

## Perancangan Sistem Informasi *Point of Sale* Berbasis Website pada Toko Arpan Electric

Dean Abner Julian Gerung

Fakultas Teknologi Informasi, Sistem Informasi, Universitas Nusa Mandiri, Jakarta Pusat, Indonesia

### INFORMASI ARTIKEL

Diterima Redaksi: 17 Oktober 2022  
Revisi Akhir: 22 Oktober 2022  
Diterbitkan Online: 23 Oktober 2022

### KATA KUNCI

*Point of Sale*, *Laravel*, *Vue.JS*, Pemrograman Web

### KORESPONDENSI

Phone: +62 812 1856 261  
E-mail: 11211289@nusamandiri.ac.id

### A B S T R A K

Dalam era globalisasi sekarang ini teknologi informasi melaju dengan cepatnya. Penggunaan teknologi informasi sangat dibutuhkan dalam pengembangan suatu sistem untuk suatu bisnis, termasuk dalam bisnis perdagangan. Saat ini, masih cukup banyak Perusahaan yang bergerak dalam bisnis perdagangan yang menggunakan sistem yang masih manual, sementara beberapa Perusahaan lain yang berhasil telah melakukan transformasi menggunakan dukungan teknologi informasi, seperti penggunaan teknologi internet dan aplikasi berbasis web, untuk memberikan nilai tambah dalam penyimpanan data, mengelola data persediaan barang, pemasok, dan pelanggan, mengelola penjualan/pembelian, serta penyajian laporan. Kekurangan dalam sistem manual di antaranya adalah pihak operasi bisnis menjadi sulit dalam mengecek persediaan barang dagang serta riwayat transaksi yang terjadi dalam rentang waktu tertentu. Penulis melakukan penelitian ini untuk mengembangkan aplikasi *Point of Sale* yang diharapkan dapat membantu dan memberikan solusi terhadap permasalahan yang dihadapi bisnis.

### PENDAHULUAN

Era globalisasi seperti saat ini, perkembangan teknologi informasi melaju dengan cepat. Begitu juga dengan penerapannya dalam berbagai aspek. Teknologi informasi sangat dibutuhkan dalam pengembangan sistem informasi untuk mendukung suatu bisnis, termasuk dalam bisnis perdagangan. Sistem adalah sekelompok komponen yang saling bekerja sama menuju tujuan bersama dengan menerima *input* dan *output* dalam suatu proses transformasi yang terorganisir [1]. Sistem memiliki tiga komponen dasar, yaitu a) input, yaitu menangkap elemen yang memasuki sistem untuk diproses, misalnya bahan baku, energi, data, dan usaha manusia; b) *processing*, yaitu proses transformasi yang mengkonversi input ke output, misalnya proses manufaktur, atau perhitungan matematis; c) output, yaitu mentransfer elemen yang telah diproduksi oleh proses transformasi ke tujuan akhir, misalnya informasi manajemen harus dikirimkan ke pengguna. Sistem informasi adalah hasil dari data yang telah diproses kemudian sudah menjadi bentuk yang memiliki arti dan lebih bermanfaat sesuai dengan keperluan tertentu serta dapat digunakan untuk pengambilan keputusan bagi penerimanya [2].

Toko Arpan *Electric* merupakan toko yang bergerak dalam bidang penjualan alat-alat kebutuhan listrik. Toko mendapatkan pasokan barang dari vendor, baik dari perusahaan besar maupun kecil, untuk pencatatannya akan dilakukan oleh kasir. Sementara itu, proses penjualan yang dilakukan pada Toko Arpan *Electric* dilakukan secara langsung di toko, dengan sistem pembayaran secara lunas dan dengan pembayaran dimuka (secara bertahap). Dengan adanya transaksi penjualan dengan pembayaran di muka ini, Kasir memerlukan sistem pencatatan yang baik dan tersimpan dengan rapih agar dapat dengan mudah mencari riwayat maupun sisa tagihan penjualan tersebut. Selain itu, Kasir juga butuh menyajikan informasi secara cepat dan akurat sesuai dengan kebutuhan. Saat ini seluruh proses bisnis pada Arpan *Electric* dilakukan secara manual, baik dalam mengelola data-data maupun pencatatan transaksi yang dilakukan. Hal ini menyebabkan Kasir mengalami kesulitan, misalnya dalam mencari stok barang yang tersedia, sehingga dapat menyebabkan toko kehilangan potensi penjualan atau bahkan kerugian.

Di dalam perkembangan sistem dan teknologi informasi, telah dikembangkan *Point of Sales* (POS) untuk mendukung komputerisasi dalam proses transaksi penjualan. *Point of sales* merupakan perangkat elektronik yang digunakan untuk menghitung dan merekam seluruh transaksi penjualan dan mengeluarkan bukti pembayaran yang dilakukan oleh pembeli [3]. Setiap POS terdiri dari *hardware* (Terminal/PC, *receipt printer*, *cash drawer*, terminal pembayaran, *barcode scanner*) dan *software* (*inventory management*, pelaporan, *purchasing*, *customer management*, standar keamanan transaksi, *retur processing*). Kedua komponen tersebut digunakan untuk setiap proses transaksi [4].

Berdasarkan paparan di atas, Penulis memandang penerapan Sistem *Point of Sales* (POS) dapat menjadi solusi bagi permasalahan yang saat ini dihadapi Toko Arpan Electric. Hal ini sejalan dengan beberapa penelitian terdahulu yang telah dilakukan sebelumnya, seperti penelitian yang dilakukan oleh [3], [5], dan penelitian [4].

## TINJAUAN PUSTAKA

### *Konsep Dasar Sistem Informasi*

*Heading* pada level kedua dituliskan dengan *boldface italics* dengan menggunakan huruf besar dan huruf kecil. *Heading* dituliskan rata kiri.

#### *Pengertian Sistem*

Sistem adalah sekumpulan prosedur yang saling berhubungan untuk mencapai tujuan tertentu.[6]

#### *Pengertian Informasi*

Informasi adalah hasil dari data yang telah diproses kemudian sudah menjadi bentuk yang memiliki arti dan lebih bermanfaat sesuai dengan keperluan tertentu serta dapat digunakan untuk pengambilan keputusan bagi penerimanya.[2]

#### *Pengertian Sistem Informasi*

Sistem informasi adalah suatu kombinasi manusia, fasilitas atau alat teknologi, media, prosedur dan pengendalian bermaksud menata jaringan komunikasi yang penting bagi pengguna atau penerima. [7]

*Heading* pada level ketiga mengikut *style* dari *heading* level kedua. Hindari penggunaan *heading* lebih dari tiga level.

### *Point Of Sales (POS)*

Pengertian dari *Point Of Sales* (POS) yaitu merupakan kegiatan yang berorientasi pada penjualan serta sistem yang membantu proses transaksi[4]. Setiap POS terdiri dari hardware berupa (Terminal/PC, *Receipt Printer*, *Cash Drawer*, Terminal pembayaran, *Barcode Scanner*) dan software berupa (*Inventory Management*, Pelaporan, *Purchasing*, *Customer Management*, Standar Keamanan Transaksi, *Retur Processing*) dimana kedua komponen tersebut digunakan untuk setiap proses transaksi.

*Point Of Sale* merupakan tempat dimana transaksi selesai, yang dimana pelanggan melakukan pembayaran atas pertukaran barang atau jasa, yang nantinya akan mengeluarkan tanda terima untuk transaksi pembelian atau pembayaran jasa. [5] Sehingga *point of sale* akan mempermudah transaksi dimana pemanfaatan teknologi dengan membuat sistem yang dapat menghitung data secara otomatis dan menulis ulang dibuku besar yang suatu nanti data bisa hilang jika tidak di persiapkan penyimpanannya dengan matang.

### *Pengertian Website*

Website adalah sekumpulan halaman yang berisi informasi berbentuk digital. Informasi tersebut bisa berupa teks, gambar, audio, video, animasi atau gabungan dari semuanya[8]. Website pada umumnya dapat diakses oleh banyak orang diseluruh dunia selama ia memiliki koneksi internet. Website dibentuk dari tiga komponen yang saling melengkapi. Ketiga komponen tersebut adalah *words*, *pictures*, dan *code*.

### *Bahasa Pemrograman*

Pada Perancangan Aplikasi Point of Sales ini, penulis menggunakan beberapa bahasa pemrograman sebagai berikut:

## PHP

Untuk mempercepat pembuatan sistem penulis menggunakan bantuan *Framework Laravel* yang dimana pondasi dari bahasa pemrogramannya adalah bahasa pemrograman PHP dan terdapat alat-alat yang sangat membantu dalam pembuatan *Model, Controller* dan lain-lain.

*PHP (Hypertext Preprocessor)* adalah bahasa *script* tujuan umum *open source* yang banyak digunakan yang sangat cocok untuk pengembangan web dan dapat disematkan ke dalam *HTML*. [9]

Sistem kerja dari *PHP* diawali dengan permintaan yang berasal dari halaman website oleh *browser*. Berdasarkan *URL* atau alamat website dalam jaringan internet, *browser* akan menemukan sebuah alamat dari *web-server*, kemudian mengidentifikasi halaman yang dikehendaki, dan menyampaikan semua informasi yang dibutuhkan oleh *web-server*. Selanjutnya *web-server* akan mencarikan berkas yang diminta dan menampilkan isinya di *browser*. *Browser* yang mendapatkan informasi sesuai kebutuhan kemudian segera menerjemahkan kode *HTML* dan menampilkannya. Pada prinsipnya sama dengan memanggil kode *HTML*, namun pada saat permintaan dikirim ke *web-server*, *web-server* akan memeriksa tipe file yang diminta oleh user. [1]

## JavaScript

*JavaScript (JS)* adalah bahasa pemrograman terkompilasi yang ringan, ditafsirkan, atau tepat waktu dengan fungsi kelas satu. Meskipun paling terkenal sebagai bahasa scripting untuk halaman Web, banyak lingkungan *non-browser* juga menggunakannya, seperti *Node.js, Apache CouchDB* dan *Adobe Acrobat*. *JavaScript* adalah bahasa berbasis *prototipe, multi-paradigma*, utas tunggal, dinamis, mendukung gaya berorientasi objek, imperatif, dan deklaratif (misalnya pemrograman fungsional). [10]

## Vue,Js

*Vue* (diucapkan /vju:/, seperti *view*) adalah kerangka kerja *JavaScript* untuk membangun antarmuka pengguna. Dibangun di atas *HTML* standar, *CSS* dan *JavaScript*, dan menyediakan model pemrograman deklaratif dan berbasis komponen yang membantu Anda mengembangkan antarmuka pengguna secara efisien, baik itu sederhana atau kompleks. [11]

## Web Server

*Web server* adalah *software* yang memberikan layanan berbasis data yang berfungsi menerima request atau permintaan atau *HTTP request* dari *client* dalam bentuk *HTTP* yang berasal dari *web browser* dan akan mengirimkan data yang diminta atau *HTTP respond* dalam bentuk halaman *web* beserta konten-konten yang diinginkan dalam bentuk dokumen *HTML*. Fungsi utama dari *web server* sendiri adalah melakukan *transfer* permintaan atau *request client* melalui aturan atau protokol yang telah ditentukan lalu mengirimkan data yang diminta *client* kembali dalam bentuk halaman *web* serta konten yang di-*request*. [12]

Disini penulis menggunakan *web server Apache*. *Apache* adalah *web server* yang berbasis *open source* dan banyak digunakan karena memiliki pengguna yang sudah banyak sehingga juga memiliki dokumentasi yang lebih lengkap dibandingkan *web server* lainnya. Ciri spesifik dari *Apache* adalah *web server* ini menggunakan arsitektur *keep-alive* atau *forked user* dalam melayani suatu *request*, sehingga lebih memakan konsumsi sumber daya. *Apache* memiliki beberapa fitur seperti kontrol akses, *CGI, PHP, SSI, URL Rewriting* dan lainnya. *Apache* dapat berjalan baik secara *multi* proses maupun *multi thread* arsitektur dimana dapat menjalankan proses yang berjalan secara konkuren. Dalam memproses halaman dinamik, *Apache* merupakan *web server* yang memiliki performa terbaik dibandingkan dengan *web server* lain. [12]

## Database (Basis Data)

*Database* atau basis data adalah kumpulan data yang disimpan secara sistematis didalam computer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data yang dimaksud meliputi cakupan tipe data, struktur, dan juga batasan-batasan data yang akan disimpan. Basis data merupakan aspek yang tidak kalah pentingnya dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat menghindari kesamaan data, relasi antar data yang tidak terstruktur, organisasi data, dan juga rumit dalam merubah data. [1]

## MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau *DBMS* yang banyak rangkaian (*multi-thread*), banyak pengguna (*multi-user*), dengan setidaknya 6 juta pengguna di seluruh dunia. *MySQL AB* menyediakan *MySQL* sebagai perangkat lunak gratis di bawah *GNU General Public License (GPL)*, tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus tertentu dimana penggunaannya tidak cocok dengan penggunaan *GPL*. *MySQL* pada dasarnya merupakan turunan dari salah satu konsep penting database sejak lama, yaitu *SQL (Structured Query Language)*. *SQL* merupakan konsep operasi basis data, terutama untuk menyeleksi dan memasukkan data, yang memudahkan untuk melakukan manipulasi data secara otomatis. [1]

## Unified Modeling Language (UML)

*Unified Modeling Language (UML)* merupakan Bahasa pemodelan yang berbentuk grafis yang digunakan untuk memvisualisasi, menspesifikasikan suatu sistem perangkat lunak. [13]

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. [14]

Berikut adalah jenis-jenis diagram UML beserta contoh diagramnya:

### Use Case Diagram.

*Use case diagram* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* bekerja dengan mendeskripsikan tipikal interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem itu dipakai.

### Class Diagram

*Class diagram* merupakan gambaran struktur sistem dari segi pendefinisian kelas- kelas yang akan dibuat untuk membangun sistem. *Class diagram* terdiri dari atribut dan operasi dengan tujuan pembuat pembuat program dapat membuat hubungan antara dokumentasi perancangan dan perangkat lunak sesuai.

### Sequence Diagram

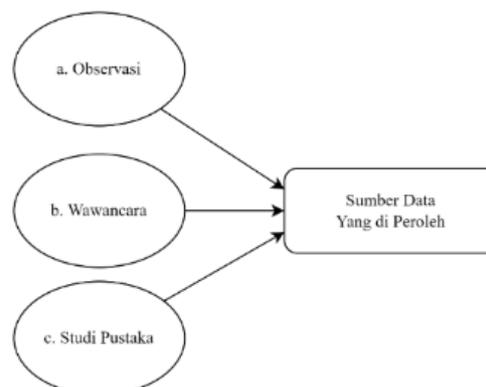
*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Gambaran *sequence diagram* dibuat minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *sequence diagram* sehingga semakin banyak *use case* yang didefinisikan, maka *sequence diagram* yang harus dibuat juga semakin banyak.

### Activity Diagram

*Activity diagram* merupakan diagram yang menggambarkan *workflow* atau aktivitas dari sebuah sistem yang ada pada perangkat lunak.

## METODOLOGI

### Teknik Pengumpulan Data



Gambar 1. Ilustrasi Pengumpulan Data yang digunakan Peneliti dalam Penelitian

Penulis melakukan pengamatan secara langsung sehingga memperoleh data yang akurat dengan tema penelitian yang di tulis:

#### *Observasi*

Penulis melakukan pengamatan dan penelitian secara langsung terhadap alur bisnis agar mendapatkan informasi yang akurat.

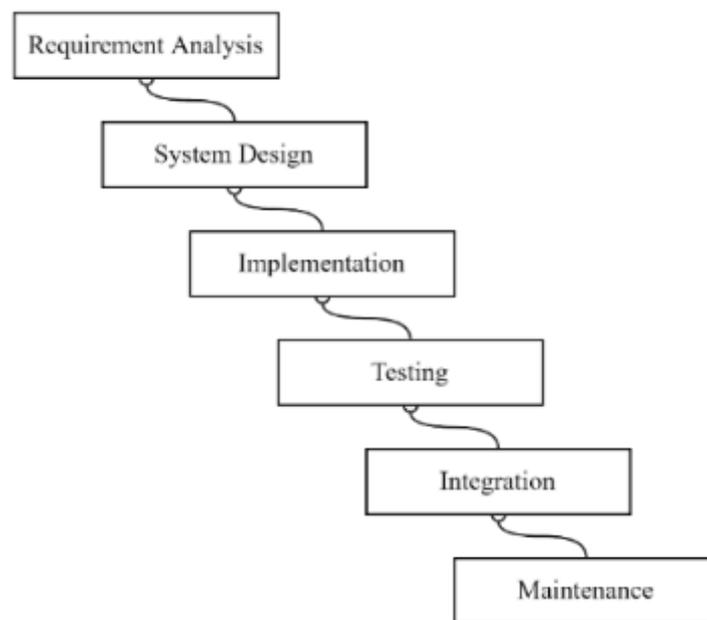
#### *Wawancara*

Penulis melakukan pengumpulan data dengan cara tatap muka dan tanya jawab dengan calon pengguna sistem yang akan dibuat (kasir toko) guna memperoleh informasi kebutuhan sistem yang lebih rinci.

#### *Studi Pustaka*

Mengumpulkan data-data yang diperlukan dengan mencari referensi dari jurnal yang ada di internet atau buku-buku yang berhubungan dengan tema penelitian yang dilakukan saat ini.

#### **Model Pengembangan Sistem**



Gambar 2. Model pengembangan *Waterfall*

Dalam perancangan aplikasi penulis menggunakan model *Waterfall*, hal ini karena model *waterfall* mudah di aplikasikan dan memiliki proses yang teratur untuk aplikasi yang akan dibuat. Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian. Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang runtut: *requirement* (analisis kebutuhan), *design sistem* (*system design*), *Coding & Testing*, Penerapan Program, Pemeliharaan. [1]

#### *Analisa Kebutuhan*

Pada tahap ini, penulis menganalisa apa saja kebutuhan dari perancangan aplikasi *Point of Sales* yang akan dibuat seperti peng-input an data barang, data suplier serta data transaksi penjualan dan pemesanan barang

#### *System Design*

Penulis merancang basis data menggunakan *MySQL*, serta *software* yang dapat digunakan untuk memudahkan mengelola basis data selama masa *development* aplikasi *database client* seperti *PhpMyAdmin* dan *HeidiSql*.

#### *Coding (Implementasi)*

Membuat kode program sebagaimana harusnya aplikasi yang akan di bangun dapat bekerja serta sesuai dengan harapan penggunanya, pada tahap ini penulis menggunakan beberapa bahasa pemrograman yaitu: *PHP*, *JavaScript*, *HTML*, *CSS*.

Penulis juga menggunakan *Framework* dan *Library* untuk membantu perancangan website yaitu *Laravel*, *Vue.JS*, dan *Bootstrap*.

### *Testing*

Pada tahap pengujian aplikasi yang telah dibuat akan dilakukan pengujian kelayakan program dengan menggunakan *blackbox testing*. Tahap ini nantinya di uji oleh pengguna dan *developer*.

### *Penerapan Program (Integration)*

Tahap implementasi yaitu tahap setelah sistem selesai diuji dan siap untuk dipakai oleh objek penelitian sesuai dengan analisa teknologi yang akan dipakai.

### *Pemeliharaan (Maintenance)*

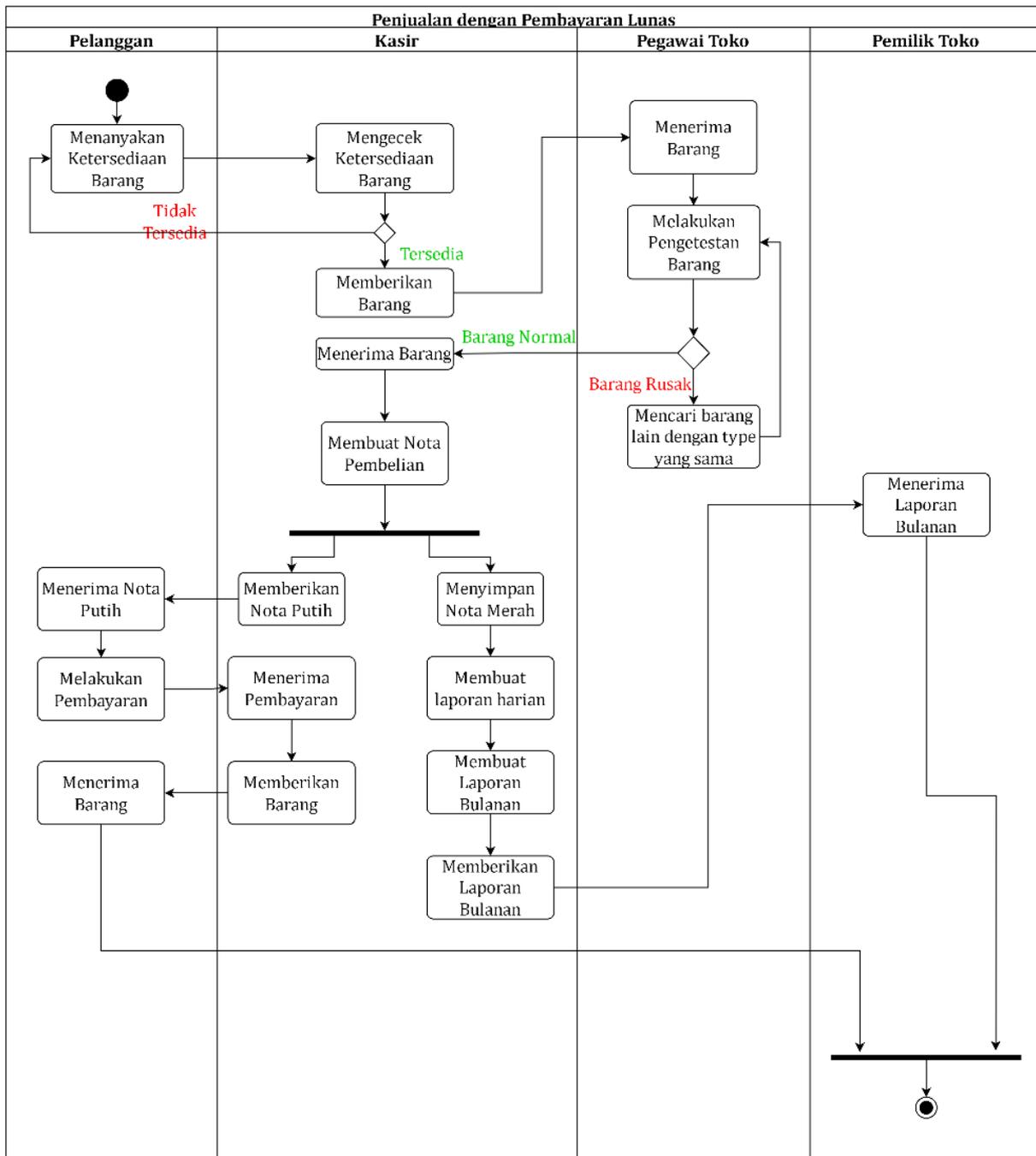
Tidak menutup kemungkinan adanya perubahan yang terjadi pada aplikasi yang mungkin karena adanya kesalahan yang tidak terdeteksi pada saat pengujian.

## **HASIL DAN PEMBAHASAN**

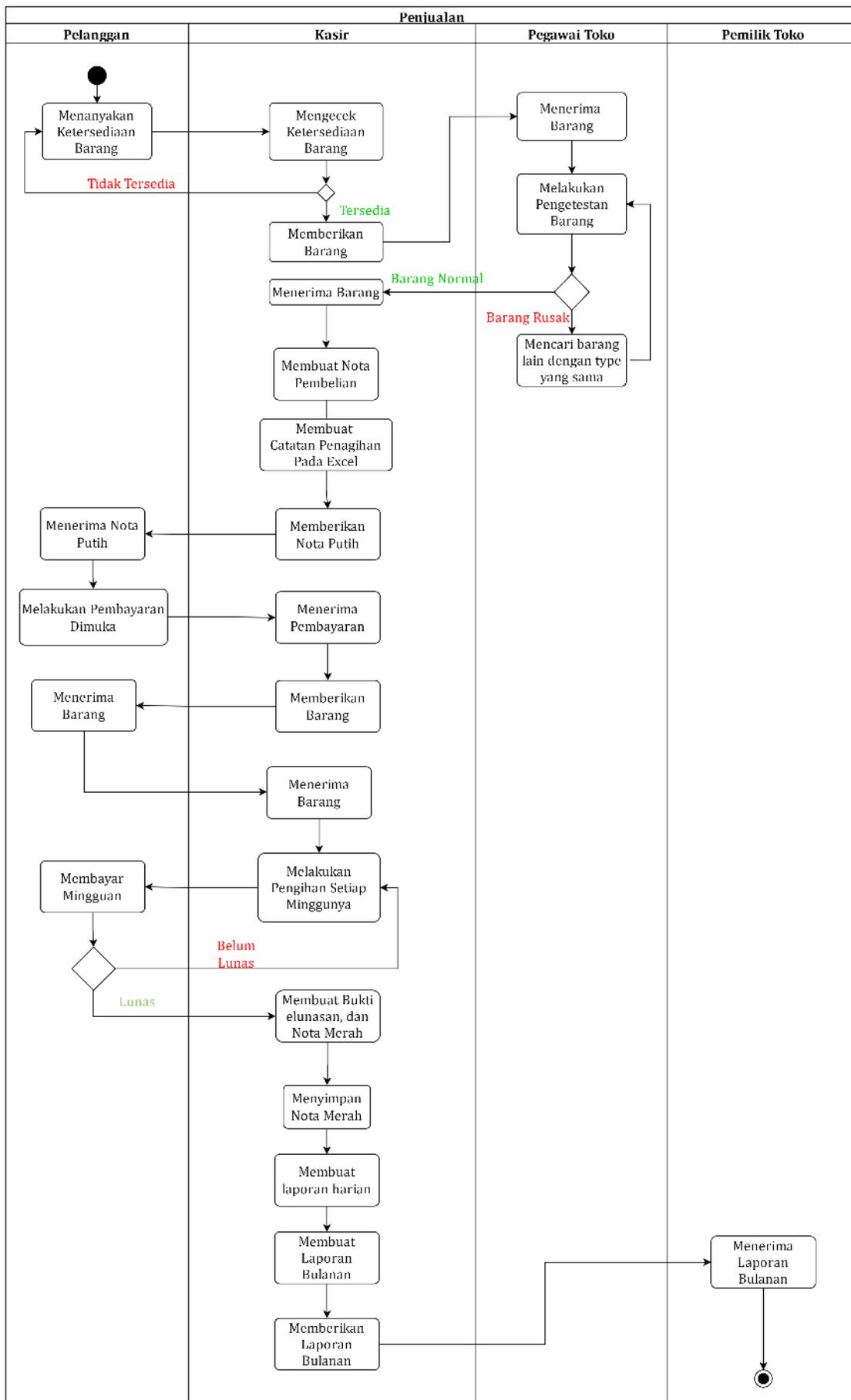
### ***Rancangan Alur dan Proses Sistem***

Toko Arpan *Electric* adalah sebuah usaha yang bergerak di bidang penjualan alat-alat dan perlengkapan elektronik. Dalam proses penjualan dimulai dari pelanggan datang ke bagian kasir untuk menanyakan ketersediaan barang yang ingin di beli, kemudian kasir memberikan penawaran harga yang diinginkan pelanggan. Jika harga telah disetujui maka barang akan dilakukan pengetest-an oleh pegawai toko, jika ada kegagalan fungsi dari alat yang di-test, maka pegawai toko akan mengambilkan barang/alat yang lain dengan tipe yang sama dan akan dilakukan *test* ulang, jika tidak ada kegagalan fungsi maka barang akan di berikan kepada kasir, kasir akan mencatat detail barang yang di beli dinota, nota berwarna putih akan diberikan kepada pembeli, sedangkan nota berwarna merah disimpan untuk pembuatan laporan penjualan harian. Dalam proses pembayaran Toko Arpan *Electric* menyediakan 2 metode pembayaran, yaitu pembayaran lunas dan pembayaran dimuka. Jika pembeli ingin melakukan pembayaran penuh/lunas, setelah pembeli menerima nota berwarna Putih, pembeli selanjutnya melakukan pembayaran sesuai dengan harga yang tertulis pada nota Putih, kemudian kasir akan menerima pembayaran dan memberikan barang yang dibeli kepada pembeli, kemudian kasir membuat laporan penjualan harian sekaligus membuat catatan laporan keuangan bulanan, Jika pembeli ingin melakukan pembayaran dimuka, kasir akan memberikan nota berwarna Merah sebagai bukti pembayaran belum lunas lalu mencatatnya pada aplikasi Excel untuk menyimpan riwayat tagihan pembeli dikemudian harinya, setelah proses pencatatan selesai kasir akan memberikan barang dan note Merah kepada pembeli, dan membayar dengan total harga dimuka sesuai dengan yang ada pada nota Merah yang dibuat oleh kasir, proses ini selanjutnya akan dilakukan penagihan setiap minggunya dengan cara kasir menagih jumlah tagihan mingguan ke tempat pembeli, setelah tagihan lunas catatan tagihan akan di pindahkan kedalam laporan penjualan dan memberikan nota Putih kepada pembeli sebagai bukti lunas.

Pada proses pembelian barang dimulai dari kasir menghubungi *supplier* dan menanyakan ketersediaan barang, jika barang yang di pesan tersedia dan harganya sesuai kemudian kasir memesan barang lalu *supplier* mengirimkan ke toko setelah barang sampai di tempat kasir melakukan pembayaran kemudian menghitung detail pembelian barang lalu mencatatnya ke laporan pembelian barang.



Gambar 2. Activity Diagram Sistem Berjalan Penjualan dengan Pembayaran Lunas pada Toko Arpan Electric

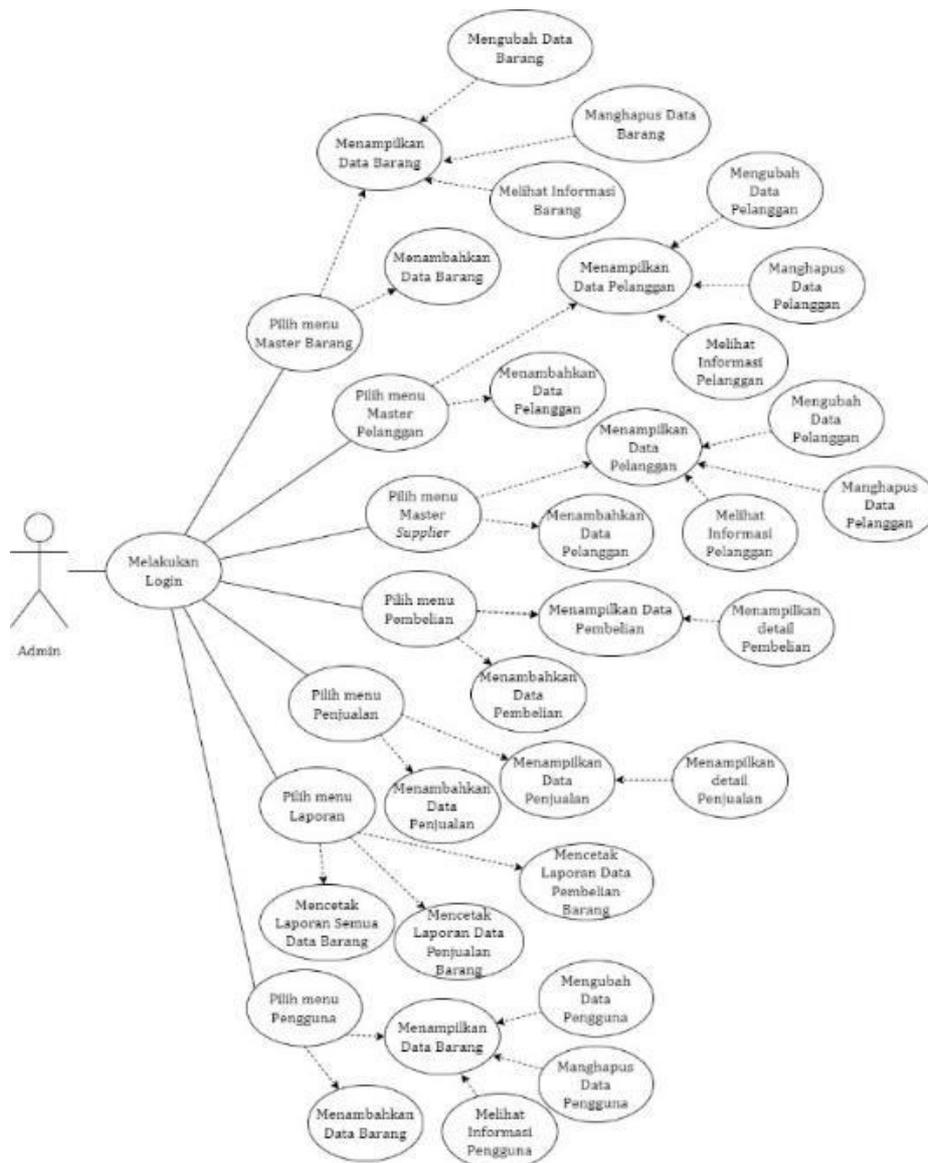


Gambar 4. Activity Diagram Sistem Berjalan Penjualan dengan Pembayaran di Muka pada Toko Arpan Electric

Aplikasi *Point of Sales* yang akan dibuat nantinya akan mampu merekam semua transaksi yang terjadi di Toko Arpan Electric melalui jaringan internal, sehingga tidak dapat di akses oleh publik melalui internet. Berikut adalah spesifikasi kebutuhan (*System Requirement*) dari aplikasi *Point of Sales* yang penulis usulkan:

- A.1 Admin dapat melakukan *Login*
- A.2 Admin dapat mengelola barang keluar (Transaksi Penjualan)
- A.3 Admin dapat mengelola barang masuk (Transaksi Pembelian)
- A.4 Admin dapat mengelola Data Barang
- A.5 Admin dapat mengelola Data Pelanggan
- A.6 Admin dapat mengelola Data Pemasok
- A.7 Admin dapat mencetak Laporan
- A.8 Admin dapat mengelola pengguna sistem

*Use Case Diagram*

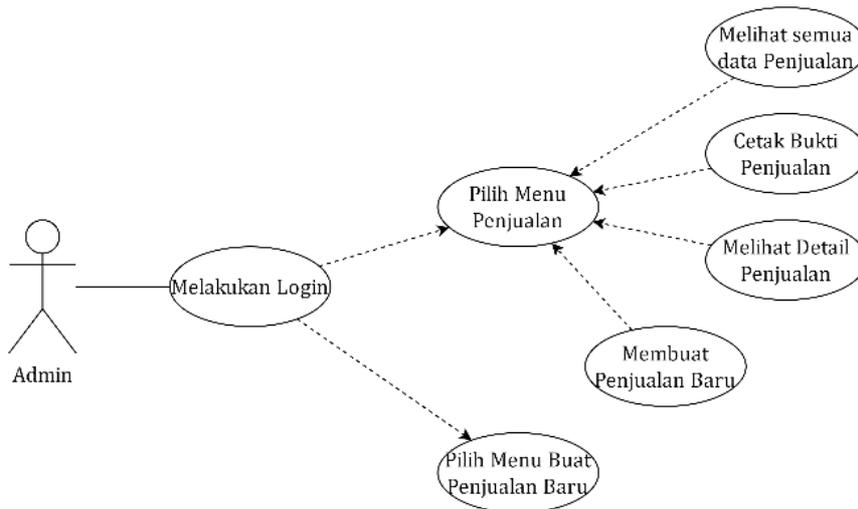


Sumber: Penelitian, 2022

Gambar 5. Use Case Diagram Halaman Admin

Tabel 1. Deskripsi Use Case Diagram Halaman Admin

<i>Use Case Name</i>	Halaman Admin
<i>Requirement</i>	A1 – A8
<i>Description</i>	Use case ini menggambarkan hal-hal yang dapat dilakukan oleh admin di dalam sistem
<i>Pre-condition</i>	Login
<i>Post-condition</i>	<ul style="list-style-type: none"> <li>• Mengelola Data Barang, <i>Customer &amp; Supplier</i></li> <li>• Membuat transaksi pembelian dan penjualan</li> <li>• Dapat mencetak laporan transaksi</li> </ul>
<i>Failed End Condition</i>	Gagal melakukan Login
<i>Condition</i>	
<i>Primary Actor</i>	Admin

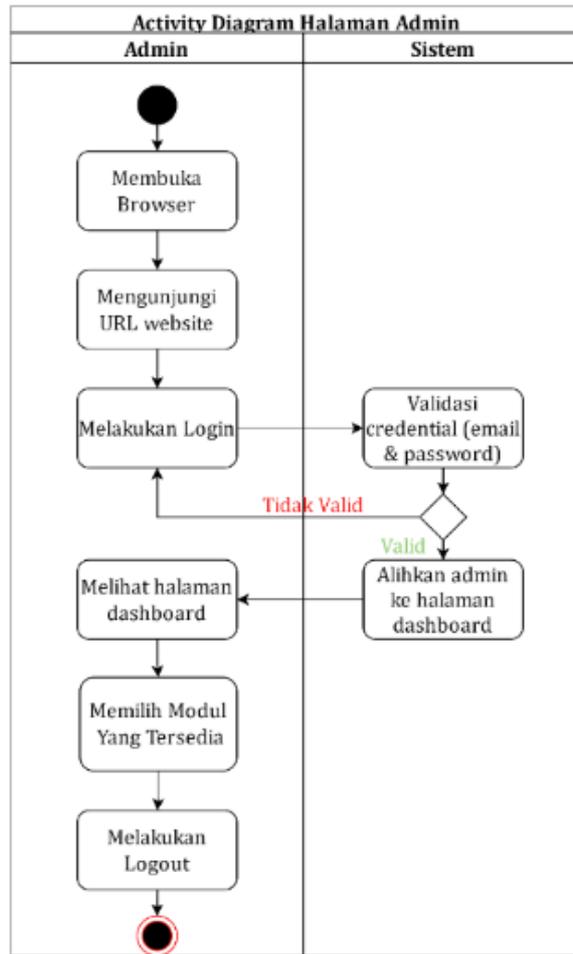


Gambar 6. Use Case Diagram Halaman Transaksi Penjualan

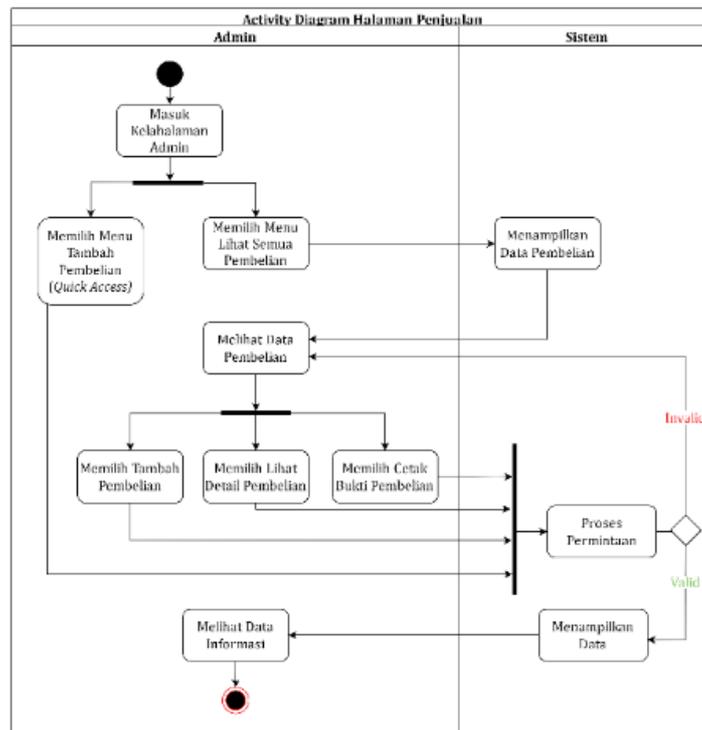
Tabel 2. Deskripsi Use Case Diagram Halaman Transaksi Penjualan

<i>Use Case Name</i>	Transaksi Penjualan
<i>Requirement</i>	A2
<i>Description</i>	Use case ini menggambarkan hal-hal yang dapat dilakukan oleh admin pada modul Penjualan
<i>Pre-condition</i>	Login
<i>Post-condition</i>	<ul style="list-style-type: none"> <li>• Melihat data penjualan</li> <li>• Membuat Transaksi Penjualan Baru</li> <li>• Melihat Detail Penjualan</li> <li>• Cetak Bukti Penjualan</li> </ul>
<i>Failed End Condition</i>	Gagal melakukan Login, Gagal validasi data penjualan baru
<i>Primary Actor</i>	Admin

Activity Diagram



Gambar 7. Activity Diagram Halaman Admin

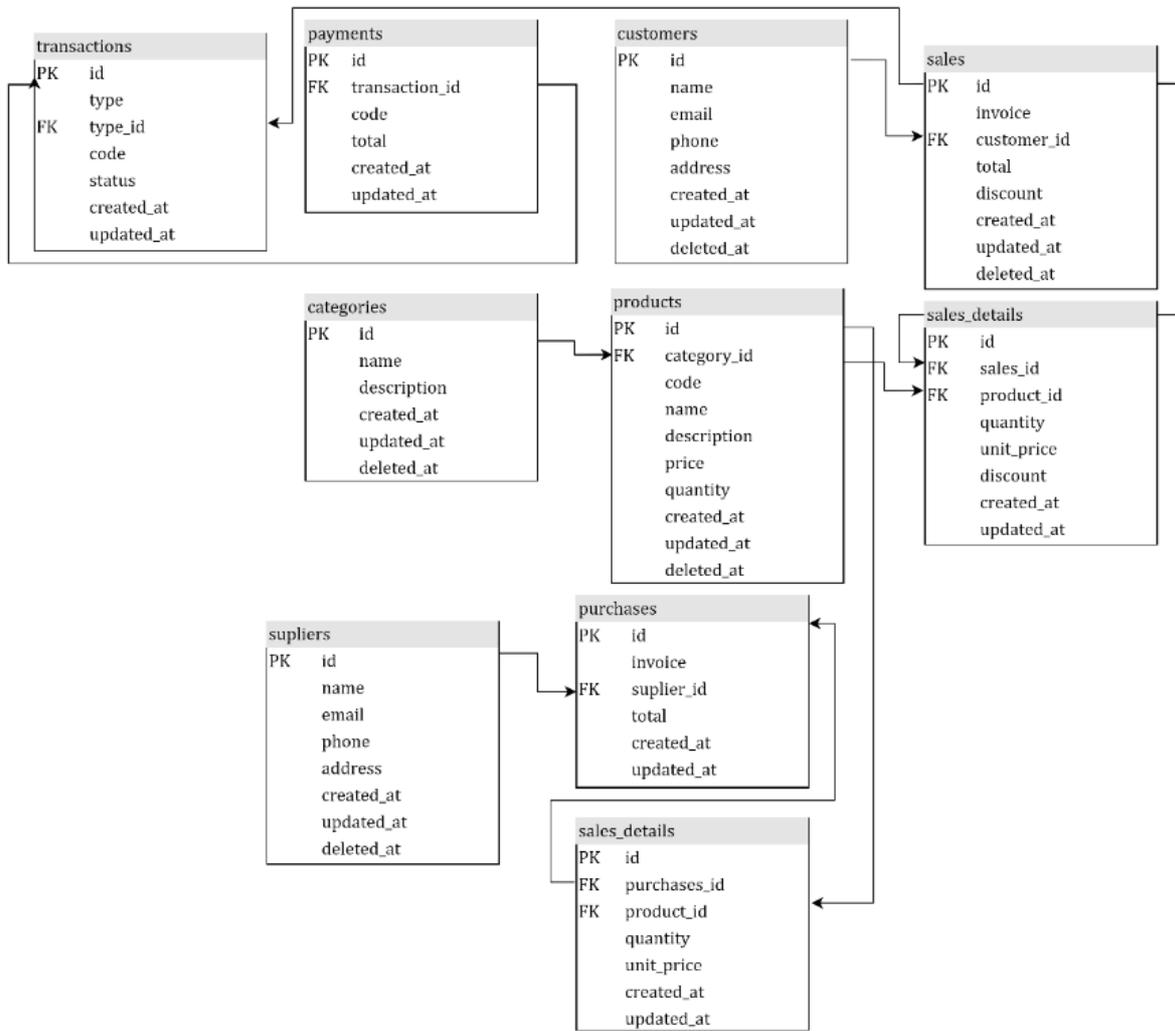


Gambar 8. Activity Diagram Transaksi Penjualan

**Rancangan Database**



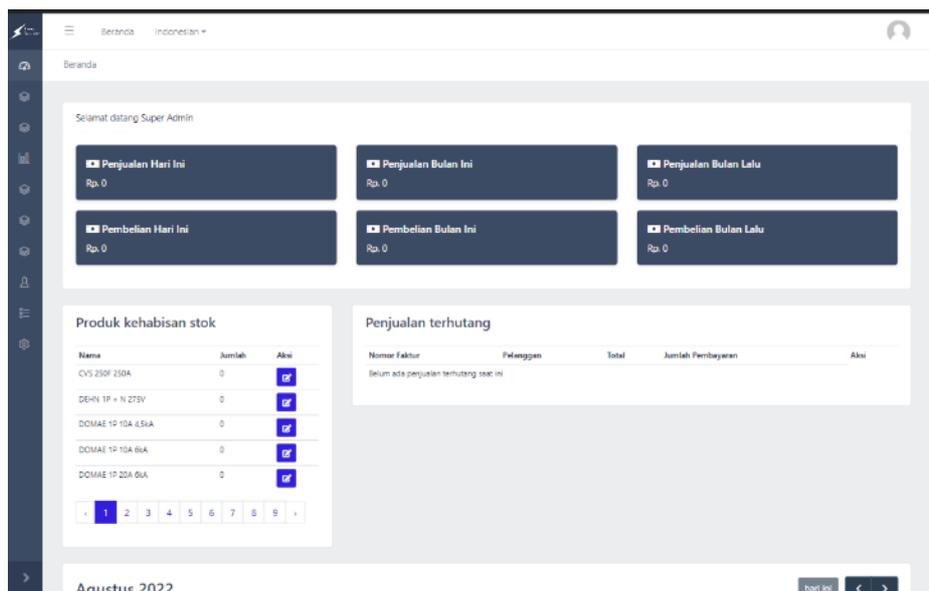
Gambar 9. Entity Relationship Diagram (ERD) Sistem Point of Sale



Gambar 10. Logical Record Structure

**Rancangan Interface**

*Halaman Dashboard*

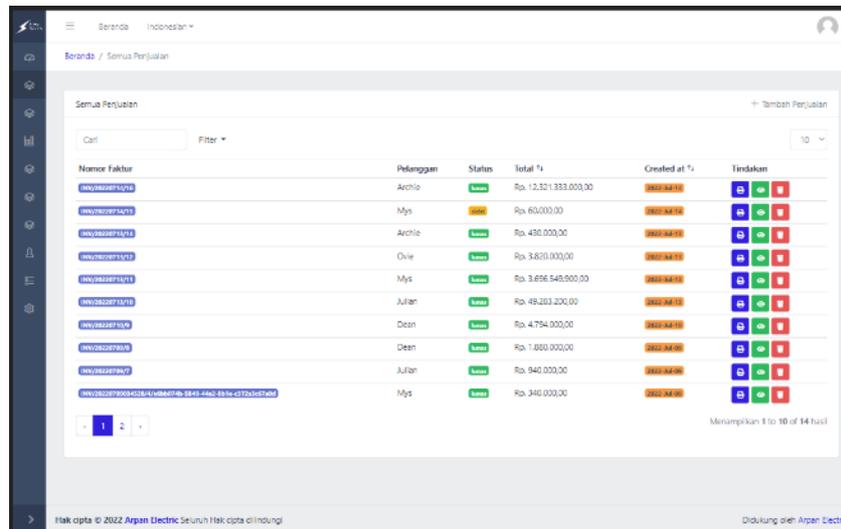


Gambar 11. Halaman Dashboard

Pada halaman *dashboard* diatas, pengguna dapat melihat informasi penjualan bulanan, pembelian bulanan, penjualan terhutang, dan informasi stok barang yang menipis. Selain terdapat informasi, pada halaman ini adalah halaman utama aplikasi yang terdapat menu untuk menuju ke modul modul kebutuhan pengguna seperti Modul penjualan, pembelian, master barang, master pelanggan, master pembelian hingga modul laporan.

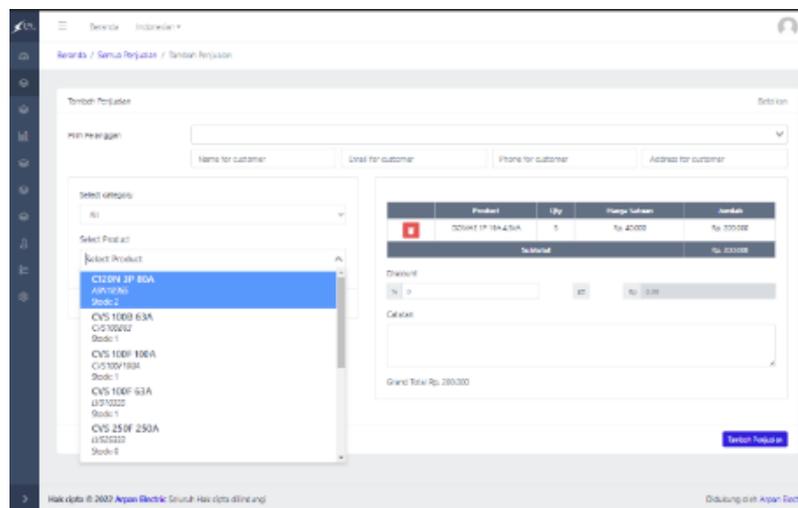
*Halaman Modul Penjualan*

Pada halaman modul penjualan, pengguna dapat melihat daftar penjualan yang terjadi dalam perusahaan/yang telah dimasukkan oleh pengguna, dalam modul ini pengguna juga dapat melihat detail penjualan, menghapus penjualan, serta mencetak *invoice* penjualan apabila dibutuhkan.



Gambar 12. Halaman Modul Penjualan

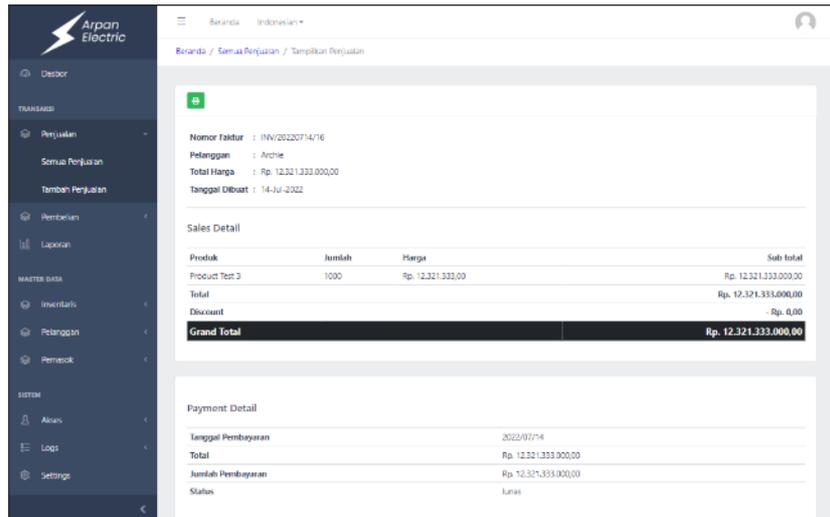
*Halaman Tambah Penjualan*



Gambar 13. Halaman Tambah Penjualan

Halaman tambah penjualan berfungsi sebagai input data penjualan, pengguna dapat memilih barang kemudian memasukkan jumlah barang, mengubah diskon penjualan (dapat memilih diskon dalam unit harga, mauput *percent*). Setelah data penjualan disimpan, data penjualan baru akan disimpan dalam database agar dapat menghasilkan laporan penjualan yang informatif bagi pengguna kemudian stok dari barang yang di pilih akan berkurang sesuai dengan jumlah barang yang di masukkan oleh pengguna sebelumnya.

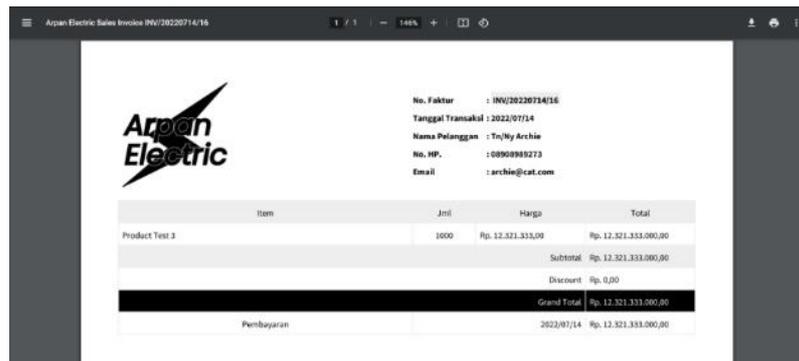
Halaman Detail Penjualan



Gambar 14. Halaman Detail Penjualan

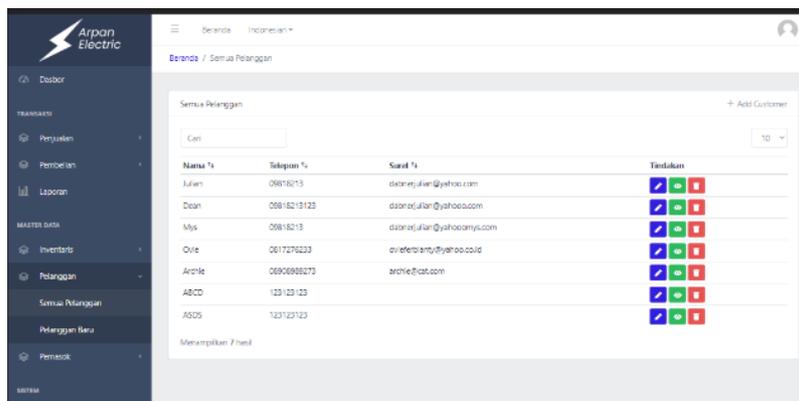
Pada halaman detail penjualan, pengguna dapat melihat detail penjualan yang telah dimasukkan sebelumnya, pada halaman ini terdapat informasi pelanggan/pembeli, detail harga barang dan jumlah barang yang di pesan. Pengguna juga dapat mencetak *invoice* penjualan melalui halaman detail penjualan.

Tampilan Invoice Penjualan



Gambar 15. Tab Invoice Penjualan

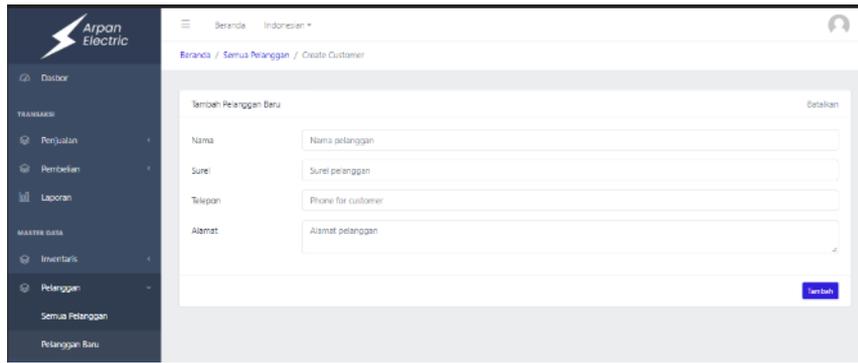
Halaman Master Pelanggan



Gambar 16. Halaman Master Pelanggan

Pada halaman master pelanggan, pengguna dapat melihat data seluruh pelanggan dan penghapusan data pelanggan.

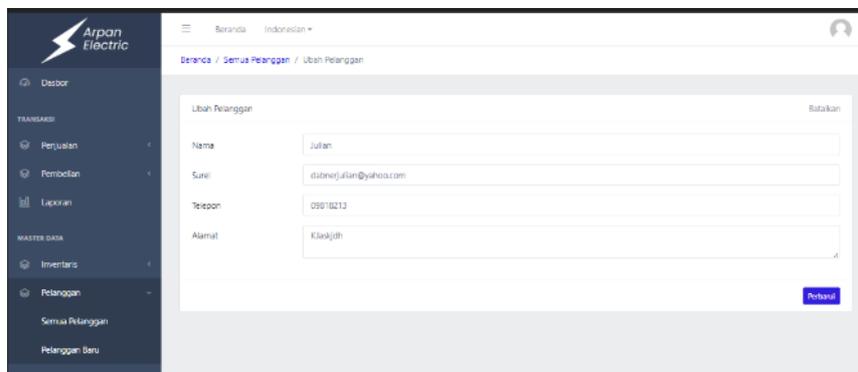
*Halaman Tambah Pelanggan*



Gambar 17. Halaman Tambah Pelanggan

Halaman tambah pelanggan berfungsi sebagai saran untuk pengguna menambahkan pelanggan baru.

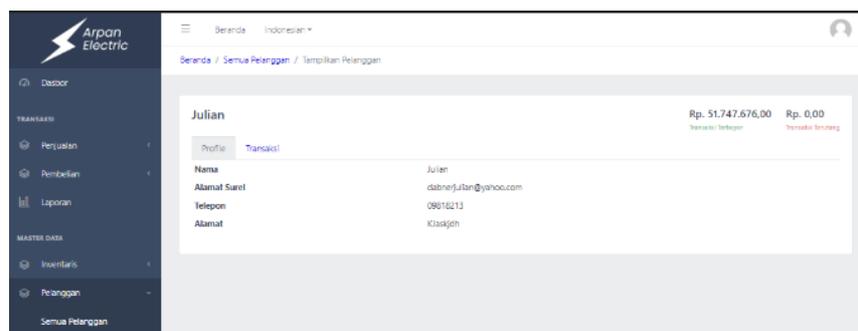
*Halaman Ubah Pelanggan*



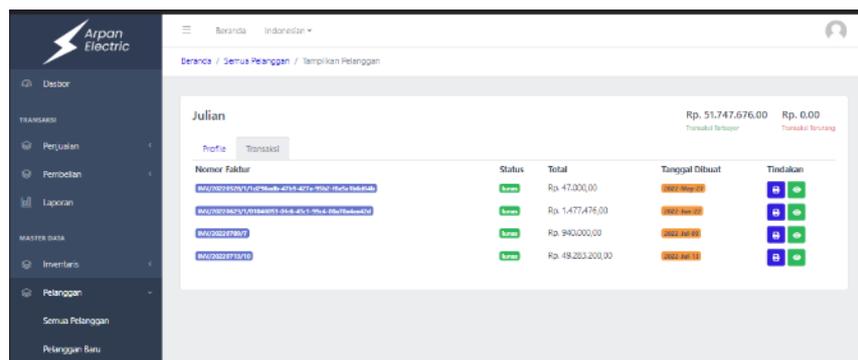
Gambar 18. Halaman Ubah Pelanggan

Pada halaman ini, pengguna dapat mengubah data pelanggan.

*Halaman Detail Pelanggan*



Gambar 19. Halaman Detail *Profile* Pelanggan

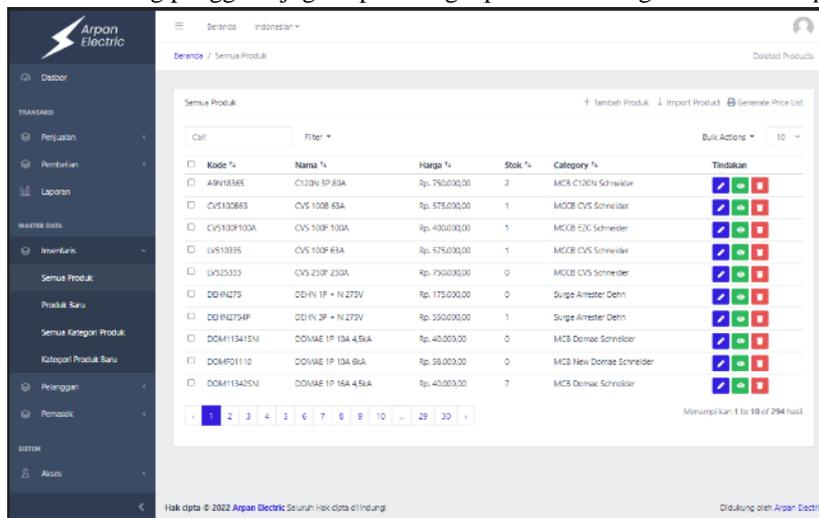


Gambar 20. Halaman Detail Transaksi Pelanggan

Pada halaman detail pelanggan terdapat 2-tab informasi: Detail *Profile* untuk menampilkan informasi umum pelanggan, dan Detail Transaksi untuk menampilkan riwayat transaksi pelanggan

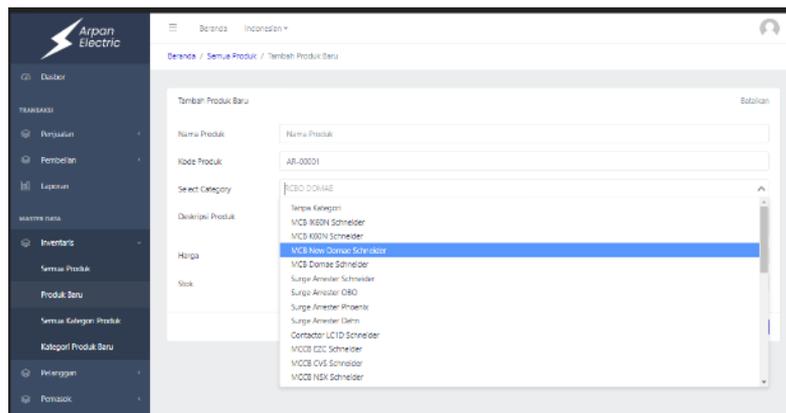
*Halaman Data Barang*

Pada halaman ini, pengguna dapat melihat informasi singkat data barang, seperti stok, kode, harga, nama dan kategori, selain melihat informasi data barang pengguna juga dapat menghapus data barang dan mencetak *pricelist* data barang.



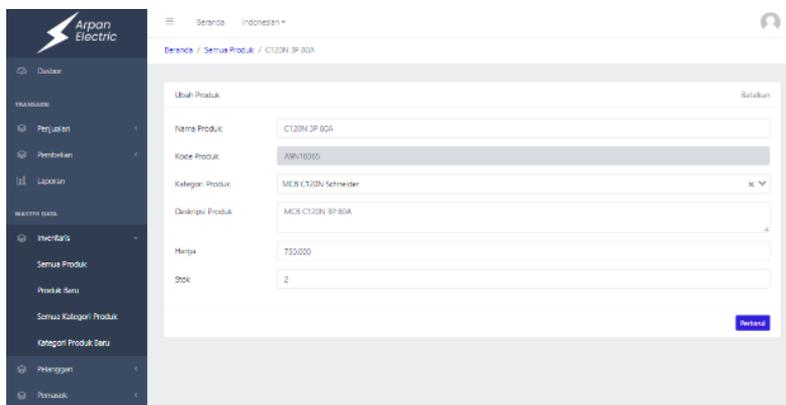
Gambar 21. Data Barang

*Halaman Tambah Barang*



Gambar 22. Halaman Tambah Barang

*Halaman Ubah Barang*



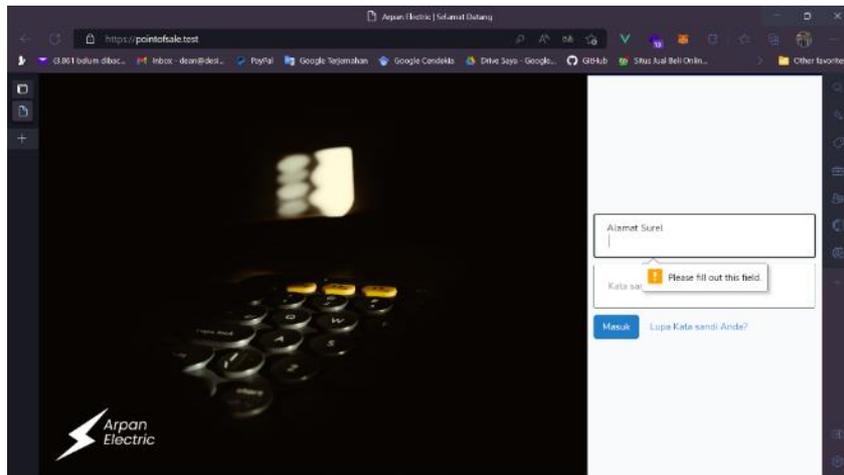
Gambar 23. Halaman Ubah Barang

### Testing

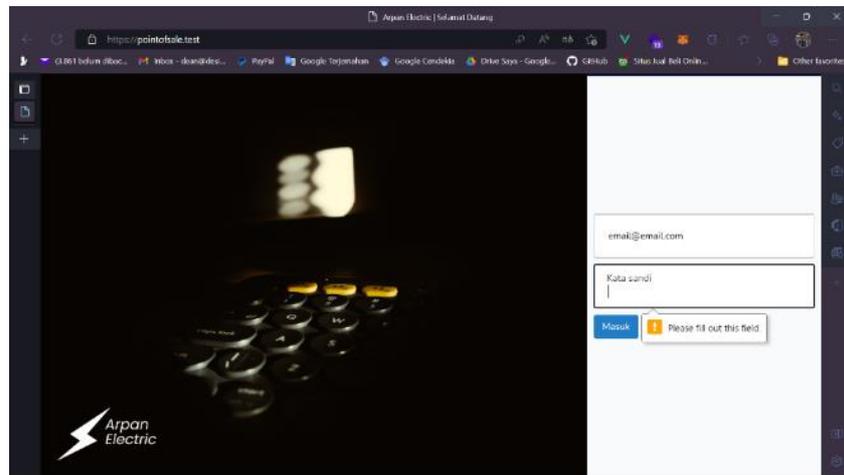
Dalam penelitian ini penulis melakukan *black box testing* pada aplikasi yang sudah dirancang, guna mengetahui kesesuaian data yang dimasukkan oleh pengguna dan keluaran yang harusnya diterima oleh pengguna dari aplikasi yang dirancang. Berikut daftar hasil dari *black box testing* yang sudah di implementasikan:

Table 3. Hasil Pengujian *Black Box Testing Login dan Logout*

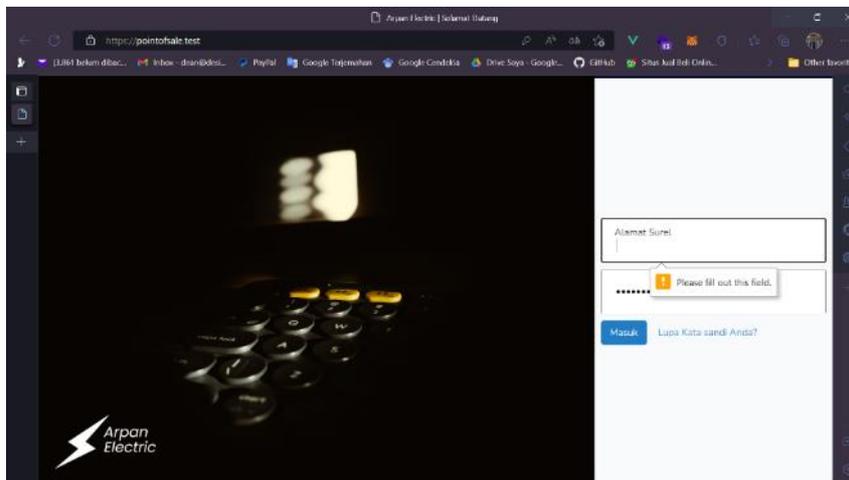
Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Kesimpulan
Mengosongkan field email dan password pada halaman login, lalu submit	Email: NULL Password: NULL	Sistem akan memvalidasi data di berikan kemudian mengembalikan notifikasi <i>Error</i>	Valid
Mengosongkan field password namun mengisi field email, lalu submit	Email: email@email.com Password: NULL	Sistem akan memvalidasi data di berikan kemudian mengembalikan notifikasi <i>Error</i>	Valid
Mengosongkan field email namun mengisi field password, lalu submit	Email: NULL Password: password	Sistem akan memvalidasi data di berikan kemudian mengembalikan notifikasi <i>Error</i>	Valid
Memasukkan field email dengan format bukan email), lalu submit	Email: loremipsum Password: NULL	Sistem akan memvalidasi data di berikan kemudian mengembalikan notifikasi <i>Error</i>	Valid
Memasukkan Password yang salah. lalu submit	Email: email@email.com Password: Invalid password	Sistem akan memvalidasi data di berikan kemudian mengembalikan notifikasi <i>Error</i>	Valid
Memasukkan Email & Password dengan benar, lalu submit	Email: email@email.com Password: Password	Sistem akan memvalidasi data yang diberikan dan mengarahkan admin ke halaman dashboard	Valid
Admin yang sudah login klik logout	Klik Logout di halaman login	Sistem akan memproses permintaan logout dan menghapus session admin	Valid



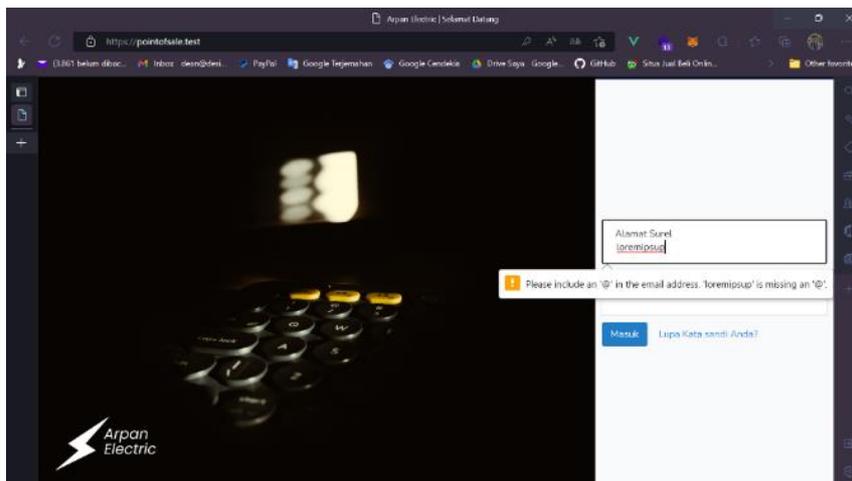
Gambar 24. *Test Case* - Mengosongkan field email dan password pada halaman login.



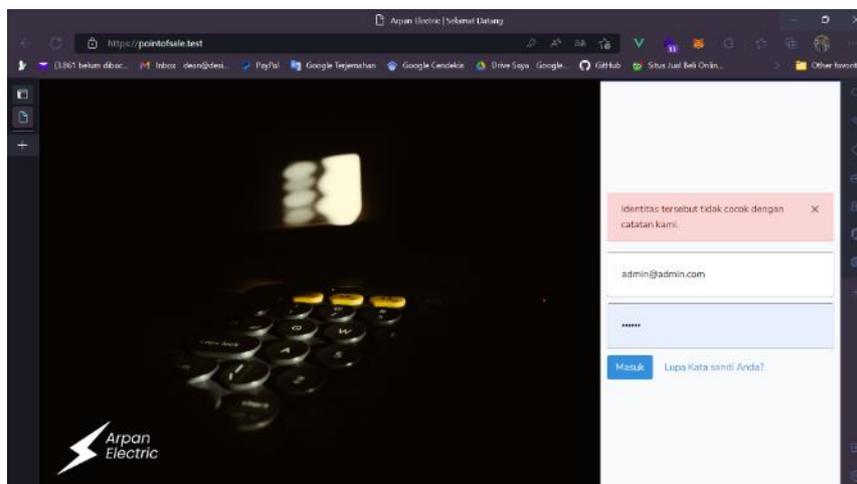
Gambar 25. *Test Case* - Mengosongkan field password namun mengisi field email.



Gambar 26. *Test Case* - Mengosongkan field email namun mengisi field password



Gambar 27. Test Case Memasukkan field email dengan format bukan email)

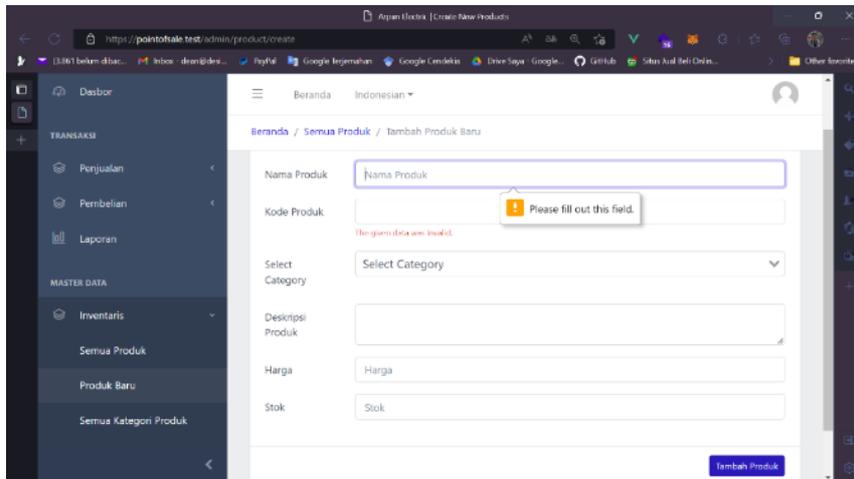


Gambar 28. Test Case - Memasukkan Password yang salah

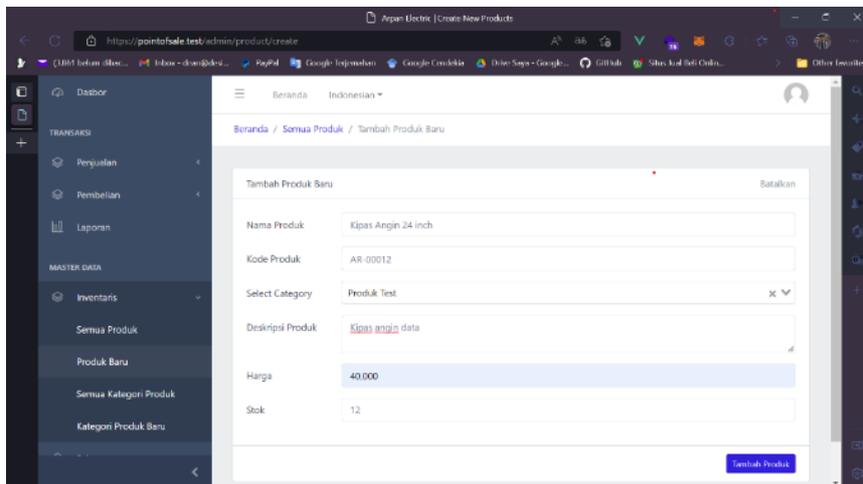
Table IV.20 Hasil Pengujian *Black Box Testing Product Management*

Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Kesimpulan
Pada saat menambahkan produk baru, semua field di kosongkan, lalu submit	Semua field kosong	Sistem akan memvalidasi data di berikan kemudian mengembalikan notifikasi <i>Error</i>	Valid
Mengisi semua field data produk dengan benar, lalu submit	Semua field terisi dengan benar	Sistem akan memvalidasi data yang di berikan kemudian menyimpan data produk ke database	Valid
Pada halaman ubah produk mengosongkan field name dll.	Mengosongkan <i>required field</i>	Sistem akan memvalidasi data di berikan kemudian mengembalikan notifikasi <i>Error</i>	Valid
Pada halaman ubah produk memasukkan semua data dengan benar.	Semua field terisi dengan benar	Sistem akan memvalidasi data di berikan kemudian menyimpan data produk ke database	Valid
Klik hapus produk	Hapus data produk	Sistem akan menyimpan data produk ke trash, agar dapat di	Valid

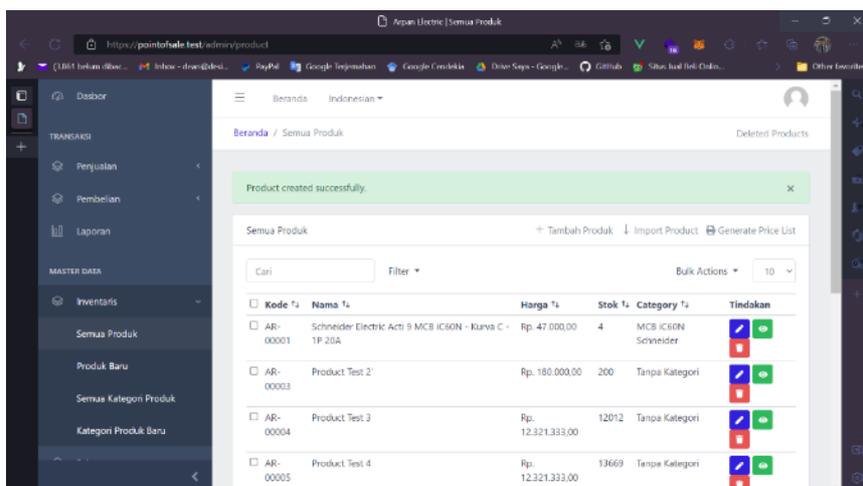
kembalikan lagi jika di  
butuhkan



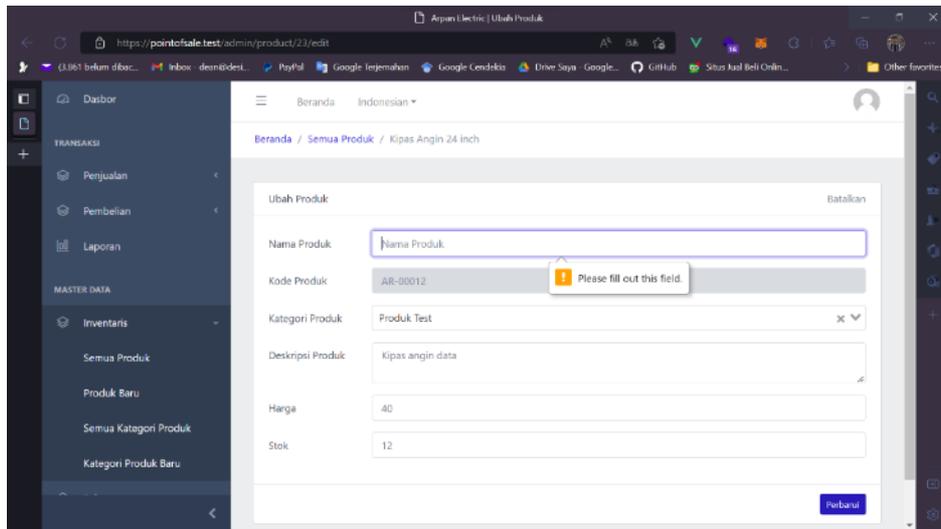
Gambar 29. Test Case - Pada saat menambahkan produk baru, semua field di kosongkan



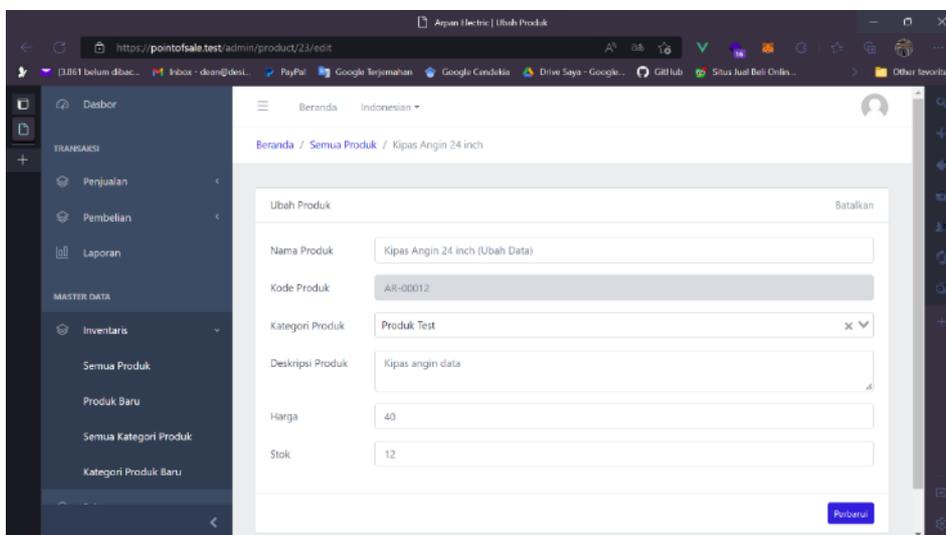
Gambar 30. Test case - Mengisi semua field data produk dengan benar



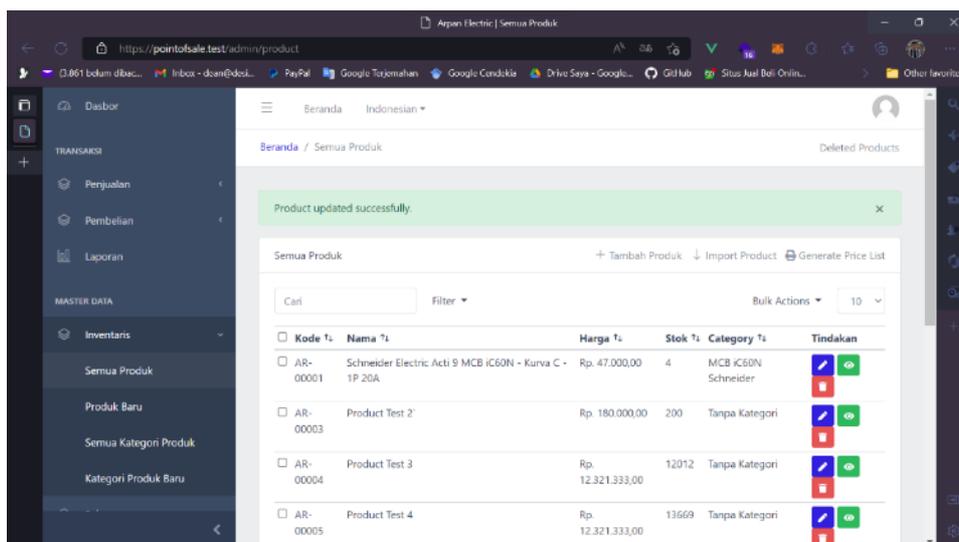
Gambar 31. Test Case – Sistem telah menyimpan data produk baru setelah memvalidasi field data produk



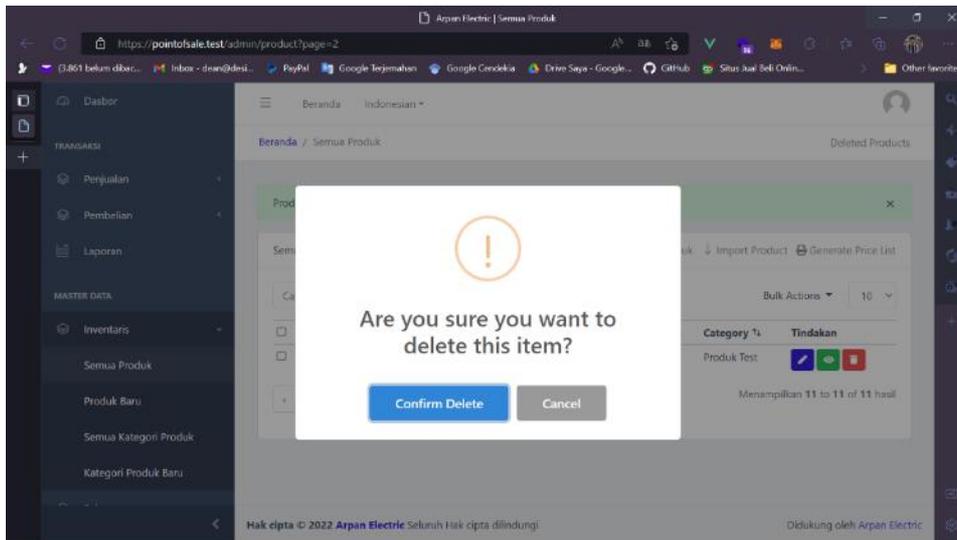
Gambar 32. Test Case - Pada halaman ubah produk mengosongkan field name dll.



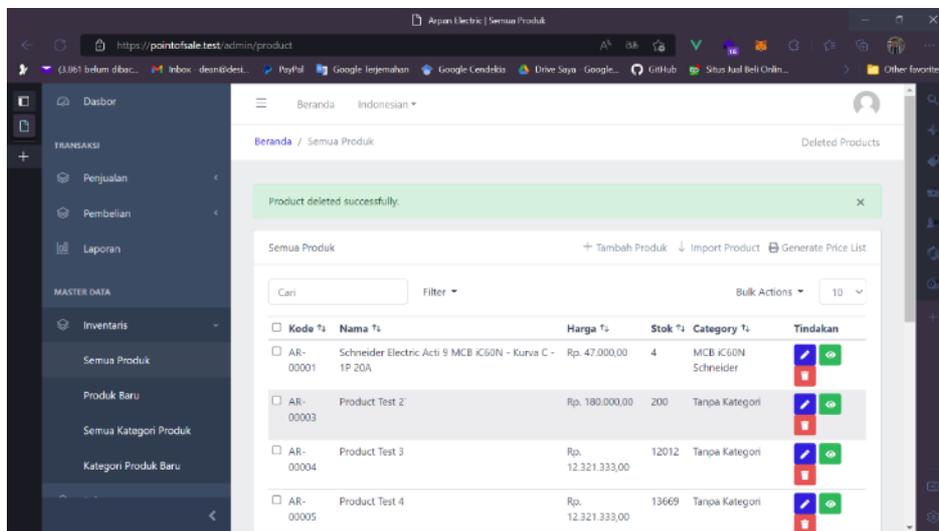
Gambar 33. Test Case - Pada halaman ubah produk memasukkan semua data dengan benar.



Gambar 34. Test Case - Sistem telah menyimpan data produk setelah memvalidasi field data ubah produk



Gambar 35. Test Case - Klik Hapus Produk



Gambar 36. Test Case – Sistem menampilkan alert bahwa produk telah terhapus.

**KESIMPULAN DAN SARAN**

Sistem *Point of Sales* (POS) yang dibangun untuk Toko Arpan *Electric* dapat digunakan untuk membantu Kasir pada Toko Arpan *Electric* melakukan pencatatan transaksi penjualan yang dilakukan. Sistem *Point of Sales* (POS) yang dibangun untuk Toko Arpan *Electric* dapat membantu Kasir untuk mencari dan mengelola data barang. Sistem *Point of Sales* (POS) yang dibangun untuk Toko Arpan *Electric* dapat membantu Kasir dalam menyusun laporan dengan lebih mudah, tepat waktu dan informasi yang akurat.

Modul Pembayaran transaksi yang dibuat masih menggunakan fitur pembayaran penuh dengan cash. Diharapkan ini akan segera dapat di kembangkan lagi jika kebutuhan di Toko Arpan *Electric* ingin menambahkan modul pembayaran dengan metode metode lainnya. Penulis akan terus mengembangkan aplikasi ini agar dapat melengkapi kebutuhan setiap penggunaanya, seperti penambahan Logging setiap transaksi data yang terjadi di dalam sistem.

**DAFTAR PUSTAKA**

[1] C. Trisianto, “PENGUNAAN METODE WATERFALL UNTUK PENGEMBANGAN SISTEM MONITORING DAN EVALUASI PEMBANGUNAN PEDESAAN,” *J. Teknol. Inf. ESIT*, vol. XII, no. 01, 2018, Diakses: 16 April 2022. [Daring]. Tersedia pada: <http://openjournal.unpam.ac.id/index.php/ESIT/article/view/18174/9335>

[2] D. S. K. M. K. Firmansyah dan R. Nugraha, “PENGEMBANGAN SISTEM INFORMASI SUMBER DAYA

- MANUSIA BERBASIS WEB,” vol. 1, no. 1, Mei 2018.
- [3] L. A. Sari *et al.*, “Pengembangan Point Of Sales & Inventory Manajemen Pada Aplikasi e-Apotik Dengan Metode Waterfall ( Studi Kasus Klinik Medika 24 ),” 2016. Diakses: 15 April 2022. [Daring]. Tersedia pada: <https://ejournal-binainsani.ac.id/index.php/ISBI/article/view/391>
- [4] A. Septian Nugraha, ) ; Sapri, ) ; Arius, S. Kurniawansyah, A. S. J. Komitek, dan ) Doi, “Building an Online-Based Point of Sale Application at the Anton Store Membangun Aplikasi Point of Sale Berbasis Online di Toko Anton,” 2021. doi: <https://doi.org/10.53697/jkomitek.v1i2.287>.
- [5] A. Christian, K. Rizal, dan N. Alam, “PERANCANGAN SISTEM INFORMASI JASA CUCI MOBIL DAN MOTOR,” *INTI NUSA MANDIRI*, vol. 14, no. 1, hal. 65–70, 2019, [Daring]. Tersedia pada: [www.bsi.ac.id](http://www.bsi.ac.id)
- [6] B. Rudianto, Y. E. Achyani, I. Ariyati, dan U. N. Mandiri, “Rancang Bangun Sistem Informasi Persediaan Obat Berbasis Web Menggunakan Model RAD,” *J. Tek. Komput. AMIK BSI*, vol. 7, no. 2, 2021, doi: 10.31294/jtk.v4i2.
- [7] R. Ahmad dan N. Hasti, “SISTEM INFORMASI PENJUALAN SANDAL BERBASIS WEB,” *J. Teknol. dan Inf.*, vol. 8, no. 1, Mar 2018, doi: 10.34010/jati.v8i1.911.
- [8] S. S. . M. D. Abdurahman Sidik, *Teori, Strategi, dan Evaluasi Merancang Website dalam Perspektif Desain*. Banjarmasin: Universitas Islam Kalimantan Muhammad Arsyad Al banjari, 2019. Diakses: 16 April 2022. [Daring]. Tersedia pada: <http://repository.uniska-bjm.ac.id/id/eprint/504>
- [9] “PHP: What is PHP? - Manual.” <https://www.php.net/manual/en/intro-what-is.php> (diakses 23 Mei 2022).
- [10] “JavaScript | MDN.” <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (diakses 23 Mei 2022).
- [11] “Introduction | Vue.js.” <https://vuejs.org/guide/introduction.html#what-is-vue> (diakses 23 Mei 2022).
- [12] S. Dwiyatno, E. Rakhmat, dan O. Gustiawan, “IMPLEMENTASI VIRTUALISASI SERVER BERBASIS DOCKER CONTAINER,” vol. 7, no. 2, 2020.
- [13] Y. Heriyanto, “PERANCANGAN SISTEM INFORMASI RENTAL MOBIL BERBASIS WEB PADA PT.APM RENT CAR,” *J. Intra-Tech*, vol. 2, no. 2, hal. 64–77, 2018.
- [14] D. Wira, T. Putra, dan R. Andriani, “Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD,” vol. 7, no. 1, 2019.