosong. | Gudang tidak berhasil *login* dan muncul pemberitahuan untuk melengkapi *field* yang masih kosong. | Sesuai |

***Halaman ini sengaja dikosongkan***

# BAB V PENUTUP

## **Kesimpulan**

Berdasarkan hasil penelitian yang telah dilakukan pada Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel, maka dapat diambil kesimpulan sebagai berikut:

1. Telah dihasilkan suatu Sistem Informasi Eksekutif Penjualan Kerajinan Tas Rotan Pada Balinese Classic Menggunakan *Framework* Laravel.
2. Sistem ini telah melalui proses perancangan dengan menggunakan *Data* *Flow* *Diagram* (DFD), perancangan basis data dengan menggunakan *Entity* *Relationship* *Diagram* (ERD), Basis Data Konseptual dan Struktur Tabel.
3. Sistem berhasil dibangun berbasis Bahasa pemrograman PHP menggunakan *Framework* Laravel dikombinasikan dengan Bahasa pemrograman Javascript Ajax yang dimaksimalkan dengan *Framework* CSS yaitu Bootstrap.
4. Sistem ini memiliki beberapa *fitur* yaitu mengelola data seperti data *users*, data *product*, data transaksi, dan data laporan.
5. Sistem ini menghasilkan *output* berupa laporan penjualan dan laporan pembelian.
6. Sistem ini telah diuji menggunakan metode *Blackbox* Testing dan seluruh pengujian yang dilakukan telah memperoleh hasil yang sesuai.

## **Saran**

Adapun saran yang dapat diberikan untuk menyempurnakan perancangan ini agar sistem menjadi lebih baik adalah sebagai berikut:

1. Sistem ini dapat dikembangkan menjadi aplikasi berbasis *Mobile* *Application* sehingga lebih praktis dalam penggunaannya.
2. Sistem ini dapat dikembangkan agar memiliki fitur *scan* *barcode* untuk setiap *product*.
3. Menambahkan fitur lacak pesanan agar lebih mudah dalam mengetahui posisi pesanan yang sedang dikirim.

***Halaman ini sengaja dikosongkan***

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | Y. Maulana, "Sistem Informasi Penjualan Kayu Ulin Pada UD Karya Perdana Berbasis Web Menggunakan Framework Codeigniter", Denpasar: (STIMIK) STIKOM Bali, 2018. |
| [2] | I. K. A. Adnyana, "Sistem Informasi Eksekutif Penjualan Jajanan Sagu Pada Bagia Suli Menggunakan Framework Laravel", Denpasar: (STMIK) STIKOM Bali, 2018. |
| [3] | K. O. E. Artawan, "Sistem Informasi Penyewaan Busana Pada Denata Kebaya Menggunakan Framwork Laravel", Denpasar: (STMIK) STIKOM Bali, 2019. |
| [4] | I. P. Kawiyana, "Sistem Informasi Eksekutif Pengadaan Barang Pada Saturday Cloth Menggunakan Framework Laravel", Denpasar: (STIMIK) STIKOM Bali, 2019. |
| [5] | S. Setyaningrum, Konsep dan Perancangan Basis Data, Yoyakarta: PT. Skripta Media Creativ, 2011. |
| [6] | J. S. P. Tyoso, Sistem Informasi Manajemen, Yogyakarta: CV BUDI UTAMA, 2016. |
| [7] | A. N. Wayan, "Sistem Informasi Eksekutif Bantuan Kelompok Usaha Bersama (KUBE) Pada Dinas SOsial Provinsi Bali", Denpasar: (STMIK) STIKOM Bali, 2017. |
| [8] | R. Budi, Belajar Otodidak MySQL, Bandung: Informatika Bandung, 2015. |
| [9] | N. K. D. A. Jayanti, Teori Basis Data, Denpasar: CV Andi Offset, 2018. |
| [10] | A. Subagia, Membuat Web dengan PHP 7 dab Database PDO MySQLi, Jakarta: PT Elex Komputindo, 2016. |
| [11] | Sudarmaji, "Migrasi dan Optimalisasi Database Sistem Informasi berbasis E-Learning Program Diploma III Manajemen Informatika", Universitas Muhammadiyah Metro, 2017. |
| [12] | A. Rohman, Mengenal Framwork Laravel, Ilmu TI, 2014. |
| [13] | J. Spurlock, Responsive Web Development, O'Reilly Media, 2013. |
| [14] | Pratama. E, Sistem Informasi dan Implementasinya, Bandung: Informatika Bandung, 2014. |
| [15] | F. H. U. Asnawati, Rekayasa Perangkat Lunak, Yogyakarta: CV Budi Utama, 2015. |