

LAPORAN HASIL PENELITIAN

Pembuatan Aplikasi *Point Of Sale* (POS) Berbasis Web Untuk Sistem Pembayaran di Waroeng Nongkrong Keboen Coffee



Pelaksana Penelitian :

Risdiandri Iskandar, SKom., MM (0325047802)
Lyvia Tedja (53417391)

FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS GUNA DARMA
JAKARTA

2021

HALAMAN PENGESAHAN

Judul Penelitian : Pembuatan Aplikasi Point Of Sale (POS)
Berbasis Web Untuk Sistem Pembayaran di
Waroeng Nongkrong Keboen Coffee

Ketua Peneliti

Nama : Risdandri Iskandar, SKom., MM

NIDN : 0325047802

Anggota 1

Nama : Lyvia Tedja

NIDN / NPM : 53417391

Anggota 2

Nama :

NIDN / NPM :

Program Studi : Teknik Informatika

Fakultas : Teknologi Industri

Skema Penelitian (Dasar / Terapan) : Terapan

Tahun Pelaksanaan : 2021

Lama Kegiatan (bulan) : 6 bulan

Biaya Penelitian : Rp. 4.000.000,-

Sumber Biaya (Institusi/Mandiri) : Internal Perguruan Tinggi

Jakarta, 20 Desember 2021

Mengetahui,
Ketua Lembaga Penelitian

Dr. Bertalya, SKom., DEA
NIP : 920312

Ketua Peneliti


Risdandri Iskandar, SKom., MM
NIP : 010433

Abstrak

Teknologi Informasi tentunya tidak asing lagi bagi masyarakat umum khususnya pemilik usaha yang sudah memanfaatkan teknologi informasi untuk membantu kelangsungan bisnis mereka. Dalam dunia bisnis, mesin kasir atau Point of Sale (POS) sudah menjadi bagian penting dari operasional berbagai usaha. Namun, seiring dengan berkembangnya teknologi, cara konvensional dalam pencatatan transaksi toko dianggap sudah tidak relevan. Selain kurang efisien, membutuhkan banyak biaya seperti kertas dan pulpen. Untuk meningkatkan penjualan toko, Point of Sale (POS) digunakan untuk mempermudah proses transaksi penjualan. Waroeng Nongkrong Keboen Coffee merupakan bisnis kuliner yang berlokasi di Kabupaten Bogor. Salah satu kendala yang sering dihadapi oleh pemilik usaha Waroeng Nongkrong Keboen Coffee adalah terlalu banyaknya waktu yang dihabiskan untuk menghitung total pemasukan harian karena pencatatan transaksi masih dilakukan secara manual. Untuk mengatasi masalah tersebut, maka penulis ingin merancang dan membangun sebuah aplikasi Point of Sale (POS) berbasis web dengan menggunakan bahasa pemrograman PHP dengan bantuan framework Laravel. Proses pembuatan aplikasi ini dimulai dari analisis kebutuhan data, perancangan aplikasi, perancangan antarmuka aplikasi, implementasi dan uji coba. Berdasarkan pengujian yang telah dilakukan aplikasi ini dapat mempermudah dalam melakukan proses transaksi, pencatatan transaksi, dan menghitung total pemasukan harian atau bulanan berdasarkan periode tertentu. Selain itu, Aplikasi web ini dapat diakses melalui situs <https://wnkeboencoffee.com>.

Kata Kunci: *Point of Sale, PHP, Laravel*

Kata Pengantar

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan bimbingan, petunjuk serta hidayah-Nya, hingga penelitian ini dapat terselesaikan dengan baik. Bagaimanapun juga terselesaikannya penelitian ini berkat dukungan dan kerjasama yang baik diantara berbagai pihak, untuk itu pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ibu Prof. Dr. E.S. Margianti, SE, MM selaku Rektor Universitas Gunadarma
2. Bapak Prof. Suryadi Harmanto, SSi, MMSI selaku Pembantu Rektor II Universitas Gunadarma
3. Bapak Prof. Dr.-Ing Adang Suhendra, SSi., SKom., MSc selaku Dekan Fakultas Teknologi Industri Universitas Gunadarma
4. Ibu Dr. Lintang Yuniar Banowosari, SKom., MSc selaku Ketua Program Studi Teknik Informatika Universitas Gunadarma
5. Ibu Dr. Bertalya, SKom., DEA selaku Ketua Lembaga Penelitian Universitas Gunadarma
6. Rekan-rekan dan semua pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu hingga terselesaikannya penelitian ini.

Penelitian ini masih jauh dari kesempurnaan, untuk itu penulis mengharapkan kritik dan saran dari pembaca yang bersifat membangun guna kesempurnaannya. Akhir kata semoga penelitian ini dapat memberikan manfaat bagi para pembaca, khususnya bagi penulis untuk referensi penelitian berikutnya.

Jakarta, Desember 2021

Tim Penulis

DAFTAR ISI

Cover	Error!
Bookmark not defined.	
Halaman Pengesahan.....	ii
Abstrak	iii
Kata Pengantar	iError!
Bookmark not defined.	
Daftar Isi.....	v
Daftar Gambar	vi
Daftar Tabel.....	vii
BAB I Pendahuluan	Error!
Bookmark not defined.	
BAB II Landasan Teori	2
BAB III Metode Penelitian.....	14
BAB IV Hasil Penelitian	16
BAB V Kesimpulan dan Saran	23
Daftar Pustaka.....	24

DAFTAR GAMBAR

Gambar 2.1 Struktur Navigasi Linier	9
Gambar 2.2 Struktur Navigasi Hierarki	9
Gambar 2.3 Struktur Navigasi Non Linier	9
Gambar 2.4 Struktur Navigasi Campuran	10
Gambar 4.1 Struktur Navigasi Admin	16
Gambar 4.2 Struktur Navigasi Karyawan	17
Gambar 4.3 <i>Use Case Diagram</i> Aplikasi <i>Point of Sale</i> (POS)	18
Gambar 4.4 <i>Activity diagram Login</i>	19
Gambar 4.5 <i>Activity Diagram</i> Melihat Halaman <i>Dashboard</i>	19
Gambar 4.6 <i>Activity Diagram</i> Mengelola Data Produk	19
Gambar 4.7 <i>Activity Diagram</i> Mengelola Data <i>Supplier</i>	20
Gambar 4.8 <i>Activity Diagram</i> Melakukan Proses Transaksi.....	20
Gambar 4.9 <i>Activity Diagram</i> Melihat Data Laporan Harian	20
Gambar 4.10 <i>Activity Diagram</i> Melihat Data Laporan Bulanan	20
Gambar 4.11 <i>Activity Diagram</i> Mengelola Data Karyawan	21
Gambar 4.12 <i>Activity Diagram</i> <i>Logout</i>	21

DAFTAR TABEL

Tabel 2.1 Fitur Laravel.....	5
Tabel 2.2 MVC	7
Tabel 2.3 <i>Use Case Diagram</i>	11
Tabel 2.4 <i>Activity Diagram</i>	13
Tabel 3.1 Hasil Pengujian Fungsionalitas	23

BAB I

PENDAHULUAN

Sekarang ini teknologi informasi sudah menjadi istilah yang tidak asing lagi khususnya bagi para pemilik usaha yang telah memanfaatkan teknologi informasi dalam membantu kelangsungan bisnis mereka. Namun, masih banyak pemilik usaha yang belum mengetahui kegunaan teknologi informasi untuk bisnis mereka.

Dalam dunia bisnis, mesin kasir atau *Point of Sale* (POS) sudah menjadi bagian penting dari operasional berbagai usaha. Namun, seiring dengan berkembangnya teknologi, cara konvensional dalam pencatatan transaksi toko dianggap sudah tidak relevan. Selain kurang efisien, membutuhkan banyak biaya seperti kertas dan pulpen. Dalam meningkatkan penjualan toko, *Point of Sale* (POS) atau aplikasi kasir digunakan untuk mempermudah proses transaksi penjualan. Tidak hanya itu, laporan transaksi penjualan juga dapat diketahui berdasarkan periode tertentu serta mengubah harga secara cepat dan mudah.

Waroeng Nongkrong Keboen Coffee merupakan bisnis kuliner yang berada di Kabupaten Bogor. Salah satu kendala yang sering dihadapi oleh pemilik usaha Waroeng Nongkrong Keboen Coffee adalah terlalu banyaknya waktu yang dihabiskan untuk menghitung total pemasukan harian atau bulanan karena pencatatan transaksi masih dilakukan secara manual. Karena masih dilakukan secara manual pula sering terjadi kesalahan saat mencatat transaksi.

Dengan adanya permasalahan tersebut, maka penulis ingin merancang dan membangun aplikasi *Point of Sale* (POS) berbasis web untuk mempermudah proses dan pencatatan transaksi pada Waroeng Nongkrong Keboen Coffee, serta menghitung total pemasukan harian atau bulanan berdasarkan periode tertentu.

BAB II

LANDASAN TEORI

Waroeng Nongkrong Keboen Coffee

Waroeng Nongkrong Keboen Coffee merupakan bisnis kuliner yang berlokasi di Citeureup, Kabupaten Bogor. Waroeng Nongkrong Keboen Coffee didirikan oleh Steven Tanuwijaya pada tahun 2021. Menu yang disajikan oleh Waroeng Nongkrong Keboen Coffee berupa makanan, cemilan dan minuman.

Point of Sale (POS)

Dalam dunia bisnis, *Point of Sale (POS)* dapat diartikan sebagai tempat kasir (*check-out counter*) dengan mesin kasir (*cash register*). Sesuai dengan namanya, *Point of Sale* merupakan titik penjualan (*check-out*) di mana transaksi dapat dikatakan selesai. Ini adalah di mana pembeli dan penjual melakukan pembayaran atas barang/jasa yang sudah diterima. Pada POS penjual akan menghitung seluruh jumlah harga yang dibeli konsumen dan memberikan pilihan bagi pembeli untuk melakukan pembayaran serta mengeluarkan tanda terima transaksi pembelian yang biasa disebut dengan Struk. (Moka, 2020)

PHP

Hypertext Preprocessor atau lebih akrab dengan sapaan PHP merupakan bahasa pemrograman *script server-side* yang didesain untuk pengembangan web. PHP disebut bahasa pemrograman *server-side* karena diproses pada komputer *server*. (M Jannah, Sarwandi dan C Creative, 2019)

Alasan mengapa memilih PHP: (D Krisbiantoro dan PD Abda'u, 2021)

- a. *Life cycle* yang sangat singkat, sehingga PHP selalu *up to date* mengikuti perkembangan teknologi internet.
- b. *Cross platform*, yakni PHP dapat dipakai di hampir semua *web server* yang ada di pasaran (terutama Apache dan Microsoft IIS) dan dijalankan pada berbagai sistem operasi (Linux, Windows, FreeBSD).
- c. PHP mendukung koneksi ke banyak *database* baik yang gratis maupun komersil, seperti MySQL, Oracle, Microsoft SQL Server, InterBase, dan banyak lagi.
- d. PHP bersifat *open source* dan gratis.

- e. Kemudahan dalam mendapatkan dokumentasi. Di Internet, tidak sulit untuk mencari baik itu referensi, kode-kode PHP yang sudah jadi dan juga mengajukan pertanyaan pada grup-grup diskusi yang di dalamnya banyak sekali para master PHP.

Laravel

Laravel yang dikembangkan Taylor Otwell adalah salah satu *framework* PHP terbaik. *Platform* ini luar biasa karena dikembangkan secara bersama-sama dalam komunitas. Para *developer* sepakat, jika Anda ingin membuat aplikasi dengan cepat maka Laravel adalah pilihan yang tepat. Walau Laravel tidak seperti kerangka kerja CMS atau *Ecommerce* yang akan memuaskan kebutuhan spesifik Anda, tetapi Laravel ini serba bisa. Berikut adalah alasan mengapa kita harus memilih Laravel di antara sekian banyak *framework* lainnya: (Y Yudhanto dan HA Prasetyo, 2019)

1. Simpel atau Sederhana

Menulis di Laravel ibaratnya seperti menulis puisi. Kode yang biasanya ditulis panjang dan rumit dapat diringkas begitu saja tanpa menghilangkan fungsi aslinya. Contohnya seperti ini

```
1. $koneksi = mysqli_connect("localhost","user","pas  
s","database");  
2. $query = "SELECT * FROM mahasiswa";  
3. $tampil = mysqli_query($koneksi, $query);  
4. $data = mysqli_fetch_array($tampil);  
5. {$data['Nama Mahasiswa']}
```

Bisa diringkas dengan:

```
1. DB::table('mahasiswa')->all();
```

2. Kualitas

Banyak sekali kualitas yang diberikan Laravel di antaranya adalah adanya hal-hal berikut ini:

Tabel 2.1 Fitur Laravel
(Sumber: Y Yudhanto dan HA Prasetyo, 2019)

1.	<i>Artisan</i>	<p>Menjalankan perintah-perintah PHP. Di dalam Laravel sudah terdapat fitur PHP sehingga tidak perlu <i>web server</i> seperti Apache untuk menjalankannya.</p> <p>Artisan membantu mengelola <i>website</i> kita seperti membuat <i>table database</i>, atau melihat <i>route website</i> kita, dan masih banyak lagi. Untuk melihat fitur lainnya yang tersedia, tinggal ketikkan perintah dibawah ini dalam <i>command line</i>:</p> <pre>php artisan list</pre>
2.	<i>Authentication</i>	<p><i>Authentication</i> atau autentifikasi merupakan kewajiban bagi sebuah <i>website</i> yang memiliki <i>user</i>, proses pengecekan apakah <i>user</i> tersebut valid atau tidak.</p> <p>Fungsi autentifikasi akan otomatis muncul di dalam <i>website</i> beserta halaman <i>login</i> dan <i>register</i>. Cukup menuliskan kode di bawah dalam <i>command line</i>:</p> <pre>php artisan make:auth</pre>
3.	<i>Controller</i>	<p>Setiap <i>framework</i> biasanya dilengkapi dengan <i>controller</i>. Nah, di Laravel berbeda hanya pada cara membuatnya. Dengan menuliskan satu baris kode, seluruh fungsi CRUD (<i>Create</i>, <i>Read</i>, <i>Update</i>, <i>Delete</i>) akan muncul secara otomatis dan disertai penggunaan yang sangat mudah dalam <i>route</i>-nya. Kodenya sebagai berikut:</p> <pre>php artisan make:controller nama-controller</pre>

3. Keamanan

Laravel sendiri mengklaim bahwa keamanannya sudah terjamin. Berbagai keamanan Laravel tentunya sudah otomatis, hanya saja kesalahan dari *developer* saat pembuatan tentu bisa membuat celah yang berbahaya.

4. Minimalisir Perulangan

Contohnya, *blade* sebagai sebuah *template engine*. Kita bisa memotong-motong *template* dari *website* sesuai keinginan kita dengan *blade*. Perulangan kode-kode seperti dalam *header* atau *footer* tak diperlukan karena dapat digunakan berulang-ulang tanpa perlu menuliskannya kembali. Tentu saja pengaturan *blade* dapat dilakukan melalui *controller* sehingga memudahkan kita dalam membuat *website* yang dinamis.

5. Lebih Cepat, Hemat Waktu, dan Biaya

Tentunya, dengan penggunaan *framework* Laravel, *project* yang sedang kita kerjakan akan terasa lebih cepat serta tidak memakan waktu yang banyak. Selain itu, dibanding *framework* lainnya, Laravel juga termasuk *framework* yang hemat biaya.

6. Mendukung Kemudahan Bekerja

Penulisan kode yang taat aturan akan otomatis memudahkan proses bekerja sama di antara para pengembang/*developer*.

7. Dokumentasi Lengkap

Sebagian bisa diakses secara bebas di tautan berikut:

- ✓ id-laravel.com
- ✓ learninglaravel.net
- ✓ laracasts.com
- ✓ laravel-news.com

8. Mudah Digunakan

Karena Laravel merupakan *framework* yang mudah untuk digunakan/*user friendly*, hal ini akan menguntungkan para pengembang untuk memberikan UI/UX yang elegan.

9. Adanya Dukungan Komunitas

Banyaknya pengguna Laravel membuat *framework* ini disokong oleh komunitas yang kuat.

10. *Support* MVC

MVC adalah konsep arsitektur dalam pengembangan aplikasi berbasis web yang membagi aplikasi web menjadi 3 bagian, di mana setiap bagian memiliki tugas-tugas serta tanggung jawab masing-masing. Tiga bagian itulah yang disebut dengan *Model*, *View*, dan *Controller*.

Tabel 2.2 MVC

(Sumber: Y Yudhanto dan HA Prasetyo, 2019)

Model	Bertugas untuk mengatur, menyiapkan, memanipulasi, dan mengorganisasikan data (dari <i>database</i>) sesuai dengan instruksi dari <i>controller</i> .
View	Bertugas untuk menyajikan informasi (yang mudah dimengerti) kepada <i>user</i> sesuai dengan instruksi dari <i>controller</i> .
Controller	Bertugas untuk mengatur apa yang harus dilakukan <i>model</i> dan <i>view</i> mana yang harus ditampilkan berdasarkan permintaan dari <i>user</i> . Namun, terkadang permintaan dari <i>user</i> tidak selalu memerlukan aksi dari <i>model</i> . Misalnya seperti menampilkan halaman <i>form</i> untuk registrasi <i>user</i> .

Basis Data

Basis data yang juga dikenal sebagai *database*, terdiri dari kata basis dan data. Data merupakan catatan atas kumpulan fakta yang mewakili suatu objek. Data memiliki ciri bersifat mentah dan tidak memiliki konteks. Sedangkan dapat diartikan sebagai markas, tempat berkumpul dari suatu objek atau representasi objek. (NKDA Jayanti dan NK Sumiari, 2018)

Basis data (*database*) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. (Y Yudhanto dan HA Prasetyo, 2019)

MySQL

MySQL merupakan *server* yang melayani *database*. Untuk membuat dan mengolah *database*, kita dapat mempelajari pemrograman khusus yang disebut *query* (perintah) SQL. *Database* sendiri dibutuhkan jika kita ingin meng data dari *user* menggunakan *form* HTML untuk kemudian diolah PHP agar bisa disimpan ke dalam *database* MySQL. (J Enterprise, 2018)

Structured Query Language (SQL)

Structured Query Language atau yang lebih dikenal sebagai SQL dibuat untuk proses *Relational Database Management System* (RDBMS) dalam proses operasinya. Dari sisi paradigma pemrograman, SQL merupakan bahasa pemrograman deklaratif, yaitu bahasa pemrograman ini tidak memiliki konsep *data flow* dan terbagi menjadi beberapa bagian, seperti: *clauses*, *expression*, *predicates*, *queries*, *statements*, dan lainnya. Sebagai catatan, untuk mengakhiri *syntax* dalam SQL biasanya menggunakan tanda *semicolon* (;).

Bahasa SQL secara umum dibagi dua, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). DDL adalah bahasa SQL yang berkenaan dengan perintah terhadap struktur *database*, seperti: *create*, *rename*, *drop*, dan lainnya. Sedangkan DML adalah bahasa SQL yang berhubungan dengan manipulasi tabel yang telah dibuat dalam suatu struktur *database*, Contoh DML: *select*, *insert*, *join*, dan lainnya. (Budi Aribowo, 2021)

XAMPP

XAMPP adalah kompilasi program aplikasi gratis terfavorit di kalangan *developer/programmer* yang berguna untuk pengembangan *website* berbasis PHP dan MySQL. Dengan satu aplikasi ini kita akan mendapatkan paket *software* komplet yang bisa dijalankan pada Windows ataupun Linux sehingga *programmer* dapat dengan mudah melakukan simulasi pada komputer *local* sebelum diunggah ke internet. *Software* XAMPP dibuat dan dikembangkan oleh Apache Friends dan bisa didapatkan dengan *download* di <https://www.apachefriends.org/>.

Perangkat ini memiliki kelebihan, yaitu berperan sebagai *server* web Apache untuk simulasi pengembangan *website*. Tool pengembangan web ini mendukung teknologi web populer seperti PHP, MySQL, dan Perl.

Melalui program ini, *programmer* web dapat menguji aplikasi web yang dikembangkan dan mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet. XAMPP juga dilengkapi fitur manajemen *database* PHPMyAdmin seperti pada

server hosting sungguhan sehingga pengembang web dapat mengembangkan aplikasi web berbasis *database* secara mudah. Selain XAMPP adapula LAMP, WAMP, dan MAMP. (Y Yudhanto dan HA Prasetyo, 2019)

PhpMyAdmin

PhpMyAdmin adalah aplikasi web untuk mengelola *database* MySQL dan *database* MariaDB dengan mudah melalui antarmuka (*interface*) grafis. Aplikasi web ini ditulis menggunakan bahasa pemrograman PHP. Sebagaimana aplikasi-aplikasi lain untuk lingkungan web (aplikasi yang dibuka atau dijalankan menggunakan browser), phpMyAdmin juga mengandung unsur HTML/XHTML, CSS, dan juga kode JavaScript.

Aplikasi web ini ditujukan untuk memudahkan pengelolaan basis data MySQL dan MariaDB dengan penyajian antarmuka web yang lengkap dan menarik. Keberadaan phpMyAdmin yang dianggap sangat penting juga sifatnya yang terbuka menjadikannya salah satu aplikasi yang selalu ada di dalam cPanel (aplikasi populer untuk pengontrol *website*). Hal ini menunjukkan bahwa penyedia web *hosting* (*web hosting provider*) menaruh kepercayaan yang sangat besar pada phpMyAdmin sebagai salah satu aplikasi web yang dipasang (instal) di *server*.

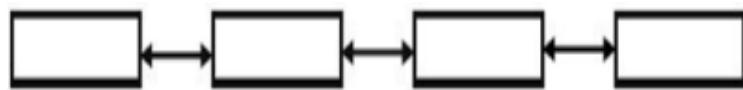
Keuntungan dengan hadirnya phpMyAdmin tidak saja dapat dinikmati oleh penyedia web *hosting*, kita juga bisa menginstal phpMyAdmin di *server* kita sendiri (*server lokal*) asalkan syarat minimumnya (*minimum requirements*) telah terpenuhi. (Y Yudhanto dan HA Prasetyo, 2019)

Struktur Navigasi

Struktur navigasi adalah alur yang digunakan dalam aplikasi yang dibuat. Sebelum menyusun aplikasi multimedia ke dalam sebuah *software*, kita harus menentukan terlebih dahulu alur apa yang akan digunakan dalam aplikasi yang dibuat. Bentuk dasar dari struktur navigasi yang biasa digunakan dalam proses pembuatan aplikasi multimedia ada empat macam, yaitu struktur navigasi linier, hierarki, non linier, dan campuran. (Papong Setiawati, 2018)

Struktur Navigasi Linier

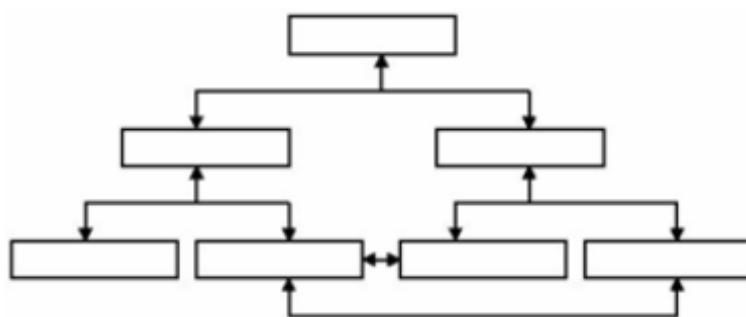
Struktur navigasi linier merupakan struktur navigasi yang mempunyai satu rangkaian cerita berurutan. Struktur ini menampilkan satu demi satu tampilan *layer* secara berurutan menurut aturannya.



Gambar 2.1 Struktur Navigasi Linier
 (Sumber: Papong Setiawati, 2018)

Struktur Navigasi Hierarki

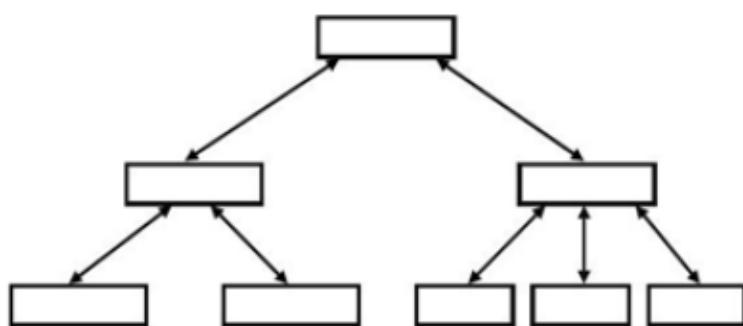
Struktur navigasi hierarki sering disebut struktur navigasi bercabang, yaitu merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data atau gambar pada *layer* dengan kriteria tertentu. Tampilan pada menu utama disebut *master page* (halaman utama satu), halaman tersebut mempunyai halaman percabangan yang disebut *slave page* (halaman pendukung) dan jika dipilih akan menjadi halaman kedua begitu seterusnya.



Gambar 2.2 Struktur Navigasi Hierarki
 (Sumber: Papong Setiawati, 2018)

Struktur Navigasi Non Linier

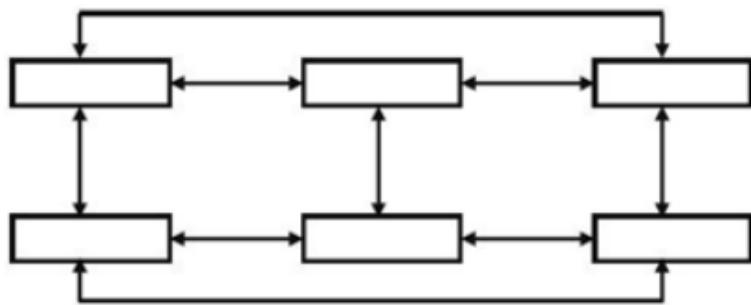
Struktur navigasi non linier (tidak terurut) merupakan pengembangan dari struktur navigasi linier, hanya saja pada struktur ini diperkenankan untuk membuat percabangan. Percabangan pada struktur non linier berbeda dengan percabangan pada struktur hierarki, pada struktur ini kedudukan semua *page* sama, sehingga tidak dikenal adanya *master* atau *slave page*.



Gambar 2.3 Struktur Navigasi Non Linier
 (Sumber: Papong Setiawati, 2018)

Struktur Navigasi Campuran

Struktur navigasi campuran (*composite*) merupakan gabungan dari struktur sebelumnya dan disebut juga struktur navigasi bebas, maksudnya adalah jika suatu tampilan membutuhkan percabangan maka dibuat percabangan. Struktur ini paling banyak digunakan dalam pembuatan aplikasi multimedia.



Gambar 2.4 Struktur Navigasi Campuran

(Sumber: Papong Setiawati, 2018)

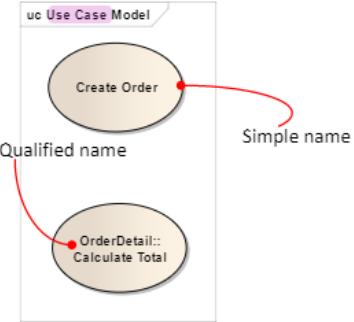
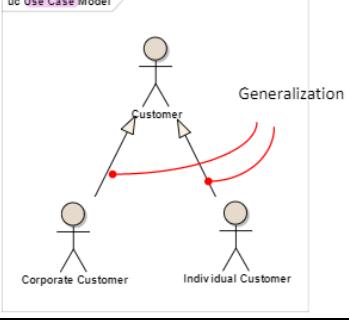
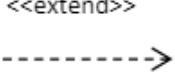
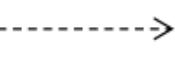
Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah bahasa pemodelan visual yang dapat digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan rancangan dari suatu sistem perangkat lunak. Pemodelan memberikan gambaran yang jelas mengenai sistem yang akan dibangun baik dari sisi struktural ataupun fungsional. UML dapat diterapkan pada semua model pengembangan, tingkatan siklus sistem, dan berbagai macam domain aplikasi. Dalam UML terdapat konsep semantik, notasi, dan panduan masing-masing diagram. UML juga memiliki bagian statis, dinamis, ruang lingkup, dan organisasional. UML bertujuan menyatukan teknik-teknik pemodelan berorientasi objek menjadi terstandarisasi. (Ibnu Akil, 2018)

Use Case Diagram

Diagram *use case* menangkap tingkah laku sistem, subsistem, kelas, atau komponen yang tampak kepada eksternal entity (*actor*). Diagram *use case* membagi fungsionalitas sistem menjadi transaksi-transaksi yang memiliki arti bagi si *actor*. Setiap potongan dari fungsi yang interaktif disebut *use case*. (Ibnu Akil, 2018)

Tabel 2.3 *Use Case Diagram*
 (Sumber: Ibnu Akil, 2018)

Gambar	Keterangan
	Sebuah <i>use case</i> adalah sebuah unit eksternal dari sistem (berupa antarmuka) yang akan menerima perintah dari seorang aktor berupa sebuah <i>event</i> . Setiap <i>use case</i> memiliki nama yang membedakannya dengan <i>use case</i> lain. Dalam praktiknya, <i>use case</i> adalah frase kata kerja aktif yang singkat. Nama itu sendiri ada yang sederhana (<i>simple name</i>), dan ada yang berkualifikasi (<i>qualified name</i>).
	Sebuah <i>actor</i> (baca: aktor) merepresentasikan suatu set peranan yang dimainkan oleh orang luar, unit kerja, atau hal-hal yang berinteraksi dengan sistem, subsistem atau kelas. Setiap aktor berpartisipasi pada satu atau lebih <i>use case</i> . Aktor tersebut berinteraksi dengan <i>use case</i> . Aktor tersebut berinteraksi dengan <i>use case</i> dengan cara mengeksekusi perintah-perintah dalam bentuk suatu <i>event</i> .
—	<i>Association</i> , merupakan sebuah jalur komunikasi antara seorang <i>actor</i> dengan satu <i>use case</i> di mana <i>actor</i> tersebut berpartisipasi.
	<i>Extend</i> , merupakan penyisipan sebuah fungsionalitas tambahan ke dalam sebuah <i>use case</i> dasar yang bersifat opsional.
	<i>Include</i> , merupakan penyisipan sebuah fungsionalitas tambahan ke dalam sebuah <i>use case</i> dasar di mana fungsionalitas tersebut bersifat <i>mandatory</i> (wajib).
→	<i>Use case generalization</i> , merupakan sebuah hubungan antara <i>use case</i> umum dengan <i>use case</i> yang lebih spesifik di mana <i>use case</i> yang lebih spesifik mewarisi <i>use case</i> umum tersebut.

Activity Diagram

Activity Diagram menggambarkan arus dari satu aktivitas ke aktivitas lainnya dari suatu fungsional sistem. *Activity diagram* dapat digunakan untuk menggambarkan arus kerja dan *business process*. Kelebihan *activity diagram* dibandingkan dengan *flowchart* adalah *activity diagram* mendukung proses yang berjalan secara parallel, sedangkan *flowchart* tidak.

Tabel 2.4 *Activity Diagram*
 (Sumber: Ibnu Akil, 2018)

Gambar	Keterangan
	<i>Initial node</i> adalah titik awal serangkaian <i>activity</i> atau <i>actions</i> .
	<i>Final node</i> adalah titik akhir dari serangkaian <i>activity</i> atau <i>actions</i> .
	<i>Action</i> adalah unit fungsional dasar di dalam sebuah <i>activity diagram</i> . Sebuah <i>activity</i> dapat berisi banyak <i>action</i> di dalamnya. Sebuah <i>activity</i> adalah unit organisasional di dalam <i>activity diagram</i> . Secara umum <i>activity</i> dapat berisi <i>group-group</i> dari <i>action-action</i> atau kumpulan <i>activity</i> yang lain.
	Sebuah <i>control flow</i> menunjukkan arus kendali dari satu <i>action</i> ke <i>action</i> lainnya. Ketika suatu <i>action</i> atau <i>activity</i> selesai dieksekusi <i>flow</i> akan berlanjut ke <i>action</i> atau <i>activity</i> lainnya.
	<i>Decision</i> dan <i>merge</i> node memiliki notasi yang sama, yaitu belah ketupat. Sebuah <i>decision</i> akan membagi <i>flow</i> menjadi dua di mana hanya satu <i>flow</i> yang akan dilalui apabila kondisi <i>guard</i> terpenuhi, sedangkan <i>merge</i> adalah tempat bergabung kembali <i>flow</i> yang terpisah karena <i>decision</i> .
	<i>Fork</i> dan <i>join</i> mengindikasikan proses yang <i>parallel</i> atau <i>concurrent</i> . <i>Fork</i> untuk percabangan proses yang dieksekusi secara <i>parallel</i> atau bersamaan, sedangkan <i>join</i> sebagai titik temu proses-proses yang <i>parallel</i> menjadi satu <i>flow</i> , di mana <i>flow</i> ini tidak akan berjalan sebelum semua proses selesai.
	<i>Partition</i> biasa juga disebut sebagai <i>swimlane</i> , yang berfungsi untuk menunjukkan <i>actor</i> atau <i>object</i> yang bertindak atau suatu <i>activity</i> atau <i>actions</i> dan menempatkan aktivitas-aktivitas tersebut pada <i>partition</i> yang bersangkutan.

Pengujian *Blackbox*

Blackbox Testing merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. *Blackbox Testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain. *Blackbox Testing* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program. (Jaya, 2018)

Keuntungan penggunaan metode *Blackbox Testing* adalah:

1. Penguji tidak perlu memiliki pengetahuan umum tentang bahasa pemrograman tertentu.
2. Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan.
3. *Programmer* dan *Tester* keduanya saling bergantung satu sama lain.

BAB III

METODE PENELITIAN

Studi Literatur

Mencari sumber dan referensi terkait dengan topik penulisan ilmiah meliputi buku, jurnal, dan internet. Studi literatur yang dicari mengenai aplikasi *Point of Sale* (POS), bahasa pemrograman PHP dan *framework* Laravel.

Analisis

Menganalisis kebutuhan aplikasi pada *Point of Sale* (POS) Waroeng Nongkrong Keboen Coffee.

Desain

Setelah melakukan analisis kebutuhan aplikasi *Point of Sale* (POS), kemudian dilakukan desain berdasarkan kebutuhan yang telah dianalisis. Desain yang dilakukan dalam penulisan ilmiah ini adalah desain *user interface*. Pada bagian ini akan dilakukan desain *layout* pada aplikasi *Point of Sale* (POS) Waroeng Nongkrong Keboen Coffee menggunakan Figma (*software*).

Evaluasi Desain

Setelah dilakukan desain selanjutnya akan dilakukan evaluasi oleh *client* untuk mengetahui apakah desain telah sesuai dengan keinginan *client*.

Implementasi

Desain yang telah disetujui oleh *client* selanjutnya akan diimplementasi-kan menggunakan bahasa pemrograman PHP dengan bantuan Laravel sebagai framework, Sublime Text 3 sebagai text editor, XAMPP sebagai web server, Composer, Git, dan MySQL sebagai basis data.

Pengujian

Setelah dilakukan implementasi *User Interface* aplikasi *Point of Sale* (POS) selanjutnya dilakukan pengujian aplikasi dengan membuat *test scenario* untuk mengetahui apakah aplikasi telah sesuai dengan kebutuhan dan berhasil menjalankan setiap fungsinya.

Evaluasi Aplikasi

Setelah dilakukan pengujian selanjutnya akan dilakukan evaluasi oleh *client* untuk mengetahui apakah aplikasi telah sesuai dengan yang diharapkan.

Menggunakan Aplikasi

Aplikasi yang telah diuji dan disetujui oleh *client* siap untuk digunakan.

BAB IV

HASIL PENELITIAN

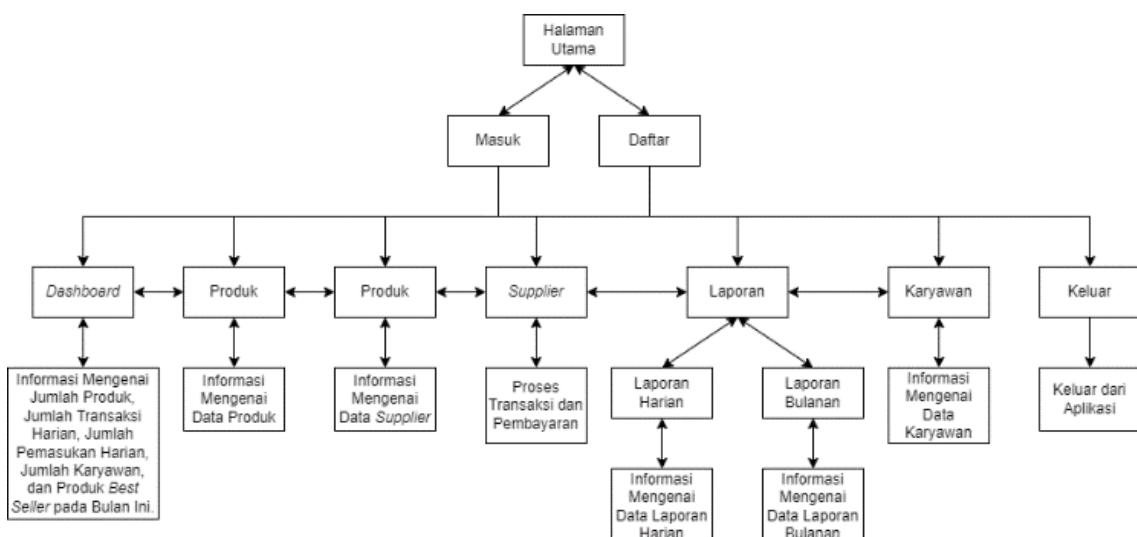
Gambaran Umum Aplikasi

Aplikasi yang akan dibangun berupa sebuah aplikasi berbasis web yang akan digunakan oleh admin atau karyawan Waroeng Nongkrong Keboen Coffee yang ingin mencatat transaksi penjualan. Aplikasi ini bertujuan untuk mempermudah admin atau karyawan dalam proses dan pencatatan transaksi penjualan, serta mengetahui laporan penjualan harian atau bulanan berdasarkan periode tertentu. Pengguna dengan mudah melakukan proses transaksi dengan menambahkan menu ke dalam keranjang yang nantinya akan tersimpan di menu laporan harian atau menu laporan bulanan jika proses transaksi sudah berhasil dilakukan.

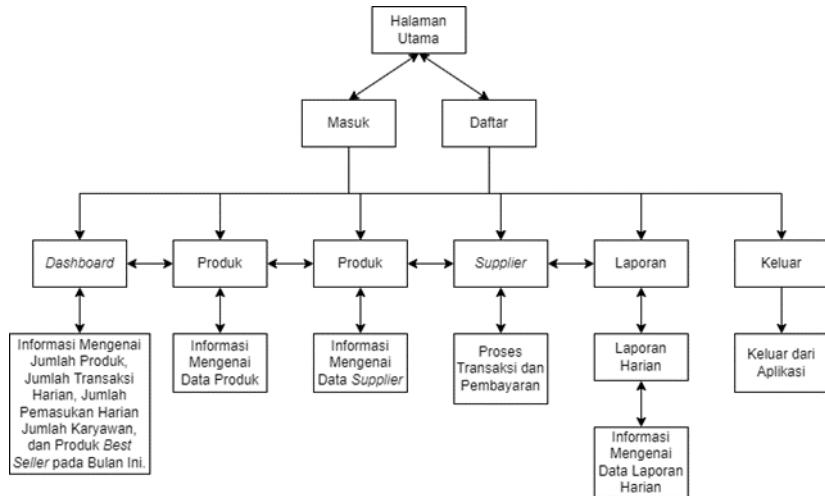
Perancangan Struktur Navigasi

Struktur navigasi adalah alur yang digunakan dalam aplikasi yang dibuat. Sebelum menyusun aplikasi multimedia ke dalam sebuah software, kita harus menentukan terlebih dahulu alur apa yang akan digunakan dalam aplikasi yang dibuat. Bentuk dasar dari struktur navigasi yang biasa digunakan dalam proses pembuatan aplikasi multimedia ada empat macam, yaitu struktur navigasi linier, hierarki, non linier, dan campuran (Setiawati, 2018).

Struktur navigasi yang akan digunakan oleh aplikasi Point of Sale (POS) adalah struktur navigasi campuran antara struktur navigasi hierarki dan non linier. Gambar 4.1 dan Gambar 4.2 adalah gambaran struktur navigasi yang akan dibangun:



Gambar 4.1. Struktur Navigasi Admin



Gambar 4.2. Struktur Navigasi Karyawan

Perancangan Diagram UML

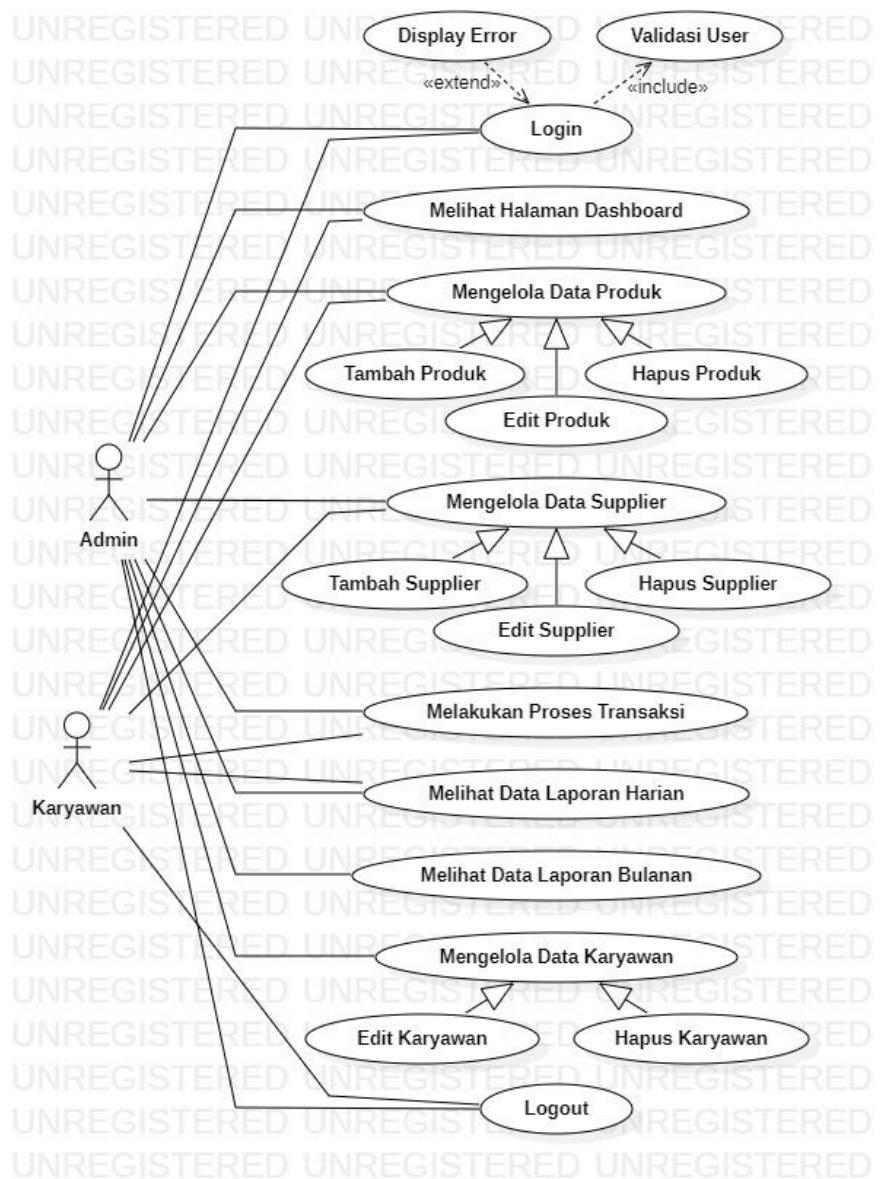
Unified Modelling Language (UML) adalah bahasa pemodelan visual yang dapat digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan rancangan dari suatu sistem perangkat lunak. Pemodelan memberikan gambaran yang jelas mengenai sistem yang akan dibangun baik dari sisi struktural ataupun fungsional. UML dapat diterapkan pada semua model pengembangan, tingkatan siklus sistem, dan berbagai macam domain aplikasi. Dalam UML terdapat konsep semantik, notasi, dan panduan masing-masing diagram. UML juga memiliki bagian statis, dinamis, ruang lingkup, dan organisasional. UML bertujuan menyatukan teknik-teknik pemodelan berorientasi objek menjadi terstandarisasi (Akil, 2018).

Pada penelitian ini dibuat dua buah diagram yaitu Use Case Diagram dan Activity Diagram.

Use Case Diagram

Gambar 4.3 menjelaskan bahwa aplikasi Point of Sale (POS) memiliki 2 aktor, yaitu admin dan karyawan. Use case di atas menunjukkan hanya admin yang bisa mengakses menu laporan bulanan dan menu karyawan. Pada aplikasi Point of Sale (POS) terdapat 8 menu utama, yaitu dashboard, produk, supplier, transaksi, laporan harian, laporan bulanan, dan karyawan. Pada menu dashboard admin atau karyawan dapat melihat informasi jumlah produk, jumlah transaksi harian, jumlah pemasukan harian, jumlah karyawan dan produk best seller pada bulan ini. Pada menu produk admin atau karyawan dapat melakukan penambahan, pengubahan dan

penghapusan data produk. Pada menu supplier admin atau karyawan dapat melakukan penambahan, pengubahan dan penghapusan data supplier. Pada menu transaksi karyawan atau admin dapat melakukan proses transaksi. Pada menu laporan harian admin atau karyawan dapat melihat data laporan harian, sedangkan pada menu laporan bulanan hanya admin yang dapat melihat data laporan bulanan berdasarkan periode tertentu. Pada menu karyawan hanya admin yang dapat melakukan pengubahan dan penghapusan data karyawan.

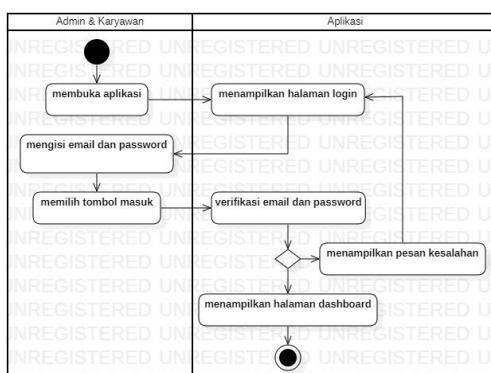


Gambar 4.1. Use Case Diagram

Activity Diagram

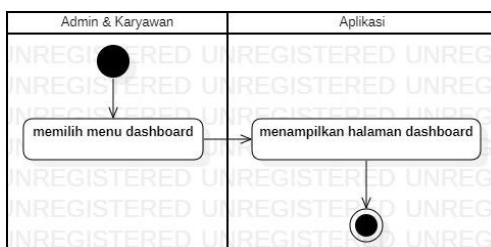
Activity Diagram menggambarkan arus dari satu aktivitas ke aktivitas lainnya dari suatu fungsional sistem. *Activity diagram* dapat digunakan untuk menggambarkan arus kerja dan *business process*. Kelebihan *activity diagram* dibandingkan dengan *flowchart* adalah *activity diagram* mendukung proses yang berjalan secara parallel, sedangkan *flowchart* tidak (Akil, 2018).

1. Activity Diagram Login



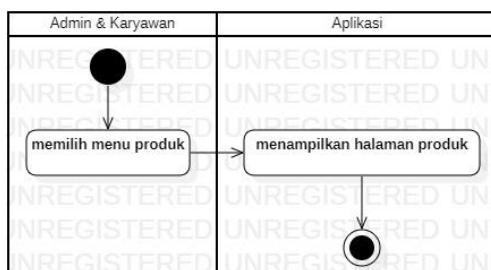
Gambar 4.2. Activity Diagram Login

2. Activity Diagram Halaman Dashboard



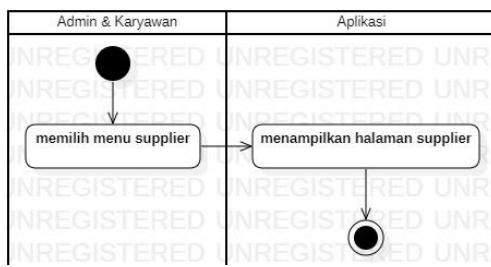
Gambar 4.3. Activity Diagram Halaman Dashboard

3. Activity Diagram Mengelola Data Produk



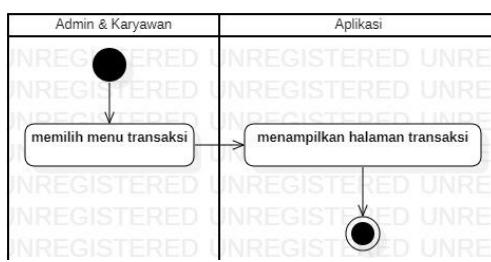
Gambar 4.4. Activity Diagram Mengelola Data Produk

4. *Activity Diagram* Mengelola Data Supplier



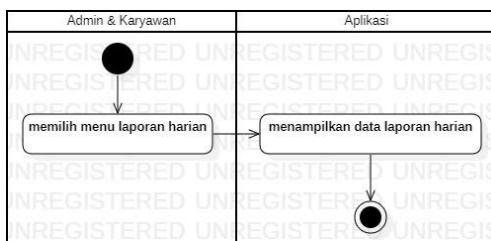
Gambar 4.5. Activity Diagram Mengelola Data Supplier

5. *Activity Diagram* Melakukan Proses Transaksi



Gambar 4.6. Activity Diagram Melakukan Proses Transaksi

6. *Activity Diagram* Melihat Data Laporan Harian



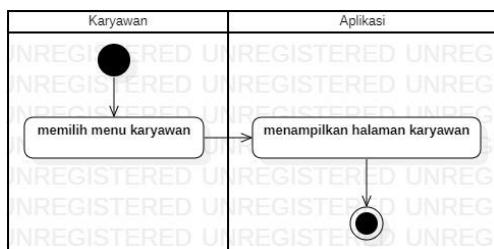
Gambar 4.7. Activity Diagram Melihat Data Laporan Harian

7. *Activity Diagram* Melihat Data Laporan Bulanan



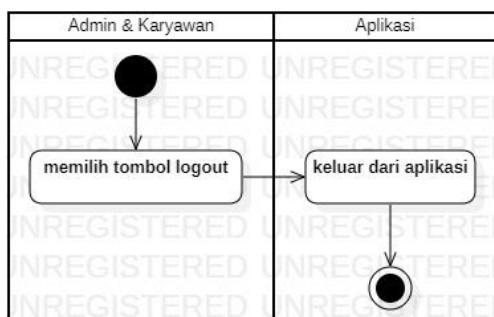
Gambar 4.8. Activity Diagram Melihat Data Laporan Bulanan

8. Activity Diagram Mengelola Data Karyawan



Gambar 4.9. Activity Diagram Mengelola Data Karyawan

9. Activity Diagram Logout



Gambar 4.10. Activity Diagram Logout

Uji Coba

Blackbox Testing merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. Blackbox Testing bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain. Blackbox Testing memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. (Jaya, 2018).

Pada tahap ini dilakukan sebuah pengujian *blackbox* terhadap aplikasi *Point of Sale* (POS) yang telah dirancang untuk memastikan bahwa aplikasi yang dibuat telah sesuai dengan kebutuhan dan telah lolos dari *bug* atau *error*.

Hasil Pengujian Fungsionalitas

Hasil pengujian fungsionalitas setiap menu dapat dilihat pada Tabel 1. Berdasarkan Tabel 1, dapat dilihat bahwa setiap menu pada aplikasi berhasil dijalankan.

Tabel 2.5. Hasil Pengujian Fungsionalitas

No.	Menu	Pengujian	Keterangan
1.	Halaman Masuk	Menampilkan halaman masuk untuk masuk ke dalam aplikasi.	Berhasil
2.	Halaman Daftar	Menampilkan halaman pendaftaran bagi karyawan yang belum memiliki akun.	Berhasil
3.	Halaman <i>Dashboard</i>	Menampilkan informasi mengenai jumlah produk, jumlah transaksi harian, jumlah pemasukan harian, jumlah karyawan, dan produk <i>best seller</i> pada bulan ini.	Berhasil
4.	Halaman Produk	Menampilkan informasi mengenai data produk, serta melakukan operasi CRUD	Berhasil
5.	Halaman <i>Supplier</i>	Menampilkan informasi mengenai data <i>supplier</i> .	Berhasil
6.	Halaman Transaksi	Menampilkan halaman transaksi dan pembayaran.	Berhasil
7.	Halaman Laporan Harian	Menampilkan informasi mengenai data laporan harian	Berhasil
8.	Halaman Laporan Bulanan	Menampilkan informasi mengenai data laporan bulanan.	Berhasil
9.	Halaman Karyawan	Menampilkan informasi mengenai data karyawan	Berhasil

BAB V

KESIMPULAN DAN SARAN

Kesimpulan

Beberapa kesimpulan yang dapat diambil dari penulisan ilmiah ini adalah sebagai berikut:

1. Aplikasi *Point of Sale* (POS) Waroeng Nongkrong Keboen Coffee berbasis web telah berhasil dibuat dengan menggunakan bahasa pemrograman PHP dengan bantuan *framework* Laravel sesuai dengan rancangan yang diharapkan.
2. Aplikasi ini dapat diakses melalui situs <https://wmkeboencoffee.com>
3. Aplikasi ini dapat mempermudah admin dan karyawan Waroeng Nongkrong Keboen Coffee dalam melakukan proses transaksi, pencatatan transaksi, dan menghitung total pemasukan harian atau bulanan berdasarkan periode tertentu.
4. Berdasarkan hasil uji coba yang telah dilakukan, aplikasi ini telah sesuai dengan kebutuhan dan telah lolos dari *bug* atau *error*

Saran

Beberapa saran untuk penelitian selanjutnya yang dapat penulis berikan dalam Penulisan Ilmiah ini adalah:

1. Aplikasi ini hanya berbasis web, untuk pengembangan lebih lanjut diharapkan dapat ditambahkan *platform* yang beragam seperti android, iOS, dll.
2. Aplikasi ini hanya menyediakan pembayaran secara tunai, untuk pengembangan lebih lanjut diharapkan dapat menyediakan pembayaran secara nontunai.

DAFTAR PUSTAKA

- Budi Aribowo. 2021. *Dasar-Dasar SQL MariaDB*. Yogyakarta: Mirra Buana Media.
- D Krisbiantoro dan PD Abda'u. 2021. *DASAR PEMROGRAMAN WEB dengan bahasa HTML, PHP, dan Database MySQL*. Banyumas: Zahira Media Publisher.
- Henderi. 2020. *Sistem Basis Data: Model Relasional, SQL, dan Object Oriented Database*. Yogyakarta: Bintang Pustaka Madani.
- Ibnu Akil. 2018. *Referensi dan Panduan UML 2.4 Singkat Tepat Jelas*. Jakarta: Penerbit Andi.
- J Enterprise. 2018. *HTML, PHP, dan MySQL untuk Pemula*. Jakarta: Elex Media Komputindo.
- M Jannah, Sarwandi dan C Creative. 2019. *Mahir Bahasa Pemrograman PHP*. Jakarta: Elex Media Komputindo.
- Moka. 2020. Definisi dan Fungsi dari Point of Sale (POS). <https://blog.mokapos.com/point-of-sale-adalah> [Tanggal Akses: 21 Oktober 2021]
- NKDA Jayanti dan NK Sumiari. 2018. *Teori Basis Data*. Yogyakarta: Penerbit Andi.
- Papong Setiawati. 2018. Analisa dan Perancangan Sistem Informasi Penyedia Lowongan Pekerjaan yang Direkomendasi Berdasarkan Standar Kompetensi Kerja Nasional Indonesia (SKKNI). *Jurnal Ilmu Komputer*, 3(2), 136-147.
- TS Jaya. 2018. Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). *Jurnal Informatika: Jurnal Pengembangan IT*, 3(1), 45-48.

Y Yudhanto dan HA Prasetyo. 2019. *Mudah Menguasai Framework Laravel*. Jakarta: Elex Media Komputindo.