



Aplikasi P.O.S (*Point of Sales*) Dengan Fitur Notifikasi Piutang Melalui *E-mail* Dan *Whatsapp* (Studi Kasus: Toko Fendymart Bondowoso)

Savira Ainun Mardiyah Wijayantoro^{1*}, Habibatul Azizah Al Faruq², Amalina Maryam Zakiyyah³

Program Studi Manajemen Informatika, Universitas Muhammadiyah Jember^{1,2,3}

Email: saviraainun00@gmail.com^{1*}, habibatulazizah@unmuhjember.ac.id², amalinamaryam@unmuhjember.ac.id³

ABSTRAK

Toko Fendymart merupakan salah satu toko retail di Bondowoso yang menjual berbagai jenis barang seperti sembako, peralatan rumah tangga, kebutuhan sehari-hari, makanan ringan, minuman kaleng, es krim, sayur-mayur hingga bumbu dapur. Permasalahan yang sering terjadi di Toko Fendymart yaitu piutang pelanggan yang gagal dibayar, tercatat pada bulan Juni sampai bulan Juli 2021 ada sekitar 10 orang pelanggan yang memiliki piutang dan tidak dapat tertagih sesuai dengan batas jatuh tempo hal ini dikarenakan tidak ada pengingat untuk pembayaran piutang sesuai jatuh tempo, tentu hal tersebut dapat membuat kerugian bagi Toko Fendymart. Penelitian ini bertujuan untuk membangun sebuah aplikasi P.O.S (*Point of Sales*) dengan fitur notifikasi piutang melalui *e-mail* dan *whatsapp* dengan harapan dapat membantu Toko Fendymart. Hasil penelitian menunjukkan bahwa aplikasi sudah sesuai dengan kebutuhan dan dapat digunakan untuk kegiatan operasional di Toko Fendymart serta fitur notifikasi piutang juga telah membantu meningkatkan terbayarnya piutang pelanggan sesuai dengan jatuh tempo dengan peningkatan sebesar 18.51 % dari sebelum adanya aplikasi.

Kata Kunci: Aplikasi Web, Aplikasi *Point of Sales*, Aplikasi Kasir, Notifikasi Piutang

ABSTRACT

Fendymart store is one of the retail stores in Bondowoso that sells various types of goods such as basic-necessities, household appliances, daily necessities, snacks, canned drinks, ice cream, vegetables to kitchen spices. Problems that often occur at the Fendymart Store are customer receivables that fail to be paid, recorded in June to July 2021 there are about 10 customers who have receivables and cannot be collected according to the due date this is because there are no reminders for payment of receivables as they fall tempo, of course this can make a loss for the Fendymart Store. This study aims to build a P.O.S (Point of Sales) application with a receivable notification feature via e-mail and whatsapp in the hope of helping Fendymart Stores. The results show that the application is in accordance with the needs and can be used for operational activities at the Fendymart Store and the receivable notification feature has also helped increase the payment of customer receivables according to maturity with an increase of 18.51 % from before the application

Keywords: Web Application, Point of Sales Application, Cashier Application, Accounts Receivable Notification

1. PENDAHULUAN

Pada era globalisasi teknologi berkembang sangat pesat khususnya dalam bidang teknologi informasi. Perkembangan teknologi informasi saat ini sudah menjadi salah satu aspek penting yang menjangkau berbagai macam aspek kehidupan mulai dari pendidikan, *entertainment* atau hiburan, hingga bisnis. Penggunaan teknologi informasi memiliki banyak keuntungan dalam menghemat waktu dan tenaga. Saat ini, teknologi informasi tidak hanya digunakan di perusahaan-perusahaan besar saja tetapi digunakan juga di toko retail.

Toko retail merupakan toko yang menjual berbagai jenis produk dari berbagai produsen pemasok, dan produk yang dijual biasanya merupakan produk yang siap digunakan, seperti produk makanan, minuman, serta produk untuk kebutuhan sehari-sehari, dimana penjualan dilakukan secara langsung kepada *customer* (Sugianto & Tjandra, 2016).

Toko Fendymart merupakan salah satu toko retail di Bondowoso, yang memiliki 5 pegawai. Pada awal berdirinya hanya sebuah usaha kecil yang beralamat di Jl. Rengganis No. 5 Kembang Kecamatan Bondowoso Kabupaten Bondowoso. Toko Fendymart menjual berbagai jenis barang yang beranekaragam seperti sembako, peralatan rumah tangga, kebutuhan sehari – hari, makanan ringan, minuman kaleng, es krim, sayur-mayur hingga bumbu dapur.

Fendymart, sebagai sebuah toko retail dengan jumlah pelanggan sebanyak 50 sampai 60 pelanggan per harinya, ada permasalahan yang sering terjadi di Toko Fendymart yaitu piutang

pelanggan yang gagal dibayar, tercatat pada bulan Juni sampai bulan Juli 2021 ada sekitar 10 orang pelanggan yang memiliki piutang dan tidak dapat tertagih sesuai dengan batas jatuh tempo hal ini dikarenakan tidak ada pengingat untuk pembayaran piutang sesuai jatuh tempo, tentu hal tersebut dapat membuat kerugian bagi Toko Fendymart.

Dari permasalahan yang telah diuraikan sebelumnya, maka diperlukan sebuah aplikasi *Point of Sales* dengan fitur notifikasi piutang melalui *e-mail* dan aplikasi *whatsapp* dengan harapan dapat membantu Toko Fendymart dalam segi operasional sehingga setiap piutang dan data transaksi dapat tercatat dengan baik. Selain itu dengan adanya aplikasi ini nantinya juga dapat mempermudah dan mempercepat proses pelayanan serta dapat memberikan solusi terhadap masalah-masalah yang terjadi di Toko Fendymart.

P.O.S atau *Point of Sales* merupakan sebuah aplikasi yang digunakan untuk mengolah data penjualan, pembelian, hutang, dan retur pembelian serta laporan-laporan yang dibutuhkan sebagai alat pendukung keputusan bagi pelaku bisnis (Sugianto & Tjandra, 2016).

Terdapat beberapa penelitian terdahulu yang dijadikan acuan dalam membangun aplikasi P.O.S (*Point of Sales*) di Toko Fendymart diantaranya, pada tahun 2016 oleh Aslam Fatkhudin berjudul “Sistem Informasi Penjualan dan Pembelian Pada Toko Elektronik Lubada Jaya Kajen Dengan Menggunakan Java” dengan lokasi penelitian di Toko Elektronik Lubada Jaya Kajen. Hasil dari penelitian ini adalah aplikasi yang dibuat dapat memberikan kemudahan dalam mencatat transaksi penjualan, transaksi pembelian serta transaksi retur barang. Selain itu juga mempermudah untuk mendapatkan laporan keuangan, pembelian, penjualan serta laporan retur yang langsung dihasilkan dari data transaksi yang diinputkan.

Pada tahun 2017, Hidayat dkk., merancang *Point of Sales* (P.O.S) pada Usaha Dagang Hada Putra Berbasis Java Neatbeans dengan judul “Aplikasi *Inventory* Barang, Pengelolaan Data Hutang-Piutang Dagang dan Prediksi Kebutuhan Barang dengan Metode *Single Exponential Smoothing* (Studi Kasus: Usaha Dagang Hada Putra)”. Hasil dari penelitian ini adalah menyediakan informasi berupa SMS untuk pemilik Hada Putra yaitu pemberitahuan stok barang menipis, untuk *supplier* yaitu informasi pemesanan barang, penerimaan barang dan laporan kondisi barang serta untuk pelanggan yaitu pemberitahuan batas waktu jatuh tempo pembayaran piutang.

2. KAJIAN PUSTAKA

A. Aplikasi

Menurut Sutarman (2009), aplikasi adalah sebuah program komputer yang dibuat oleh suatu pihak tertentu untuk para pengguna yang beroperasi dalam bidang tertentu, seperti komunikasi, penerbangan, perdagangan, pertokoan, dan lain sebagainya. Dengan kata lain aplikasi merupakan salah satu bagian dari perangkat lunak komputer yang menggunakan kemampuan komputer langsung untuk melakukan tugas-tugas tertentu sesuai dengan keinginan pengguna.

B. Website

Menurut (Enterprise, 2018), Website merupakan sebuah halaman yang dapat dilihat oleh siapapun yang terkoneksi jaringan internet, dimana halaman tersebut berisi informasi berupa gambar, teks dan video atau gabungan dari semuanya.

C. P.O.S (*Point of Sales*)

Menurut (Tahir dkk., 2019), P.O.S atau *Point of Sales* adalah kegiatan yang berorientasi pada penjualan di sebuah toko maupun perusahaan dengan menggunakan sistem untuk membantu proses transaksi. Sistem yang ada biasanya terdiri dari perangkat keras seperti PC/Komputer, *Barcode Scanner*, Printer, dan perangkat lunak berupa (Pelaporan, Manajemen Barang, Manajemen Pelanggan, Manajemen Pembelian, Manajemen Penjualan serta Standar Keamanan Transaksi).

D. HTML

Menurut Anhar (2010), *Hypertext Markup Language* (HTML) merupakan aturan-aturan dalam membuat kode program sehingga kode tersebut dapat dibaca dan ditampilkan oleh *browser*. HTML juga bisa disebut sekumpulan simbol atau tag yang disimpan dalam sebuah *file* yang dapat di baca oleh *browser* guna menampilkan sebuah halaman *website*.

E. Java Script

Java Script adalah kode program yang dapat diimplementasikan kedalam halaman web menggunakan *tag Script* dan dapat berjalan pada hampir semua *browser modern* yang ada sekarang (Sunnyoto, 2007). Beberapa hal tentang *Java Script*:

1. Digunakan untuk menambah interaktif dengan pengguna pada halaman *website*.
2. Merupakan bahasa *scripting* yang sangat populer digunakan pada saat ini.
3. Bahasa *scripting* ialah bahasa pemrograman yang tidak memerlukan *resource* terlalu banyak untuk dijalankan serta mudah dipahami.
4. Biasanya terdiri dari kumpulan kode yang dijalankan di *browser* pada sisi *client*.
5. Biasanya ditambahkan pada kode HTML.
6. Merupakan bahasa interpreter (tidak perlu dikompilasi untuk dapat dijalankan).
7. Bersifat *open source* atau tidak perlu membayar lisensi untuk menggunakannya.

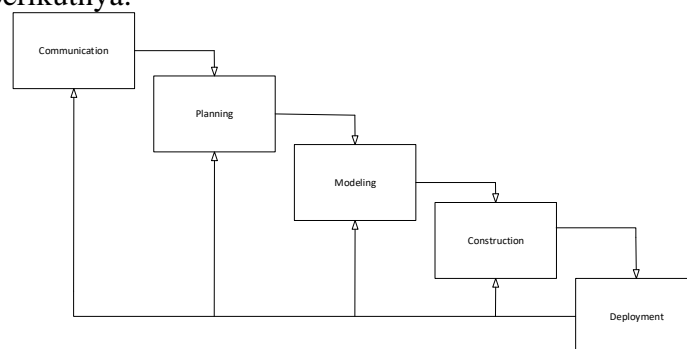
F. PHP

Menurut (Enterprise, 2017) PHP adalah sebuah bahasa pemrograman yang bisa digunakan untuk mengatur kapan sebuah data ditambah, ditampilkan, dan dihapus. Data yang diolah biasanya disimpan dalam sebuah *server database* yang bernama *MySQL Server*. PHP adalah bahasa pemrograman *server-side*, yang berarti setiap proses dilakukan melalui pemrosesan dari sisi *server*, sehingga kode PHP tidak bisa diberikan langsung ketika ada *request* dari *client (browser)*.

3. METODE PENELITIAN

A. Metode Waterfall

Metode penelitian yang digunakan dalam penelitian ini yaitu metode *waterfall*. Menurut Pressman (2015), metode *waterfall* adalah model pengembangan sistem yang berurutan dan sistematis. Disebut dengan *waterfall* karena tahap yang dilalui harus menunggu sampai dengan selesai untuk bisa melanjutkan ke tahap berikutnya.



Gambar 1. Metode *Waterfall* (Pressman, 2015)

Tahapan-tahapan dari metode *waterfall* adalah *communication (Project Initiation & Requirements Gathering)*, *planning (Estimating, Scheduling, Tracking)*, *modelling (Analysis & Design)*, dan *Construction (Code & Test)* (Pressman, 2015).

1. *Communication (Project Initiation & Requirements Gathering)*

Tahap awal dari metode ini yaitu melakukan komunikasi dengan *customer* dengan tujuan untuk menggali dan mendefinisikan kebutuhan dari sistem, baik kebutuhan fungsional maupun kebutuhan non fungsional. Dalam tahap ini peneliti melakukan pengumpulan data dengan observasi dan wawancara secara langsung ke pemilik Toko Fendymart. Observasi dilakukan dengan mengunjungi

langsung Toko Fendymart di Bondowoso dan serta melihat alur kerja yang diterapkan serta menganalisa secara langsung permasalahan yang berkaitan dengan tema yang diangkat. Selain itu wawancara juga dilakukan untuk memperjelas inti dari permasalahan yang ada dan menemukan solusi yang tepat yang harus diterapkan. Setelah proses observasi dan wawancara dilakukan, terakhir yaitu melakukan *collecting* data yang dibutuhkan untuk melakukan penelitian terhadap permasalahan yang akan diteliti.

2. *Planning (Estimating, Scheduling, Tracking)*

Pada tahap ini peneliti merencanakan dan menentukan estimasi pengerjaan dari sistem, menentukan sumber daya yang diperlukan serta mengukur resiko-resiko yang dapat terjadi selama proses pembuatan dan pengembangan sistem.

3. *Modeling (Analysis & Design)*

Tahap selanjutnya melakukan pemodelan dari sistem berdasarkan dari kebutuhan awal yang telah didefinisikan. Pemodelan yang dibuat seperti rancangan *database*, tampilan *mockup* serta alur kerja dari sistem yang nantinya akan dikembangkan.

4. *Construction (Code & Test)*

Tahap berikutnya yaitu pembuatan sistem menggunakan kode program. Setelah sistem dibuat, kemudian dilakukan *test* untuk memastikan setiap fitur yang ada pada sistem berjalan sesuai dengan fungsi masing-masing. *Test* dilakukan dengan menggunakan metode *blackbox testing*.

5. *Deployment (Operation and maintenance)*

Tahap terakhir dari metode ini yaitu mengoperasikan sistem di sisi *user*, dengan tujuan untuk mendapatkan *feedback* dari *user* sehingga sistem yang dibangun dapat sesuai dengan kebutuhan. Tidak ada ukuran pasti dalam menentukan jumlah responden untuk melakukan pengujian *usability testing*, namun menurut Nielsen (2000), 5 responden dalam melakukan *usability testing* akan menemukan 85% masalah dari suatu sistem. Sedangkan menurut Faulkner (2003), 5 responden dalam melakukan *usability testing* hanya akan menemukan 55% masalah dari sistem dan untuk menemukan 95% masalah dari sistem dibutuhkan responden sebanyak 15.

Dalam penelitian Az-zahra dkk (2015) yang berjudul “*Usability Evaluation of Mobile Application in Culinary Recommendation System*” menggunakan 5 orang sebagai responden dan penelitian yang dilakukan oleh Yuliyana dkk. (2019) dengan judul “*Usability Testing pada Aplikasi Potwis*” menggunakan 20 orang sebagai responden. Maka dalam penelitian ini akan menggunakan responden sebanyak 31 orang responden untuk mendapatkan hasil yang optimal. Setiap responden nantinya akan melakukan beberapa skenario pengujian yang telah disusun dan mengisi kuisisioner setelah selesai melakukan skenario pengujian. Kuisisioner yang dibagikan mewakili 5 kriteria *usability testing* yang meliputi: *Learnability, Efficiency, Memorability, Error* dan *Satisfaction*. Berikut daftar skenario dan kuisisioner yang telah dirancang untuk melakukan proses pengujian *usability testing*.

B. SOP dan Alur Bisnis Toko Fendymart

1. SOP penjualan barang

SOP penjualan yang dilakukan di toko Fendymart yaitu pembeli memilih barang yang dibutuhkan, kemudian kasir melayani pembelian dengan menghitung total belanja secara manual, setelah pembeli melakukan pembayaran, hasil transaksi dimasukkan kedalam kas penjualan yang nantinya akan direkap sebagai omset di akhir hari penjualan.

2. SOP pembelian barang

Untuk melakukan pembelian barang, SOP yang pertama dilakukan yaitu melakukan pengecekan terhadap ketersediaan stok barang, jika barang kurang dari stok minimum maka akan dilakukan proses *order* kepada *supplier* barang tersebut. Setelah barang datang, barang akan dicatat kembali kemudian ditempatkan di rak barang serta diberikan label harga jual.

3. SOP piutang

Untuk memberikan piutang kepada pembeli, toko Fendymart memberlakukan beberapa persyaratan yaitu, yang pertama pembeli harus merupakan pelanggan tetap dari toko Fendymart,

dimana orang tersebut setiap hari harus melakukan pembelian di toko Fendymart, yang kedua yaitu tidak boleh ada piutang yang masih belum terbayar sebelumnya, dan yang ketiga jika piutang yang diajukan melebihi Rp.150.000 (seratus lima puluh ribu rupiah), maka pembeli harus menyertakan KTP serta nomor telepon sebagai jaminan piutang, jika kurang dari itu, pembeli hanya akan dimintai nama, alamat serta nomor telepon yang bisa dihubungi untuk dimasukkan kedalam buku piutang. Jatuh tempo untuk piutang sendiri adalah 2 minggu, jika lebih dari itu, pihak Fendymart akan melakukan penagihan kepada yang bersangkutan, baik melalui telepon maupun *whatsapp*. Setelah semua persyaratan terpenuhi, transaksi piutang akan dicatat kedalam buku piutang yang nantinya akan direkap sebagai laporan piutang di akhir hari penjualan.

C. Perencanaan Kebutuhan dan Alur Bisnis Aplikasi P.O.S (*Point of Sales*)

1. Kebutuhan Sistem Fungsional

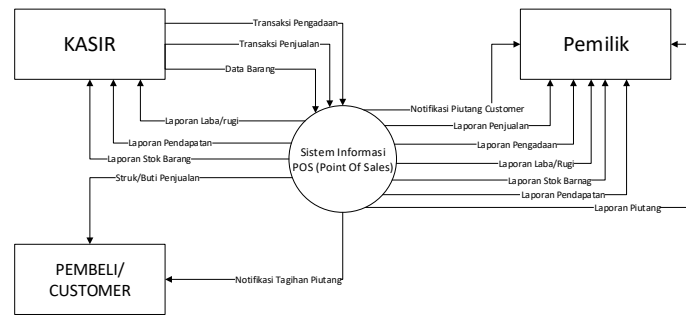
- a) Sistem dapat menambahkan barang baru.
- b) Sistem dapat menampilkan data barang serta stok yang tersisa.
- c) Sistem dapat menghapus data barang.
- d) Sistem dapat menyimpan data pembelian barang dari *supplier*.
- e) Sistem dapat menyimpan data penjualan kepada *customer*.
- f) Sistem dapat menyimpan data piutang secara otomatis.
- g) Sistem dapat memberikan notifikasi piutang pelanggan kepada pemilik melalui *e-mail*.
- h) Sistem dapat memberikan notifikasi tagihan piutang kepada pelanggan melalui *whatsapp*.
- i) Sistem dapat memberikan laporan penjualan berupa grafik.
- j) Sistem dapat memberikan laporan rugi/laba.
- k) Sistem dapat melakukan cetak struk penjualan.
- l) Sistem dapat melakukan *scanner barcode* barang.

2. Kebutuhan Non-Fungsional

- a) Sistem operasi yang digunakan adalah *windows 10*.
- b) Kebutuhan RAM minimal 4 GB.
- c) *Barcode scanner*.
- d) *Printer* termal untuk cetak struk belanja.
- e) Kebutuhan *harddisk* minimal 500GB.
- f) Kebutuhan *processor* minimal *core i3*.

3. *Data Flow Diagram* (DFD)

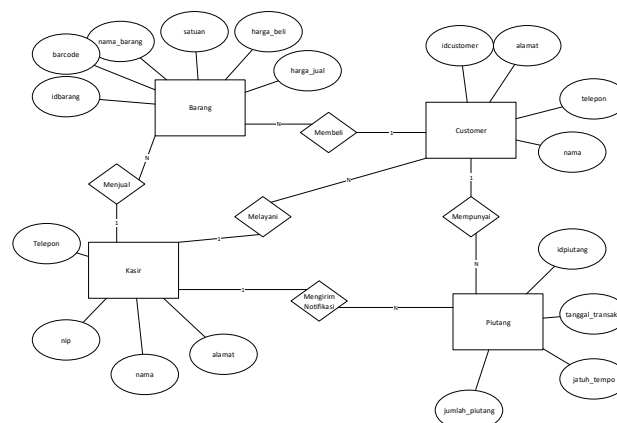
Sistem terdiri dari 3 pengguna utama yaitu kasir, pemilik dan pembeli atau *customer*. Pengguna pertama yaitu kasir, yang bertugas untuk menginputkan data barang, data transaksi penjualan dan data transaksi pengadaan barang ke dalam sistem informasi *Point of Sales* yang ada di Toko Fendymart, dimana data yang telah di inputkan oleh kasir akan menghasilkan beberapa output setelah dilakukan proses pengolahan data dari sistem. *Output* yang dapat diberikan oleh sistem diantaranya laporan laba/rugi laporan pendapatan, laporan penjualan, laporan pengadaan, laporan piutang dan struk/bukti penjualan serta notifikasi piutang *customer*. Laporan atau *output* yang telah disediakan oleh sistem dapat diakses oleh setiap pengguna sesuai dengan kebutuhan masing-masing.



Gambar 2. DFD Level 0 Sistem

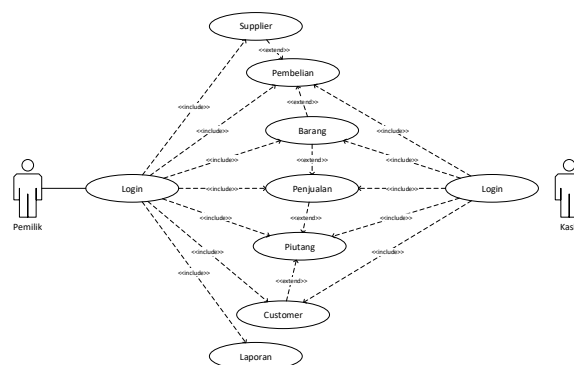
4. ER Diagram

Terdapat 4 entity utama yang nantinya akan ada di dalam sistem yaitu barang, *customer*, kasir, dan piutang. Kasir nantinya akan melayani *customer*, kasir dapat melayani beberapa *customer* sekaligus karena menggunakan relasi *one to many*, selain itu kasir juga dapat menjual dan menginputkan data barang ke dalam sistem. Dalam proses bisnis yang ada, *customer* dapat melakukan pembelian barang yang digambarkan dalam ERD menggunakan relasi *one to many*, yang artinya satu *customer* dapat membeli beberapa barang, dari proses pembelian yang terjadi dapat timbul sebuah piutang, sehingga digambarkan di dalam ERD bahwa *entity customer* dapat mempunyai piutang. Piutang yang dimiliki oleh *customer* dapat di ingatkan oleh kasir dengan mengirimkan notifikasi piutang kepada *customer* melalui *whatsapp* sesuai dengan jatuh tempo yang berlaku.



Gambar 3. ER Diagram

5. Use Case Diagram



Gambar 4. Usecase Diagram

Di dalam sistem ini terdapat dua aktor utama yaitu pemilik toko dan juga kasir. Pemilik toko nantinya dapat mengakses beberapa fitur yang berada dalam sistem diantaranya *login*, *supplier*, fitur pembelian barang, data barang, penjualan barang, piutang pelanggan, daftar *customer* dan laporan.

Sebelum dapat mengakses fitur yang ada, kedua aktor diharuskan untuk melakukan *login* terlebih dahulu untuk dapat mengakses fitur-fitur yang berada di dalam aplikasi. Fitur *supplier* digunakan untuk melihat keseluruhan data *supplier* serta untuk melakukan input data *supplier* baru, *edit* data *supplier*, dan hapus data *supplier*. Fitur pembelian barang digunakan untuk memasukkan data pembelian barang dari *supplier*. Fitur data barang digunakan untuk melihat keseluruhan data barang serta untuk melakukan *input* data barang baru, *edit* data barang, dan hapus data barang. Fitur peramalan penjualan barang digunakan untuk menginputkan data penjualan barang. Fitur piutang digunakan untuk melihat keseluruhan data piutang pelanggan serta untuk mengirim notifikasi piutang kepada pelanggan sesuai dengan jatuh tempo. Fitur *customer* digunakan untuk melihat data *customer* yang ada, menambah data *customer*, merubah data *customer* serta menghapus data *customer*. Fitur laporan digunakan untuk melihat laporan-laporan yang ada yang berkaitan dengan proses bisnis yang ada di toko, seperti laporan stok, laporan penjualan dan laporan pendapatan.

Sedangkan untuk kasir nantinya hanya akan mengakses beberapa fitur diantaranya fitur pembelian, barang, penjualan, piutang pelanggan dan *customer*. Fitur yang didapat oleh kasir jauh lebih sedikit dibandingkan dengan pemilik toko, dan untuk bisa mengakses fitur yang ada, kasir juga harus melakukan *login* terlebih dahulu.

D. Mengukur Nilai Efisiensi Fitur Notifikasi Piutang

Tahap terakhir dalam penelitian ini adalah dengan melakukan pengujian untuk mengukur nilai efisiensi notifikasi piutang yang ada di dalam aplikasi. Pengujian dilakukan dengan menghitung persentase jumlah hutang terbayar sesuai dengan jatuh tempo pada periode tertentu yaitu pada periode sebelum adanya aplikasi dan periode setelah adanya aplikasi. Berikut tahapan-tahapan yang dilakukan untuk mengukur nilai efisiensi fitur notifikasi piutang.

Pertama menghitung nilai persentase jumlah piutang yang tertagih sesuai dengan jatuh tempo menggunakan rumus:

$$\text{persentase} = \frac{(\text{jumlah piutang tertagih})}{(\text{jumlah piutang})} \times 100\% \quad (1)$$

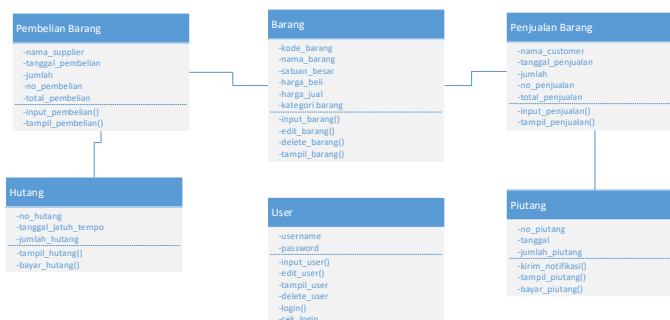
Selanjutnya untuk menghitung nilai efisiensi menggunakan rumus:

$$\text{efisiensi} = \text{persentase periode 2} - \text{persentase periode 1} \quad (2)$$

Dimana jika nilai efisiensi bernilai negatif berarti tidak ada peningkatan dari sebelum adanya dengan sesudah adanya aplikasi P.O.S dengan fitur notifikasi piutang.

4. HASIL DAN PEMBAHASAN

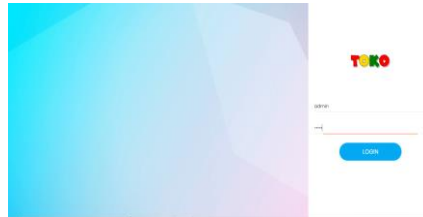
Pengkodean dilakukan dengan menggunakan *framework PHP codeigniter* dengan pendekatan pemrograman berbasis objek (*Object Oriented Programing*). Gambaran dari pengkodean dapat direpresentasikan dengan menggunakan *class diagram*. Berikut adalah *class diagram* sebagai gambaran dari pengkodean sistem yang dilakukan, dimana pemrograman dibuat dengan dibentuk beberapa *class* dan didalamnya ada *function* dan properti yang dapat dipanggil ketika dibutuhkan.



Gambar 5. Class Diagram

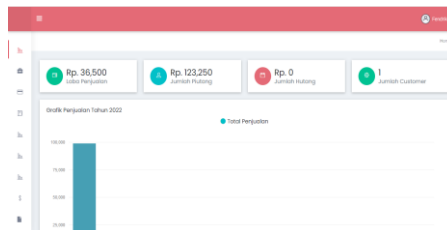
A. Hasil *Interface Login*

Fitur yang pertama kali tampil saat sistem diakses, berfungsi untuk membatasi pengguna, sehingga yang dapat mengakses sistem hanya pengguna yang mempunyai hak terhadap sistem. Pengguna nantinya akan diberikan *username* dan *password* yang dapat digunakan untuk masuk ke dalam sistem, *username* dan *password* bersifat rahasia sesuai dengan hak akses masing-masing pengguna.



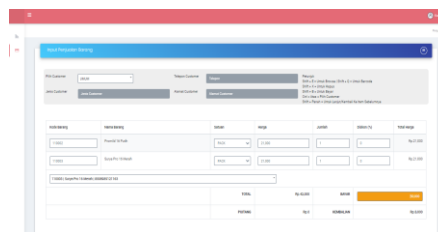
Gambar 6. Tampilan Halaman *Login*

B. Hasil *Interface Dashboard*



Gambar 7. Tampilan Halaman *Dashboard*

C. Hasil *Interface* Fitur Penjualan



Gambar 8. Tampilan Halaman Penjualan

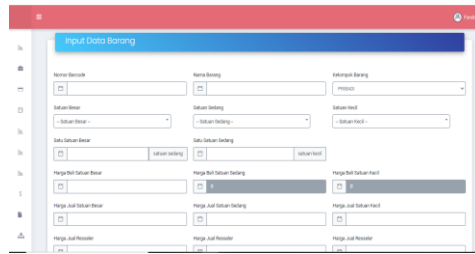
D. Hasil *Interface* List Data Barang



No	Kode Barang	ID/No Barcode	Nama Barang	Stok	Harga Beli	Harga Jual	Aksi
1	110700	00001234	BB Hitam 10	0	Rp. 100 / LEMBE Rp. 120.000 / PCS Rp. 120.000 / PREDS	Rp. 1000 / LEMBE Rp. 140.000 / PCS Rp. 100.000 / PREDS	[Edit]
2	110207	00000000000000	Lumut L04 20w	0	Rp. 4.000 / PCS Rp. 4.000 / PCS Rp. 4.000 / PCS	Rp. 12.000 / PCS Rp. 12.000 / PCS Rp. 12.000 / PCS	[Edit]
3	110108	000702000400	Dairy Besar	0	Rp. 20.000 / KALINDO Rp. 20.000 / KALINDO Rp. 10.000 / KALINDO	Rp. 20.000 / KALINDO Rp. 20.000 / KALINDO Rp. 10.000 / KALINDO	[Edit]

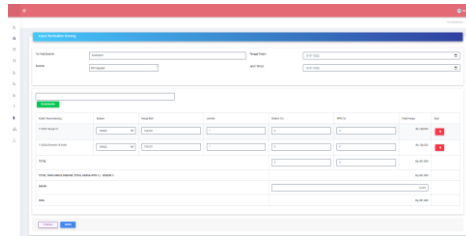
Gambar 9. Tampilan Halaman Data Barang

E. Hasil *Interface* Form Tambah Barang



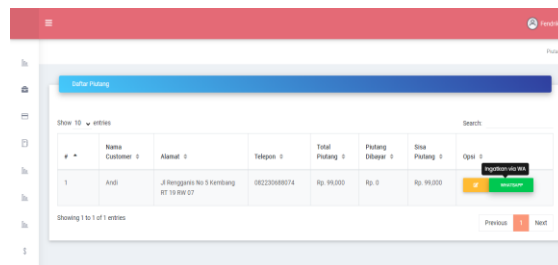
Gambar 10. Tampilan Halaman Tambah Barang

F. Hasil *Interface Form* Pengadaan Barang



Gambar 11. Tampilan Halaman Pengadaan Barang

G. Hasil *Interface Fitur* Notifikasi Piutang



Gambar 12. Tampilan Halaman Notifikasi Piutang

H. Pengujian Aplikasi dengan *Blackbox Testing*

Pengujian dilakukan dengan tujuan untuk memastikan sistem sudah berjalan sesuai dengan yang diharapkan dan meminimalisir adanya *error* maupun *bug*. Pengujian dilakukan dengan menggunakan metode pengujian *blackbox testing*. *Blackbox testing* digunakan untuk memastikan bahwa fungsional dari sistem berjalan sesuai dengan yang diharapkan dan dapat memenuhi kebutuhan. Berikut pengujian *blackbox testing* yang telah dilakukan.

Tabel 9. Skenario Pengujian *Blackbox Testing* pada *Login*

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil
1	Mengosongkan <i>username</i> dan <i>password</i> , kemudian langsung klik tombol <i>login</i>	Sistem akan menolak dan meminta untuk memasukkan <i>username</i> dan <i>password</i>	[v] Sesuai [] Tidak Sesuai
2	Memasukkan <i>username</i> yang tidak terdaftar di sistem	Sistem akan menolak dan menampilkan pesan “ <i>username</i> tidak ditemukan”	[v] Sesuai [] Tidak Sesuai
3	Memasukkan <i>username</i> yang terdaftar dengan <i>password</i> yang tidak sesuai	Sistem akan menolak dan menampilkan pesan “ <i>password</i> anda salah”	[v] Sesuai [] Tidak Sesuai
4	Mesmasukkan <i>username</i> yang terdaftar di sistem dan <i>password</i> yang sesuai	Sistem menerima akses <i>login</i> dan mengarahkan ke halaman <i>dashboard</i>	[v] Sesuai [] Tidak Sesuai

Tabel 10. Skenario Pengujian *Blackbox Testing* pada Fitur Data Barang

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil
1	Tambah data barang tanpa memasukkan data	Sistem akan menolak dan meminta memasukkan data dengan benar	[v] Sesuai [] Tidak Sesuai
2	Tambah data barang dan mengisi <i>form</i> dengan lengkap	Sistem akan menyimpan data dan menampilkan pesan “berhasil tersimpan”.	[v] Sesuai [] Tidak Sesuai
3	Klik <i>edit</i> data pada salah satu data yang akan di <i>edit</i> .	Sistem akan menampilkan data yang siap untuk <i>diedit</i> .	[v] Sesuai [] Tidak Sesuai
4	Lakukan perubahan pada saat <i>edit</i> data kemudian klik simpan	Sistem akan menyimpan perubahan dan menampilkan pesan “berhasil <i>update</i> data”.	[v] Sesuai [] Tidak Sesuai

Tabel 11. Skenario Pengujian *Blackbox Testing* pada Fitur Pengadaan Barang

No	Skenario pengujian	Hasil Yang Diharapkan	Hasil
1	Tambah data pengadaan barang dan simpan tanpa memasukkan data <i>supplier</i>	Sistem akan menolak dan meminta memasukkan data <i>supplier</i>	[v] Sesuai [] Tidak Sesuai
2	Tambah data pengadaan barang dan klik simpan tanpa memasukkan data barang	Sistem akan melakukan proses dan menampilkan pesan “gagal <i>input</i> data”	[v] Sesuai [] Tidak Sesuai
3	Tambah data barang dan klik cari barang	Menampilkan daftar barang yang siap dipilih	[v] Sesuai [] Tidak Sesuai
4	Tambah data pengadaan barang kemudian memilih barang	Data barang akan tampil di tabel untuk di inputkan	
5	Memasukkan jumlah barang kurang dari satu pada saat <i>input</i> pengadaan barang	Sistem akan menolak dan meminta agar jumlah barang yang diinputkan minimal adalah satu.	[v] Sesuai [] Tidak Sesuai
6	Menyimpan data yang sudah sesuai dengan prosedur	Sistem menyimpan data dan menampilkan pesan berhasil	[v] Sesuai [] Tidak Sesuai

Tabel 12. Skenario Pengujian *Blackbox Testing* pada Fitur Penjualan barang

No	Skenario pengujian	Hasil yang Diharapkan	Hasil
1	Tambah data penjualan barang dan simpan tanpa memasukkan data <i>customer</i>	Sistem akan menolak dan meminta memasukkan data <i>customer</i>	[v] Sesuai [] Tidak Sesuai
2	Tambah data penjualan barang dan simpan tanpa memasukkan data barang	Sistem akan melakukan proses dan menampilkan pesan “gagal <i>input</i> data”	[v] Sesuai [] Tidak Sesuai
3	Tambah data penjualan barang kemudian cari dan memilih barang	Data barang akan tampil di tabel untuk di inputkan	[v] Sesuai [] Tidak Sesuai
4	Memasukkan jumlah barang kurang dari satu pada saat <i>input</i> penjualan barang	Sistem akan menolak dan meminta agar jumlah barang yang diinputkan minimal adalah satu.	[v] Sesuai [] Tidak Sesuai
5	Menyimpan data yang sudah sesuai dengan prosedur	Sistem menyimpan data dan menawarkan untuk mencetak nota transaksi	[v] Sesuai [] Tidak Sesuai
6	Menginputkan data penjualan dengan jumlah bayar sebesar nol rupiah dengan data <i>customer</i> terdaftar	Sistem akan otomatis mencatat data piutang <i>customer</i> dan mengirimkan notifikasi <i>e-mail</i> kepada	[v] Sesuai [] Tidak Sesuai

No	Skenario pengujian	Hasil yang Diharapkan	Hasil
		pemilik bahwa ada <i>customer</i> yang melakukan piutang	
7	Menginputkan data penjualan dengan jumlah bayar sebesar nol rupiah dengan data <i>customer</i> tidak terdaftar	Sistem akan menolak dan memunculkan notifikasi bahwa <i>customer</i> tidak terdaftar tidak dapat memiliki piutang	[v] Sesuai [] Tidak Sesuai

Tabel 13. Skenario Pengujian *Blackbox Testing* pada Fitur Piutang

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil
1	Melakukan pembayaran piutang	Sistem menyimpan data pembayaran dan mengurangi otomatis jumlah piutang	[v] Sesuai [] Tidak Sesuai
2	Memberikan notifikasi piutangan melalui <i>whatsapp</i> dengan mengklik kirim notifikasi	Sistem mengarahkan ke <i>whatsapp web</i> dengan pesan otomatis untuk memberikan notifikasi piutang	[v] Sesuai [] Tidak Sesuai

Tabel 14. Skenario Pengujian *Blackbox Testing* pada Fitur Laporan

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil
1	Mengakses menu laporan stok dan klik cetak laporan	Sistem akan menampilkan laporan stok dan dapat melakukan cetak kedalam format pdf	[v] Sesuai [] Tidak Sesuai
2	Mengakses menu laporan laba penjualan dan cetak laporan	Sistem akan menampilkan laporan laba penjualan dan dapat melakukan cetak kedalam format pdf	[v] Sesuai [] Tidak Sesuai

I. Pengujian Aplikasi Dengan *Usability Testing*

Untuk mendapatkan *feedback* dari pengguna dilakukan dengan menggunakan metode *usability testing*. Pengujian *usability* dibantu oleh 31 responden yang diambil secara random, terdiri mahasiswa TI, mahasiswa NON-TI dan masyarakat umum yang dimininta untuk mencoba mengoperasikan sistem sesuai dengan skenario yang telah dibuat.

Dari hasil *usability testing* yang telah dilakukan pada pertanyaan ke 1, 65% menjawab sangat mudah, 32% mudah, 3% cukup mudah dan 0% menjawab tidak mudah. Pada pertanyaan ke 2, 34% menjawab sangat mudah, 42% mudah, 16% cukup mudah dan 6% menjawab tidak mudah. Pada pertanyaan ke 3, 61% menjawab sangat mudah, 29% mudah, 10% cukup mudah dan 0% menjawab tidak mudah. Pada pertanyaan ke 4, 48% menjawab sangat mudah, 35% mudah, 13% cukup mudah dan 3% menjawab tidak mudah. Pada pertanyaan ke 5, 58% menjawab sangat mudah, 32% mudah, 6% cukup mudah dan 3% menjawab tidak mudah. Pada pertanyaan ke 6, 55% menjawab sangat mudah, 32% mudah, 13% cukup mudah dan 0% menjawab tidak mudah. Pada pertanyaan ke 7, 35% menjawab sangat mudah, 45% mudah, 13% cukup mudah dan 6% menjawab tidak mudah. Pada pertanyaan ke 8, 52% menjawab sangat mudah, 39% mudah, 6% cukup mudah dan 3% menjawab tidak mudah. Pada pertanyaan ke 9, 48% menjawab sangat mudah, 42% mudah, 10% cukup mudah dan 0% menjawab tidak mudah. Pada pertanyaan ke 10, 48% menjawab sangat mudah, 39% mudah, 10% cukup mudah dan 3% menjawab tidak mudah. Pada pertanyaan ke 11, 52% menjawab sangat mudah, 39% mudah, 10% cukup mudah dan 0% menjawab tidak mudah. Pada pertanyaan ke 12, 45% menjawab sangat mudah, 39% mudah, 6% cukup mudah dan 10% menjawab tidak mudah. Pada pertanyaan ke 13, 45% menjawab sangat mudah, 39% mudah, 16% cukup mudah dan 0% menjawab tidak mudah. Pada pertanyaan ke 14, 35% menjawab sangat mudah, 35% mudah, 26% cukup mudah

dan 3% menjawab tidak mudah. Pada pertanyaan ke 15, 39% menjawab sangat mudah, 45% mudah, 48% cukup mudah dan 0% menjawab tidak mudah.

Dari hasil pengujian yang telah dikumpulkan dapat disimpulkan bahwa sistem telah layak digunakan dan dioperasikan dengan nilai rata-rata 48% menjawab sangat mudah, 38% menjawab mudah, 14 persen menjawab cukup mudah dan 3% menjawab tidak mudah. Namun perlu diperhatikan pada pertanyaan ke 12, yang berkaitan dengan fitur piutang pelanggan mendapat nilai paling besar pada jawaban tidak mudah, yaitu 10%, hal ini menunjukkan bahwa perlu adanya perbaikan pada fitur tersebut.

J. Menghitung Nilai Efisiensi

Untuk menghitung nilai efisiensi fitur notifikasi piutang dilakukan dengan melakukan pengamatan terhadap proses penagihan piutang mulai dari sebelum adanya aplikasi sampai dengan adanya aplikasi. Untuk mengukur nilai efisiensi ini dilakukan dengan mengumpulkan data piutang pada bulan Juni – Juli 2021 yaitu sebelum adanya aplikasi, dimana terdapat 41 data piutang dan 31 piutang dibayarkan sesuai dengan tanggal jatuh tempo, sehingga nilai persentase piutang dapat dibayar sesuai dengan tanggal jatuh tempo pada periode ini dapat dihitung dengan rumus dibawah:

$$\text{Persentase Periode 1} = \frac{\text{jumlah piutang tertagih}}{\text{jumlah data piutang}} \times 100\% \quad (3)$$

$$\text{Persentase Periode 1} = \frac{31}{41} \times 100\% \quad (4)$$

$$\text{Persentase Periode 1} = 75.61\% \quad (5)$$

Jadi persentase piutang yang dapat dibayar sesuai dengan tanggal jatuh tempo pada periode 1 (sebelum adanya aplikasi notifikasi piutang) yaitu sebesar 75.61%.

Pada pengamatan kedua setelah adanya aplikasi, dilakukan pengumpulan data piutang pada bulan November-Desember 2021, terdapat 34 data piutang pelanggan dan 32 data piutang dapat dibayarkan sesuai dengan tanggal jatuh tempo, sehingga nilai persentase piutang dapat dibayar sesuai dengan tanggal jatuh tempo pada periode ke 2 dapat dihitung dengan rumus dibawah:

$$\text{Persentase Periode 2} = \frac{\text{jumlah piutang tertagih}}{\text{jumlah data piutang}} \times 100\% \quad (6)$$

$$\text{Persentase Periode 2} = \frac{32}{34} \times 100\% \quad (7)$$

$$\text{Persentase Periode 2} = 94.12\% \quad (8)$$

Jadi persentase piutang yang dapat dibayar sesuai dengan tanggal jatuh tempo pada periode 2 (setelah adanya aplikasi notifikasi piutang) yaitu sebesar 94.12%. Kemudian untuk menghitung nilai efisiensi dari fitur notifikasi ini dapat menggunakan rumus sebagai berikut:

$$\text{Nilai efisiensi} = \text{persentase periode 2} - \text{persentase periode 1} \quad (9)$$

$$\text{Nilai efisiensi} = 94.12\% - 75.61\% \quad (10)$$

$$\text{Nilai efisiensi} = 18.51\% \quad (11)$$

Dari hasil pengamatan ini, maka dapat disimpulkan adanya peningkatan pembayaran piutang sesuai dengan jatuh tempo sebesar 18.51% setelah adanya aplikasi dengan fitur notifikasi piutang melalui *whatsapp*.

5. KESIMPULAN

Berdasarkan penelitian dan hasil analisa yang telah dilakukan, dapat disimpulkan proses perancangan aplikasi menggunakan metode *waterfall* sudah sesuai dengan kebutuhan, perancangan disesuaikan dengan alur dari metode *waterfall* yaitu analisis kebutuhan, membuat *planning* untuk pembuatan aplikasi, melakukan modeling sistem dengan membuat ERD, *usecase diagram*, *activity diagram*, dan *mockup*, serta melakukan pengkodean dan *testing* dengan menggunakan *blackbox testing*

dan *usability testing*. Selain itu, aplikasi *Point of Sales* (P.O.S) yang dibangun sudah sesuai dengan kebutuhan dan dapat digunakan kegiatan operasional baik pengadaan barang, penjualan barang dan penagihan piutang melalui *whatsapp* di Toko Fendrymart. Dengan hasil dari *blackbox testing* yang telah dilakukan yaitu semua kondisi pengujian menghasilkan nilai yang sesuai dengan kondisi yang diharapkan, dan hasil dari *usability testing* yang telah dilakukan menunjukkan bahwa sistem sudah sangat layak untuk dapat digunakan.

Dengan adanya aplikasi *Point of Sales* (P.O.S) dengan fitur notifikasi piutang juga telah membantu meningkat terbayarnya piutang pelanggan sesuai dengan jatuh tempo, hal ini dibuktikan dengan meningkatnya jumlah pelanggan yang dapat membayar piutang yaitu pada bulan November 2021 atau bulan kedua setelah aplikasi diimplementasikan, pelanggan yang membayar piutang sesuai dengan jatuh tempo meningkat sebesar 18.51% dari sebelum adanya aplikasi.

Adapun saran yang dapat dilakukan untuk penelitian selanjutnya guna memberikan hasil yang lebih maksimal adalah sebagai berikut:

1. Perlu adanya perbaikan pada fitur piutang pelanggan karena berdasarkan hasil *usability testing* memiliki nilai jawaban “Tidak Mudah” paling tinggi. Hal ini menandakan bahwa fitur tersebut masih cukup sulit untuk digunakan oleh pengguna.
2. Pengembangan sistem dapat dilakukan dengan menambahkan beberapa fitur yang nantinya mungkin dibutuhkan sesuai dengan perkembangan teknologi yang ada.
3. Perlu adanya pengembangan sebuah sistem lanjutan untuk memanfaatkan data yang ada di aplikasi P.O.S, seperti halnya sistem peramalan penjualan yang nantinya akan sangat berguna dalam memperkirakan jumlah barang yang harus diadakan pada periode selanjutnya.

6. DAFTAR PUSTAKA

- Anhar. (2010). *Panduan Menguasai PHP dan MYSQL secara Otodidak*. Jakarta: PT Transmedia.
- Az-zahra, H. M., Pinandito, A., & Tolle, H. (2015). Usability Evaluation of Mobile Application in Culinary Recommendation System, *Conference: Wireless and Mobile (APWiMob)*, hal 89–94.
- Enterprise, J. (2017). *Mengenal Pemrograman PHP7 Database untuk Pemula*. Jakarta: PT Elex Media Komputindo.
- Enterprise, J. (2018). *Pemrograman Database dengan Python dan MySQL*. Jakarta: PT Elex Media Komputindo.
- Fatkhudin, A. (2016). Sistem Informasi Penjualan Dan Pembelian Pada Toko Elektronik Lubada Jaya Kajen Dengan Menggunakan Java, *Jurnal Digit: Digital of Information Technology*, 6(1), 23–36.
- Faulkner, L. (2003). *Beyond the five-user assumption: Benefits of increased sample sizes in usability testing*, 35(3), 379–383.
- Hidayat, D. D., Wikusna, W., & Telnoni, P. A. (2017). Aplikasi Inventory Barang, Pengelolaan Data Hutang-piutang Dagang Dan Prediksi Kebutuhan Barang Dengan Metode Single Exponential Smoothing (Studi Kasus: Usaha Dagang Hada Putra), 3(2), 653–660.
- Nielsen, J. (2000). Why You Only Need to Test with 5 Users, 1–5.
- Pressman, R. S. (2015). *Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku*. Yogyakarta: Andi.
- Sugianto, Y., & Tjandra, S. (2016). Aplikasi Point of Sale Pada Toko Retail Dengan Menggunakan Dynamic Software Development Method, 8(1), 1–8.
- Sunyoto, A. (2007). *AJAX Membangun Web dengan Teknologi Asynchronous JavaScript dan XML*. Yogyakarta: CV ANDI.
- Sutarman. (2009). *Pengantar Teknologi Informasi*. Jakarta: PT Bumi Aksara.
- Tahir, T. Bin, Rais, M., & HS, M. A. (2019). Aplikasi Point of Sales Menggunakan Framework Laravel Point OF Sales Application using Laravel Framework, 2(2), 55–60.
- Yuliyana, T., Arthana, I. K. R., & Agustini, K. (2019). Usability Testing Pada Aplikasi Potwis, 8(1), 12–22.