

---

# Rancang Bangun Sistem Informasi *Point of Sales* Berbasis *Website* Pada Distributor Kain Hoggly Djaya

Kevin Chandra, Universitas Ciputra UC Town, Citraland, Surabaya

---

## ABSTRAK

Sebuah sistem yang baik adalah suatu fondasi yang kokoh untuk sebuah perusahaan. Dengan adanya sistem maka segala proses bisnis akan dapat berjalan secara terstruktur dan rapi. Dengan berkembangnya teknologi, maka sistem terkomputerisasi menjadi solusi terbaik karena sangat menguntungkan bagi perusahaan. Distributor kain "Hoggly Djaya" sebelumnya hanya menggunakan sistem pencatatan secara manual yang tidak efisien. Diperlukan adanya pembaharuan sistem dimana segala proses bisnis sebaiknya dimigrasikan supaya dapat dibuat terkomputerisasi. Seiring dengan perkembangan teknologi, *website* juga mengalami banyak kemajuan, yang membuat *website* bisa menjadi sebuah solusi sistem informasi. Banyak teknologi baru yang mendukung sebuah *website* untuk dapat dijadikan sebuah aplikasi. Perkembangan terbaru (sejak penulisan skripsi ini) adalah desain *website* responsif dimana sebuah *website* bisa dibuatkan tampilan untuk *smartphone* dan *tablet*. Selain itu, *website* juga didukung dengan bahasa pemrograman yang kuat yaitu PHP sehingga dapat digunakan sebagai sebuah aplikasi yang kuat, sehingga bukan tidak mungkin bahwa dengan dasaran *website* akan dapat dihasilkan sebuah aplikasi *point of sales* untuk mendukung kinerja perusahaan. Dengan adanya aplikasi *point of sales* yang berbasis *website* dapat memudahkan dan mengoptimalkan proses bisnis pada perusahaan.

Kata kunci: sistem, teknologi, *website*, *point of sales*

---

## 1. Pendahuluan

Sebuah sistem pembukuan merupakan sebuah dasar dari perusahaan yang baik. Demikian pula untuk perusahaan yang bergerak pada bidang retail/distribusi. Pencatatan mutasi barang sangat diperlukan bagi kepentingan audit. Semua itu dapat dikerjakan pada sebuah catatan fisik berupa buku. Seiring dengan berkembangnya teknologi, maka catatan fisik tersebut dinilai sudah tidak lagi dikatakan efisien. Muncul sistem baru yaitu, khususnya pada bidang retail yaitu Point of Sales atau POS. POS akan sangat membantu dan mengoptimalkan kinerja perusahaan karena sistemnya yang serba terkomputerisasi.

Perusahaan "Hoggly Djaya" adalah sebuah perusahaan yang bergerak pada bidang distribusi tekstil. Sebagai sarana pembukuan, perusahaan "Hoggly Djaya" menggunakan sistem manual untuk mencatat semua transaksi. Semua transaksi dicatat pada sebuah nota atau buku.

Dapat disimpulkan bahwa permasalahan utama dari perusahaan "Hoggly Djaya" adalah untuk dihasilkannya

sebuah sistem POS yang dapat berfungsi sebagai sebuah bentuk pembukuan secara terkomputerisasi.

Solusi terbaik adalah untuk membangun sebuah aplikasi pembukuan transaksi berbasis *website*. Solusi ini dapat berguna untuk meringankan proses pembukuan pada perusahaan "Hoggly Djaya" agar dapat terlaksana dengan cepat dan tepat.

Sistem POS yang dibuat harus dapat melakukan beberapa hal, antara lain adalah sistem tersebut harus dapat mencatat pengeluaran dan pemasukan barang serta melakukan deduksi jumlah barang pada stok. Selain itu, sistem juga perlu merekam pembayaran piutang yang dilakukan oleh pelanggan kepada pemilik perusahaan.

Sistem POS yang dibuat juga diharapkan untuk dapat mengelola data secara cepat dan tepat, aplikasi dapat digunakan dengan mudah oleh pemilik, dan efisiensi waktu dalam melakukan pembukuan.

Pada akhirnya, tulisan ini bertujuan untuk melakukan perancangan dan pembuatan sebuah aplikasi sistem informasi *point of sales* yang berbasis *website* agar dapat memudahkan pemilik dari perusahaan "Hoggly Djaya"

---

\* Kevin Chandra.

E-mail: kchandra@student.ciputra.ac.id

dalam mengoperasikan aplikasi tersebut untuk mendapatkan hasil pembukuan dengan cepat dan tepat.

## 2. Landasan Teori

### 2.1 Sistem Informasi

Menurut McLeod (McLeod Jr., 2007), sistem merupakan sebuah kumpulan elemen-elemen yang saling berinteraksi dan memiliki tujuan tertentu. Sebuah sistem memiliki suatu tujuan pencapaian tertentu; biasa disebut *goal*.

### 2.2 Point of Sales (POS)

*Point of Sales* atau disingkat sebagai POS, secara sederhananya dapat diartikan sebagai sebuah perangkat lunak yang melakukan pencatatan akan transaksi jual beli yang terjadi pada perusahaan (ST, 2010). Sebelum sistem POS dikembangkan, orang-orang banyak menggunakan yang kita kenal sebagai mesin *Cash Register* atau bahkan pencatatan manual menggunakan nota dalam menjalankan usahanya. Seiring dengan kemajuan teknologi, *Cash Register* ini dianggap sudah tidak cukup memadai lagi, karena sistem *Cash Register* hanya dapat mencatat/menghitung transaksi. Sehingga muncul pengembangan baru yaitu perangkat lunak POS. Beberapa kegunaan system POS adalah sebagai berikut:

1. Peningkatan kualitas pelayanan
2. Peningkatan citra usaha
3. *Competitive advantage*
4. Kemudahan proses control & pengambilan keputusan

### 2.3 Pengembangan Web

Pada jaman dahulu, pengguna memiliki banyak keterbatasan dalam mengakses *web* yaitu hanya dengan menggunakan aplikasi *browser* pada komputer, namun jika melihat perkembangan teknologi yang cukup pesat saat ini yang sekaligus membuat perkembangan dunia *web* semakin maju, maka banyak alternatif baru dalam sarana mengakses *web*, antara lain dengan menggunakan perangkat handphone dan *smartphone*.

Menurut Gary B. Thomas J dan Misty E (Shelly, Cashman, & Vermaat, 2007), *Smartphone* adalah sebuah telepon yang *internet enabled* yang biasanya menyediakan fungsi *Personal Digital Assistant*, seperti fungsi kalender, buku agenda, buku alamat, kalkulator dan catatan. Dikarenakan penggunaan telepon genggam semakin canggih, banyak orang beralih memilih menggunakan telepon genggam dalam mengakses *web*, selain ukurannya yang lebih kecil dari komputer, telepon genggam juga sangat efisien dalam mengakses *web* kapan dan dimanapun.

Namun ada kendala yang banyak dihadapi oleh pengguna, yaitu desain *web* yang tidak sesuai dengan tampilan pada layar telepon genggam yang cenderung kecil, yang menyulitkan pengguna untuk mengakses *web* tersebut, sehingga munculah solusi dalam teknologi baru yang disebut sebagai *Responsive Web Design*.

### 2.4 Responsive Web Design (RWD)

*Responsive Web Design* (RWD) merupakan sebuah inovasi dalam pengembangan desain *front-end* sebuah *website* yang menentukan penataan elemen pada *web* yang berbeda pada setiap ukuran layar. Pertama kali konsep ini dicetuskan oleh Ethan Martcotte (Marcotte, 2011) pada artikelnya yang berjudul "Responsive Web Design" Ethan menuliskan bahwa konsep desain *web* responsif ada 3 faktor utama yaitu:

- a. Kerangka dasar yang fleksibel
- b. Gambar yang sinkron dengan posisi lebar layar
- c. Melalui media query, membuat sebuah *web* dapat menyesuaikan dengna ukuran layar.

### 2.5 Twitter Bootstrap

Twitter Bootstrap atau kerap disebut Bootstrap, adalah sebuah *front-end framework* yang membantu pengembang *website* untuk memulai proses pengembangan sebuah *website* (Rahman, 2014). Ditujukan untuk membantu pengembang *website* yang fokus pada bidang *back-end* yang cenderung sulit untuk dapat memunculkan sebuah desain yang tampak menarik. Dengan adanya Bootstrap, pengembang *website* hanya perlu memikirkan isi/konten dan mengandalkan Bootstrap sebagai sarana untuk desain.

Bootstrap pertama kali dikembangkan oleh Mark Otto dan Jacob Thornton, sepasang pengembang *website* pada perusahaan Twitter (media sosial), dan dirilis pada pertengahan tahun 2011 (Rahman, 2014). Tujuan dikembangkannya Bootstrap adalah untuk mendatangkan konsistensi dan kode yang terawat.

Syed Fazole Rahman (Rahman, 2014) mengatakan bahwa bootstrap membawa beberapa keuntungan pada dunia pengembangan *website*:

1. Kemudahan dalam membuat elemen-elemen *website* yang sudah didesain, meskipun bukan desainer *website*
2. Menyediakan *grid system* yang berfungsi sebagai sarana pengembangan *responsive web design* untuk menghasilkan desain yang *mobile-friendly*
3. Bootstrap menyediakan fleksibilitas bagi para pengembang *front-end* untuk melakukan kustomisasi terhadap style bawaan sehingga Bootstrap dapat digunakan menjadi sebuah dasaran yang dapat dikembangkan lebih luas.

### 2.6 Basis Data

Bambang Hariyanto (Hariyanto, 2004) menyatakan bahwa sebuah basis data adalah sekumpulan data yang secara logika berkaitan dalam merepresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi pada sistem tertentu.

### 2.7 Database Management System (DBMS)

Database Management System, biasa disingkat sebagai DBMS merupakan sebuah perangkat lunak yang khusus dirancang untuk melakukan pengelolaan dan pengkoleksian data dalam jumlah besar. DBMS ini juga dirancang untuk dapat melakukan manipulasi data dengan mudah.

Menurut Kadir (Kadir, 2003), pengertian DBMS adalah yaitu satu program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memanipulasi dan memperoleh data/informasi yang dilakukan dengan cara praktis dan efisien.

Menurut C.J. Date (Date, 2003), DBMS adalah software yang menangani seluruh akses pada sebuah basis data untuk melayani kebutuhan pengguna.

Menurut Ramez Elmasri dan Shamkant Navathe (Elmasri & Navathe, 2010), DBMS adalah sebuah koleksi program yang dapat memberikan pengguna kemampuan untuk membuat dan mengurus sebuah basis data. DBMS adalah sebuah sistem perangkat lunak dengan tujuan umum yang memberikan fasilitas untuk melakukan proses mendefinisikan, membuat, memanipulasi, dan membagikan sebuah basis data kepada banyak pengguna dan aplikasi.

Dari ketiga pernyataan itu, dapat disimpulkan bahwa DBMS adalah semua peralatan komputer yang dilengkapi dengan bahasa yang berorientasi pada data yang sering disebut sebagai bahasa generasi ke-4.

## 2.8 MySQL

Menurut Allen dan Hornberger (Allen & Hornberger, 2002), MySQL merupakan bahasa pemrograman open-source yang paling populer dan banyak digunakan di lingkungan Linux. Kepopuleran ini karena ditunjang dengan performansi query dari basis datanya yang jarang bermasalah.

MySQL sebenarnya sebuah produk yang berjalan pada platform Linux. Karena sifatnya yang open-source, MySQL dapat berjalan pada berbagai platform seperti Windows maupun Macintosh. Selain itu MySQL juga bersifat sebagai program pengakses basis data yang bersifat jaringan sehingga dapat digunakan untuk aplikasi multi-user (banyak pengguna).

Kelebihan dari MySQL adalah penggunaan bahasa query yang dimiliki SQL (Structured Query Language). SQL merupakan suatu bahasa permintaan yang terstruktur dan sudah distandarisasi untuk semua program pengakses basis data seperti Oracle, PostgreSQL, termasuk MySQL.

Sebagai sebuah program yang menghasilkan basis data, MySQL tidak dapat berjalan sendiri tanpa aplikasi penghubung antarmuka. Hampir semua program dan aplikasi yang baik mendukung kemampuan MySQL, aplikasi yang open-source seperti PHP, maupun aplikasi yang ada di platform pengguna seperti Delphi, Visual Basic, dan lainnya.

## 2.9 Unified Modeling Language (UML)

Unified Modeling Language (berikutnya disebut UML) merupakan sebuah bahasa yang dipergunakan untuk memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. Penggunaan UML dalam industri makin meningkat dan merupakan standar terbuka untuk sebuah bahasa pemodelan yang umum, terutama pada industri perangkat lunak dan pengembangan sistem. Menurut Adi Nugroho (Nugroho, 2009), UML adalah sebuah alat bantu analisis serta perancangan perangkat

lunak yang berbasis orientasi objek.

Tujuan dibuatnya UML ini adalah sebagai berikut:

1. Dapat memberikan sebuah bahasa pemodelan yang bebas dari bahasa pemrograman dan proses rekayasa.
2. Memberi model yang siap dipakai. Ini disebabkan karena UML adalah sebuah bahasa pemodelan yang visual dan ekspresif untuk dikembangkan dan saling menukar model
3. Menyatukan praktek terbaik yang ada pada proses pemodelan
4. Dapat berfungsi sebagai dasar (blueprint) sebuah aplikasi karena bersifat sangat lengkap dan detail. Dengan adanya cetak biru atau blueprint akan dapat dilakukan reverse engineering pada aplikasi.

Macam-macam diagram yang ada pada UML ini beraneka ragam. Untuk kepentingan skripsi ini, sudah dipilih beberapa yang umum dan paling berdampak. Berikut adalah beberapa penjelasan mengenai diagram yang digunakan pada tulisan ini:

### 1. Use Case Diagram

Use Case Diagram adalah sebuah lapisan abstraksi dari interaksi antara sistem dengan aktor. Cara kerja use case ini adalah dengan mendeskripsikan setiap interaksi yang ada. Use Case adalah sebuah konstruksi dimana sebuah sistem akan dapat dideskripsikan bagaimana sebuah sistem tersebut akan terlihat di mata pengguna.

### 2. Activity Diagram

Activity Diagram berguna untuk dapat menggambarkan aliran dari aktivitas yang ada pada sebuah sistem. Diagram ini digunakan dalam mendeskripsikan alur pada suatu operasi serta alur pada interaksi use case

### 3. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku sebuah skenario. Diagram ini berfungsi untuk menunjukkan rangkaian pesan dan data yang dikirim antara objek dan interaksi antar objek, sesuatu yang terjadi pada titik tertentu pada eksekusi sistem.

## 3. Analisis dan Rancangan Sistem

### 3.1 Analisis

Sebuah *Point of Sales* (POS) secara umum akan memiliki sistem sederhana untuk melakukan penjualan. Pada dasarnya, sistem tersebut akan memiliki sebuah aksi untuk menambah barang ke penjualan dan aksi untuk melakukan pembayaran.

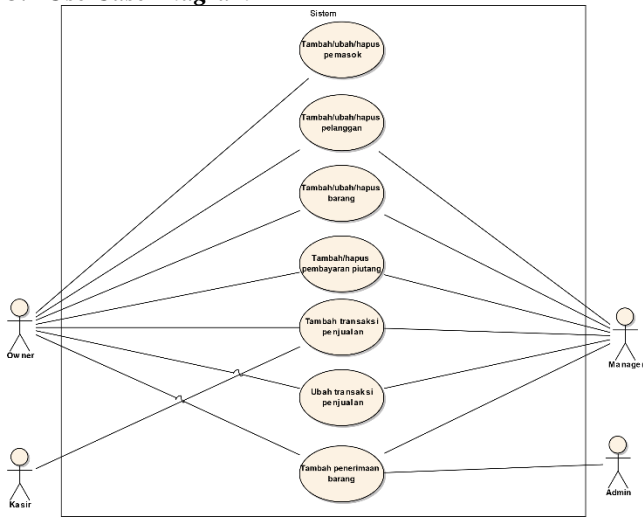
Sistem tersebut mengandalkan basis data untuk dapat melakukan transaksi penjualan. Hal tersebut dikarenakan oleh sebuah sistem penjualan akan membutuhkan untuk membaca basis data sebagai referensi untuk melakukan pengecekan ketersediaan barang dan merekam pengeluaran barang.

Perusahaan “Hoggy Djaya” bergerak dalam bidang distribusi. Sementara ini pemilik “Hoggy Djaya” melakukan pencatatan semua transaksi secara manual melalui pencatatan pada buku besar. Kesulitan yang dihadapi adalah saat melakukan rekap ulang, karena pencarian data yang di-input pada buku tersebut sulit ditemukan, membuat pekerjaan jadi rumit. Disamping itu, karena semua dilakukan secara manual, maka tidak dapat dihindari bahwa dapat terjadi adanya kesalahan input data.

Kami (perusahaan “Inertia”) menawarkan jasa kami untuk membantu meringankan pekerjaan manual tersebut. Pemilik dari perusahaan “Hoggy Djaya” memberikan beberapa rincian permintaan untuk sistem POS bagi perusahaannya:

1. Dapat melakukan penambahan barang ke transaksi.
2. Dapat melakukan transaksi untuk per-pelanggan.
3. Dapat melakukan perekaman pembayaran.
4. Dapat melihat beberapa transaksi terakhir dari pelanggan terpilih.
5. Dapat melakukan penundaan transaksi

3.2 Use Case Diagram

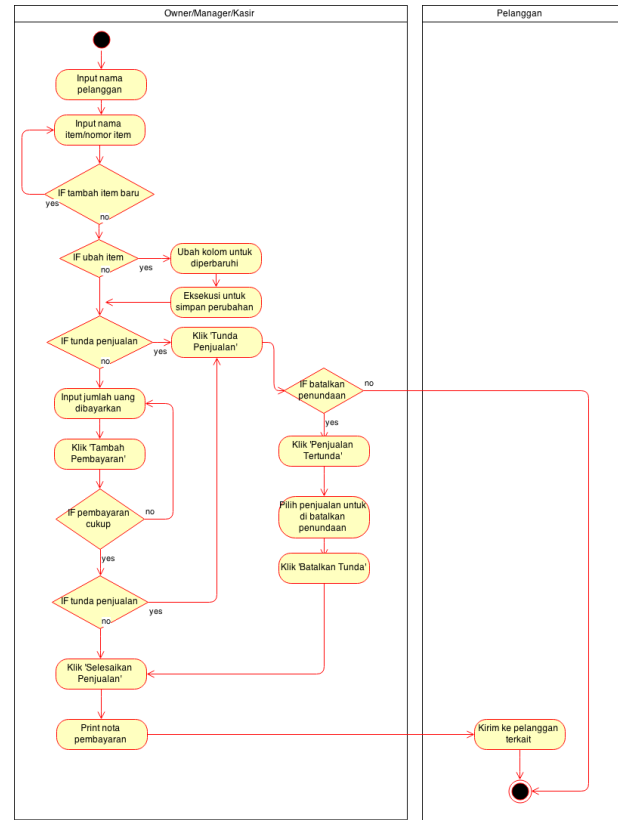


Gambar 1 Use Case Diagram

Penjelasan mengenai diagram:

1. Owner memiliki semua hak untuk mengakses semua modul
2. Manager akan memiliki beberapa akses yang tidak diperbolehkan, seperti melakukan pengelolaan pemasok.
3. Kasir hanya dapat melakukan transaksi penjualan.
4. Admin hanya memiliki tanggung jawab untuk modul tambah penerimaan barang.

3.3 Flowchart



Gambar 2 Flowchart

Penjelasan diagram:

Aktor: Owner/Manager/Kasir

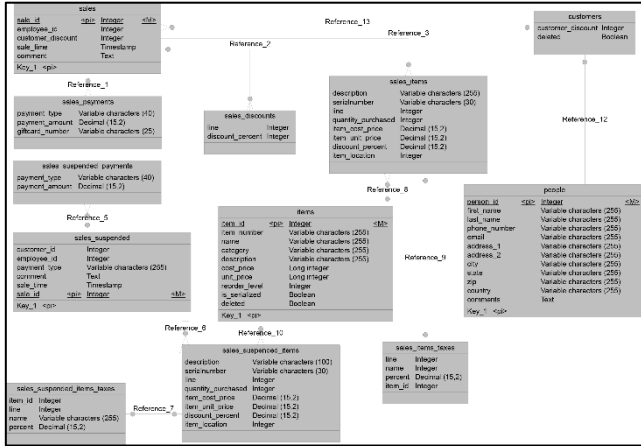
1. Agar dapat melakukan transaksi yang merekam pelanggan, aktor akan diharuskan untuk menentukan kepada siapa penjualan tersebut ditujukan
2. Setelah menentukan pelanggan, aktor akan memasukkan barang/item pada penjualan.
3. Apabila item memiliki tambahan kuantitas atau informasi lain, aktor akan dapat melakukan perubahan pada barang dan menambah kuantitas.
4. Jika diperlukan penambahan item, maka aktor akan diminta untuk memasukkan nama barang atau kode barang untuk dimasukkan ke penjualan.
5. Apabila penjualan diharuskan untuk ditunda, maka aktor akan melakukan penundaan transaksi dengan mengeksekusi tombol `tunda penjualan`.
6. Untuk membatalkan penundaan, aktor akan mengeksekusi tombol `penjualan tertunda` dan memilih penjualan yang akan dibatalkan penundaannya.
7. Untuk menyelesaikan transaksi penjualan, aktor harus memasukkan jumlah yang dibayarkan oleh pelanggan.
8. Apabila jumlah yang dibayarkan belum mencukupi, maka aktor diharuskan untuk menambahkan pembayaran hingga mencukupi total.

- 9. Saat pembayaran sudah mencukupi, maka tombol `selesaikan penjualan` harus dieksekusi untuk menyelesaikan transaksi dan merekam semua data transaksi ke basis data.
- 10. Aktor akan dibawa ke tampilan faktur yang harus dicetak sebagai bukti fisik akan penyelesaian transaksi.

transaksi dapat memiliki pajak, saat penjualan ditunda, maka informasi tersebut akan disimpan pada Tabel `sales\_suspended\_items\_taxes`, menghasilkan relasi one to one antara `sales\_suspended\_items` dengan `sales\_suspended\_items\_taxes` → Reference\_7

8. Setiap barang akan dapat digunakan dalam banyak transaksi. Untuk itu relasi yang diperlukan antara Tabel `items` dengan `sales\_items` adalah relasi one to many → Reference\_8

- 9. Setiap barang pada penjualan akan dapat dikenakan pajak, maka data perpajakan tersebut disimpan pada Tabel `sales\_items\_taxes` dengan relasi antara tabel tersebut dengan `sales\_items` adalah one to one karena tiap baris item akan memiliki satu informasi pajak → Reference\_9
- 10. Setiap barang yang ditunda penjualannya, akan disimpan pada tabel di basis data. Relasi yang dihasilkan dengan data barang pada Tabel `items` adalah one to many dimana satu barang dapat digunakan untuk berbagai transaksi tertunda → Reference\_10
- 11. Setiap pelanggan akan memiliki data pribadi yang disimpan pada Tabel `people` (hasil normalisasi). Sehingga relasi antara Tabel `customers` dengan `people` adalah one to one karena tiap informasi pelanggan hanya akan memiliki satu kolerasi dengan informasi pribadi yang dimilikinya → Reference\_12
- 12. Setiap penjualan akan ditujukan kepada hanya satu pelanggan, yang menghasilkan relasi one to one antara Tabel `sales` dengan `customers` → Reference\_13



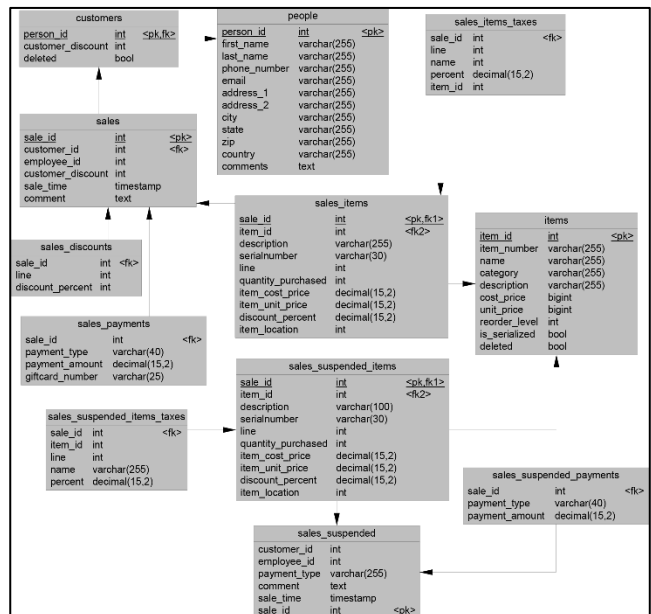
Gambar 3 Conceptual Data Model

### 3.4 Conceptual Data Model (CDM)

Beberapa asumsi untuk relasi tabel yang tertera:

- 1. Tabel `sales` akan memiliki banyak barang, karena itu akan diperlukan relasi one to many untuk relasi antara `sales` dengan `sales\_items` → Reference\_3
- 2. Tiap `sales\_items` akan memiliki satu data perpajakan barang, menghasilkan relasi one to one antara `sales\_items` dengan `sales\_items\_taxes` → Reference\_9
- 3. Untuk setiap `sales` akan memiliki banyak macam pembayaran. Sehingga relasi yang dihasilkan adalah one to many kepada Tabel `sales\_payments` → Reference\_1
- 4. Setiap `sales` akan memiliki banyak diskon. Diskon akan direkam pada Tabel `sales\_discounts` yang menghasilkan relasi one to many untuk `sales` kepada `sales\_discounts` → Reference\_2
- 5. Untuk transaksi penjualan yang ditunda, semua informasi transaksi akan disimpan pada basis data untuk mempermudah penyimpanan. Relasi yang dihasilkan kepada `sales\_suspended\_items` adalah one to many dimana satu `sales\_suspended` dapat memiliki banyak barang → Reference\_6
- 6. Untuk transaksi yang ditunda, jika sudah ada pembayaran yang dimasukkan, maka pembayaran tersebut akan disimpan pada Tabel `sales\_suspended\_payments`, maka relasi yang dihasilkan adalah one to many, dimana satu `sales\_suspended` akan memiliki banyak `sales\_suspended\_payments` → Reference\_5
- 7. Setiap barang yang sudah dimasukkan ke dalam

### 3.5 Physical Data Model (CDM)

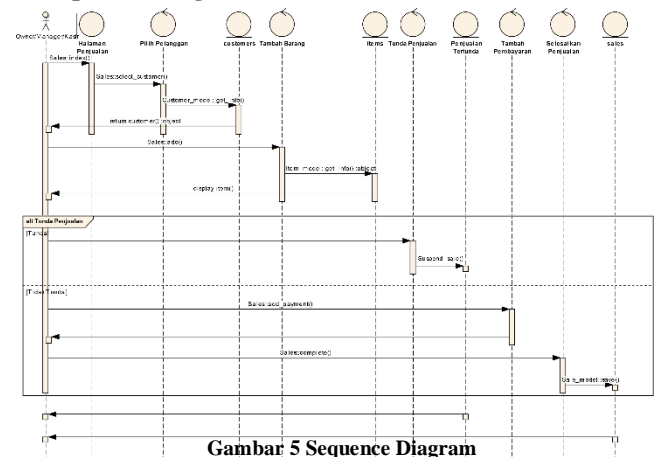


Gambar 4 Physical Data Model

Penjelasan atribut tabel:

1. Tabel `sales` memiliki atribut `sale\_id` sebagai Primary Key (PK) yang digunakan untuk mengidentifikasi setiap rekaman data.
2. Tabel `sales\_items` memiliki atribut `sale\_id` sebagai Foreign Key (FK) untuk menghubungkan tiap `sales\_item` kepada rekaman `sales`.
3. Tabel `sales\_items\_taxes` memiliki `sale\_id` sebagai FK untuk menghubungkan tabel tersebut pada rekaman `sales\_items` yang bersangkutan.
4. Tabel `sales\_discounts` memiliki atribut `sale\_id` sebagai FK untuk menghubungkan diskon yang direkam kepada rekaman `sales` yang bersangkutan.
5. Tabel `sales\_payments` menggunakan referensi `sale\_id` sebagai atribut FK yang digunakan untuk mereferensikan rekaman pada Tabel `sales`.
6. Tabel `sales\_suspended` merekam `sale\_id` sebagai atribut PK untuk mengidentifikasi setiap rekaman data.
7. Tabel `sales\_suspended\_items` mereferensikan `sale\_id` dari Tabel `sales\_suspended` untuk digunakan sebagai FK untuk mengidentifikasi tiap rekaman.
8. Tabel `sales\_suspended\_items\_taxes` memiliki atribut `sale\_id` yang digunakan sebagai FK untuk menghubungkan tiap `sales\_suspended\_items\_taxes` kepada rekaman `sales\_suspended\_items`.
9. Tabel `sales\_suspended\_payments` menggunakan atribut `sale\_id` sebagai FK untuk mereferensi Tabel `sales\_suspended`.
10. Tabel `customers` memiliki atribut `person\_id` yang bersifat FK dari Tabel `people` hal ini karena hasil dari normalisasi yang menyebabkan Tabel `customers` dipisah dari data pribadi tiap orang yang ada pada Tabel `people`.
11. Tabel `people` memiliki atribut `person\_id` yang bersifat PK yang digunakan untuk mengidentifikasi tiap data.
12. Tabel `items` memiliki atribut PK yaitu `item\_id` yang digunakan sebagai referensi pada Tabel `sales\_items` dan `sales\_suspended\_items` untuk mengidentifikasi data barang yang ada pada Tabel `items`.

### 3.6 Sequence Diagram



Gambar 5 Sequence Diagram

Penjelasan diagram:

Aktor: Owner/Manager/Kasir

1. Aktor mengakses halaman penjualan setelah berhasil melakukan proses autentikasi.
2. Aktor memilih pelanggan untuk transaksi yang akan dilakukan.
  - sistem akan mengakses lapisan Controller (interaksi antara pengguna aplikasi dengan sistem) untuk mencari informasi pelanggan yang dipilih.
  - sistem akan mengakses lapisan Model (interaksi dengan basis data) `customers` untuk mendapatkan informasi mengenai pelanggan yang terpilih.
3. Aktor menambah barang pada transaksi
  - sistem akan mengakses lapisan Controller untuk melakukan penambahan barang.
  - sistem akan mengakses lapisan Model `items` untuk mendapatkan informasi mengenai barang yang akan ditambahkan.
4. Apabila aktor mengambil keputusan untuk menunda transaksi penjualan:
  - sistem akan mengakses lapisan Controller untuk melakukan penundaan transaksi.
  - sistem akan mengakses lapisan Model `sales\_suspended` untuk menyimpan data transaksi yang akan ditunda pada basis data.
5. Apabila aktor mengambil keputusan untuk melanjutkan transaksi, maka selanjutnya aktor akan melakukan penambahan metode pembayaran.
  - sistem akan mengakses lapisan Controller untuk menambahkan metode pembayaran.
6. Aktor akan menyelesaikan transaksi penjualan.
  - sistem akan mengakses lapisan Controller untuk menyelesaikan penjualan.
  - sistem akan mengakses lapisan Model `sales` untuk menyimpan data penjualan pada basis data.

## 4. Implementasi Sistem

### 4.1 User Acceptance Test (UAT)

Uji Penerimaan atau *User Acceptance Test* (berikutnya disebut UAT) adalah sebuah tahap pada langkah akhir pengembangan suatu aplikasi sebelum aplikasi tersebut dipasarkan. UAT adalah sebuah metode pengujian yang dapat menentukan apakah suatu aplikasi sudah layak dipublikasikan atau *launching*.

Suatu aplikasi akan dinyatakan sudah layak untuk dirilis apabila aplikasi tersebut dapat lulus melewati semua kasus UAT. Apabila masih terdapat adanya kegagalan pada hasil UAT, maka aplikasi akan dinyatakan belum siap rilis dan akan ditahan untuk dilakukan perbaikan dan sesi UAT ulang untuk memastikan bahwa semua fungsi aplikasi sudah berjalan dengan sempurna.

Berikut akan dicantumkan *User Acceptance Test* yang dilaksanakan pada aplikasi ini:

**Tabel 1. Use Case Tambah Penjualan**

<b>Deskripsi</b>	<i>User akan membuat penjualan baru</i>	
<b>Pengguna</b>	<i>Owner/Manager/Kasir</i>	
<b>Hasil yang diharapkan</b>	yang	<ul style="list-style-type: none"> <li>User berhasil membuat penjualan baru</li> <li>Berhasil mencapai halaman faktur</li> <li>Data penjualan disimpan di basis data</li> </ul>
<b>Hasil yang didapat</b>	yang	<ul style="list-style-type: none"> <li>Halaman faktur berhasil muncul dan di print</li> <li>Data penjualan sudah di cek dan ada di laporan penjualan</li> </ul>
<b>Status</b>	LOLOS	

**Tabel 2. Use Case Ubah Penjualan**

<b>Deskripsi</b>	<i>User akan merubah data penjualan</i>	
<b>Pengguna</b>	<i>Owner/Manager</i>	
<b>Hasil yang diharapkan</b>	yang	<ul style="list-style-type: none"> <li>Data penjualan berhasil dirubah dan disimpan di basis data</li> </ul>
<b>Hasil yang didapat</b>	yang	<ul style="list-style-type: none"> <li>Data penjualan berubah</li> </ul>
<b>Status</b>	LOLOS	

**Tabel 3. Use Case Tambah Pembelian**

<b>Deskripsi</b>	<i>User akan menambah pembelian baru</i>	
<b>Pengguna</b>	<i>Owner/Admin</i>	
<b>Hasil yang diharapkan</b>	yang	<ul style="list-style-type: none"> <li>User berhasil membuat pembelian baru</li> <li>Berhasil mencapai halaman faktur</li> <li>Data pembelian disimpan di basis data</li> </ul>
<b>Hasil yang didapat</b>	yang	<ul style="list-style-type: none"> <li>Faktur pembelian sudah muncul dan di print</li> <li>Data pembelian sudah masuk di laporan pembelian</li> </ul>
<b>Status</b>	LOLOS	

**Tabel 4. Use Case Pembayaran Piutang**

<b>Deskripsi</b>	<i>User akan melakukan pembayaran piutang pelanggan</i>	
<b>Pengguna</b>	<i>Owner/Manager</i>	
<b>Hasil yang diharapkan</b>	yang	<ul style="list-style-type: none"> <li>Pembayaran piutang berhasil disimpan di basis data</li> <li>Data pembayaran piutang dapat dilihat di data penjualan</li> <li>Data pembayaran penjualan ditambahkan</li> </ul>
<b>Hasil yang didapat</b>	yang	<ul style="list-style-type: none"> <li>Pembayaran piutang muncul pada halaman info penjualan</li> <li>Pembayaran penjualan sudah diperbarui dengan nominal baru</li> </ul>
<b>Status</b>	LOLOS	

**Tabel 5. Use Case Hapus Pembayaran Piutang**

<b>Deskripsi</b>	<i>User akan menghapus/membatalkan pembayaran piutang</i>	
<b>Pengguna</b>	<i>Owner/Manager</i>	
<b>Hasil yang diharapkan</b>	yang	<ul style="list-style-type: none"> <li>Pembayaran piutang berhasil dihapus dari basis data</li> <li>Data pembayaran piutang tidak muncul</li> <li>Data pembayaran penjualan sudah dikurangi</li> </ul>
<b>Hasil yang didapat</b>	yang	<ul style="list-style-type: none"> <li>Pembayaran piutang berhasil dihapus</li> <li>Data pembayaran penjualan sudah dikurangi sesuai nominal</li> </ul>
<b>Status</b>	LOLOS	

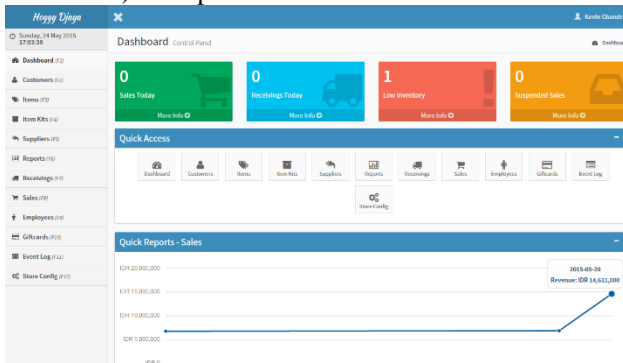
### 4.2 Hasil User Acceptance Test

Menurut hasil UAT yang telah dilakukan, dapat dinyatakan bahwa aplikasi dengan versi 0.20.0b ini siap untuk didistribusikan. Karena setiap fungsi sudah dilakukan percobaan dan sudah berhasil mencapai hasil yang diharapkan.

Tahap selanjutnya dari proses pengembangan aplikasi ini adalah untuk mempersiapkan aplikasi menjadi sebuah aplikasi yang siap untuk didistribusikan, ini termasuk melakukan pengoptimalan kode aplikasi.

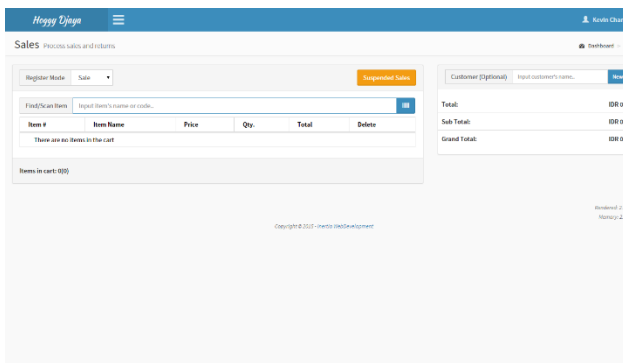
### 4.3 Tampilan Antar Muka

Berikut akan ditampilkan berbagai tampilan antar muka (user-interface) dari aplikasi:



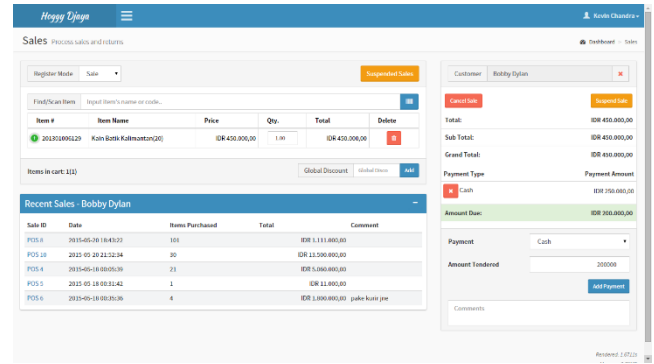
Gambar 6 Halaman Beranda

Pada halaman utama atau sering disebut 'beranda', akan ditampilkan beberapa komponen, antara lain adalah komponen informasi, komponen akses cepat, dan komponen grafik singkat.



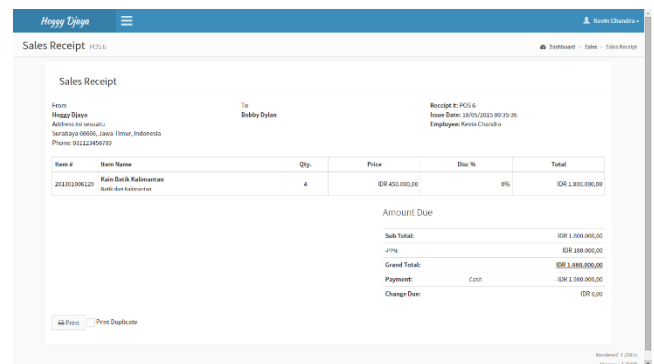
Gambar 7 Halaman Transaksi Penjualan

Pada halaman transaksi penjualan, terbagi menjadi dua kolom utama, dimana kolom sebelah kiri akan berisi daftar barang yang terdaftar pada penjualan dan kolom sebelah kanan berisi informasi mengenai transaksi yang sedang dilaksanakan. Informasi tersebut mencakup total, pembayaran, hingga informasi pelanggan.



Gambar 8 Halaman Transaksi Penjualan Dalam Proses

Halaman ini merupakan halaman dimana sebuah transaksi sedang berlangsung. Di bawah tabel daftar barang, ditampilkan sebuah tabel yang berisi daftar transaksi yang dilakukan oleh pelanggan terkait.



Gambar 9 Halaman Faktur Penjualan

Halaman faktur penjualan pada bagian atas ditampilkan informasi mengenai pelanggan serta informasi mengenai faktur. Akan ditampilkan sebuah tabel berisi daftar barang yang terkait pada transaksi penjualan. Pada bagian bawah ada sebuah daftar yang berisi mengenai informasi pembayaran yang dilakukan pelanggan. Halaman faktur sudah siap untuk dicetak.

### 4.4 Kesimpulan dan Saran

#### 4.4.1 Kesimpulan

Dari hasil analisis kebutuhan dan rancangan sistem *Point of Sale* untuk Hoggy Djaya, dapat ditarik beberapa kesimpulan utama yaitu:

1. Sistem *Point of Sale* berbasis *website* sudah berhasil dibuat dan disetujui oleh calon pengguna.
2. Beberapa kebutuhan yang klien inginkan telah dapat direalisasikan.
3. Fitur khusus diluar permintaan klien sudah ditambahkan sebagai fitur pendukung kinerja aplikasi.
4. Aplikasi sudah berhasil memenuhi UAT yang menyatakan bahwa aplikasi siap dirilis ke public/didistribusikan.



#### 4.4.2 Saran

Berdasarkan hasil kajian dan penelitian terhadap kebutuhan pelanggan dan perancangan sistem Point of Sale untuk Hoggy Djaya, penulis memberikan beberapa saran sebagai berikut:

1. Kepada pembaca, diharapkan skripsi ini dapat menjadi referensi untuk dasar pengembangan aplikasi Point of Sale berbasis website. Skripsi ini mengandung rancangan dan analisis sistem untuk membuat sebuah aplikasi Point of Sales, sehingga pembaca dapat memanfaatkan untuk mendapatkan ilmu serta pengetahuan akan rancangan sistem Point of Sale.
2. Kepada perusahaan Hoggy Djaya (pelanggan) diharapkan dapat bisa memanfaatkan aplikasi Point of Sale dengan sepenuhnya. Fitur yang diberikan pada sistem sangat bermanfaat bagi kemudahan dan guna mempersingkat serta meringankan pekerjaan pencatatan transaksi.
3. Kepada calon pengembang aplikasi, diharapkan dapat menjadikan aplikasi Point of Sale berbasis website ini sebagai dasar pengembangan aplikasi baru atau dikembangkan dan ditambahkan berbagai fitur yang lebih canggih. Sebagai saran utama, penulis menyarankan untuk dapat menulis ulang kode program dari dasaran kerangka kerja CodeIgniter dan dimigrasikan pada kerangka kerja yang lebih modern dan kuat seperti Laravel.

#### DAFTAR PUSTAKA

- 
- Allen, J., & Hornberger, C. (2002). *Mastering PHP 4.1*. Alameda, CA: Sybex.
- Date, C. (2003). *An Introduction to Database Systems (8th Edition)*. New York: Pearson.
- Elmasri, R., & Navathe, S. B. (2010). *Fundamentals of Database Systems (6th Edition)*. USA: Pearson.
- Kadir, A. (2003). *Pengenalan Sistem Informasi*. Yogyakarta: Andi.
- Marcotte, E. (2011). *Responsive Web Design*. New York: A Book Apart.
- McLeod Jr., R. (2007). *Sistem Informasi Manajemen*. Indonesia: Indeks.
- Nugroho, A. (2009). *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. Yogyakarta: Andi.
- Rahman, S. F. (2014). *Jump Start Bootstrap*. Australia: SitePoint Pty, Limited.
- Shelly, G. B., Cashman, T. J., & Vermaat, M. E. (2007). *Discovering Computers: Fundamentals, 3rd ed. (terjemahan)*. Jakarta: Salemba Infotek.
- ST, H. (2010). *Membangun Aplikasi Point of Sale dengan VB 6.0, MySQL, dan PHP*. Jakarta: PT Elex Media Komputindo.