

DAFTAR ISI

HALAMAN Sampul	i
HALAMAN Judul.....	ii
HALAMAN Persetujuan.....	iii
HALAMAN Pengesahan.....	iv
HALAMAN Surat Pernyataan Keaslian	v
HALAMAN Persembahan.....	vi
HALAMAN Motto	vii
Kata Pengantar	viii
DAFTAR ISI.....	x
DAFTAR Tabel.....	xi
DAFTAR Gambar	xii
DAFTAR Lampiran.....	xiii
Intisari.....	xiv
ABSTRACT.....	xv
BAB I PENDAHULUAN	
A. Latar Belakang Masalah.....	1
B. Rumusan Masalah.....	2
C. Batasan Masalah.....	3
D. Tujuan Penelitian	3
E. Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA	
A. Landasan Teori	6
1. Pengertian Implementasi	5
2. Pengertian Framework Laravel	6
3. Pengertian Aplikasi.....	7
4. Pengertian Web	7
5. Pengertian <i>PHP</i>	8
6. Pengertian <i>Mysql</i>	9

	7. Definisi <i>XAMPP</i>	9
	8. <i>UML (Unified Modeling Language)</i>	10
	9. <i>Entity Relationship Diagram (ERD)</i>	17
	10. Pengertian Transaksi	21
	B. Penelitian Sebelumnya	22
BAB III	METODE PENELITIAN	
	A. Tempat Dan Waktu Penelitian	25
	B. Metode Pengumpulan Data	25
	C. Alat Dan Bahan Penelitian	26
	D. Konsep Penelitian.....	27
BAB IV	HASIL DAN PEMBAHASAN	
	A. Analisis Hasil	32
	1. Analisis Identifikasi Masalah	32
	2. Analisis Kebutuhan Sistem.....	33
	a. Analisis Kebutuhan Fungsional	33
	b. Analisis Kebutuhan Non Fungsional.....	34
	B. Desain.....	35
	1. Perancangan <i>Unified Modelling Language (UML)</i>	35
	a. Use Case Diagram	35
	b. <i>Activity Diagram</i>	37
	c. <i>Sequence Diagram</i>	40
	d. <i>Class Diagram</i>	42
	2. <i>Entity Relationship Diagram (ERD)</i>	43
	3. Perancangan <i>Database</i>	44
	a. Struktur Tabel.....	44
	4. Perancangan Tampilan Antarmuka	47
	C. Pengkodean	51
	1. Implementasi Halaman Antarmuka	52
	D. Pengujian.....	55
	1. Rancangan Pengujian <i>Black-Box</i>	55
	2. Hasil Pengujian <i>Black-Box</i>	56

BAB V	PENUTUP	
	A. Kesimpulan	58
	B. Saran.....	59
	DAFTAR PUSTAKA	
	LAMPIRAN	



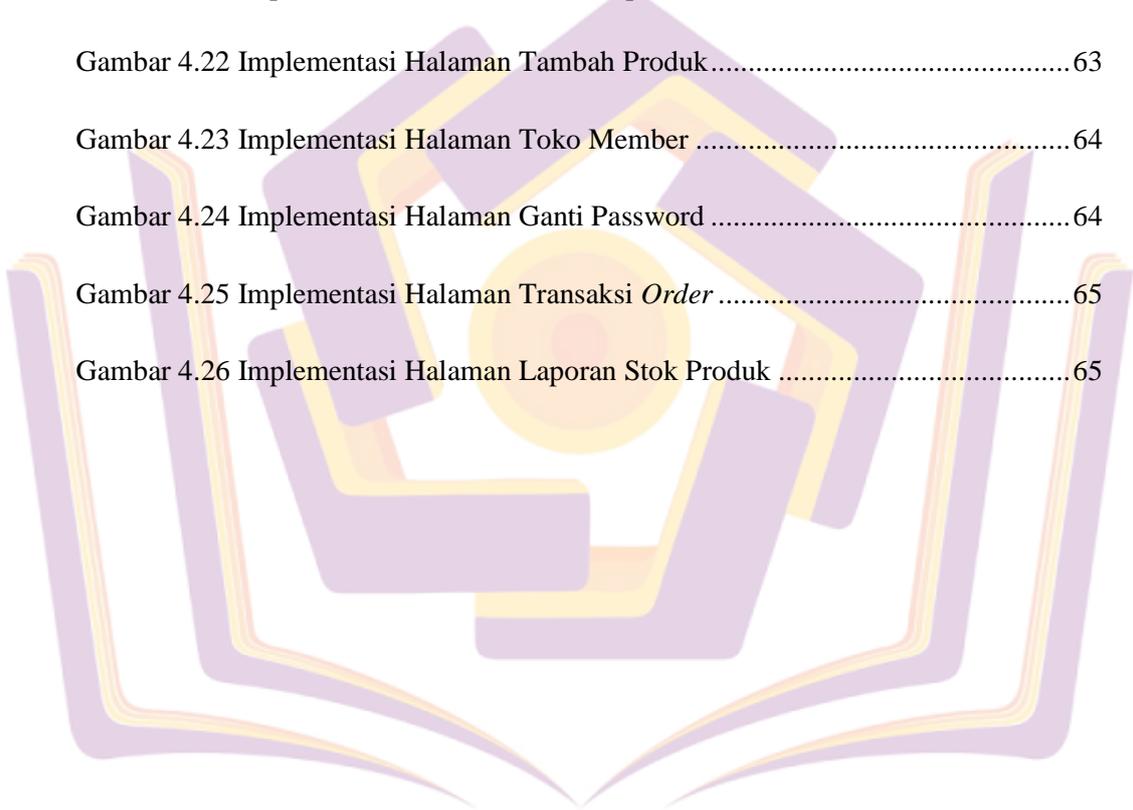
DAFTAR TABEL

Tabel 2.1	Simbol-simbol <i>Use Case Diagram</i>	12
Tabel 2.2	Simbol-simbol <i>Activity Diagram</i>	14
Tabel 2.3	Simbol-simbol <i>Sequence Diagram</i>	15
Tabel 2.4	Simbol-simbol <i>Class Diagram</i>	17
Tabel 2.5	Simbol-simbol ERD.....	18
Tabel 2.6	Hasil Penelitian Sebelumnya.....	23
Tabel 4.1	Tabel Definisi Aktor.....	39
Tabel 4.2	Struktur Tabel Users.....	53
Tabel 4.3	Struktur Tabel Shops.....	53
Tabel 4.4	Struktur Tabel Products.....	54
Tabel 4.5	Struktur Tabel Orders.....	54
Tabel 4.6	Struktur Tabel Migrations.....	54
Tabel 4.7	Struktur Tabel Fax Orders.....	55
Tabel 4.8	Struktur Tabel Categories.....	55
Tabel 4.9	Struktur Tabel Rencana Pengujian.....	56
Tabel 4.10	Struktur Tabel Pengujian Halaman Login.....	56
Tabel 4.11	Struktur Tabel Pengujian Halaman Tambah Produk.....	57
Tabel 4.12	Struktur Tabel Pengujian Halaman Transaksi Order.....	57

DAFTAR GAMBAR

Gambar 2.1 Kardinalitas Relasi <i>One To One</i>	28
Gambar 2.1 Kardinalitas Relasi <i>One To Many</i>	28
Gambar 2.3 Kardinalitas Relasi <i>Many To One</i>	28
Gambar 2.4 Kardinalitas Relasi <i>Many to Many</i>	28
Gambar 3.1 Diagram Tahapan dalam Metode Waterfall.....	28
Gambar 4.1 <i>Use Case Diagram</i> Aplikasi	40
Gambar 4.2 <i>Activity Diagram</i> Login.....	41
Gambar 4.3 <i>Activity Diagram</i> Input Data Produk	42
Gambar 4.4 <i>Activity Diagram</i> Input Data Toko	42
Gambar 4.5 <i>Activity Diagram</i> Input Barang Penjualan	43
Gambar 4.6 <i>Activity Diagram</i> Pembuatan Faktur.....	44
Gambar 4.7 <i>Sequence Diagram</i> Login	45
Gambar 4.8 <i>Sequence Diagram</i> input data produk.....	46
Gambar 4.9 <i>Sequence Diagram</i> Input data toko.....	47
Gambar 4.10 <i>Sequence Diagram</i> Transaksi Penjualan Barang	47
Gambar 4.11 <i>Class Diagram</i> Aplikasi	48
Gambar 4.12 <i>Entity Relationship Diagram</i>	48
Gambar 4.13 Desain Halaman <i>Login</i>	49
Gambar 4.14 Desain Halaman Utama Aplikasi.....	49
Gambar 4.15 Desain Halaman Tambah Produk	50

Gambar 4.16 Desain Halaman Tambah Data Toko Member.....	51
Gambar 4.17 Desain Halaman Ganti Password.....	52
Gambar 4.18 Desain Halaman Transaksi <i>Order</i>	60
Gambar 4.19 Desain Halaman Laporan Data Produk.....	61
Gambar 4.20 Implementasi Halaman Login.....	62
Gambar 4.21 Implementasi Halaman Utama Aplikasi	62
Gambar 4.22 Implementasi Halaman Tambah Produk.....	63
Gambar 4.23 Implementasi Halaman Toko Member	64
Gambar 4.24 Implementasi Halaman Ganti Password	64
Gambar 4.25 Implementasi Halaman Transaksi <i>Order</i>	65
Gambar 4.26 Implementasi Halaman Laporan Stok Produk	65



DAFTAR LAMPIRAN

Lampiran 1. Kartu Bimbingan Skripsi

Lampiran 2. Koding Program



BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Dengan semakin ketatnya persaingan bisnis di dalam dunia usaha, kecepatan dan ketepatan dalam bertindak merupakan suatu hal yang utama. Pengelolaan yang baik pada suatu toko atau warung sangat diperlukan untuk memperlancar kinerja pada proses jual beli di sebuah toko ataupun warung. Ada beberapa sistem yang dapat di terapkan di sebuah toko atau warung, salah satunya adalah sistem *inventory* barang, yang berfungsi untuk mengetahui jumlah barang pada gudang dan manajemen pemasaran barang.

Toko “Kuat” merupakan sebuah usaha dalam bidang penjualan barang-barang elektronik, sembako, dan barang-barang kebutuhan sehari-hari yang terletak di jalan raya pasar tambak Karangpucung Rt.03 Rw.03 Kabupaten Banyumas Provinsi Jawa Tengah. Saat ini sistem yang berjalan masih memiliki banyak kekurangan dan kelemahan seperti pada proses pemesanan barang, transaksi barang, pengiriman barang, penerimaan barang ke pelanggan, pengelolaan stok barang sembako. Pemilik toko menghadapi banyak kendala dalam pelayanan terhadap konsumennya dikarenakan dalam proses penjualan barang sembako tersebut, seperti proses pemesanan yang memakan cukup waktu untuk satu pelanggan, banyaknya pelanggan yang memesan dalam satu waktu, menyiapkan barang untuk pelangganya, menghitung biaya yang harus dibayar, mengantar barang ke tempat pelanggan, serta proses pembayaran pelanggan yang

tidak selalu membayar lunas saat penerimaan barang. Transaksi yang berjalan selama ini masih menggunakan kertas dan bolpoin sehingga dalam proses tersebut masih terdapat kesalahan penghitungan dan pencatatan laporan.

Untuk memecahkan masalah tersebut maka dapat diupayakan sebuah solusi untuk dapat mengurangi kelemahan-kelemahan yang ada, juga untuk mencapai target pelayanan yang maksimal maka penulis tertarik untuk mengembangkan sebuah sistem informasi yang dapat memanajemen permasalahan-permasalahan yang ada di toko tersebut, sistem ini akan mempermudah pemilik toko dalam mengelola stok barang secara elektronik atau online, mengelola proses pemesanan pelanggan ke toko dan membuat pelaporan produk, pembelian produk dan pendistribusian produk ke pelanggan.

Berdasarkan latar belakang yang dikemukakan di atas, maka penulis menganalisis dan merancang aplikasi **“Implementasi Framework Laravel untuk membangun aplikasi Manajemen Transaksi pada Toko Kuat Karangpucung Tambak Banyumas”**.

B. Rumusan Masalah

Berdasarkan latar belakang yang dikemukakan di atas maka dapat diambil suatu rumusan masalah yaitu bagaimana mengimplementasikan Framework Laravel untuk membangun aplikasi manajemen transaksi pada Toko Kuat Karangpucung, Tambak, Banyumas.

C. Batasan Masalah

Dari rumusan masalah tersebut, maka permasalahan yang akan dibahas dibatasi hanya mencakup hal-hal sebagai berikut:

1. Stok barang sembako.
2. Proses pemesanan pelanggan ke toko.

D. Tujuan Penelitian

Adapun tujuan dari implementasi Framework Laravel untuk membangun aplikasi manajemen transaksi pada Toko Kuat Karangpucung, Tambak, Banyumas adalah:

1. Memberi kemudahan pada pemilik toko dalam melakukan proses pencatatan stok barang secara online.
2. Memberi kemudahan pada pemilik toko dalam melakukan proses pemesanan pelanggan ke toko.
3. Memberi kemudahan pada pemilik toko dalam melakukan proses pendistribusian barang-barang ke pelanggan
4. Memberi kemudahan pada pemilik toko dalam pembuatan pelaporan setiap bulan.

E. Manfaat Penelitian

Adapun kegunaan yang dapat diambil dan diharapkan dari penyusunan perancangan aplikasi ini adalah:

1. Manfaat Teoritik

a) Dapat digunakan sebagai sumber informasi mengenai objek penelitian dan sebagai sarana untuk menambah wawasan, yang terkait dengan mengimplementasikan framework Laravel untuk membangun aplikasi transaksi.

b) Penelitian ini juga dapat digunakan sebagai bahan perbandingan dengan pengetahuan yang sudah ada dan terkait dengan pembuatan laporan sehingga diharapkan pula dapat digunakan sebagai contoh dan acuan bagi para peneliti selanjutnya.

2. Manfaat Aplikatif

a) Untuk menghasilkan suatu aplikasi penjualan yang mempermudah proses penjualan.

b) Menerapkan aplikasi *website* agar dapat digunakan sesuai kebutuhan pemakai sehingga bisa berjalan secara efisien.

c) Mempermudah pemilik toko melakukan proses penjualan.

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Pengertian Implementasi

Arti implementasi menurut KBBI (Kamus Besar Bahasa Indonesia) yaitu pelaksanaan/ penerapan. Sedangkan pengertian umum adalah suatu tindakan atau pelaksana rencana yang telah disusun secara cermat dan rinci (matang).

Kata iplementasi sendiri berasal dari bahasa Inggris "*to implement*" artinya mengimplementasikan. Tak hanya sekedar aktivitas, implementasi merupakan suatu kegiatan yang direncanakan serta dilaksanakan dengan serius juga mengacu pada norma-norma tertentu guna mencapai tujuan kegiatan.

Haryanto (2007) mengemukakan bahwa implementasi mencakup melakukan, membantu, atau mengarahkan kinerja aktivitas sehari-hari. Dengan kata lain implementasi adalah melakukan rencana tindakan yang telah ditentukan untuk mengatasi masalah klien.

Fidianti (2018) menjelaskan implementasi secara sederhana dapat diartikan sebagai pelaksanaan atau penerapan, sebagaimana yang ada di dalam kamus besar bahasa Indonesia, implementasi berarti penerapan..

2. Pengertian Framework Laravel

Menurut Yuniar dan Suleman (2019) menjelaskan bahwa “*Laravel* adalah sebuah *framework PHP* yang dirilis di bawah lisensi *MIT* dan dibangun dengan konsep *MVC (Model View Controller)*”.

Menurut Yuniar dan Suleman (2019) menjelaskan bahwa “*Laravel* merupakan pengembangan *website* berbasis *MVP* yang ditulis dalam *PHP* yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, serta untuk meningkatkan pengalaman bekerja dengan aplikasi yang menyediakan sintaks yang ekspresif, jelas dan menghemat waktu”.

Framework Laravel dibuat oleh Taylor Otwell dan proyek *Laravel* dimulai pada april 2011. Awal mula proyek ini dibuat karena Otwell sendiri tidak menemukan *framework* yang *up-to-date* dengan versi *PHP*. Mengembangkan *framework* yang sudah ada juga bukan merupakan ide bagus karena keterbatasan sumber daya. Dikarenakan beberapa keterbatasan tersebut, Otwell membuat sendiri *framework* dengan nama *Laravel*. Oleh karena itu *Laravel* mensyaratkan *PHP* versi 5.3 ke atas.

3. Pengertian Aplikasi

Aplikasi adalah perangkat lunak yang dikembangkan untuk menyelesaikan masalah-masalah tertentu yakni dapat mengerjakan tugas tertentu. (Yakub, 2012).

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa aplikasi merupakan program atau sekumpulan program yang dibuat dengan tujuan untuk menyelesaikan masalah tertentu berdasarkan perintah-perintah yang diberikan *user* melalui komputer.

4. Pengertian Web

Menurut Puspitosari (2010) menjelaskan bahwa “*Website* adalah halaman informasi yang disediakan melalui jalur *internet* sehingga bisa diakses diseluruh dunia, selama terkoneksi dengan jaringan *internet*”.

Secara garis besar menurut Puspitosari (2010) menggolongkan *website* menjadi beberapa jenis yaitu:

a) *Website Statis*

Website Statis adalah *web* yang mempunyai halaman tidak berubah. Artinya, untuk melakukan perubahan pada suatu halaman dilakukan secara manual dengan mengedit *code* yang menjadi struktur dari *website* tersebut.

b) *Website Dinamis*

Website Dinamis merupakan *website* yang secara terstruktur diperuntukan untuk *update* sesering mungkin, biasanya selain halaman utama bisa diakses oleh *user* pada umumnya, juga disediakan halaman *back end* untuk mengedit konten dari *website*.

c) *Website Interaktif*

Website Interaktif adalah *web* yang saat ini memang sedang ‘*booming*’. Salah satu contoh *website* interaktif adalah *blog* dan forum. Di *website* ini *user* bisa berinteraksi dan beradu argument mengenai apa yang menjadi pemikiran mereka. Biasanya *website* seperti memiliki moderator untuk mengatur supaya topik yang diperbincangkan tidak keluar jalur.

5. Pengertian HTML (*Hypertext Markup Language*)

Menurut Nugroho (2008) mengemukakan bahwa “*HTML* adalah kependekan dari (*Hyper Text Markup Language*), merupakan sebuah bahasa *Scripting* yang berguna untuk menuliskan halaman *Web*”.

Menurut Nugroho (2008) menerangkan bahwa “Semua bahasa *Scripting* yang berjalan di bawah *Web* dapat didukung oleh *HTML*, biasanya bahasa-bahasa tersebut melakukan *Embedded Script* pada *tag-tag HTML*.”

Sebagai sebuah elemen dasar di dalam pemrograman *Web*, *HTML* memiliki beberapa aturan-aturan dalam penulisannya yang kesemuanya itu

harus dipenuhi dalam penulisannya, sehingga dengan adanya aturan-aturan tersebut, sebagai seorang *programmer Web* harus konsisten dalam hal penulisan semua *tag-tag HTML*.

HTML memiliki beberapa *sintaks* dasar yang hampir mirip dengan semua pemrograman baik yang berbasis *Web* maupun *Visual*. Kemiripan itu adalah bahwa semua struktur pemrograman harus ada *sintaks* yang menyatakan program itu dimulai dan akhir *sintaks*. Sebagai contoh dalam pemrograman *Web PHP* mengenal tanda `<?php` sebagai tanda yang mengenalkan bahwa semua bentuk setelah tanda `<?php` merupakan halaman isi program *PHP* dan kemudian untuk menutupnya diakhiri dengan tanda `?>`.

6. Pengertian PHP

Menurut dokumen resmi *PHP* dalam Kadir (2018) mengemukakan bahwa: *PHP* merupakan singkatan dari *PHP Hypertext Preprocessor*. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses di *server*, hasilnya adalah yang dikirimkan ke klien, tempat pemakai menggunakan *browser*.

Pada saat ini *PHP* cukup populer sebagai piranti pemrograman *web*, terutama di lingkungan *Linux*. Walaupun demikian, *PHP* sebenarnya juga dapat berfungsi pada *server-server* yang berbasis *UNIX*, *Windows* dan *Macintosh*. Pada awalnya, *PHP* dirancang untuk diintegrasikan dengan

web server Apache, namun belakangan *PHP* juga dapat bekerja dengan *web server* seperti *PWS (Personal Web Server)*, *IIS (Internet Information Server)* dan *Xitami*.

7. Pengertian MySQL

Menurut Kadir (2018) menerangkan bahwa:

Mysql adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses databasenya, selain itu ia bersifat *open source* pada berbagai *platform* (kecuali untuk jenis *Enterprise*, yang bersifat komersial).

8. UML (Unified Modeling Language)

Menurut Rosa A. dan M. Shalahuddin (2013) menjelaskan bahwa: UML (*Unified Modeling Language*) adalah salah satu standard bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

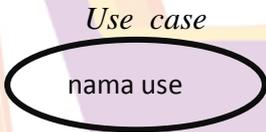
Diagram UML terdiri dari 13 macam diagram. Namun dalam penyusunan skripsi ini penulis hanya menggunakan 4 macam diagram diantaranya *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*. Berikut penjelasan dari keempat diagram tersebut menurut Rosa A. dan M. Shalahuddin (2013) :

a. *Use Case Diagram*

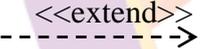
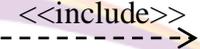
Diagram *Use Case* menyajikan interaksi antara *Use Case* dan *actor*. Dimana, *actor* dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use Case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai.

Berikut adalah symbol-simbol yang ada pada diagram *Use Case* :

Tabel 2.1 Simbol-simbol *Use Case Diagram*

Simbol	Deskripsi
	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i></p>

Tabel 2.1 Simbol-simbol *Use Case Diagram* (Lanjutan)

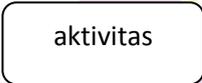
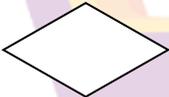
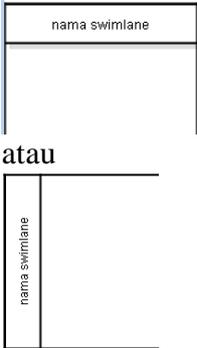
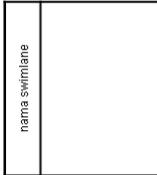
Simbol	Deskripsi
<p>Aktor / <i>actor</i></p>  <p>nama aktor</p>	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Asosiasi / <i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>
<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p>Menggunakan / <i>include</i> / <i>uses</i></p>  	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ul style="list-style-type: none"> - <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan. - <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan. <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

(Sumber : Rosa A.S – M.Shalahuddin, 2013)

b. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis, diagrama aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis (*business work flow*). Dapat juga digunakan untuk menggambarkan aliran kejadian (*flow of event*) dalam *Use Case*.

Tabel 2.2 Simbol – Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i>  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

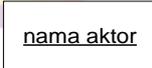
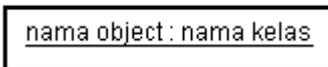
(Sumber : Rosa A.S – M.Shalahuddin, 2013)

c. *Sequence Diagram*

Sequence Diagram digunakan untuk menunjukkan aliran fungsionalitas dalam *Use Case*. Diagram sequensial adalah diagram yang disusun berdasarkan urutan waktu. Kita membaca diagram sekuensial dari atas ke bawah. Setiap diagram sekuensial mempresentasikan suatu aliran dari beberapa aliran didalam *Use Case*. Kita dapat membaca diagram ini dengan memperhatikan obyek-obyek dan pesan-pesan yang ada di diagram. Obyek yang terlibat dalam aliran ditunjukkan dengan bujur sangkar yang ada di atas diagram.

Berikut adalah simbol-simbol yang ada pada diagram sekuen :

Tabel 2.3 Simbol-simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p>  <p>Atau</p>  <p>tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, biasanya dinyatakan menggunakan kata benda di awal <i>frase</i> nama aktor.</p>
<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan.</p>

Tabel 2.3 Simbol-simbol *Sequence Diagram* (Lanjutan)

Simbol	Deskripsi
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
Pesan tipe <i>create</i> <code><<create>></code> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe <i>call</i> <code>1 : nama_metode()</code> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
Pesan tipe <i>send</i> <code>1 : masukan</code> 	Menyatakan bahwa suatu objek mengirim data/masukan/informasi ke objek lain, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe <i>return</i> <code>1 : keluaran</code> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

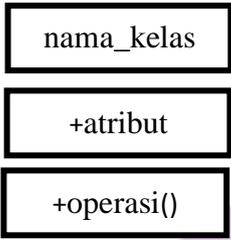
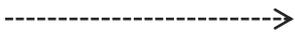
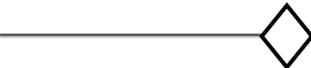
(Sumber : Rosa A.S – M.Shalahuddin, 2013)

d. *Class Diagram*

Class diagram yang digunakan untuk menampilkan kelas-kelas atau paket-paket didalam sistem dan relasi antar mereka. Ia memberikan gambaran sistem secara statis. Biasanya dibuat beberapa diagram kelas untuk satu sistem.

Berikut adalah simbol-simbol yang ada pada diagram kelas :

Tabel 2.4 Simbol-simbol *Class Diagram*

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem
<p>Antarmuka / <i>interface</i></p> <p>Nama_interface</p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
<p>Asosiasi / <i>association</i></p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Asosiasi berarah / <i>directed association</i></p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Generalisasi / <i>generalization</i></p> 	Relasi antarkelas dengan makna generalisasi - spesialisasi (umum khusus)
<p>Kebergantungan / <i>dependency</i></p> 	Relasi antarkelas dengan makna kebergantungan antarkelas
<p>Agregasi / <i>aggregation</i></p> 	Relasi antarkelas dengan makna semua – bagian (<i>whole – part</i>)

(Sumber : Rosa A.S – M.Shalahuddin, 2013)

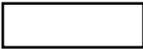
9. Entity Relationship Diagram (ERD)

a. Definisi Entity Relationship Diagram

ERD untuk mendokumentasikan data perusahaan dengan mengidentifikasi jenis entitas dan hubungannya. ERD merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak. ERD juga menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas yang lain dalam suatu sistem yang terintegrasi (Yakub, 2012).

ERD digunakan oleh perancang sistem untuk memodelkan data yang nantinya akan dikembangkan menjadi basis data. Model data ini juga akan membantu pada saat melakukan analisis dan perancangan basis data, karena model data ini menunjukkan bermacam-macam data yang dibutuhkan dan hubungan antar data. ERD ini juga merupakan model konseptual yang dapat mendeskripsikan hubungan antara file yang digunakan untuk memodelkan struktur data serta hubungan antar data. ERD terbagi atas tiga komponen, yaitu entitas (*entity*), atribut (*attribute*) dan relasi (*relation*). Berikut adalah symbol-simbol yang ada didalam Entity Relationship Diagram (ERD) (Yakub, 2012).

Tabel 2.5 adalah symbol-simbol yang ada didalam *Entity Relationship Diagram* (ERD).

Simbol	Keterangan
	Entitas yaitu kumpulan dari objek yang dapat diidentifikasi secara unik.
	Relasi yaitu hubungan yang terjadi antara satu atau lebih entitas, jenis hubungan antara lain satu ke satu, satu ke banyak dan banyak ke banyak.
	Atribut yaitu karakteristik dari <i>entity</i> atau relasi yang merupakan penjelasan detail tentang entitas.
	Hubungan antara <i>entity</i> dengan atributnya dan hubungan entitas dengan himpunan relasinya.
	Collaboration merupakan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur.
	Realisation merupakan operasi yang benar-benar dilakukan oleh suatu objek.
	Depedency merupakan hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.

(Sumber : Yakub, 2012)

b. Entitas

Yakub (2012) entitas (*entity*) menunjukkan objek-objek dasar yang terkait didalam sistem. Objek dasar dapat berupa orang, benda atau hal lain yang keterangannya perlu disimpan dalam basis data.

Untuk menggambarkan entitas dilakukan dengan mengikuti aturan-aturan sebagai berikut:

- 1) Entitas dinyatakan dengan symbol persegi panjang.
- 2) Nama entitas berupa kata benda tunggal.
- 3) Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami dan menyatakan maknanya dengan jelas.

c. Atribut

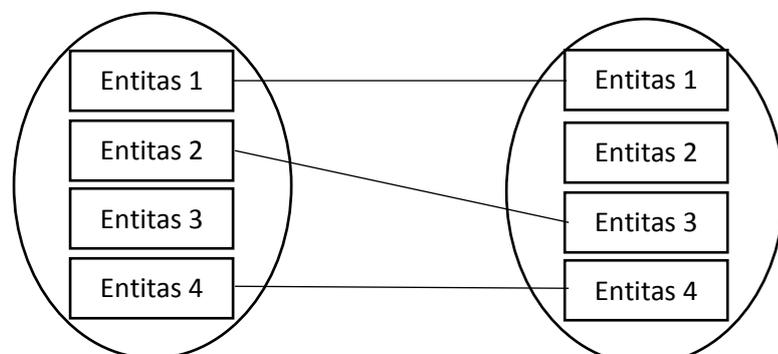
Yakub (2012) atribut (*attribute*) disebut juga sebagai *property*, merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan sebagai basis data. Atribut berfungsi sebagai penjelas sebuah entitas untuk menggambarkan atribut yang dilakukan dengan mengikuti aturan sebagai berikut:

- 1) Atribut dinyatakan dengan simbol *elipps*.
- 2) Nama atribut dituliskan dengan simbol *elipps*.
- 3) Nama atribut berupa kata benda tunggal.
- 4) Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.
- 5) Atribut dihubungkan dengan entitas yang bersesuaian dengan menggunakan garis.

d. Derajat Relasi

Yakub (2012) derajat relasi (kardinalitas) relasi menunjukkan maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas relasi yang terjadi antara lain:

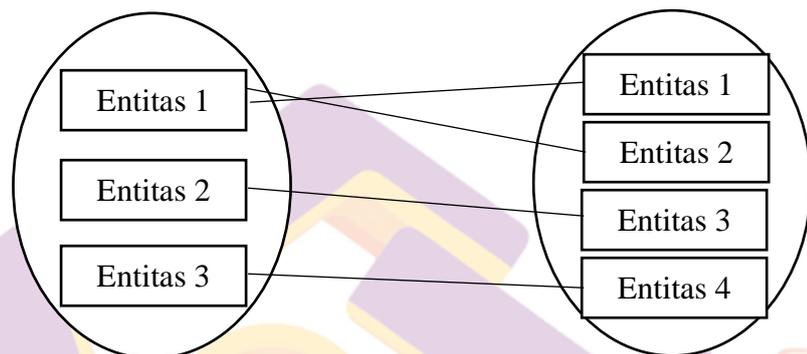
- 1) Satu ke satu (*one to one*)



Gambar 2.1 Kardinalitas Relasi *One To One*
(Yakub, 2012)

Kesimpulan gambar 2.1, kardinalitas relasi tersebut adalah setiap data pada entitas A berhubungan dengan maksimal satu data pada entitas B maupun sebaliknya.

2) Satu ke banyak (*one to many*)

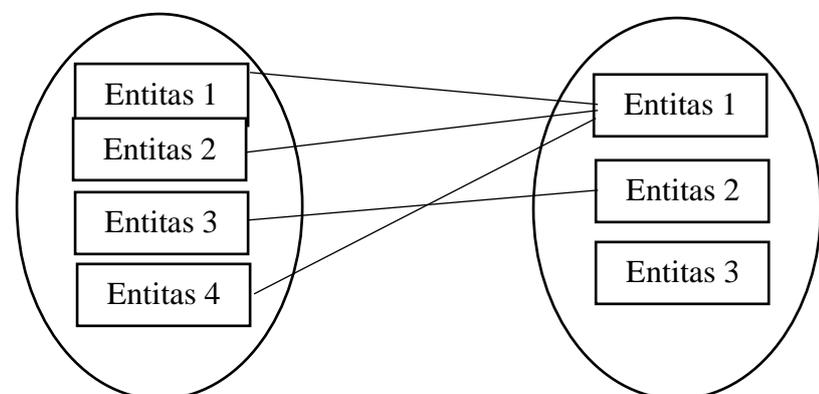


Gambar 2.2 Kardinalitas Relasi *One To Many*

(Yakub, 2012)

Kesimpulan gambar 2.2, kardinalitas tersebut adalah setiap data pada entitas A bisa berhubungan dengan banyak data pada entitas B, tetapi data pada entitas B berhubungan maksimal hanya dengan sebuah data di A.

3) Banyak ke satu (*many to one*)

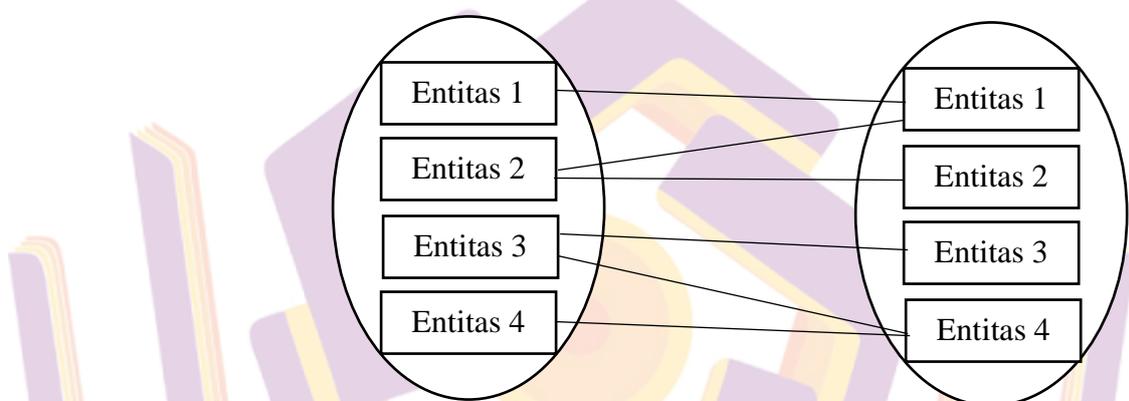


Gambar 2.3 Kardinalitas Relasi *Many To One*

(Yakub, 2012)

Kesimpulan gambar 2.3,kardinalitas tersebut adalah setiap data pada entitas B bisa berhubungan dengan banyak data pada entitas A, tetapi data pada entitas A berhubungan maksimal hanya dengan sebuah data di B.

4) Banyak ke banyak (*many to many*)



Gambar 2.4 Kardinalitas Relasi *Many to Many*

(Yakub, 2012)

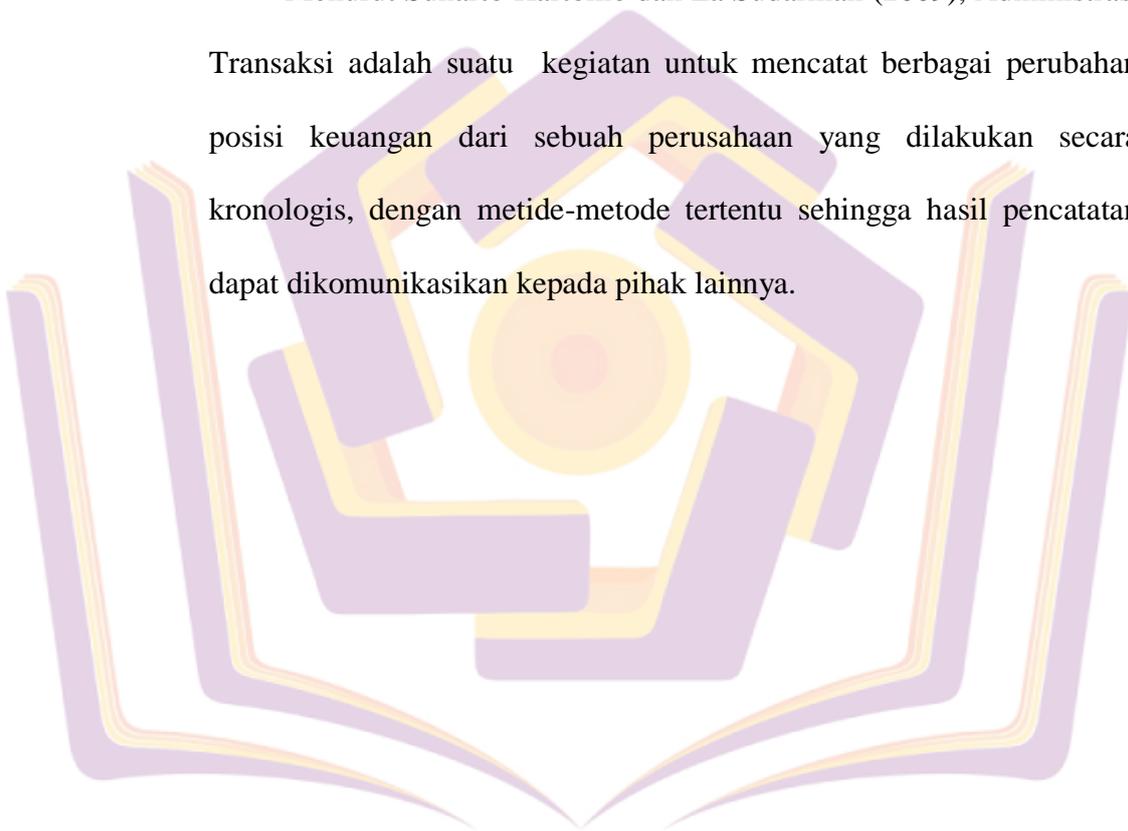
Yang entitas pada himpunan A dapat berhubungan dengan banyak entitas pada himpunan entitas B, demikian juga sebaliknya.

10. Pengertian Transaksi

Transaksi adalah suatu aktivitas yang dilakukan oleh individu atau organisasi dan dapat menimbulkan perubahan terhadap harta atau keuangan yang dimiliki, baik itu bertambah ataupun berkurang.

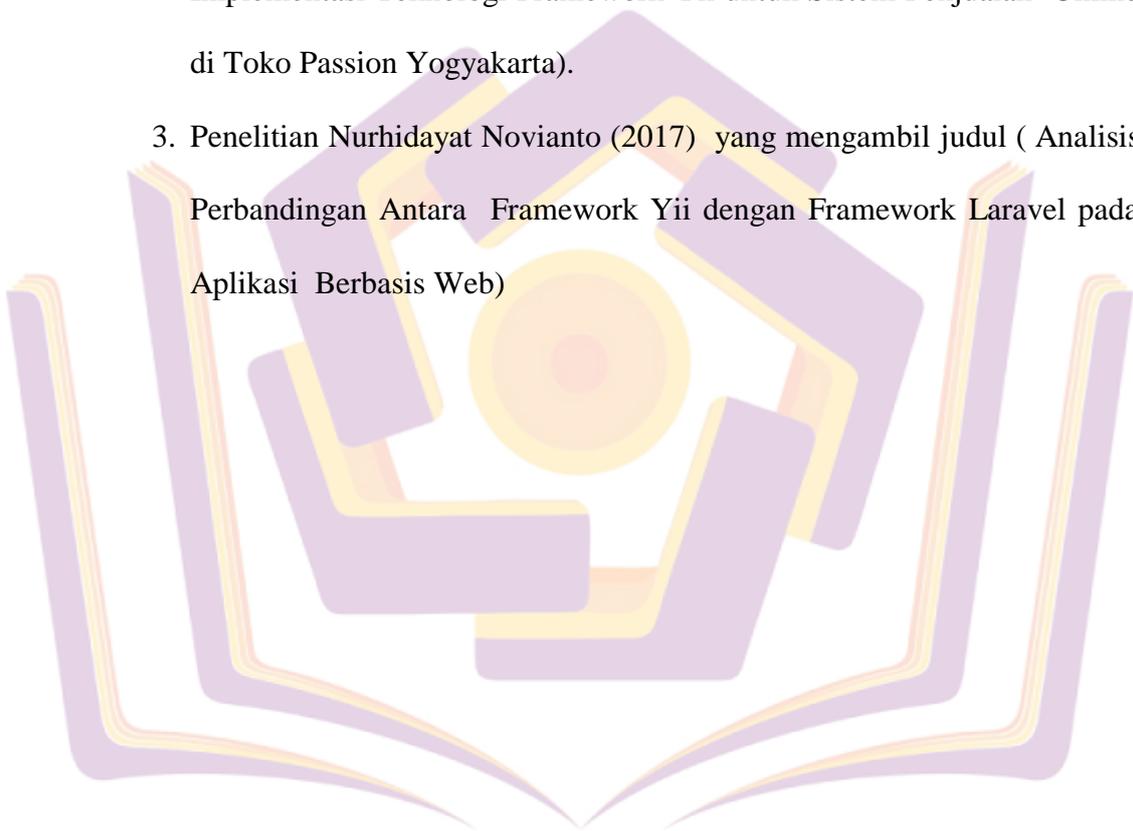
Menurut Kartomo dan La Sudarman (2019), pengertian transaksi adalah suatu aktivitas perusahaan yang menimbulkan perubahan terhadap posisi harta keuangan perusahaan, misalnya seperti menjual, membeli, membayar gaji, serta membayar berbagai macam biaya lainnya.

Menurut Sunarto Kartomo dan La Sudarman (2009), Administrasi Transaksi adalah suatu kegiatan untuk mencatat berbagai perubahan posisi keuangan dari sebuah perusahaan yang dilakukan secara kronologis, dengan metode-metode tertentu sehingga hasil pencatatan dapat dikomunikasikan kepada pihak lainnya.



B. Penelitian Sebelumnya

1. Penelitian oleh Mirwandi Ngidiho (2018) pada penelitiannya mengambil judul Sistem Informasi Penjualan Berbasis Web Menggunakan Framework Yii (Studi Kasus Toko Beta Musik Di Kota Masohi-Maluku)
2. Penelitian oleh Gisma Gerry Kurniawan (2018) yang mengambil judul (Implementasi Teknologi Framework Yii untuk Sistem Penjualan Online di Toko Passion Yogyakarta).
3. Penelitian Nurhidayat Novianto (2017) yang mengambil judul (Analisis Perbandingan Antara Framework Yii dengan Framework Laravel pada Aplikasi Berbasis Web)



Tabel 2.6 Perbedaan dengan penelitian sebelumnya

No	Penelitian/Tahun	Judul Penelitian	Perbedaan	
			Penelitian Terdahulu	Penelitian Sekarang
1	Mirwandi Ngidiho (2018)	Sistem Informasi Penjualan Berbasis Web Menggunakan Framework Yii (Studi Kasus Toko Beta Musik Di Kota Masohi-Maluku)	Pada penelitian sebelumnya penelitian tersebut berbasis android, fitur pada aplikasi android terdapat kekurangan OS(Operasi Sistem) tidak semua android memiliki OS di atas versi aplikasi tersebut.	Penelitian sekarang berbasis <i>website</i> sehingga dapat di akses oleh computer dan <i>handphone</i> yang memiliki akses internet
2	Maslan, Setiono dan Alfazri(2016)	Pengembangan <i>Smart Application Translation</i> Aneka Bahasa Sulawesi Berbasis Android	Penelitian sebelumnya tidak ada fitur keterangan jika ada kata yang belum dimasukkan ke dalam <i>database</i> .	Penelitian sekarang terdapat fitur bila ada kata yang belum ada di <i>database</i> maka akan muncul keterangan bahwa kata tersebut belum ada dalam <i>database</i> .
3.	Bintari(2015)	Aplikasi Pembelajaran Bahasa Bali Alus dengan <i>Phonegap</i> (Melajah Basa Bali)	Penelitian sebelumnya masih <i>offline</i> sehingga bila ada kata yang baru di <i>database</i> maka harus mengupdate aplikasi terlebih dahulu.	Penelitian sekarang tidak perlu mengupdate aplikasi terlebih dahulu untuk mendapatkan kata yang baru dimasukkan di <i>database</i> .

BAB III

METODE PENELITIAN

A. Tempat dan Waktu Penelitian

Penelitian ini dilakukan Toko “Kuat” yang beralamat di Jl.Raya Pasar Tambak Karangpucung Rt.03 Rw.03 Kabupaten Banyumas Provinsi Jawa Tengah dengan lama waktu penelitian selama 6 bulan dimulai dari 1 Februari 2020 sampai dengan 20 Juli 2020.

B. Metode Pengumpulan Data

Dalam melaksanakan kegiatan penelitian, pengumpulan data dilakukan dengan menggunakan metode sebagai berikut:

1. Observasi (*observation research*)

Menurut Sutrisno Hadi (1986) dalam Sugiyono (2015) menjelaskan bahwa “Observasi merupakan suatu proses yang kompleks, suatu proses yang tersusun dari pelbagai proses biologis dan psikologis”.

Observasi dilakukan dengan pengamatan langsung terhadap proses transaksi penjualan di toko “Kuat”. Dalam kegiatan observasi Penulis mempelajari dan menganalisa sistem yang sedang berjalan dan mengidentifikasi semua permasalahan yang timbul pada proses tersebut.

2. Studi Kepustakaan

Studi kepustakaan dilakukan untuk mencari landasan teori dari berbagai literatur yang berkaitan dengan masalah penelitian. Studi kepustakaan ini dilakukan dengan membaca buku-buku perpustakaan, panduan, jurnal, serta literature lain yang berkaitan dengan penelitian (Yakub,2012).

Studi kepustakaan dilakukan dengan mengamati data-data menggunakan buku-buku, literatur, artikel-artikel, karya ilmiah dan sumber lain.

3. Wawancara (*interview*)

Wawancara digunakan sebagai teknik pengumpulan data apabila peneliti ingin melakukan studi pendahuluan untuk menemukan permasalahan yang harus diteliti, dan juga apabila peneliti ingin mengetahui hal-hal dari responden yang lebih mendalam dan jumlah respondennya sedikit atau kecil (Sugiyono, 2015).

Untuk mendapatkan keterangan tentang proses transaksi di took “Kuat”, Penulis melakukan kegiatan tanya jawab dengan narasumber yakni pemilik toko tersebut. Wawancara dilakukan dengan pemilik toko dan karyawan untuk mengetahui alur transaksi pembelian, penjualan dan pendistribusian ke pelanggan.

C. Alat dan Bahan Penelitian

Dalam proses pembuatan penelitian ini, penulis menggunakan alat dan bahan penelitian berupa sumber data, *hardware* dan *software*, yaitu :

1. Hardware yang digunakan

a. Laptop *ACER V3-371-57V3*

Spesifikasi laptop yang digunakan :

- 1) *Processor Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz*
- 2) *RAM 4,00 GB*
- 3) *Hardisk 500 GB*

b. Printer *EPSON L220*

2. Software yang digunakan

- a. Sistem Operasi *Windows 8.1 Pro 64-bit*
- b. Bahasa Pemrograman *PHP Version 5.4.45*
- c. *XAMPP Versi 5.5.28*

3. Bahan yang digunakan

Bahan yang digunakan untuk penelitian adalah data-data yang didapatkan dari hasil wawancara langsung dengan pemilik toko Kuat.

Dari wawancara tersebut dapat diketahui apa saja kebutuhan yang diperlukan untuk membangun aplikasi.

D. Konsep Penelitian

Metode pengembangan sistem yang digunakan untuk pembangunan aplikasi Implementasi Framework Laravel untuk membangun aplikasi

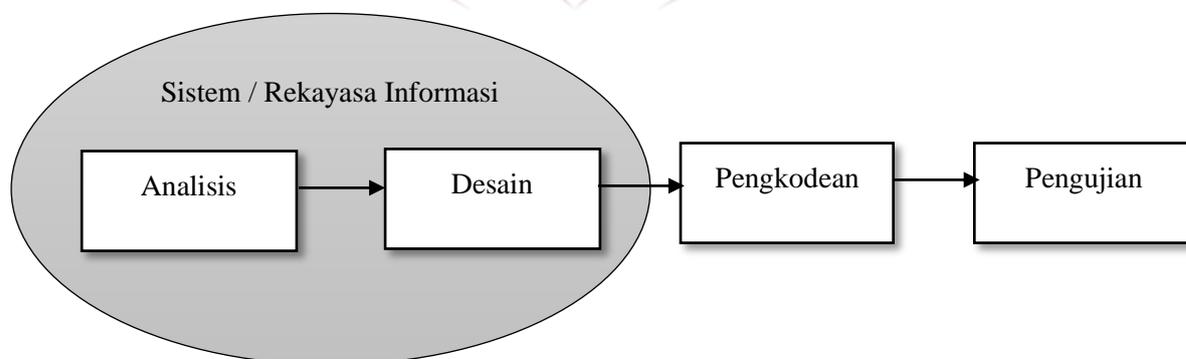
Manajemen Transaksi pada Toko Kuat Karangpucung Tambak Banyumas ini adalah SDLC. Siklus Hidup Pengembangan Sistem atau disebut *System Development Life Cycle* (SDLC) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model

dan metodologi yang digunakan orang untuk mengembangkan sistem – sistem perangkat lunak sebelumnya.

Siklus hidup pengembangan pengembangan sistem terdiri dari 5 model. Namun dalam penyusunan skripsi ini penulis hanya menggunakan 1 model yaitu model air terjun (*waterfall*).

Model air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, implementasi (pengodean), pengujian dan tahap pendukung (*maintenance/support*).

Berikut adalah gambar model air terjun :



Gambar 3.1 Diagram Tahapan dalam Metode Waterfall (Rosa A.S dan M. Shalahuddin, 2013)

1. *Analysis (Analisis Kebutuhan)*

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. *Design (Desain)*

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak, representasi antar muka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan juga perlu didokumentasikan.

Tahap perancangan pembuatan aplikasi sebagai berikut :

a. Perancangan UML

Tahapan perancangan UML dimulai dengan membuat *Use Case Diagram*, *Class Diagram*, *Activity Diagram* dan *Sequence Diagram* guna menspesifikasi, menggambarkan, membangun dan dokumentasi Aplikasi.

b. Perancangan Tampilan Antar Muka

Perancangan tampilan antar muka dilakukan dengan menggambar desain tampilan untuk setiap halaman aplikasi yang dikembangkan. Hasil rancangan tampilan antar muka akan diimplementasikan pada tahap pengkodean agar menjadi halaman *web* yang dapat diakses oleh pengguna.

3. **Coding (Pengodean)**

Tahap Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. **Testing (Pengujian)**

Pengujian fokus pada perangkat lunak secara dari segi *logic* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

Adapun metode yang digunakan dalam proses pengujian aplikasi ini yaitu dengan menggunakan metode pengujian *black box testing*.

a. Metode Pengujian *Black box testing*

Black box testing memungkinkan pengembangan *software* untuk membuat himpunan kondisi input yang meliputi seluruh syarat-syarat fungsional suatu program.

Uji coba *black box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

- 1) Fungsi-fungsi yang salah atau hilang
- 2) Kesalahan *interface*
- 3) Kesalahan dalam struktur data atau akses *database* eksternal
- 4) Kesalahan performa
- 5) Kesalahan inisialisasi dan terminasi

Pengujian dengan menggunakan metode *black box testing* dilakukan dengan cara memberi sejumlah masukan (*input*) pada program aplikasi yang kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah program aplikasi ini menghasilkan keluaran (*output*) yang diinginkan sesuai dengan fungsi dari program aplikasi tersebut.

BAB IV

HASIL DAN PEMBAHASAN

A. Analisis Hasil

1. Analisis Identifikasi Masalah

Proses penjualan dan pemesanan barang di toko “Kuat” saat ini masih bersifat konvensional, dimana proses penjualan dan pemesanan barang yang dilakukan masih dicatat di selembar kertas dan penjual masih harus menghitung total harga barang secara manual, sehingga proses tersebut memakan banyak waktu yang cukup lama, serta proses pencatatan pelaporan juga masih sangat sederhana dimana penjual mengumpulkan kertas pemesanan dan penjualan dalam laporan harian, bulanan dan tahunan di rekap satu per satu, tidak menutup kemungkinan presentase kesalahan saat perhitungan sangatlah besar karena proses pencatatan tersebut dilakukan secara manual di tulis di buku laporan dan perhitungannya menggunakan alat bantu kalkulator.

Untuk memecahkan masalah tersebut diharapkan dengan adanya aplikasi manajemen transaksi pada toko “Kuat” dapat memberikan sebuah solusi untuk dapat mengurangi kesalahan-kesalahan yang ada, juga untuk mencapai target pelayanan yang maksimal, sistem ini akan bekerja atau berfungsi , dimana admin membuat surat pesanan maupun penjualan di aplikasi dan setelah pemesanan maupun penjualan dilakukan admin dapat mencetak invoice dan laporan bulanan penjualan dan sebelum proses transaksi penjualan dilakukan administrator terlebih dahulu mengisi data

master barang, kategori barang dan nama toko member langganannya. Setelah data lengkap admin dapat melakukan proses transaksi dan mencetak invoice dan pelaporan.

2. Analisis Kebutuhan Sistem

a. Analisis Kebutuhan Fungsional

Pengguna dalam aplikasi manajemen transaksi pada toko adalah:

1) *Administrator*

Administrator merupakan sebuah halaman khusus untuk *admin* agar dapat mengelola semua data-data yang berkaitan dengan sistem yang ada di aplikasi manajemen toko “Kuat”.

Secara garis besar *Administrator* membutuhkan beberapa fitur yang dapat memberi kemudahan dalam mengelola semua data yang ada di aplikasi tersebut diantaranya sebagai berikut:

- a) *Administrator* dapat menambah, mengubah, menghapus atau mengelola data master seperti produk dan kategori produk.
- b) *Administrator* dapat mengelola data toko yang sudah menjadi langganan atau member toko “Kuat”
- c) *Administrator* dapat membuat pemesanan barang dan membuat invoice untuk penjualan barang.

- d) *Administrator* dapat melihat dan mencetak laporan stok produk, laporan rekap penjualan dalam bulan dan laporan pendistribusian produk.
- e) *Administrator* dapat mengelola data akun aplikasi dan mengganti password untuk masuk ke aplikasi.

b. Analisis Kebutuhan Non Fungsional

1) Kebutuhan Perangkat Lunak (*Software*)

Spesifikasi perangkat lunak yang digunakan adalah :

- a) Sistem Operasi *Windows 8.1 Pro 64-bit*
- b) Bahasa Pemrograman *PHP Version 7.3.14*
- c) *XAMPP* Versi 5.5.28
- d) Editor Notepad++ Versi 7.3.2
- e) *Editor Sublime Text*
- f) *Enterprise Architect Version 12.1*

2) Kebutuhan Perangkat Keras (*Hardware*)

Spesifikasi perangkat lunak yang digunakan adalah :

- 1) *Processor Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz*
- 2) *RAM 4,00 GB*
- 3) *Hardisk 500 GB*
- 4) *Printer EPSON L220*

B. Desain

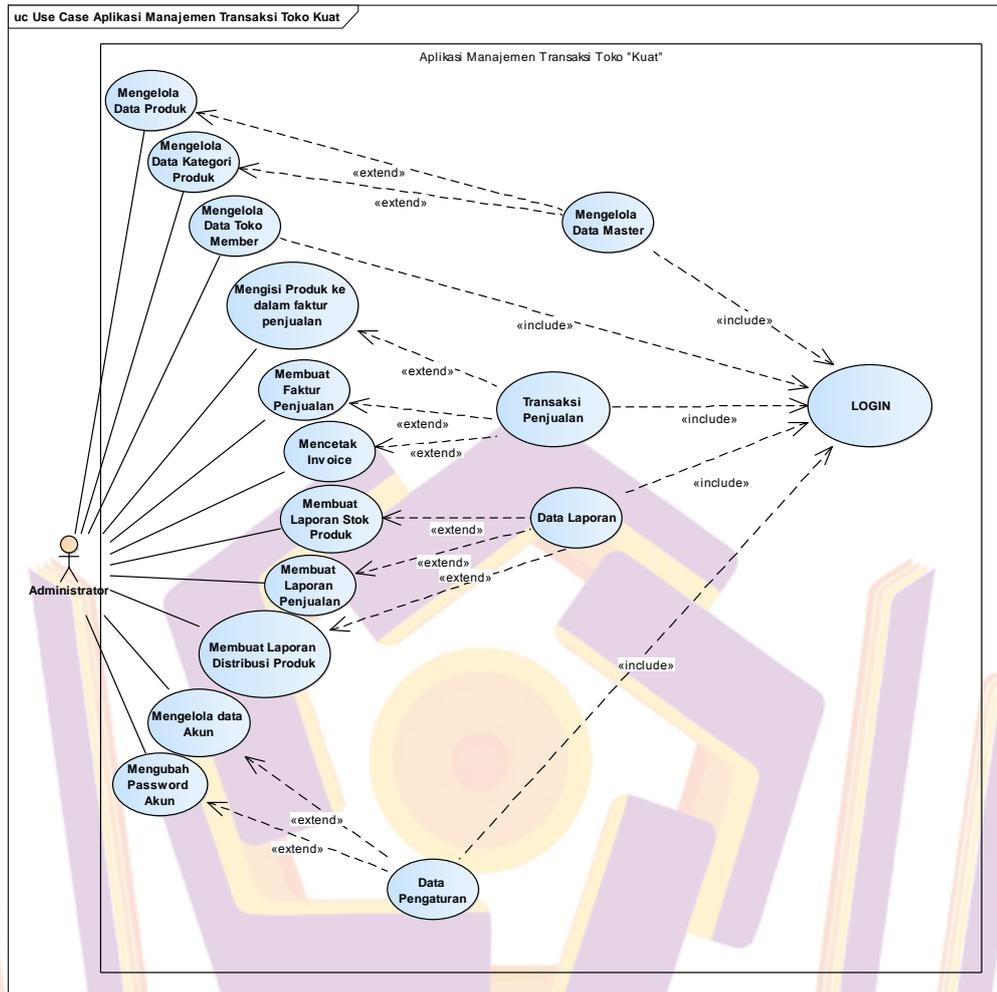
1. Perancangan *Unified Modelling Language (UML)*

a. *Use Case Diagram*

Use Case Diagram menggambarkan bagaimana aktor menggunakan sistem. Pengguna digambarkan sebagai aktor yang mengakses fasilitas yang disediakan aplikasi dalam bentuk *use case*. Pada aplikasi yang dikembangkan terdapat aktor *administrator*. *Use Case Diagram* untuk aplikasi manajemen transaksi pada toko “Kuat” adalah sebagai berikut :

Tabel 4.1 Tabel Definisi Aktor

No	Aktor	Deskripsi
1	<i>Administrator</i>	Orang yang memiliki hak akses penuh atas aplikasi, mengolah data master, data produk, data toko member, data transaksi order, pelaporan, pengaturan akun dan data password akun.



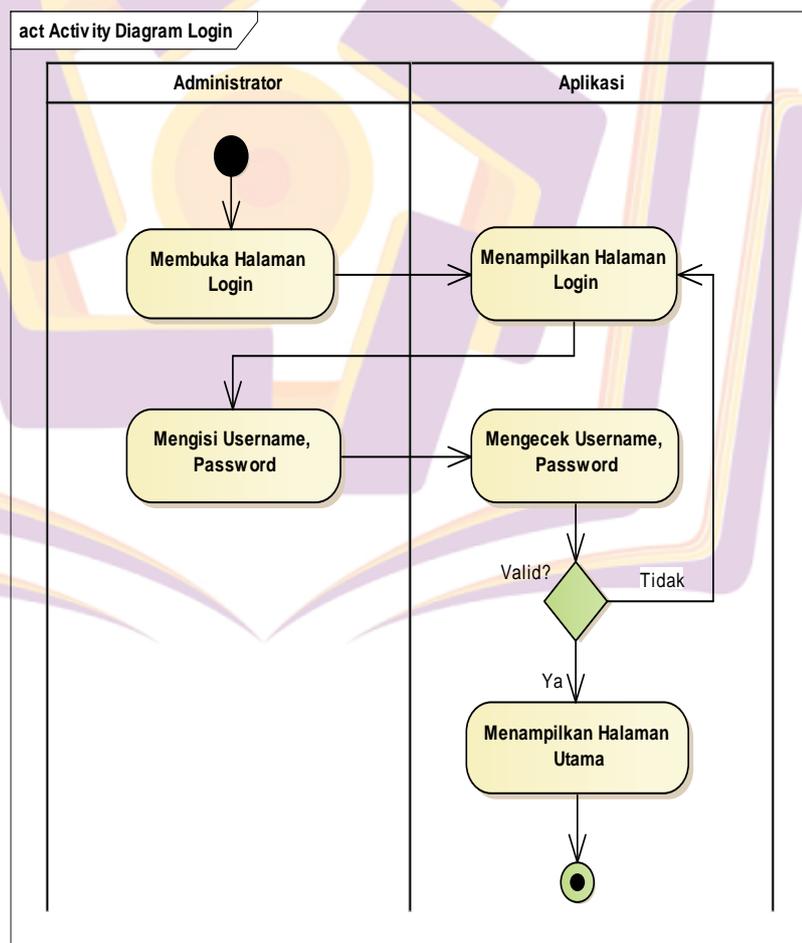
Gambar 4.1 Use Case Diagram Aplikasi

Pada gambar 4.1 di atas menjelaskan hubungan antara masing-masing pengguna dengan aplikasi. *Administrator* harus melakukan proses login terlebih dahulu agar dapat mengakses semua data yang berhubungan dengan proses transaksi di toko “Kuat”. *Administrator* memiliki hak akses untuk mengelola data barang, data kategori barang, data toko, data penjualan barang, data laporan, cetak invoice dan mengelola data akun dan mengganti password.

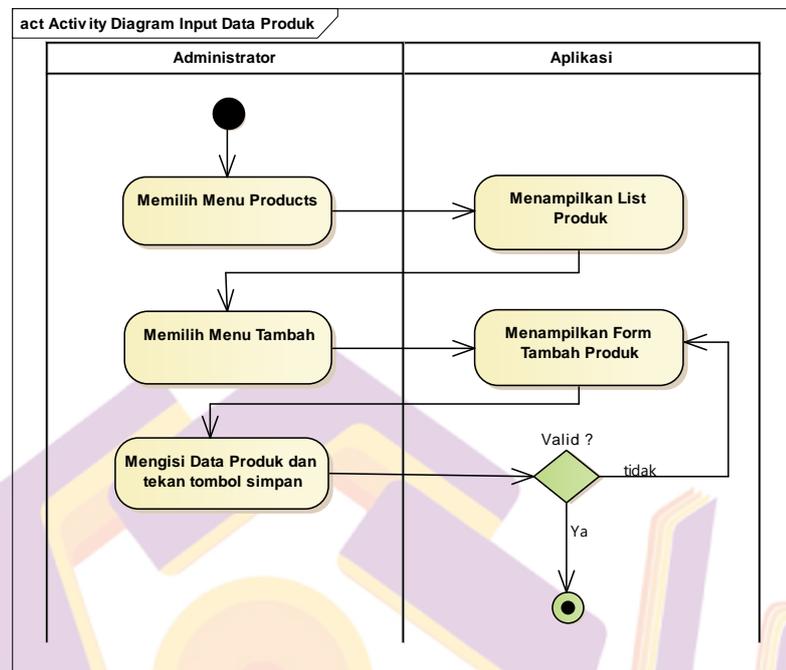
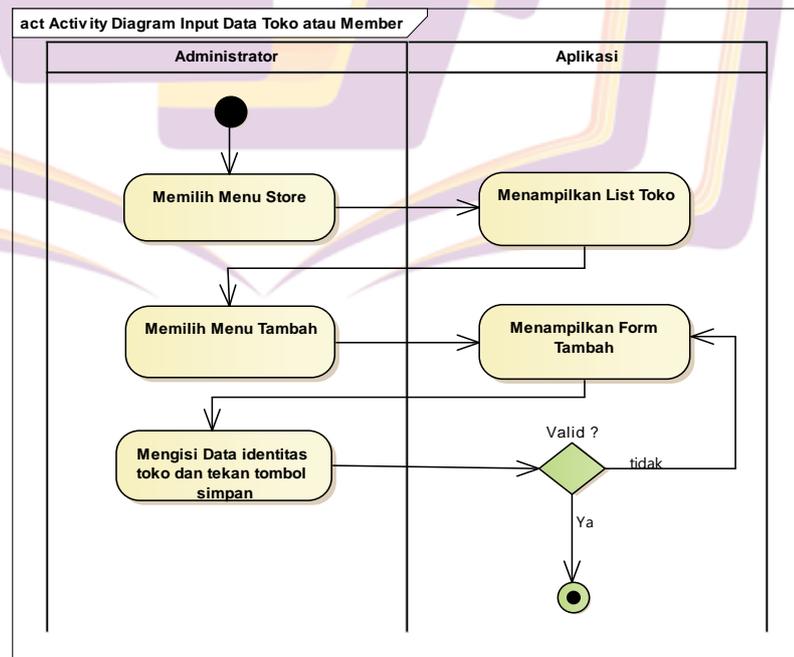
b. Activity Diagram

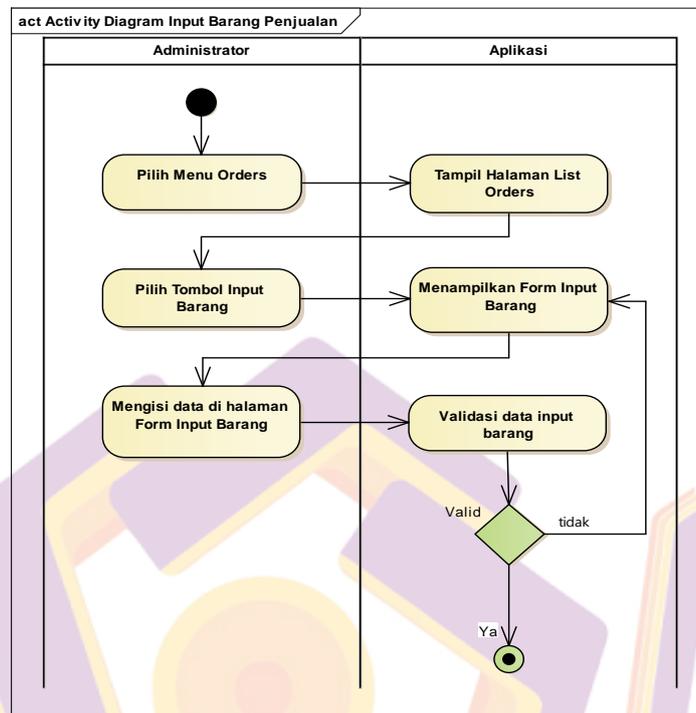
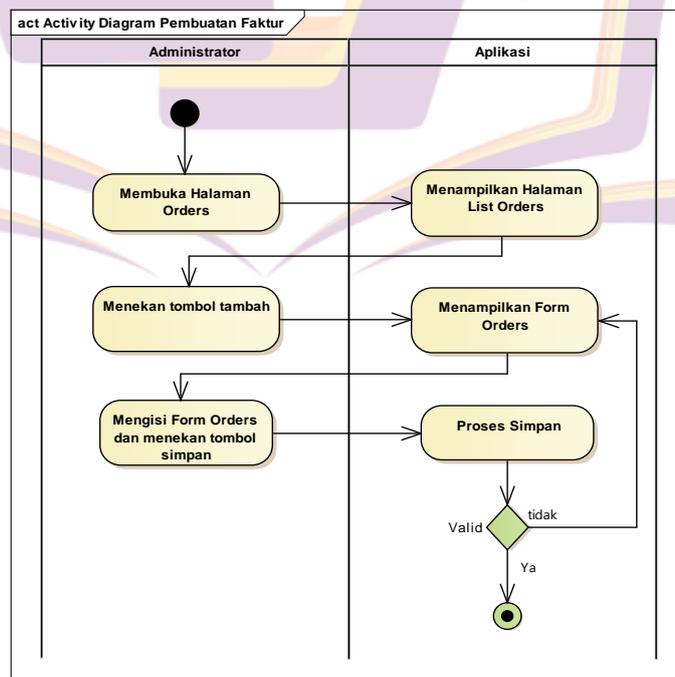
Activity diagram menggambarkan berbagai aliran aktivitas dalam sebuah sistem yang sedang dirancang dan bagaimana masing-masing aliran berawal, keputusan yang mungkin terjadi, dan bagaimana aktivitas tersebut berakhir. *Activity diagram* yang dirancang pada aplikasi manajemen toko “Kuat” adalah sebagai berikut :

1) Activity Diagram Login



Gambar 4.2 Activity Diagram Login

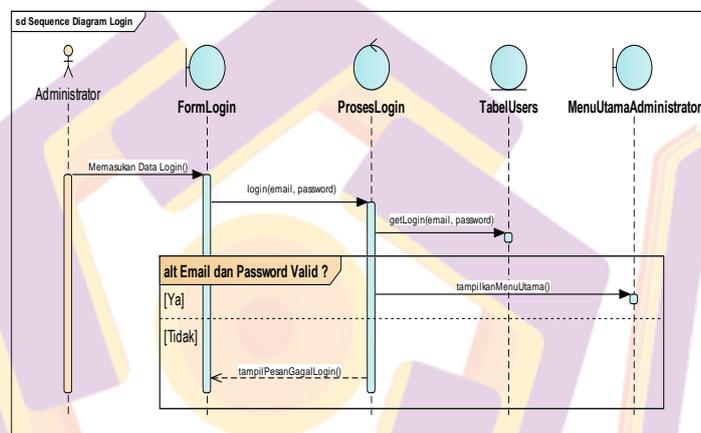
2) *Activity Diagram Input Data Produk*Gambar 4.3 *Activity Diagram Input Data Produk*5) *Activity Diagram Input Data Toko*Gambar 4.4 *Activity Diagram Input Data Toko*

6) *Activity Diagram Input Barang Penjualan*Gambar 4.5 *Activity Diagram Input Barang Penjualan*7) *Activity Diagram Pembuatan Faktur*Gambar 4.6 *Activity Diagram Pembuatan Faktur*

c. Sequence Diagram

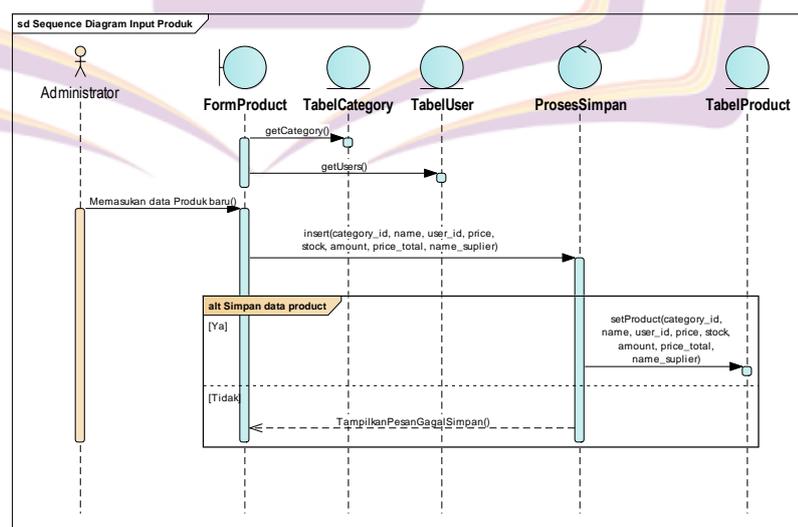
Sequence diagram menggambarkan pola hubungan antara sekumpulan objek yang saling mempengaruhi menurut urutan waktu. Berikut adalah *Sequence diagram* pada aplikasi manajemen transaksi pada toko “Kuat”:

1) Sequence Diagram Login



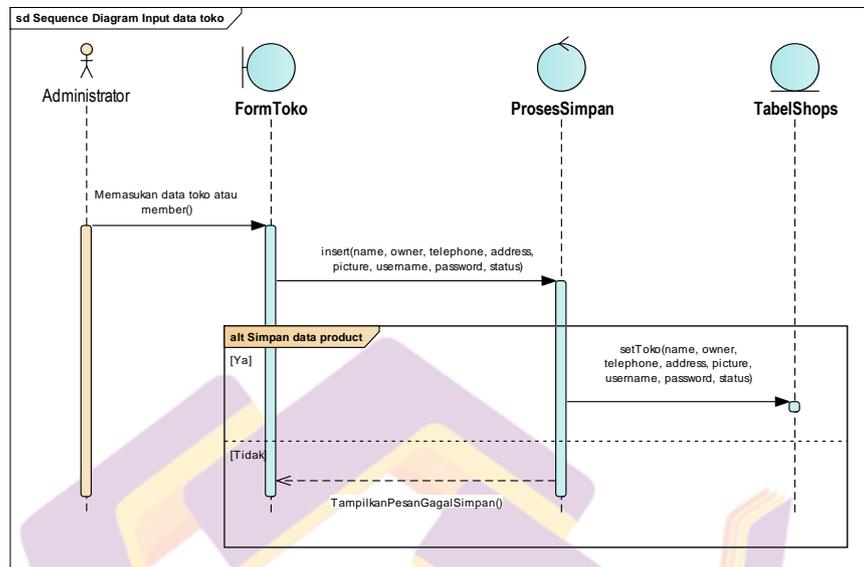
Gambar 4.7 Sequence Diagram Login

2) Sequence Diagram input data produk



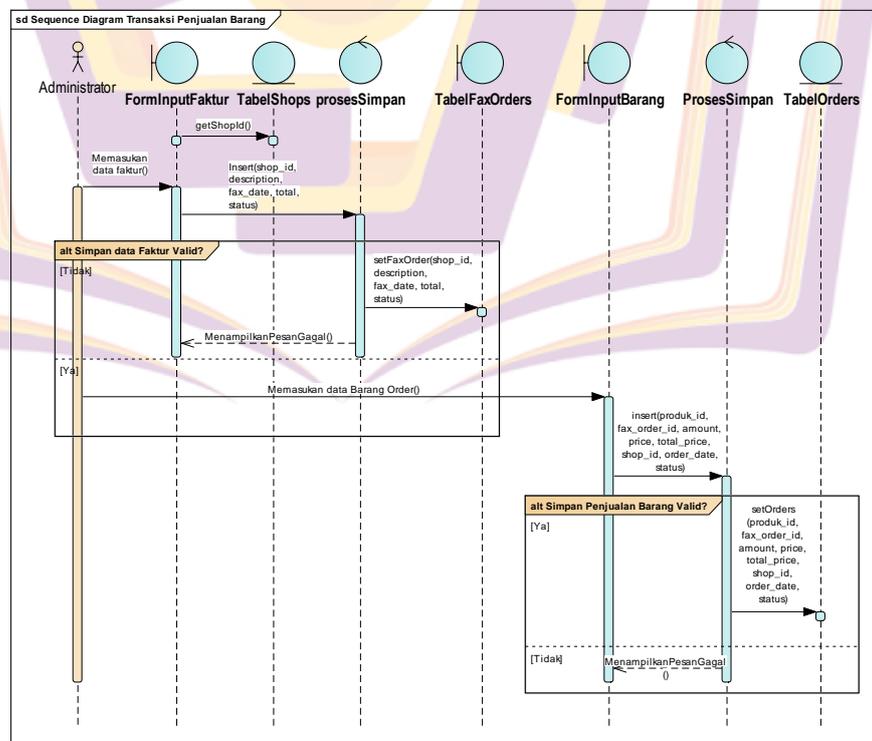
Gambar 4.8 Sequence Diagram input data produk

3) Sequence Diagram Input data toko



Gambar 4.9 Sequence Diagram Input data toko

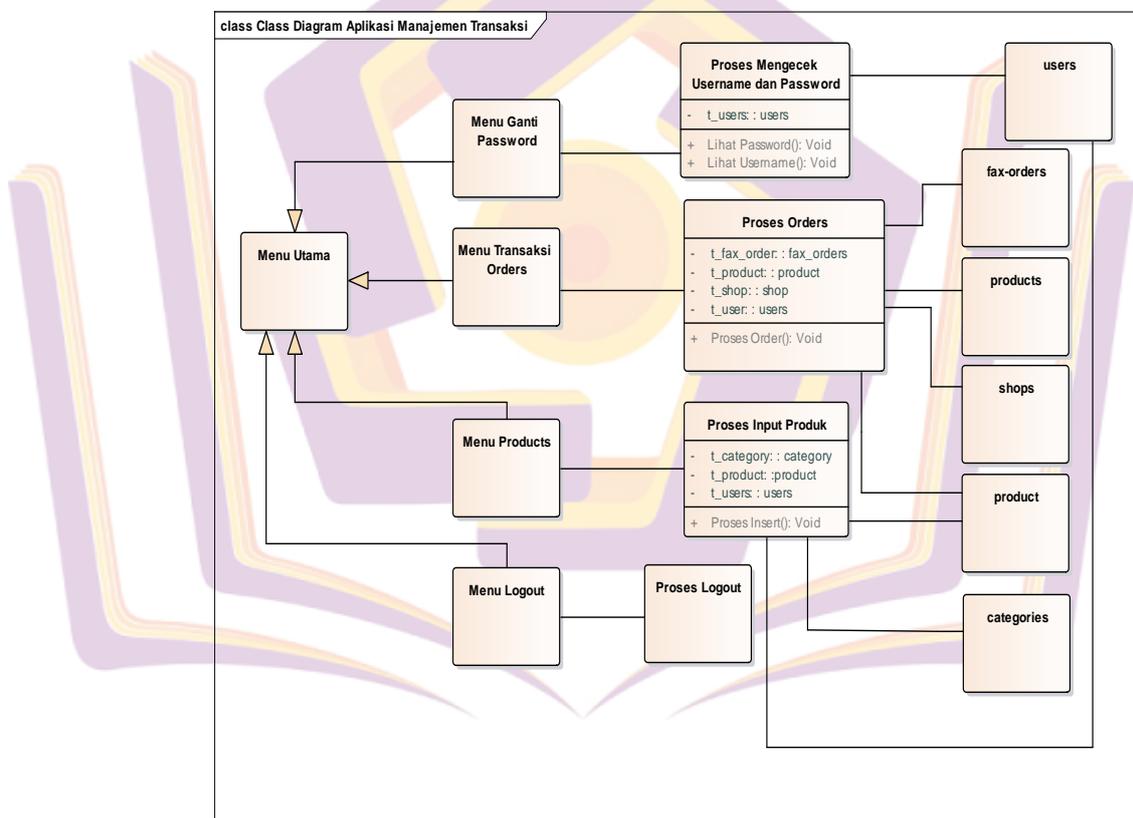
4) Sequence Diagram Transaksi Penjualan Barang



Gambar 4.10 Sequence Diagram Transaksi Penjualan Barang

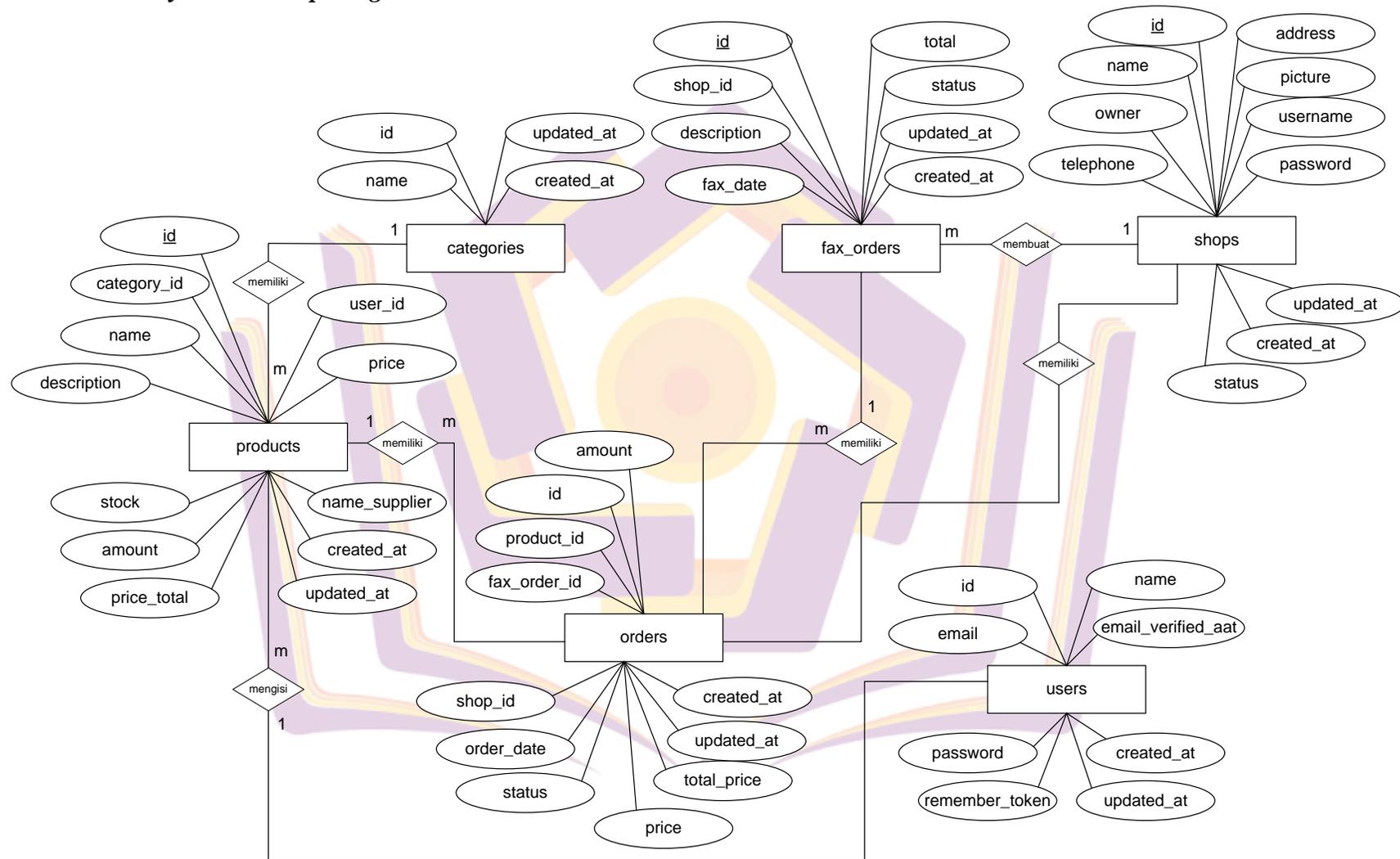
d. *Class Diagram*

Pembuatan *class diagram* merupakan langkah awal untuk merancang objek-objek dan kelas-kelas yang digunakan dalam aplikasi. *class diagram* menunjukkan kelas-kelas pada aplikasi yang dikembangkan, dimana kelas-kelas tersebut mengandung atribut dan operasi yang dibutuhkan. *class diagram* aplikasi manajemen transaksi pada toko “Kuat” sebagai berikut:



Gambar 4.11 *Class Diagram* Aplikasi

2. Entity Relationship Diagram



Gambar 4.12 Entity Relationship Diagram

Gambar 4.12 menunjukkan Entity Relationship Diagram dari aplikasi manajemen transaksi toko “Kuat”.

3. Perancangan Database

a. Struktur Tabel

1) Tabel *Users*

Tabel 4.2 Struktur Tabel *Users*

No	Nama	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>bigint</i>	20	<i>primary key</i>
2	<i>name</i>	<i>varchar</i>	255	
3	<i>email</i>	<i>varchar</i>	255	
4	<i>email_verified_at</i>	<i>timestamp</i>		
5	<i>password</i>	<i>varchar</i>	255	
6	<i>remember_token</i>	<i>varchar</i>	100	
7	<i>created_at</i>	<i>timestamp</i>		
8	<i>updated_at</i>	<i>timestamp</i>		

Tabel 4.2 berfungsi untuk menyimpan data administrator yang berfungsi untuk mengatur atau mengelola semua data data yang ada di dalam aplikasi yang terdiri dari *field id, name, email, email_verified_at, password, remember_token, created_at dan updated_at.*

2) Tabel *shops*

Tabel 4.3 Struktur Tabel *Shops*

No	Nama	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>biginteger</i>	20	<i>primary key</i>
2	<i>name</i>	<i>varchar</i>	255	
3	<i>owner</i>	<i>varchar</i>	255	
4	<i>telephone</i>	<i>varchar</i>	255	
5	<i>address</i>	<i>text</i>		
6	<i>picture</i>	<i>varchar</i>	255	
7	<i>username</i>	<i>varchar</i>	100	
8	<i>password</i>	<i>varchar</i>	255	
9	<i>status</i>	<i>enum</i>		
10	<i>created_at</i>	<i>timestamp</i>		
11	<i>updated_at</i>	<i>timestamp</i>		

Tabel 4.3 berfungsi untuk menyimpan data toko sebagai member dari toko “Kuat” yang terdiri dari *field id, name, owner, telephone, address, picture, username, password, status, created_at, updated_at*.

3) Tabel *Products*

Tabel 4.4 Struktur Tabel *Products*

No	Nama	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>biginteger</i>	20	<i>primary key</i>
2	<i>category_id</i>	<i>biginteger</i>	20	<i>foreign key</i>
3	<i>name</i>	<i>varchar</i>	255	
4	<i>description</i>	<i>text</i>		
5	<i>user_id</i>	<i>biginteger</i>	20	<i>foreign key</i>
6	<i>price</i>	<i>double</i>		
7	<i>stock</i>	<i>integer</i>	11	
8	<i>amount</i>	<i>integer</i>	11	
9	<i>price_total</i>	<i>double</i>		
10	<i>name_supplier</i>	<i>varchar</i>	255	
11	<i>created_at</i>	<i>timestamp</i>		
12	<i>updated_at</i>	<i>timestamp</i>		

Tabel 4.4 berfungsi untuk menyimpan data product yang berrelasi dengan tabel *category* dan tabel *users* yang terdiri dari *field id, category_id, name, description, user_id, price, stock, amount, price_total, name_supplier, created_at, updated_at*.

4) Tabel *Orders*

Tabel 4.5 Struktur *Orders*

No	Nama	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>biginteger</i>	20	<i>primary key</i>
2	<i>product_id</i>	<i>biginteger</i>	20	<i>foreign key</i>
3	<i>fax_order_id</i>	<i>biginteger</i>		<i>foreign key</i>
4	<i>amount</i>	<i>integer</i>	11	
5	<i>price</i>	<i>double</i>		

6	<i>total_price</i>	<i>double</i>		
7	<i>shop_id</i>	<i>biginteger</i>	20	<i>foreign key</i>
8	<i>order_date</i>	<i>date</i>		
9	<i>status</i>	<i>enum</i>		
10	<i>created_at</i>	<i>timestamp</i>		
11	<i>updated_at</i>	<i>timestamp</i>		

Tabel 4.5 berfungsi untuk menyimpan data *order* barang yang berelasi dengan tabel *products*, tabel *fax orders*, dan tabel *shops* yang terdiri dari *field id, product_id, fax_order_id, amount, price, total_price, shop_id, order_date, status, created_at, updated_at*.

5) Tabel *Migrations*

Tabel 4.6 Struktur Tabel *Migrations*

No	Nama	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>integer</i>	10	<i>primary key</i>
2	<i>migration</i>	<i>varchar</i>	255	
3	<i>batch</i>	<i>integer</i>	11	

Tabel 4.6 berfungsi untuk menyimpan data tabel *migration framework laravel* yang terdiri dari *field id, migration, batch*.

6) Tabel *Fax Orders*

Tabel 4.7 Struktur Tabel *Fax Orders*

No	Nama	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>biginteger</i>	20	<i>primary key</i>
2	<i>shop_id</i>	<i>biginteger</i>	20	<i>foreign key</i>
3	<i>description</i>	<i>varchar</i>	255	
4	<i>fax_date</i>	<i>date</i>		
5	<i>total</i>	<i>double</i>		
6	<i>status</i>	<i>enum</i>		
7	<i>created_at</i>	<i>timestamp</i>		
8	<i>updated_at</i>	<i>timestamp</i>		

Tabel 4.7 berfungsi untuk menyimpan data faktur pemesanan atau *order* yang berrelasi dengan table *shops* yang terdiri dari *field id, shop_id, description, fax_date, total, status, created_at, updated_at*.

7) Tabel *Categories*

Tabel 4.8 Struktur Tabel *Categories*

No	Nama	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>biginteger</i>	20	<i>primary key</i>
2	<i>name</i>	<i>varchar</i>	255	
3	<i>created_at</i>	<i>timestamp</i>		
4	<i>updated_at</i>	<i>timestamp</i>		

Tabel 4.8 berfungsi untuk menyimpan data kategori produk yang terdiri dari *field id, name, created_at, updated_at*.

4. Perancangan Tampilan Antarmuka

Perancangan tampilan antarmuka aplikasi manajemen transaksi pada toko “Kuat” adalah sebagai berikut :

a. Halaman *Login*

Gambar 4.13 Desain Halaman *Login*

Gambar 4.21 menggambarkan desain dari halaman *login* aplikasi yang digunakan oleh administrator ketika akan melakukan *login*.

b. Halaman Utama Aplikasi

INVENTORY	Nama_Admin
MENU	Halaman Beranda
=> Dashboard	
=> Product	
=> Store	
=> Order	
=> Report	
=> Settings	

Gambar 4.14 Desain Halaman Utama Aplikasi

Gambar 4.14 menggambarkan desain halaman dari halaman utama aplikasi dimana di dalam halaman tersebut terdapat menu-menu yang digunakan oleh administrator dalam melakukan transaksi.

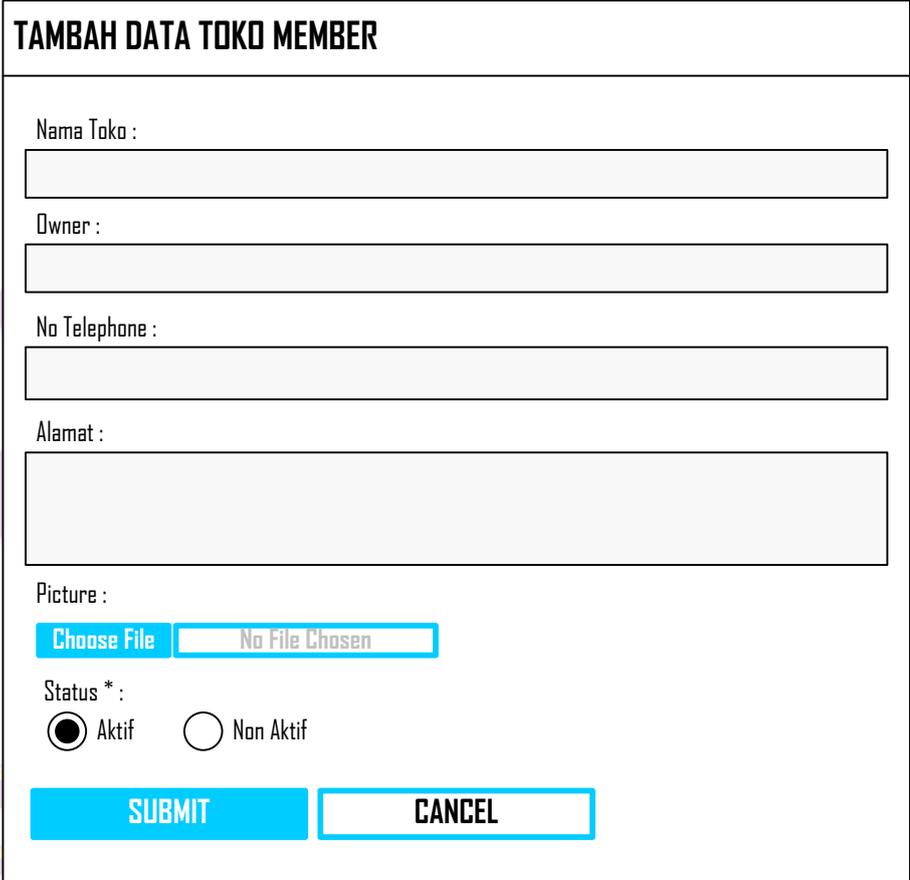
c. Halaman Data Tambah Produk

TAMBAH PRODUK	
Nama Produk :	<input type="text"/>
Kategori Produk :	<input type="text" value="▼"/>
Deskripsi Produk :	<input type="text"/>
Harga :	<input type="text"/>
Stok :	<input type="text"/>
Nama Supplier :	<input type="text"/>
<input type="button" value="SUBMIT"/> <input type="button" value="CANCEL"/>	

Gambar 4.15 Desain Halaman Tambah Produk

Gambar 4.15 menggambarkan desain dari halaman tambah produk yang digunakan oleh administrator ketika akan menambahkan data produk .

d. Halaman Tambah Data Toko Member



TAMBAH DATA TOKO MEMBER

Nama Toko :

Owner :

No Telephone :

Alamat :

Picture :

Status * :
 Aktif Non Aktif

Gambar 4.16 Desain Halaman Tambah Data Toko Member

Gambar 4.16 menggambarkan desain dari halaman tambah data toko member.

e. Halaman Ganti Password

GANTI PASSWORD

Masukan Password Lama

Masukan Password Baru

Konfirmasi Password Baru

GANTI

Gambar 4.17 Desain Halaman Ganti Password

Gambar 4.17 menggambarkan desain dari halaman ganti password.

f. Halaman Transaksi *Order*

Input Data Barang

NAMA TOKO :xxxxxxxxxxxxx
TANGGAL :xxxxxxxxxxxxx
DESKRIPSI :xxxxxxxxxxxxx
STATUS :xxxxxxxxxxxxx
TOTAL PEMBAYARAN :xxxxxxxxxxxxx

Produk :

Jumlah :

Harga :

ADD PRODUCT

DAFTAR BARANG

CETAK INVOICE

NO	NAMA PRODUK	JUMLAH	HARBA	TOTAL HARBA	AKSI
xx	xxxxxx	xxxxxx	xxxxxx	xxxxxx	Hapus

Gambar 4.18 Desain Halaman Transaksi *Order*

Gambar 4.18 menggambarkan desain dari halaman transaksi *order* yang digunakan oleh administrator untuk menyimpan data pesanan dari toko member.

g. Halaman Laporan Data Produk

LAPORAN DATA PRODUK					
"TOKO KUAT"					
Jl Raya Pasar Tambak Karangpucung Rt.03 Rw.03, Banyumas, Jawa Tengah					
NO	NAMA PRODUK	KATEGORI	STOK	HARGA	SUPPLIER
XX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX

Gambar 4.19 Desain Halaman Laporan Data Produk

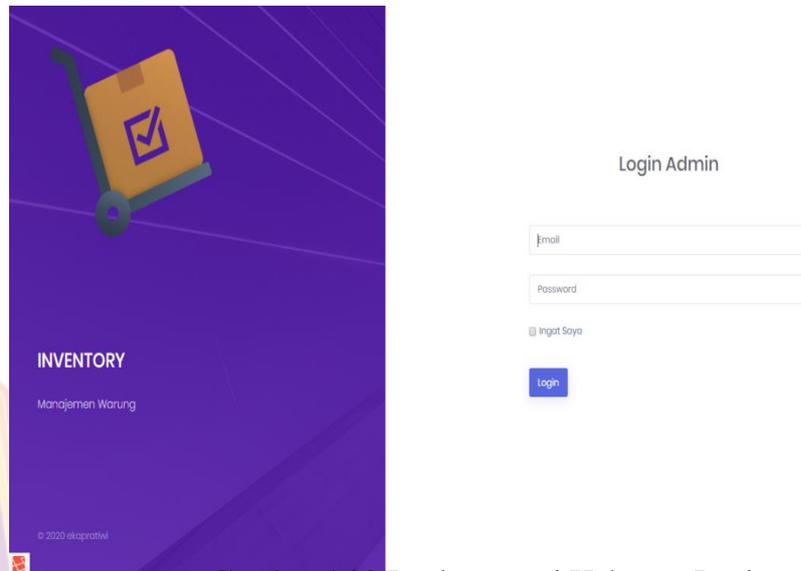
Gambar 4.19 menggambarkan desain dari halaman Laporan Data Produk digunakan oleh administrator untuk mengetahui jumlah stok produk yang masih tersedia.

D. Pengkodean

Implementasi pengkodean merupakan tahap dimana hasil rancangan diwujudkan dalam bentuk aplikasi. Rancangan diimplementasikan dengan bahasa pemrograman sehingga fungsi dapat dijalankan dengan baik.

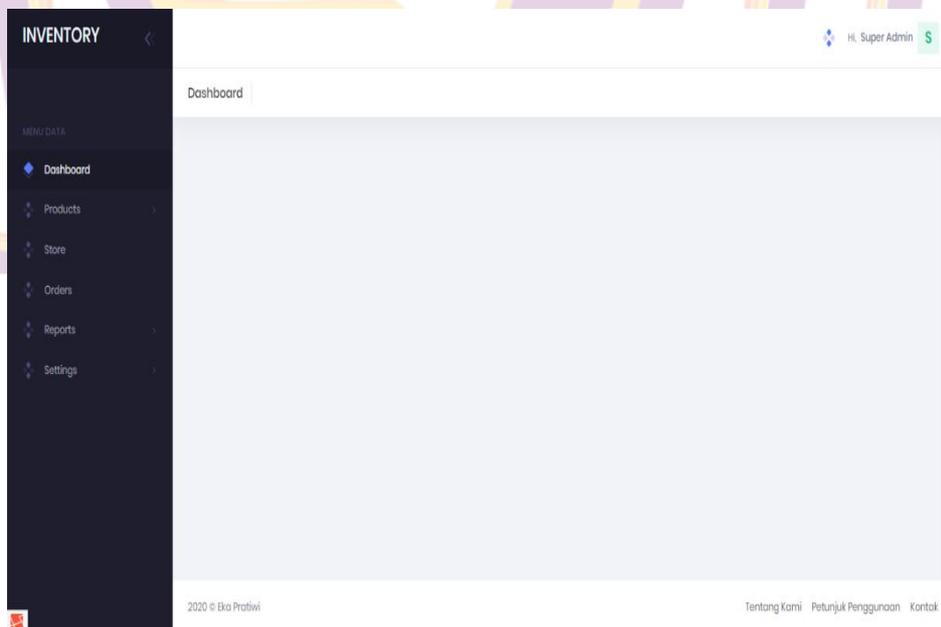
1. Implementasi Halaman Antarmuka

a. Halaman *Login*



Gambar 4.20 Implementasi Halaman Login

b. Halaman Utama Aplikasi



Gambar 4.21 Implementasi Halaman Utama Aplikasi

c. Halaman Data Tambah Produk

The screenshot shows the 'Tambah Produk' form within an 'INVENTORY' application. The form includes the following fields and elements:

- Header:** 'INVENTORY' logo, user profile 'Hi, Super Admin', and a 'Kembali' (Back) button.
- Title:** 'Tambah Produk' with a 'Kembali' button.
- Form Fields:**
 - Nama Produk:** Text input field with placeholder 'Enter nama produk'.
 - Kategori Produk:** Dropdown menu with 'Lampu LED' selected.
 - Deskripsi Produk:** Rich text editor with a toolbar containing icons for bold, italic, link, unlink, list, and other text formatting options.
 - Harga:** Text input field with placeholder 'Enter harga produk'.
 - Stock:** Text input field with placeholder 'Enter stok produk'.
 - Nama Supplier:** Text input field with placeholder 'Enter stok produk'.
- Buttons:** 'Submit' and 'Cancel' buttons at the bottom left.
- Footer:** Copyright notice '2020 © Eko Pratiwi', and links for 'Tentang Kami', 'Petunjuk Penggunaan', and 'Kontak'.

Gambar 4.22 Implementasi Halaman Tambah Produk

d. Halaman Tambah Toko Member

The screenshot shows the 'Tambah Data Toko Member' form within an 'INVENTORY' application. The form includes the following fields and elements:

- Header:** 'INVENTORY' logo, user profile 'Hi, Super Admin', and a 'Kembali' (Back) button.
- Title:** 'Tambah Data Toko Member' with a 'Kembali' button.
- Form Fields:**
 - Nama Toko:** Text input field with placeholder 'Enter nama toko'.
 - Owner:** Text input field with placeholder 'Enter nama pemilik'.
 - No Telepon / HP:** Text input field with placeholder 'Enter nomor telepon'.
 - Alamat:** Text input field with placeholder 'Enter alamat'.
 - Username:** Text input field with placeholder 'Enter username'.
 - Password:** Text input field with placeholder 'Enter password'.
 - Picture:** File upload field with a 'Choose File' button and 'No file chosen' text.
 - Status:** Radio button group with 'AKTIF' (selected) and 'Non AKTIF' options.
- Buttons:** 'Submit' and 'Cancel' buttons at the bottom left.
- Footer:** Copyright notice '2020 © Eko Pratiwi', and links for 'Tentang Kami', 'Petunjuk Penggunaan', and 'Kontak'.

Gambar 4.23 Implementasi Halaman Toko Member

e. Halaman Ganti Password

Gambar 4.24 Implementasi Halaman Ganti Password

f. Halaman Transaksi *Order*

Nama Toko	:	TOKO TOKIK'S
Tanggal	:	2020-07-07
Deskripsi	:	Faktur Pemesanan Tokik's
Status	:	Belum Di Kirim
Total Pembayaran	:	850000

#	Nama Produk	Jumlah	Harga	Total Harga	Aksi
1	Lampu LED	10	85000	850000	

Gambar 4.25 Implementasi Halaman Transaksi *Order*

g. Halaman Laporan Data Produk

#	Nama Produk	Kategori	Stock	Harga	Supplier
1	Lampu LED	Lampu LED	100	85000	WAHYU SUSLO

Gambar 4.26 Implementasi Halaman Laporan Stok Produk

E. Pengujian

Pengujian terhadap program yang dibuat menggunakan *blackbox* testing yang fokus terhadap proses masukan dan keluaran program.

1. Rancangan Pengujian *Black-Box*

Berikut ini adalah tabel rancangan rencana pengujian *black box* yang akan dilaksanakan pada aplikasi.

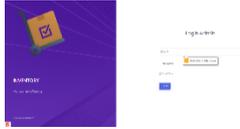
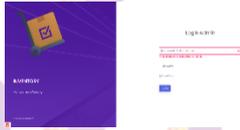
Tabel 4.9 Tabel Rencana Pengujian

Modul Uji	Detail Pengujian	Hasil yang diharapkan
<i>Login</i>	Validasi <i>login</i>	Berhasil masuk ke halaman utama jika <i>login</i> berhasil.
Data Produk	Tambah Data Produk	Proses tambah data produk berhasil dilakukan dengan pesan “Berhasil di simpan”
Transaksi <i>Order</i>	Membuat Faktur <i>Order</i> dan input <i>order</i> produk	Proses pembuatan faktur dan <i>input order</i> produk berhasil di tambahkan

2. Hasil Pengujian *Black Box*

a. Hasil pengujian terhadap *form login*

Tabel 4.10 Tabel Pengujian halaman *login*

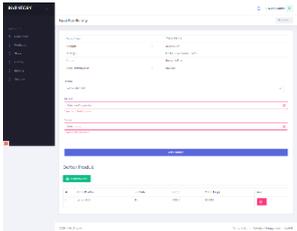
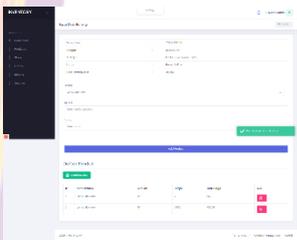
No	Role	Antarmuka yang diuji	Skenario pengujian	Capaian	Gambar
1	Administ rator	Halaman <i>Login</i>	<i>Username, Password</i> tidak di isi maka sistem akan menolak akses dan kembali ke halaman login serta meminta isian terlebih dahulu.	Sesuai	
2	Administ rator	Halaman <i>Login</i>	Memasukan <i>username, password</i> dengan salah maka sistem akan menolak akses dan kembali ke halaman <i>login</i> .	Sesuai	
3	Administ rator	Halaman <i>Login</i>	Memasukan <i>username, password</i> dengan benar maka sistem akan menerima dan akan masuk ke halaman utama dari akunnya.	Sesuai	

b. Hasil Pengujian terhadap *form* Tambah Produk

Tabel 4.11 Tabel Pengujian halaman tambah produk

No	Role	Antarmuka yang diuji	Skenario pengujian	Capaian	Gambar
1	Administ rator	Halaman Tambah Produk	Tidak mengisi salah satu <i>form</i> yang di sediakan maka sistem akan menolak akses dan meminta isian form terlebih dahulu.	Sesuai	
2	Administ rator	Halaman Tambah Produk	Mengisi semua isian dari <i>form</i> yang disediakan dengan benar maka sistem akan menerima dan menampilkan informasi bahwa data yang di masukan berhasil disimpan.	Sesuai	

c. Hasil Pengujian terhadap *form* transaksi *order*Tabel 4.12 Tabel Pengujian halaman transaksi *order*

No	Role	Antarmuka yang diuji	Skenario pengujian	Capaian	Gambar
1	Administrator	Halaman Transaksi <i>Order</i>	Tidak mengisi salah satu form yang di sediakan di formulir input produk order maka sistem akan menolak akses dan meminta isian form terlebih dahulu.	Sesuai	
2	Administrator	Halaman formulir permohonan	Mengisi semua data dengan benar maka sistem akan menerima dan akan memberikan pesan "Data berhasil ditambahkan".	Sesuai	

3. Uji Kuesioner

Dalam proses Uji Kuesioner dilaksanakan oleh 10 orang pengguna, hal ini di dasarkan pada pengujian sample deskriptif menurut Gay, Mills dan Airasian (2009: 133) adalah 10%, dimana sebagai pengguna diambil dari member toko kuat yang berjumlah kurang lebih sekitar 100 toko, sehingga di dapatkan jumlah uji kuesioner adalah 10 orang. Dengan menyebarkan kuesioner bagi minimal 10 orang untuk mengetahui sejauh mana website ini berfungsi dengan baik serta bermanfaat bagi pengguna. Kuesioner dilaksanakan pada hari senin 27 Juli 2020 selama satu minggu, kuesioner tersebut diisi oleh pemilik toko-toko rekanan dari toko “Kuat” karangpucung.

Tabel 4.13 Tabel Uji Kuesioner

NO	PERNYATAAN KUESIONER	Penilaian			
		SS	S	KS	TS
1.	Apakah tampilan aplikasi manajemen transaksi ini menarik / <i>user friendly</i> ?				
2.	Apakah aplikasi manajemen transaksi ini mudah digunakan?				
3.	Apakah dengan adanya aplikasi manajemen transaksi ini pengguna mendapatkan pelaporan sesuai dengan yang dibutuhkan?				
4.	Apakah dengan adanya aplikasi manajemen transaksi ini sudah sesuai dengan kebutuhan?				
5.	Apakah dengan adanya aplikasi manajemen transaksi ini benar-benar bermanfaat dan membantu?				
6.	Apakah aplikasi manajemen transaksi ini sudah siap untuk digunakan?				

Dari hasil skenario pengujian terhadap pengguna dapat di peroleh kesimpulan bahwa sistem aplikasi manajemen transaksi pada toko “Kuat” ini berjalan dengan baik dan sesuai dengan rancangan yang telah di buat.

Pada pengujian yang telah dilakukan semua pengguna bisa melakukan kegiatan sesuai dengan *role* masing-masing. Dari hasil pengujian yang dilakukan pada aplikasi manajemen transaksi pada toko “Kuat” ini maka dapat disimpulkan bahwa semua fungsi dapat dijalankan dan siap digunakan.



BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil analisis dan pembahasan yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut :

1. Telah berhasil dibangun aplikasi manajemen transaksi pada toko kuat karangpucung tambak banyumas.
2. Berdasarkan hasil pengujian menggunakan *blackbox testing*, dapat disimpulkan bahwa aplikasi yang dibangun dapat mempermudah pemilik toko dalam melakukan transaksi dan pelaporan penjualan.
3. Berdasarkan hasil pengujian menggunakan *blackbox testing*, dapat disimpulkan bahwa aplikasi yang dibangun dapat menyajikan laporan data transaksi toko menjadi lebih cepat, lengkap dan akurat.

B. Saran

Aplikasi yang dibangun masih memiliki banyak kekurangan yang dapat diperbaiki maupun dilengkapi oleh pengembang selanjutnya.

Adapun beberapa usulan untuk pengembangan aplikasi ini diantaranya :

1. Dalam aplikasi ini belum disediakan fasilitas untuk *back up* dan *export data*.
2. Sebaik apapun aplikasi yang dibangun, tidak akan berguna jika sumber daya manusia yang ada belum siap untuk menerima dan menggunakannya. Oleh karena itu, untuk bisa menjalankan aplikasi

ini, diharapkan diadakan sosialisasi terlebih dahulu mengenai penggunaan aplikasi kepada pihak-pihak yang berhubungan dengan aplikasi ini secara langsung.

3. Di harapkan perubahan *password* baik untuk *admin* atau pemohon sebaiknya dilakukan setiap bulan untuk mengurangi adanya kecurangan oleh orang-orang yang tidak bertanggung jawab.



DAFTAR PUSTAKA

- Fidianti, A. (2018). *Implementasi manajemen berbasis sekolah dalam meningkatkan prestasi belajar siswa*. Yogyakarta: Gre Publishing
- Gay, L.R., Geoffrey, E. M., dan Peter, W. A. (2009). *Educational research, competencies for analysis and application*. New Jersey: Pearson Education, Inc.
- Haryanto. (2007). *Konsep dasar keperawatan dengan pemetaan konsep*. Jakarta: Salemba Medika
- Kadir, A. (2018). *Dasar pemrograman web dinamis menggunakan PHP (Revisi)*. Yogyakarta: Andi Offset
- Kartomo, S. (2019). *Buku ajar dasar-dasar akuntansi*. Yogyakarta: Deepublish.
- Nugroho, B. (2008). *Aplikasi pemrograman web dinamis dengan PHP dan MySQL*. Yogyakarta: Gava Media
- Puspitosari, H. A. (2010). *Membangun website interaktif dengan adobe creative Suite 5 tingkat dasar*. Yogyakarta: Skripta Media Creative
- Sukamto, R. A., dan Shalahuddin, M. (2013). *Rekayasa perangkat lunak terstruktur dan berorientasi objek*. Bandung: Informatika.
- Sugiyono. (2015). *Metode penelitian kuantitatif, kualitatif dan R&D*. Bandung: Alfabeta
- Supardi, Y., dan Sulaeman. (2019). *Semua bisa menjadi programmer laravel basic*. Jakarta: Elex Media Computindo
- Yakub. (2012). *Pengantar sistem informasi*. Yogyakarta: Graha Ilmu

Lampiran 1. Kartu Bimbingan Skripsi



Lampiran 2. Koding Program

Koding Program Login

```
1 @extends('layouts.fullscreen')
2
3 @section('title') Login @endsection
4 @section('css')
5 <link href="{{ asset('assets/css/demo1/pages/login/login-1.css') }}" rel="stylesheet" type="text/css" />
6 @endsection
7
8 @section('content')
9 <div class="kt-grid kt-grid-hor kt-grid--root kt-login kt-login-v1" id="kt_login">
10 <div class="kt-grid_item kt-grid_item--fluid kt-grid kt-grid--desktop kt-grid--ver-desktop kt-grid--hor-tablet-and-mobile">
11
12 <!--begin:Aside-->
13 @include('layouts.auth-aside')
14 <!--end:Aside-->
15
16 <!--begin:Content-->
17 <div class="kt-grid_item kt-grid_item--fluid kt-grid_item--order-tablet-and-mobile-1 kt_login_wrapper">
18
19 <!--begin:Body-->
20 <div class="kt_login_body">
21
22 <!--begin:SignIn-->
23 <div class="kt_login_form">
24 <div class="kt_login_title">
25 <h3>Login Admin</h3>
26 </div>
27 <!--end:SignIn-->
28
29 <!--begin:Form-->
30 <form method="POST" action="{{ route('login') }}">
31 @csrf
32 <div class="form-group">
33 <input class="form-control @error('email') is-invalid @enderror" name="email" type="email" placeholder="Email"
34 value="{{ old('email') }}" required autocomplete="email" autofocus>
35 @error('email')
36 <span class="invalid-feedback" role="alert">
37 <strong>{{ $message }}</strong>
38 </span>
39 @enderror
40 </div>
41 <div class="form-group">
42 <input class="form-control @error('password') is-invalid @enderror" name="password" type="password" required
43 autocomplete="current-password" placeholder="Password">
44 @error('password')
45 <span class="invalid-feedback" role="alert">
46 <strong>{{ $message }}</strong>
47 </span>
48 @enderror
49 </div>
50 <div class="form-group row">
51 <div class="col-md-6">
52 <div class="form-check">
53 <input class="form-check-input" type="checkbox" name="remember" id="remember" {{ old('remember') ? 'checked'
54 : '' }}>
55 <label class="form-check-label" for="remember">
56 {{ __('Ingat Saya') }}
57 </label>
58 </div>
59 </div>
60 <div class="col-md-6 text-right">
61 </div>
62 </div>
63 <!--end:Form-->
64 <!--begin:Action-->
65 <div class="kt_login_actions">
66 <button id="kt_login_signin_submit" class="btn btn-primary btn-elevate">Login</button>
67 </div>
68 <!--end:Action-->
69 </form>
70 <!--end:Form-->
71 </div>
72 <!--end:Body-->
73 </div>
74 <!--end:SignIn-->
75 </div>
76 <!--end:Content-->
77 </div>
78 </div>
79 <!--end:Body-->
80 </div>
81 </div>
82 <!--end:Content-->
83 </div>
84 </div>
85 @endsection
86
87
```


Kuesioner User Acceptance Test
Aplikasi Manajemen Transaksi Pada Toko Kuat

Nama : THOFIK HIDAYAT

Nama Toko : “Berkah Jaya”

Sebagai : User

Berilah tanda ceklist (√) pada kolom yang sesuai dengan jawaban anda.

Keterangan :

SS = Sangat Setuju

KS = Kurang Setuju

S = Setuju

TS = Tidak Setuju

NO	PERNYATAAN KUESIONER	Penilaian			
		SS	S	KS	TS
1.	Apakah tampilan aplikasi manajemen transaksi ini menarik / <i>user friendly</i> ?	√			
2.	Apakah aplikasi manajemen transaksi ini mudah digunakan?	√			
3.	Apakah dengan adanya aplikasi manajemen transaksi ini pengguna mendapatkan pelaporan sesuai dengan yang dibutuhkan?	√			
4.	Apakah dengan adanya aplikasi manajemen transaksi ini sudah sesuai dengan kebutuhan?		√		
5.	Apakah dengan adanya aplikasi manajemen transaksi ini benar-benar bermanfaat dan membantu?	√			
6.	Apakah aplikasi manajemen transaksi ini sudah siap untuk digunakan?	√			
Jumlah		5	1		

Responden


 (THOFIK HIDAYAT)

Kuesioner User Acceptance Test
Aplikasi Manajemen Transaksi Pada Toko Kuat

Nama : NUNUNG SULISTYANI

Nama Toko : "NUNUNG"

Sebagai : User

Berilah tanda ceklist (✓) pada kolom yang sesuai dengan jawaban anda.

Keterangan :

SS = Sangat Setuju

KS = Kurang Setuju

S = Setuju

TS = Tidak Setuju

NO	PERNYATAAN KUESIONER	Penilaian			
		SS	S	KS	TS
1.	Apakah tampilan aplikasi manajemen transaksi ini menarik / <i>user friendly</i> ?	✓			
2.	Apakah aplikasi manajemen transaksi ini mudah digunakan?	✓			
3.	Apakah dengan adanya aplikasi manajemen transaksi ini pengguna mendapatkan pelaporan sesuai dengan yang dibutuhkan?	✓			
4.	Apakah dengan adanya aplikasi manajemen transaksi ini sudah sesuai dengan kebutuhan?	✓			
5.	Apakah dengan adanya aplikasi manajemen transaksi ini benar-benar bermanfaat dan membantu?	✓			
6.	Apakah aplikasi manajemen transaksi ini sudah siap untuk digunakan?	✓			
Jumlah		6			

Responden


 (NUNUNG SULISTYANI)

Kuesioner User Acceptance Test
Aplikasi Manajemen Transaksi Pada Toko Kuat

Nama : Bambang Sucipto
 Nama Toko : "Toko Hamzah"
 Sebagai : User

Berilah tanda ceklist (√) pada kolom yang sesuai dengan jawaban anda.

Keterangan :

SS = Sangat Setuju

KS = Kurang Setuju

S = Setuju

TS = Tidak Setuju

NO	PERNYATAAN KUESIONER	Penilaian			
		SS	S	KS	TS
1.	Apakah tampilan aplikasi manajemen transaksi ini menarik / <i>user friendly</i> ?	√			
2.	Apakah aplikasi manajemen transaksi ini mudah digunakan?	√			
3.	Apakah dengan adanya aplikasi manajemen transaksi ini pengguna mendapatkan pelaporan sesuai dengan yang dibutuhkan?	√			
4.	Apakah dengan adanya aplikasi manajemen transaksi ini sudah sesuai dengan kebutuhan?	√			
5.	Apakah dengan adanya aplikasi manajemen transaksi ini benar-benar bermanfaat dan membantu?	√			
6.	Apakah aplikasi manajemen transaksi ini sudah siap untuk digunakan?	√			
Jumlah		6			

Responden


 (BAMBANG SUCIPTO)

Kuesioner User Acceptance Test
Aplikasi Manajemen Transaksi Pada Toko Kuat

Nama : ROBY
 Nama Toko : "Toko Roby"
 Sebagai : User

Berilah tanda ceklist (√) pada kolom yang sesuai dengan jawaban anda.

Keterangan :

SS = Sangat Setuju

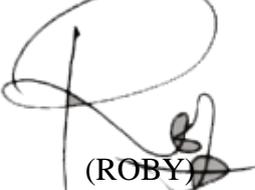
KS = Kurang Setuju

S = Setuju

TS = Tidak Setuju

NO	PERNYATAAN KUESIONER	Penilaian			
		SS	S	KS	TS
1.	Apakah tampilan aplikasi manajemen transaksi ini menarik / <i>user friendly</i> ?	√			
2.	Apakah aplikasi manajemen transaksi ini mudah digunakan?	√			
3.	Apakah dengan adanya aplikasi manajemen transaksi ini pengguna mendapatkan pelaporan sesuai dengan yang dibutuhkan?	√			
4.	Apakah dengan adanya aplikasi manajemen transaksi ini sudah sesuai dengan kebutuhan?	√			
5.	Apakah dengan adanya aplikasi manajemen transaksi ini benar-benar bermanfaat dan membantu?	√			
6.	Apakah aplikasi manajemen transaksi ini sudah siap untuk digunakan?	√			
Jumlah		6			

Responden


(ROBY)

Lampiran 4. Daftar Wawancara

Hasil Wawancara

Nama / Kode : **KUAT / K**

Tanggal : 10 Februari 2020

Isi Deskripsi Hasil Wawancara

1. P : Apa yang Bapak harapkan dari pembuatan Aplikasi manajemen Transaksi ini?
K : Harapannya dengan adanya Aplikasi Transaksi ini dapat mempermudah kami dalam mengelola penjualan dan pelaporan di toko kami.
2. P : Apa masalah yang terjadi pada proses transaksi penjualan dan pelaporan di toko anda selama ini?
K : Pengelolaan di toko kami ini kan masih manual ya mba, jadi sering sekali terjadi selisih di pelaporan penjualan, trus kami juga kesulitan dalam mendata pelanggan kami.
3. P : Apa saja fungsi-fungsi penting yang harus ada dalam Aplikasi Manajemen Transaksi ini?
I.W : Fungsi-fungsi yang harus ada aplikasi ini, yang kami harapkan ada pengelolaan data barang, pengelolaan data pelanggan dan pelaporan penjualan dan distribusi untuk setiap bulannya mba.

Lampiran 5. Metodologi Penelitian

1. Observasi



2. Studi Pustaka



3. Wawancara

