**RANCANG BANGUN APLIKASI PENDATAAN TRANSAKSI JASA PENYEBERANGAN DENGAN FRAMEWORK REACT NATIVE BERBASIS ANDROID (STUDI KASUS: B.C. BAKAUHENI)**

*The Design of Crossing Service Transaction Data Collection Application for Framework Service with React Native Framework Based on Android (Case Study: Bc. Bakauheni)*

Usulan penelitian untuk skripsi S-1

Diusulkan oleh:

Rahmad Suryadi

17312031

****

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK ILMU KOMPUTER**

**UNIVERSITAS TEKNOKRAT INDONESIA**

**BANDAR LAMPUNG**

**2022**

# KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT, karena atas berkat rahmat-Nya, penulis dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar sarjana pada Program Studi S1 Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia. Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan laporan ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Bapak Dr. H. M. Nasrullah Yusuf, S.E., M.B.A., selaku Rektor Universitas Teknokrat Indonesia
2. Bapak H. Mahathir Muhammad, S.E., M.M., selaku Dekan Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia.
3. Bapak Andi Nurkholis, S.Kom., M.Kom. selaku pembimbing yang telah meluangkan waktu untuk membimbing penulis menyelesaikan skripsi ini.
4. Bapak/Ibu selaku dosen penguji.

Akhir kata, penulis berharap semoga Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu dan semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Bandar Lampung, Mei 2022

Penulis

# DAFTAR ISI

**Hal.**

[KATA PENGANTAR i](#_Toc103335626)

[DAFTAR ISI ii](#_Toc103335627)

[DAFTAR TABEL iv](#_Toc103335628)

[DAFTAR GAMBAR v](#_Toc103335629)

[1 PENDAHULUAN 1](#_Toc103335630)

[1.1 Latar Belakang 1](#_Toc103335631)

[1.2 Rumusan Masalah 3](#_Toc103335632)

[1.3 Batasan Masalah 3](#_Toc103335633)

[1.4. Tujuan Penelitian 3](#_Toc103335634)

[1.5 Manfaat Penelitian 3](#_Toc103335635)

[2 LANDASAN TEORI 4](#_Toc103335636)

[2.1 Tinjauan Pustaka 4](#_Toc103335637)

[2.2 Pengertian Android 9](#_Toc103335638)

[2.3 Framework 9](#_Toc103335639)

[2.4 React Native 10](#_Toc103335640)

[2.5 QR-code 10](#_Toc103335641)

[2.6 Metode Pengembangan Sistem 11](#_Toc103335642)

[2.7 *Unified Modeling Language* (UML) 13](#_Toc103335643)

[2.7.1 *Use Case* 13](#_Toc103335644)

[2.7.2 *Activity Diagram* 14](#_Toc103335645)

[2.7.3 *Class* *diagram* 15](#_Toc103335646)

[2.8 *Class Responsibility Collaborator Card (CRC Card).* 16](#_Toc103335647)

[2.9 Firebase 17](#_Toc103335648)

[2.10 Pengujian ISO 25010 17](#_Toc103335649)

[2.11 Pengertian Figma 19](#_Toc103335650)

[3 METODOLOGI PENELITIAN 20](#_Toc103335651)

[3.1 Kerangka Penelitian 20](#_Toc103335652)

[3.2 *Planning* 21](#_Toc103335653)

[3.2.1 *User Stories* 21](#_Toc103335654)

[3.3.2 *Values* 23](#_Toc103335655)

[3.3.3 *Acceptance Test Criteria* 26](#_Toc103335656)

[3.3.4 *Iteration Plan* 30](#_Toc103335657)

[3.3.4.1 *Iterations* 31](#_Toc103335658)

[3.3.4.2. Analisis Kebutuhan Sistem 31](#_Toc103335659)

[3.4. *Design* 33](#_Toc103335660)

[3.4.1 *Use Case* *Diagram* 33](#_Toc103335661)

[3.4.2 *Activity Diagram* 36](#_Toc103335662)

[3.4.3 *Class Diagram* 51](#_Toc103335663)

[3.4.1 *Class Responsibility Collaborator Card* (*CRC Card*). 51](#_Toc103335664)

[3.4.2 *Prototype* *Design* 54](#_Toc103335665)

[3.5 Jadwal Penelitian 61](#_Toc103335666)

[DAFTAR PUSTAKA 62](#_Toc103335667)

[LAMPIRAN 64](#_Toc103335668)

# DAFTAR TABEL

[Tabel 2.1 Tinjauan Pustaka 4](#_Toc103339270)

[Tabel 2.2 Simbol-Simbol *Use Case* 14](#_Toc103339271)

[Tabel 2.3 Simbol – Simbol *Activity Diagram* 15](#_Toc103339272)

[Tabel 2.4 Simbol – Simbol *Class Diagram* 16](#_Toc103339273)

[Tabel 2.5 *Class Responsibility Collaborator Card (CRC Card)* 17](#_Toc103339274)

[Tabel 3.1 *User Stories* 21](#_Toc103339275)

[Tabel 3.2 *User Stories* (lanjutan 1) 22](#_Toc103339276)

[Tabel 3.3 *User Stories* (lanjutan 2) 23](#_Toc103339277)

[Tabel 3.4 *Value User Stories* 24](#_Toc103339278)

[Tabel 3.5 *Value* *User Stories* (Lanjutan 1) 25](#_Toc103339279)

[Tabel 3.6 *Value* *User Stories* (Lanjutan 2) 26](#_Toc103339280)

[Tabel 3.7 Kerangka Pengujian *Co-existence* 28](#_Toc103339281)

[Tabel 3.8 Kerangka Pengujian *Operating System Dan Device* 29](#_Toc103339282)

[Tabel 3.9 Rencana Iterasi 30](#_Toc103339283)

[Tabel 3.10 Contoh Tahapan Iterasi 31](#_Toc103339284)

[Tabel 3.11 Penjelasan *Use Case* 35](#_Toc103339285)

[Tabel 3.12 *CRC Card* AUTH 52](#_Toc103339286)

[Tabel 3.13 *CRC Card* Pegawai 52](#_Toc103339287)

[Tabel 3.14 *CRC Card* Pimpinan 52](#_Toc103339288)

[Tabel 3.15 *CRC Card* Transaksi Penyeberangan 53](#_Toc103339289)

[Tabel 3.16 *CRC Card* Kendaraan 53](#_Toc103339290)

[Tabel 3.17 *CRC Card* rekap 53](#_Toc103339291)

[Tabel 3.18 *CRC Card* Transaksi Kreditur 53](#_Toc103339292)

# DAFTAR GAMBAR

[Gambar 2.1 Logo android 9](#_Toc102022303)

[Gambar 2.2 Logo React Native 10](#_Toc102022304)

[Gambar 2.3 Metode Extreme Programming 11](#_Toc102022305)

[Gambar 2.4 Model ISO 25010 18](#_Toc102022306)

[Gambar 3.1 Kerangka Penelitian 20](#_Toc102022307)

[Gambar 3.2 *Use Case* Diagram Aplikasi Pendataan 34](#_Toc102022308)

[Gambar 3.3 *Activity Diagram* Mengelola Data Transaksi 36](#_Toc102022309)

[Gambar 3.4 *Activity Diagram* Menampilkan Data Transaksi 37](#_Toc102022310)

[Gambar 3.5 *Activity Diagram* Memfilter Data Transaksi 38](#_Toc102022311)

[Gambar 3.6 *Activity Diagram* Pencarian Data Transaksi 39](#_Toc102022312)

[Gambar 3.7 *Activity Diagram* Menampilkan Data Kendaraan dan Pegawai 40](#_Toc102022313)

[Gambar 3.8 *Activity Diagram* Menglola Data Kendaraan dan Pegawai 41](#_Toc102022314)

[Gambar 3.9 *Activity Diagram* Melakukan Transaksi Penyeberangan 42](#_Toc102022315)

[Gambar 3.10 *Activity Diagram* Melakukan Transaksi Kredit 43](#_Toc102022316)

[Gambar 3.11 *Activity Diagram* Menampilkan Rekap Harian 44](#_Toc102022317)

[Gambar 3.12 *Activity diagram* Menampilkan dan mengelola stok 45](#_Toc102022318)

[Gambar 3.13 *Activity Diagram* Menampilkan dan Mengelola Harga Pelabuhan 46](#_Toc102022319)

[Gambar 3.14 *Activity Diagram* Menambahkan Pengeluaran 47](#_Toc102022320)

[Gambar 3.15 *Activity Diagram* Filter Data Kendaraan 48](#_Toc102022321)

[Gambar 3.16 *Activity Diagram* Pencarian Data Kendaraan 49](#_Toc102022322)

[Gambar 3.17 Activity Diagram Simpan QR-Code 50](#_Toc102022323)

[Gambar 3.18 *Class diagram* aplikasi pendataan transaksi 51](#_Toc102022324)

[Gambar 3.19 Tampilan *Login* 54](#_Toc102022325)

[Gambar 3.20 Halaman Home 55](#_Toc102022326)

[Gambar 3.21 Halaman Data Transaksi 55](#_Toc102022327)

[Gambar 3.22 Halaman *Scanner* 56](#_Toc102022328)

[Gambar 3.23 Halaman Transaksi Penyeberanagan 57](#_Toc102022329)

[Gambar 3.24 Halaman Transaksi *Kredit* 57](#_Toc102022330)

[Gambar 3.25 Halaman Data Kendaraan Dan Pegawai 58](#_Toc102022331)

[Gambar 3.26 Halaman Harga Pelabuhan 59](#_Toc102022332)

[Gambar 3.27 Halaman Tambah Pengeluaran Harian 59](#_Toc102022333)

[Gambar 3.28 Halaman Rekap Harian 60](#_Toc102022334)

[Gambar 3.29 Halaman Edit Saldo 61](#_Toc102022335)

# 

# 1 PENDAHULUAN

## 1.1 Latar Belakang

Penggunaan komputer merupakan suatu hal yang penting di berbagai kehidupan dengan berbagai komunitas saat ini. Sehingga dapat disimpulkan bahwa komputer menjadi media informasi yang dapat membantu suatu pekerjaan yang rumit. Bukan hanya pada lingkungan kerja, komputer juga dapat memudahkan pengguna dalam kehidupan sehari-harinya. Seiring dalam perkembangan teknologi, kemajuan perangkat lunak dan perangkat keras berjalan dengan baik, perkembangan ini berpengaruh dalam hal pembuatan aplikasi yang semakin lebih mudah dan cepat. Hal ini pun membuat meningkatnya pemanfaatan komputer pada lingkungan masyarakat dan tentunya pada lingkungan kerja. (Loveri 2018).

Jasa merupakan suatu aktivitas bukan berupa benda yang ditawarkan oleh penyedia jasa ke pihak lain (Gultom dkk., 2014). Berdasarkan observasi peneliti pada jasa penyeberangan, layanan yang ditawarkan jasa berupa pembayaran tiket secara *online* dan juga membantu menangani kendala-kendala yang terjadi pada truk yang menghambat perjalanan. Pencatatan transaksi pada jasa penyeberangan B.C. Bakauheni masih dilakukan dengan cara konvensional yang membuat pencatatan tidak efisien yang mengharuskan pegawai mencatat data kendaraan di setiap transaksinya, rentan terjadinya kesalahan dalam pencatatan data, pencarian data transaksi yang sulit, dan kekhawatiran terjadi hilangnya data. Sebelumnya pencatatan pernah dilakukan dengan menggunakan komputer, hal ini dirasa kurang efisien karena pegawai harus mencatat pada buku catatan saat di lapangan dan kemudian melakukan pencatatan lagi setelah kembali ke kantor dan keluhan tagihan listrik yang meningkat.

Pada Penelitian sebelumnya telah banyak mengembangkan aplikasi untuk menangani permasalahan pendataan dengan memanfaatkan aplikasi *mobil*e. Pertama, pengembangan aplikasi pengelolaan uang berbasis android dengan melakukan pencatatan terperinci dan menghitung pemasukan dan pengeluaran untuk mengontrol keuangan pengguna (Juhardi and Khairullah 2019) Selanjutnya pengembangan aplikasi pendataan pada CV. Nana Beautyskin dapat berguna dalam pendataan produk, pencatatan transaksi pencatatan bonus pegawai dan keefektifan dalam pembuatan struk (Saputra, dkk. 2021). Selanjutnya perancangan aplikasi kasir *point of sales* berbasis android untuk gerai makanan yang dapat memudahkan dalam perekapan harian (Iskandar & Abdurrahman, 2020). Selanjutnya pengembangan aplikasi pengelolaan pinjaman berbasis *mobile* pada koperasi PKK sejahtera sukabumi yang membuat pembukuan simpan pinjam dan pengajuan pinjaman menjadi lebih efektif (Maranti dkk., 2018). Selanjutnya pengembangan sistem pencatatan keuangan toko berbasis android meningkatkan efisiensi pengelolaan keuangan (Romadony dkk., 2019).

Pada penelitian terdahulu yang telah dipaparkan, mendapatkan hasil bahwasanya penggunaan aplikasi berbasis android dapat membantu dalam pencatatan dan pengelolaan data transaksi sehingga proses bisnis dapat berjalan lebih efisien. Selain itu dengan adanya sistem dengan *database* *online* pengguna dapat melihat hasil transaksi kapanpun dan dimanapun. Pengembangan yang dapat dilakukan dari penelitian sebelumnya adalah menambahkan fitur pemindai *QR-code* dalam melakukan transaksinya, dan juga menampilkan secara *realtime* hasil transaksi yang telah terjadi.

Penelitian ini bertujuan mengembangkan aplikasi pendataan transaksi dengan studi kasus jasa penyeberangan B.C. Bakauheni. aplikasi dikembangkan menggunakan metode *extreme programming* dengan empat tahapan yaitu *planning*, *design*, *coding* dan *testing*. Metode ini dipilih karena fleksibilitas yang tinggi terhadap *user* dan klien dalam pengembangannya. sehingga mampu membuat aplikasi yang sesuai dengan fungsi yang dibutuhkan klien (Widodo 2008). Cakupan fungsi yang dimaksud yaitu mencakup pencatatan data dengan memindai *QR-code*, pencarian data dengan mudah, rekap harian dan menampilkan data transaksi secara *realtime*. Pada tahap *coding* dalam pengembangan aplikasi menggunakan bahasa pemrograman JavaScript dengan *framework* React Native dan pengujian aplikasi menggunakan ISO 25010. Aplikasi ini diharapkan dapat menyelesaikan masalah-masalah pendataan yang terjadi dan dapat meningkatkan kinerja pegawai dalam setiap transaksinya sehingga proses bisnis dapat berjalan lebih efisien.

## 1.2 Rumusan Masalah

Berdasarkan dari pemaparan yang telah penulis jelaskan diatas, maka rumusan masalah yang diambil penulis adalah bagaimana mengembangkan sebuah aplikasi berbasis android yang dapat membantu jasa penyeberangan dalam hal pencatatan, pencarian, penyimpanan, dan perekapan data transaksi?

## 1.3 Batasan Masalah

Dalam penelitian rancang bangun aplikasi pendataan transaksi pada jasa penyeberangan ini ditentukan batasan masalah sebagai berikut:

1. Aplikasi hanya melakukan pendataan transaksi, pencarian, dan perekapan data transaksi pada jasa penyeberangan B.C. Bakauheni.
2. Aplikasi di bangun menggunakan *framework* React Native dan *database* Firebase.

## 1.4. Tujuan Penelitian

Penelitian ini bertujuan untuk merancang dan mengembangkan sebuah aplikasi pendataan transaksi untuk jasa penyeberangan B.C Bakauheni yang dapat melakukan pencatatan, pencarian, penyimpanan, dan perekapan data transaksi untuk meningkatkan efisiensi dalam proses bisnisnya.

## 1.5 Manfaat Penelitian

Adapun manfaat yang diperoleh secara langsung maupun tidak langsung dari penelitian ilmiah ini adalah sebagai berikut:

1. Bagi penulis dapat meningkatkan kemampuan menyelesaikan masalah di suatu kasus dengan memanfaatkan teknologi dan juga meningkatkan kemampuan pengembangan perangkat lunak berbasis android khususnya menggunakan *framework* react native.
2. Bagi jasa penyeberangan B.C. Bakauheni manfaat penelitian ini diharap dapat memudahkan jasa penyeberangan dalam menyelesaikan masalah - masalah pendaataan yang ada dan dapat membantu meningkatkan efisiensi dalam proses bisnisnya.

# 

# 2 LANDASAN TEORI

## 2.1 Tinjauan Pustaka

Pada penelitian ini menggunakan tinjauan Pustaka dari penelitian sebelumnya yang nantinya akan digunakan sebagai pendukung penelitian, tinjauan Pustaka yang digunakan penulis dalam penelitian ini dapat dilihat pada Gambar 2.1.

Tabel 2.1 Tinjauan Pustaka

|  |  |  |
| --- | --- | --- |
| **Nomor** | **Detail Jurnal** | |
| **1** | Judul | Sistem Pencatatan dan Pengelolaan Keuangan Pada Aplikasi Manajemen Keuangan E-Dompet Berbasis Android |
| Tahun Terbit | 2019 |
| Penulis | Ujang Juhardi, Khairullah |
| Metode Penelitian | Waterfall |
| Analisis Masalah | Dalam penelitian ini disebutkan bahwa pencatatan pemasukan dan pengeluaran uang pribadi pada umumnya masih dilakukan dengan cara konvensional. hal tersebut sangat tidak efisien, jumlah transaksi yang tidak sedikit di setiap harinya membuat pencatatan, perhitungan, dan pembuatan laporan membutuhkan waktu yang tidak sedikit, maka dibutuhkan aplikasi pengelolaan keuang yang dapat digunakan seseorang untuk mempermudah pencatatan laporan keuangan. |
| Hasil | Hasil dari penelitian ini adalah sebuah Aplikasi Manajemen Keuangan E-Dompet Berbasis Android yang dapat melakukan pencatatan laporan keuangan secara rinci sehingga memudahkan seseorang melihat rincian laporan keuangannya. |
| Kelebihan | Penyimpanan data pada plikasi ini bersifat *offline* sehingga pengguna tidak harus menggunakan internet setiap menggunakan aplikasi. |
| Kekurangan | Karena penyimpanan yang digunakan *offline* atau local, jika pengguna menggunakan *device* yang berbeda tidak dapat membuka catatan yang sudah ada sebelumnya. |
| **2** | Judul | Perancangan Aplikasi Kasir *Point of Sales* Berbasis Android Menggunakan Metode Rapid Application Development Untuk Usaha Retail |
| Tahun Terbit | 2020 |
| Penulis | Iskandar, Umar Tsani Abdurrahman |
| Metode Penelitian | Rapid Application Development |
| Analisis Masalah | Dalam penelitian ini disebutkan bahwa kegiatan transaksi pada gerai pada umumnya dilakukan belum menggunakan sistem komputer, sehingga banyak menemukan kendala, yaitu dalam proses penghitungan sering terjadi kesalahan hitung dan prose rekap transaksi yang relatif lama karena harus menghitung ulang setelah gerai selesai berjualan sehingga membutuhkan waktu extra. |
| Hasil | Sebuah aplikasi kasir *point of sales* berbasis android yang dapat digunakan gerai dalam melakukan pencatatan transaksi dan dapat mempermudah kegiatan transaksinya. |
| Kelebihan | Aplikasi dapat terhubung dengan printer melalui bluetooth sehingga dapat Melakukan pencetakan struk penjualan |
| Kekurangan | *Interface* pada aplikasi dirasa kurang menarik dan pemilihan *color* *palette* yang kurang selaras atau terlalu kontras. |
| **3** | Judul | Rancang Bangun Aplikasi Pengelolaan Pinjaman Koperasi Berbasis Mobile Pada Koperasi Pkk Sejahtera Sukabumi |
| Tahun Terbit | 2018 |
| Penulis | Ovi Sovia Maranti, Lis Saumi Ramdhani, Rusli Nugraha, Khairul Rizal |
| Metode Penelitian | Waterfall |
| Analisis Masalah | Dalam penelitian ini disebutkan bahwa jumlah anggota yang cukup banyak pada koperasi PKK Sejahtera Sukabumi mengakibatkan sistem pelaporan, pembukuan simpan pinjam dan pengajuan pinjaman menjadi kurang efektif, karena masih menggunakan cara konvensional, terutama dalam pengajuan pinjam, hal tersebut menyebabkan perputaran modal berjalan lambat, oleh karana itu dibutuhkanya sistem terkomputerisasi agar anggota dapat mengajukan pinjaman kapanpun dan dimana saja. |
| Hasil | Hasil dari penelitian ini yaitu aplikasi pengelolaan pinjaman koperasi berbasis *mobile* yang dibuat terintegrasi dengan web server supaya memudahkan dalam pengelolaan datanya. |
| Kelebihan | Penyimpanan data pinjaman yang ditegrasikan dengan web server dapat memudahkan dalam pengolahan data. |
| Kekurangan | Halaman admin dan anggota dibuat pada platform yang berbeda sehingga tidak efisien dan user interface yang dirasa masih kurang baik. |
| **4** | Judul | Pengembangan Aplikasi Klinik Kecantikan sebagai Pengelola Transaksi berbasis Android menggunakan Metode Prototype (Studi Kasus: Klinik Kecantikan CV Nana Beautyskin). |
| Tahun Terbit | 2021 |
| Penulis | Agung Dwi Saputra, Agi Putra Kharisma, Lutfi Fanani. |
| Metode Penelitian | Prototype |
| Analisis Masalah | Dalam penelitian ini disebutkan bahwa kegiatan pencatatan transaksi yang dilakukan pada CV. Nana Beauty Skin masih dilakukan dengan cara tradisional, metode pencatatan tersebut masih dirasa efektif saat kegiatan transaksi masih sedikit, namun saat kegiatan transaksi yang mulai banyak hal tersebut memiliki beberapa resiko dalam pelaksanaanya. Sehingga dibutuhkan sistem pengelolaan data transaksi yang lebih baik dari sebelumnya. |
| Hasil | Aplikasi klinik kecantikan sebagai pengelola transaksi berbasis Android, yang dapat digunakan untuk pencatatan setiap transaksi, pencatatan bonus pegawai pada setiap pelayanan jasa, dan pembuatan struk yang lebih efektif. |
| Kelebihan | Penyimpanan yang sudah secara online, UI yang menarik dan pengujian aplikasi yang mendapat nilai yang bagus. |
| Kekurangan | Pengujian kompatibilitas aplikasi yang tidak dilakukan pada *real device*. |
| **5** | Judul | Sistem Pencatatan Keuangan Toko Berbasis Android |
| Tahun Terbit | 2019 |
| Penulis | Willi Alham Romadony, Muhammad Ardianto, Wisnu Kartiko Arie Pangestu, Didih Rizki Chandranegara, Wildan Suharso |
| Metode Penelitian | Waterfall |
| Analisis Masalah | pengelolaan keuangan dengan cara konvensional dirasa kurang efektif dan juga memerlukan waktu yang lebih untuk mencatat pengeluaran yang telah dilakukan saat itu sehingga terkadang lupa untuk melakukan pencatatan pengeluaran kecil yang telah dilakukan. |
| Hasil | aplikasi pencatatan keuangan berbasis android yang dapat digunakan dalam mengelola keuangan. |
| Kelebihan | Terdapat rekap harian yang sudah langsung terekap dan dapat di cetak |
| Kekurangan | Tidak disebutkan penyimpanan yang digunakan dan juga UI yang masih kurang menarik. |

Kesimpulan yang dapat diambil dari tinjauan literatur diatas yaitu masalah yang muncul karena pengelolaan data transaksi yang masih menggunakan cara konvensional dapat teratasi dengan sistem yang terkomputerisasi dengan memanfaatkan aplikasi *mobil*e berbasis android sehingga proses transaksi yang dilakukan akan lebih efisien. Perbedaan yang terdapat dari tinjauan literatur dengan penelitian penulis yaitu pada metode pengembangan sistem yang dipakai, pada penelitian yang akan diteliti penulis menggunakan metode Extreme Programming (XP), karena fleksibilitas yang tinggi terhadap *user* dan klien dalam pengembangannya. sehingga mampu membuat aplikasi yang sesuai dengan fungsi yang dibutuhkan klien (Widodo 2008). Kemudian terdapat juga perbedaan dalam penggunaan teknologi yang digunakan penulis dalam mengembangkan aplikasi berbasis android, dalam penelitian ini penulis menggunakan *Framework* React Native. Pada aplikasi yang akan peneliti kembangkan juga memanfaatkan fitur *QR-code* *scanner* yang berguna untuk menginputkan data pelanggan yang membuat pegawai tidak harus menginputkan data pelanggan setiap transaksinya dan fitur lainnya yang akan ada pada aplikasi yaitu dapat menampilkan *history* transaksi secara *realtime* supaya pimpinan dapat mudah mengetahui dan memantau transaksi yang terjadi kapan saja dan dimana saja. Kemudian dalam pengujian sistem penulis menggunakan pengujian ISO 25010.

## 2.2 Pengertian Android

Android adalah sebuah *software* yang digunakan pada perangkat *mobile* yang mencakup sistem operasi, *middleware*, dan aplikasi kunci yang dirilis oleh Google.hal tersebut membuat android dapat mencakup keseluruhan dari suatu aplikasi, mulai dari pengembangan aplikasi sampai sistem operasi itu sendiri. Pengembangan aplikasi yang diterapkan pada android, menggunakan dasar bahasa pemrograman Java. Tapi secara sempit, Android biasanya mengacu pada sistem operasinya saja. Sistem operasi ini bersifat open source atau terbuka, sehingga pengembang bebas mengernbangkan atau mengembangkan aplikasi dengan biaya yang sedikit, dan pengembang dapat menjual aplikasi yang diciptakan tanpa ada lisensi ke produsen atau vendor tertentu. Kemudian pengembang diperbolehkan memodifikasi atau mengubah sistem operasi android. Dari beberapa faktor tersebut lah yang membuat android menjadi salah satu sistem operasi yang popular (Tim, 2015).



Gambar 2.1 Logo android

Sumber: (Dieter, 2019)

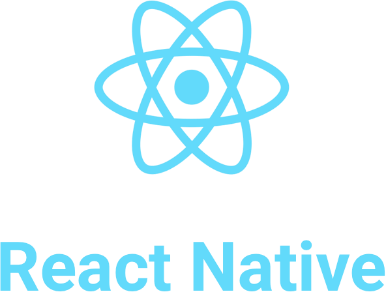
## 2.3 Framework

*Framework* merupakan sebuah kerangka kerja yang bertujuan untuk memudahkan programmer dalam pengembangan aplikasi dengan memanfaatkan library yang sudah terorganisir untuk dapat membuat suatu program lebih cepat (Solikin 2014). Dalam Bahasa Indonesia yaitu kerangka kerja dan dapat diartikan sebagai library (*class*) yang bisa diturunkan, atau dapat digunakan fungsinya oleh modul modul atau fungsi yang akan dikembangkan. (Cahyati and Murti 2018).

Berdasarkan definisi-definisi di atas dapat diambil kesimpulan bahwa framework adalah sistem yang terstruktur yang digunakan sebagai kerangka dalam mengembangkan sesuatu yang bertujuan untuk menyelesaikan suatu permasalahan atau isu – isu yang cukup kompleks lebih cepat.

## 2.4 React Native

React Native adalah salah satu *framework* yang digunakan untuk pengembangkan aplikasi *mobile* dan menggunakan Bahasa pemrograman JavaScript. Dengan menggunakan *framework* React Native, pengembang dapat merender *user interface* yang di peruntukan untuk digunakan pada platform iOS maupun Android. React Native juga merupakan *framework* yang bersifat *open source*, dan dalam penggunaannya dapat berjalan pada sistem operasi seperti Windows maupun macOS (Eisenman B. , 2016). dalam pengembanganya juga, 75% kode program dapat digunakan jika ingin mengembangkan aplikasi ke *platform* lain tanpa harus menulis ulang kode (Hansson and Vidhall 2016).

****

Gambar 2.2 Logo React Native

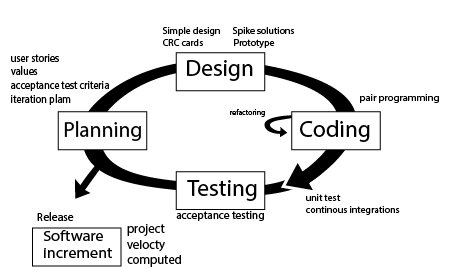
Sumber: (John3, 2019)

## 2.5 QR-code

QR *code* adalah jenis kode matriks atau kode batang dua dimensi dan dikembangkan oleh Denso Wave, salah satu divisi Denso Corporation yang merupakan perusahaan dari Jepang dan dipublikasikan pada tahun 1994. Sesuai dengan namanya, *quick response* atau respon cepat fungsi utama *QR-code* dapat dengan mudah dibaca oleh pemindai *QR-code*. tujuan adalah untuk diciptakannya *QR-code* yaitu untuk menyampaikan informasi dengan cepat dan mendapatkan respons yang cepat. Berbeda dengan kode batang, yang hanya menyimpan informasi secara *horizontal* *QR-code* dapat menampung informasi yang lebih banyak dari pada kode batang. (Ismail dkk., 2021).

## 2.6 Metode Pengembangan Sistem

*Extreme Programming* (XP)Jmerupakan metodologi yang digunakan untuk pengembangan perangkat lunak yang ditujukan dalam meningkatkan kualitas perangkat lunak terhadap perubahan serta kebutuhan pelanggan. Jenis pengembangan perangkat lunak semacam ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pro pemeriksaan dimana persyaratan pelanggan baru dapat diadopsi. Ada beberapa tahapan yang ada pada *Extreme Programming* yaitu terdiri dari Perencanaan (*Planning*) seperti memahami kriteria pengguna dan perencanaan pengembangan, designing seperti perancangan *prototype* dan tampilan, pengkodean juga termasuk dalam pengintegrasian, terakhir adalah testing (Ariyanti dkk., 2020). *Extreme Programming* adalah metode pengembangan perangkat lunak untuk menyederhanakan proses pengembangan sehingga lebih fleksibel, adaptif, dan dikerjakan oleh satu atau dua orang pengembang (Fatoni and Dwi 2016). Ada empat tahapan dalam pengembangan perangkat lunak menggunakan metode extreme programming (Pressman, 2005).



Gambar 2.3 Metode Extreme Programming

Sumber: (Pressman, 2005)

1. *Planning* (Perencanaan)

Pada tahapan ini merupakan tahapan yang diperlukan sebelum pengembang membuat sistem, tahapan ini penting karena ketika membuat sebuah sistem harus direncanakan atau dianalisis kebutuhan-kebutuhan yang diperlukan *user*. Dalam tahapan planning ini terdapat beberapa bagian yaitu:

a. *User Stories* yaitu Pengguna menggambarkan atau menceritakan permasalahan dan kebutuhan *user* yang diperlukan untuk sistem yang akan dibangun.

b. *Values* merupakan poin atau nilai-nilai yang dapat diambil dari *user* stories.

c. *AccepTable test criteria* yaitu menentukan kriteria tes sebagai acuan terhadap kebutuhan sistem yang akan dibangun.

d. *Iteration* *plan* merupakan rencana untuk menentukan berapa kali peneliti melakukan pertemuan terhadap pengguna.

2. *Design* (Perancangan)

Setelah pada tahapan perencanaan selesai, maka tahapan selanjutnya adalah perancangan. Pada tahapan ini pengembang melakukan perancangan dengan membuat sebuah pemodelan, yang dimulai dari pemodelan sistem, kemudian pemodelan arsitektur, dan yang terakhir adalah pemodelan basis data. Dalam tahapan ini terdapat beberapa bagian yaitu:

a. *Simple design*: Pengembang mengembangkan perangkat lunak dengan desain yang sederhana.

b. *Spike solution* Jika dalam praktiknya desain yang dibuat sangatlah sulit. Extreme programming akan menggunakan spike solution dimana pembuatan design akan dibuat langsung ke tujuannya.

c. *CRC card* Digunakan untuk mengidentifikasikan dan mengorganisasikan object-oriented *class*es.

d. *Prototype*: Merupakan perancangan *user interface* biasanya dalam bentuk *wireframing* untuk mempermudah pengembang dan klien dalam melihat gambaran sistem

3. *Coding* (Pengkodean)

Tahapan ini merupakan tahapan untuk menerapkan pemodelan yang sudah dirancang di tahapan perancangan yang sudah dibuat kedalam bentuk *user interface* dan menggunakan bahasa pemrograman.

a. *Pair* *programming*: Dalam proses pengembangan terdapat dua orang *programmer*, dimana seorang *programmer* membuat *coding* dan *programmer* lainnya mengoreksi code yang dibuat.

b. *Refactory*: Merupakan tahapan yang dilakukan ketika terjadi ketidaksesuaian kode program kemudian dilakukan perbaikan untuk mendapatkan hasil yang diinginkan.

4. *Testing* (Pengujian)

Pada tahapan ini pengembang sistem melakukan pengujian terhadap sistem yang sudah dibuat untuk mengetahui kesalahan yang terdapat pada sistem dan untuk mengecek sistem telah dibuat sudah sesuai dengan kebutuhan pengguna atau belum.

## 2.7 *Unified Modeling Language* (UML)

UML (*Unified Modelling Language)* adalah sebuah Bahasa berbentuk grafik atau gambar yang digunakan untuk memvisualisasikan sebuah sistem yang akan di kembangkan berbasis *Object-Oriented*. UML juga memiliki standar penulisan sebuah sistem blue print, meliputi kelas dalam Bahasa pemrograman yang spesifik, proses bisnis, skema *database* dan komponen lainya yang diperlukan sistem (Mubarak dkk., 2019). Terdapat tiga diagram UML yang memiliki fungsi masing – masing yaitu:

### 2.7.1 *Use Case*

*Use Case* adalah teknik untuk mengambarkan kebutuhan-kebutuhan fungsional dari sebuah sistem yang ingin di kembangakan atau sistem baru yang akan dibuat. Setiap *use case* memiliki satu atau lebih skenario yang menjelaskan bagaimana sebuah sistem akan berinteraksi dengan pengguna atau sistem lainya untuk tercapainya suatu sasaran bisnis tertentu. (Artina, 2006). Simbol – simbol yang terdapat pada *use case* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol-Simbol *Use Case*

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | *Use Case:* Fungsionalitas yang disediakan sistem, Biasanya diawali dengan kata kerja sebelum farse nama *use case*. |
|  | Aktor: pengguna, proses atau sistem lain yang berinteraksi dengan sistem yang akan di kembangkan, walau simbol berbentuk orang aktor belum tentu orang, nama aktor biasanya menggunakan kata benda |
|  | Asosiasi: sebuah komunikasi antar aktor dan *use case* digunakan saat *use case* memilki interaksi dengan aktor |
|  | Generalisasi: hubungan umum sampai khusus antara dua buah *use case* di mana salah satu use memiliki fungsi yang lebih umum. |
|  | *Extend*: menunjukan *use case* tambahan dapat berdiri sendiri walau tanpa *use case* tambahan. |
|  | *Include*: Menunjukan bahwa *use case* tambahan akan dipanggil saat *use case* tambahan dijalankan |

### 2.7.2 *Activity Diagram*

*Activity diagram* merupakan diagram yang berfungsi untuk menjelaskan secara prosedural alur proses sebuah sistem. Pada diagram ini memungkinkan mengevaluasi adanya lebih dari satu jalur yang terbentuk dan berjalan beriringan. pembuatan *activity diagram* dimulai dari *initial* *node* dan diakhiri dengan *end* *node*. *initial node* dalam sebuah *activity diagram* diperbolehkan lebih dari satu, hal ini berguna untuk mengakomodasi jika sistem yang dikembangkan akan memiliki input lebih dari satu (Ayu, 2017). adapun simbol-simbol yang terdapat pada *activity diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol – Simbol *Activity Diagram*

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | Status awal: simbol yang memiliki fungsi menunjukan status awal dari sebuah diagram |
|  | Aktivitas: simbol yang memiliki fungsi menunjukan aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja. |
|  | *Decision*: asosiasi percabangan, dimana jika ada pilihan aktivitas lebih dari satu |
|  | *Joint*: asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
|  | Status akhir: simbol yang menunjukan aktivitas akhir diagram |

### 2.7.3 *Class* *diagram*

*Class diagram* adalah sebuah diagram UML yang menggambarkan hubungan antar kelas dan memiliki penjelasan detail dari sebuah sistem di dalam model desain, dan juga memperhatikan entitas perilaku sistem. Pada *class diagram* memiliki atribut-atribut dan operasi yang dimiliki dari sebuah kelas dan constraint yang berhubungan dengan objek yang dikoneksikan. Komponen-komponen yang terdapat pada *class diagram* yaitu *class*, relasi, assosiasi, generalisasi dan agregasi, attribut, operasi, dan visibilitas, tingkat akses objek eksternal pada suatu operasi. Juga hubungan antar kelas disebut multiplicity dan *card*inality (Hendini 2016). Adapun simbol-simbol yang terdapat pada *class diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Simbol – Simbol *Class Diagram*

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | Menggambarkan kelas yang terdapat pada struktur |
|  | Asosiasi berarah: relasi antar kelas yang memiliki arti kelas satu digunakan oleh kelas yang lain dan biasanya disertai dengan *multiplicity*. |
|  | Generalisasi: relasi antar kelas yang memiliki makna umum khusus |
|  | Ketergantungan: *depedency* merupakan relasi antar kelas dengan arti ketergantungan antar kelas. |
|  | Agregasi: merupakan relasi antar kelas dengan arti semua bagian |
|  | Antar muka: yaitu menunjukan interface hal ini sama dengan konsep interface pada pemrograman berorientasi objek. |
|  | Asosiasi: merupakan relasi antar kelas dengan makna umum dan biasanya disertai multiplicity. |

## 2.8 *Class Responsibility Collaborator Card (CRC Card).*

Menurut Zulhalim (2018), *Class Responsibility Collaborator Card* (*CRC Card*) adalah suatu teknik yang bertujuan untuk mengecek adanya interaksi suatu objek, mengenali, dan menentukan *class* yang diperlukan. Pada *CRC Card* terdapat *Class Name* yaitu sebagai suatu kelas yang dijabarkan, *Responsibilities* yang berfungsi sebagai tugas dan fungsi dari kelas tersebut, dan *Collaborators* berfungsi sebagai objek atau kelas yang berkaitan dengan kelas tersebut. Berikut ini merupakan gambar bagian-bagian *CRC Card*.

Tabel 2.5 *Class Responsibility Collaborator Card (CRC Card)*

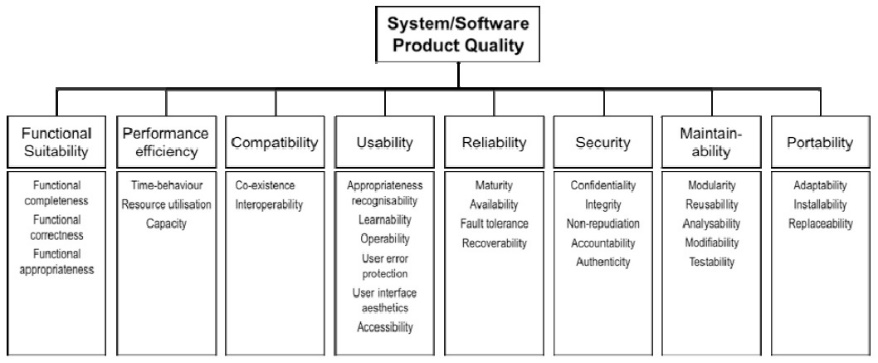
|  |  |
| --- | --- |
| ***Class Name*** | |
| ***Responsibilities*** | ***Collaborators*** |

## 2.9 Firebase

Firebase adalah penyedia layanan *cloud* dengan *back-end* sebagai servis yang berbasis di San Fransisco, California. Firebase membuat sejumlah produk untuk pengembangan aplikasi *mobile* ataupun web. Firebase didirikan oleh Andrew Lee dan James Tamplin pada tahun 2011 dan diluncurkan dengan *cloud database* secara *realtime* di tahun 2012 (Kumala & Winard, 2020).

## 2.10 Pengujian ISO 25010

Menurut (Fadli H. Wattiheluw, 2019) Pengujian ISO 25010 merupakan bagian dari *Systems and Software Quality Requirements and Evaluation* (SQuaRE) yang merupakan versi lanjutan dari ISO 91261, yang telah direvisi secara teknis dengan menambahkan beberapa struktur dan bagian dari standar model kualitas. Tujuan dari penggunaan kualitas ini adalah untuk mengukur sejauh mana produk atau sistem tersebut bisa digunakan oleh pengguna untuk memenuhi kebutuhan dalam mencapai tujuan yang diinginkan dengan efisiensi, efektivitas, kepuasan dalam konteks penggunaan yang spesifik, dan bebas dari resiko.Menurut (Harun, 2018) ISO 25010 terdiri dari delapan karakteristik yang dibagi menjadi beberapa bagian yang berhubungan dengan sifat-sifat statis perangkat lunak dan sifat dinamis dari sistem komputer, yang dapat dilihat pada Gambar 2.4



Gambar 2.4 Model ISO 25010

Sumber: (Harun, 2018)

Berdasarkan gambar diatas, dapat dijelaskan mengenai delapan karakteristik tersebut, sebagai berikut:

* 1. *Functional* *Suitability*, produk yang memberikan fungsional untuk memenuhi kebutuhan saat sistem atau produk tersebut digunakan pada keadaan tertentu.
  2. *Reliability*, sistem dapat mempertahankan kinerjanya pada level tertentu ketika digunakan pada keadaan tertentu.
  3. *Performance* *Efficiency*, sistem menyediakan performa yang baik dengan sejumlah *resource* yang akan digunakan pada sistem atau produk.
  4. *Usability*, sistem atau produk mudah dimengerti, mudah dipakai, dan menarik untuk digunakan.
  5. *Security*, sistem menyediakan layanan untuk melindungi akses, ataupun pengungkapan yang berbahaya.
  6. *Compatibility*, merupakan kemampuan pada suatu komponen atau sistem untuk bertukar informasi.
  7. *Maintainability*, merupakan tingkat suatu sistem dapat dimodifikasi, perbaikan, pengembangan untuk menyesuaikan dengan lingkungan, modifikasi pada kriteria, dan spesifikasi fungsi.
  8. *Portability*, sistem dapat dipindahkan dari satu ruang ke ruang lainnya.

Berdasarkan dari kebutuhan dalam penelitian ini, dalam pengujian aplikasi *mobil*e menggunakan empat karakteristik yang digunakan yaitu *functional suitability, compatibility, usability,* dan *performance efficiency*. (David, 2011)

## 2.11 Pengertian Figma

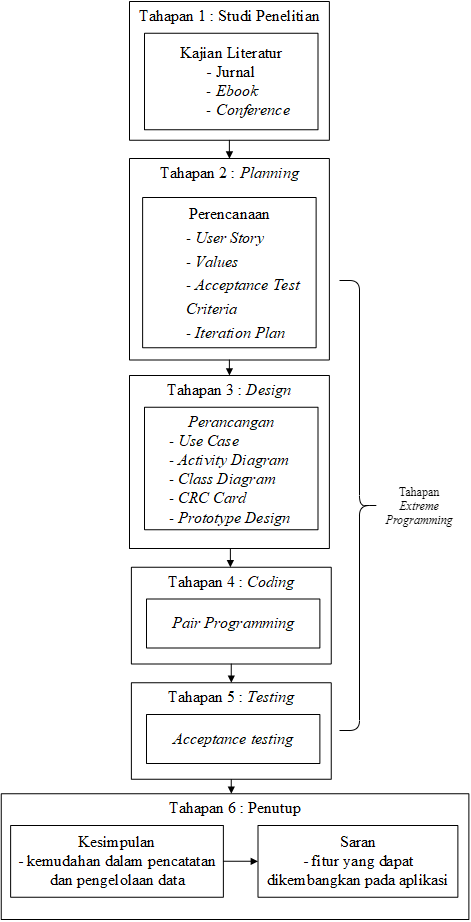
Figma adalah *software* yang digunakan untuk pembuatan desain berbasis cloud dengan konsep fungsionalitas sketch, dan dapat melakukan kolaborasi antar desainer lainnya. figma dapat bekerja pada berbagai sistem operasi karena dapat berjalan pada browser, fitur slack yang terdapat pada figma sebagai media komunikasi dari satu designer ke designer lainnya dapat memudahkan suatu tim dalam berkomunikasi secara *realtime* dalam melakukan editing, keunggulan lain yaitu pengguna dapat dengan mudah mendapatkan kode dalam bentuk css sehingga memudahkan front end dalam pengimplementasian desain yang ada (Nugraha dkk., 2020)

# 

# 3 METODOLOGI PENELITIAN

## 3.1 Kerangka Penelitian

Kerangka Penelitian adalah sebuah konsep atau gambaran yang telah dibuat oleh penulis untuk dilakukannya penelitian. Dari uraian yang sudah dijelaskan sebelumnya, maka akan menghasilkan sebuah kerangka penelitian yang dapat dilihat pada gambar 3.1.



Gambar 3.1 Kerangka Penelitian

Proses pengembangan sistem melalui empat tahapan yaitu tahap perencanaan (*planning*) yaitu bertujuan untuk mengetahui permasalahan dan mengetahui keinginan pengguna, tahapan perancangan (*design*) berupa perancangan *use case, activity diagram, class diagram*, *CRC Card* dan *interface*, pengkodean (*coding*) yaitu penerapan rancangan dalam bentuk *coding*, dan tahapan pengujian (*testing*) yaitu melakukan pengujian terhadap sistem.

## 3.2 *Planning*

Perencanaan merupakan tahapan utama yang penulis lakukan untuk mendapatkan kebutuhan pengguna, mengkaji literatur, pembuatan *user stories*, dan melakukan komunikasi secara berkala kepada pihak jasa penyeberangan B.C Bakauheni. Berikut ini merupakan tahapan pada perencanaan:

### 3.2.1 *User Stories*

Berikut ini merupakan tabel *user stories* yang telah dirangkum berdasarkan wawancara yang telah dilakukan dengan pimpinan lampiran 1. Tahap ini bertujuan untuk menggambarkan atau menceritakan permasalahan dan kebutuhan *user* yang diperlukan untuk sistem yang akan dibangun. *user* stories dapat dilihat pada Tabel 3.1 Tabel 3.2 dan Tabel 3.3.

Tabel 3. *User Stories*

|  |  |
| --- | --- |
| **Kode *Stories*** | ***User Stories*** |
| *Story*-01 | Saya ingin aplikasi dapat mencatat data transaksi dan setiap transaksi data kendaraan otomatis terinput agar memudahkan dalam melakukan transaksi dan mengurangi kesalahan pencatatan. |
| *Story*-02 | Sebagai pimpinan, saya ingin aplikasi dapat mencatat pembayaran kredit pengendara agar dapat mengetahui *history* pembayaran kredit pengendara pada sebelumnya. |

Tabel 3. *User Stories* (lanjutan 1)

|  |  |
| --- | --- |
| **Kode *Stories*** | ***User Stories*** |
| *Story*-03 | Sebagai pimpinan, saya ingin setiap transaksi dapat terlihat langsung secara *realtime* agar dapat memantau transaksi yang terjadi walau saya sedang tidak berada di kantor dan terdapat nama dari pegawai yang melakukan transaksi. |
| *Story*-04 | Sebagai pimpinan, saya ingin aplikasi dapat memisahkan data kendaraan pembayaran langsung dan bulanan agar mudah dalam membuat invoice untuk perusahaan yang bersangkutan. |
| *Story*-05 | Sebagai pimpinan, saya ingin aplikasi dapat menampilkan nama pegawai yang melakukan transaksi supaya dapat bertanya langsung jika ada kesalahan atau hal yang ingin ditanyakan dengan transaksi tersebut. |
| *Story*-06 | Sebagai pimpinan, saya ingin aplikasi dapat melakukan pencarian dan mengedit data transaksi dan data pembayaran kredit agar dapat mengubah data jika ada kesalahan input. |
| *Story*-07 | Sebagai pimpinan, saya ingin aplikasi dapat mengurutkan data sesuai perusahaan untuk bagian kendaraan yang melakukan pembayaran bulanan agar mempermudah dalam pembuatan invoice |
| *Story*-08 | Sebagai pimpinan, saya ingin aplikasi dapat mengurutkan data sesuai perusahaan untuk bagian kendaraan yang melakukan pembayaran bulanan agar mempermudah dalam pembuatan invoice. |
| *Story*-09 | Sebagai pimpinan, Saya ingin aplikasi dapat mengubah dan menambahkan data kendaraaan dan pegawai, dan juga dapat mengubah password maupun *user*name pegawai. Agar dengan mudah mengubah data jika ada data yang salah ataupun jika ada kendaraan baru atau pegawai baru. |

Tabel 3. *User Stories* (lanjutan 2)

|  |  |
| --- | --- |
| **Kode *Stories*** | ***User Stories*** |
| *Story*-10 | Sebagai pimpinan, Saya ingin aplikasi dapat melakukan edit stok untuk menambahkan atau pun mengurangi stok, karena terkadang jika stok kurang akan ditambah dan jika stok banyak akan dikurangi. |
| *Story*-11 | Sebagai Pimpinan, saya ingin aplikasi dapat melakukan rekap harian untuk mengetahui uang yang masuk dan keluar dan hasil dari rekap tersebut diambil menjadi uang stok/saldo. |
| *Story*-12 | Sebagai pimpinan, Saya ingin aplikasi dapat merubah Harga Penyeberangan karena terkadang harga naik dan turun. |
| *Story*-13 | Sebagai pimpinan, Saya ingin aplikasi dapat mencatat pengeluaran seperti buku/pena dan lainnya biaya tersebut diambil dari stok dan data masuk kedalam rekap harian |
| *Story*-14 | Sebagai pimpinan, Saya ingin pegawai tidak dapat mengubah maupun menambahkan data pegawai, data kendaraan, data kreditur, data transaksi, stok, dan harga penyeberangan, karena takut terjadi kecurangan oleh pegawai. |

### 3.3.2 *Values*

Setelah dilakukan tahapan pembuatan *user storie*s tahapan selanjutnya yang dilakukan yaitu mengambil nilai atau *value* pada setiap *stories* yang ada, hal ini berguna untuk menentukan prioritas *user stories* yang akan dibuat terlebih dahulu. Nilai yang ada berdasarkan dari tingkat kesulitan dan kompleksitas dari fitur yang akan di buat. *Value* yang dapat diambil dari *user stories* diatas dapat dilihat pada Tabel 3.4, Tabel 3.5 dan Tabel 3.6.

Tabel 3.4 *Value User Stories*

|  |  |  |
| --- | --- | --- |
| **Kode *Stories*** | ***User Stories*** | ***Values*** |
| *Story*-01 | Saya ingin aplikasi dapat mencatat data transaksi dan setiap transaksi data kendaran otomatis terinput agar memudahkan dalam melakukan transaksi dan mengurangi kesalahan pencatatan. | 4 |
| *Story*-02 | Sebagai pimpinan, saya ingin aplikasi dapat mencatat pembayaran kredit pengendara agar dapat mengetahui *history* pembayaran kredit pengendara pada sebelumnya. | 3 |
| *Story*-11 | Sebagai Pimpinan, saya ingin aplikasi dapat melakukan rekap harian untuk mengetahui uang yang masuk dan keluar dan hasil dari rekap tersebut diambil menjadi uang stok/saldo. | 3 |
| *Story*-06 | Sebagai pimpinan, saya ingin aplikasi dapat melakukan pencarian dan mengedit data transaksi dan data pembayaran kredit agar dapat mengubah data jika ada kesalahan input. | 3 |
| *Story*-08 | Sebagai pimpinan, saya ingin aplikasi dapat mengurutkan data sesuai perusahaan untuk bagian kendaraan yang melakukan pembayaran bulanan agar mempermudah dalam pembuatan invoice. | 3 |
| *Story*-02 | Sebagai pimpinan, saya ingin aplikasi dapat melihat beberapa *history* transaksi sebelumnya, sebelum melanjutkan Transaksi untuk melihat pengendara masih ada tanggungan atau tidak. | 2 |

Tabel 3.5 *Value* *User Stories* (Lanjutan 1)

|  |  |  |
| --- | --- | --- |
| **Kode *Stories*** | ***User Stories*** | ***Values*** |
| *Story*-03 | Sebagai pimpinan, saya ingin setiap transaksi dapat terlihat langsung secara *realtime* agar dapat memantau transaksi yang terjadi walau saya sedang tidak berada di kantor dan terdapat nama dari pegawai yang melakukan transaksi. | 2 |
| *Story*-04 | Sebagai pimpinan, saya ingin aplikasi dapat memisahkan data kendaraan pembayaran langsung dan bulanan agar mudah dalam membuat invoice untuk perusahaan yang bersangkutan. | 2 |
| *Story*-09 | Sebagai pimpinan, Saya ingin aplikasi dapat mengubah dan menambahkan data kendaraaan dan pegawai, dan juga dapat mengubah password maupun *user*name pegawai. Agar dengan mudah mengubah data jika ada data yang salah ataupun jika ada kendaraan baru atau pegawai baru. | 2 |
| *Story*-12 | Sebagai pimpinan, Saya ingin aplikasi dapat merubah Harga Penyeberangan karena terkadang harga naik dan turun. | 2 |
| *Story*-13 | Sebagai pimpinan, Saya ingin aplikasi dapat mencatat pengeluaran seperti buku/pena dan lainnya biaya tersebut diambil dari stok dan data masuk kedalam rekap harian | 2 |
| *Story*-05 | Sebagai pimpinan, saya ingin aplikasi dapat menampilkan nama pegawai yang melakukan transaksi supaya dapat bertanya langsung jika ada kesalahan atau hal yang ingin ditanyakan dengan transaksi tersebut. | 1 |

Tabel 3.6 *Value* *User Stories* (Lanjutan 2)

|  |  |  |
| --- | --- | --- |
| **Kode *Stories*** | ***User Stories*** | ***Values*** |
| *Story*-07 | Sebagai pimpinan, saya ingin mengetahui *history* pembayaran kreditur, data sopir yang masih memiliki tanggungan dan sopir yang melakukan pembayaran bulanan. | 1 |
| *Story*-10 | Sebagai pimpinan, Saya ingin aplikasi dapat melakukan edit stok untuk menambahkan atau pun mengurangi stok, karena terakadang jika stok kurang akan ditambah dan jika stok banyak akan dikurangi. | 1 |
| *Story*-14 | Sebagai pimpinan, Saya ingin pegawai tidak dapat mengubah maupun menambahkan data pegawai, data kendaraan, data kreditur, data transaksi, stok, dan harga penyeberangan, karena takut terjadi kecurangan oleh pegawai. | 1 |

Mengacu pada Tabel 3.2 diatas, maka *user* *story* yang memiliki tingkat kesulitan tertinggi yaitu *story*-1 sehingga *story* yang akan kerjakan dahulu yaitu *story*-1, selanjutnya *story*-2, *story*-12 dan seterusnya.

### 3.3.3 *Acceptance Test Criteria*

Pada tahapan ini bertujuan untuk membuat suatu daftar kriteria yang diambil dari *user stories*, yang bertujuan untuk menentukan batasan-batasan *user stories*, pada tahap ini juga bertujuan untuk menentukan aplikasi yang dibuat sudah berjalan sesuai kebutuhan dan keinginan pengguna. Berikut kriteria yang digunakan berdasarkan empat aspek pengujian ISO 25010 adalah sebagai berikut:

1. *Functional Suitability*

Aspek pertama yang akan dilakukan yaitu pengujian pada bagian fungsional, memiliki tujuan untuk melakukan pengecekan terhadap fungsi-fungsi yang ada. Berikut ini merupakan karakteristik pengujian yang dibagi menjadi beberapa karakteristik yaitu:

a. *Functional completeness*, sejauh mana fungsi yang telah disediakan mencakup seluruh tugas dan tujuan pengguna secara spesifik.

b. *Functional correctness*, sejauh mana produk atau sistem menyediakan hasil yang benar sesuai dengan kebutuhan pengguna.

c. *Functional appropriateness*, sejauh mana fungsi yang telah disediakan mampu memfasilitasi penyelesaian tugas atau suatu tujuan tertentu.

1. *Usability*

Tes kriteria yang akan dilakukan selanjutnya yaitu *usability* yang bertujuan untuk mengetahui aplikasi yang dibuat mudah dimengerti, mudah dipakai dan menarik. Karakteristik dalam pengujian usability ini dibagi menjadi beberapa karakteristik yaitu:

a. *Appropriateness* *recognisability*, sejauh mana pengguna mengetahui apakah sistem atau produk sesuai kebutuhan mereka.

b. *Operability*, sejauh mana produk atau sistem mudah dioperasikan dan dikendalikan.

c. *Learnability*, sejauh mana produk atau sistem dapat digunakan oleh pengguna untuk mencapai suatu tujuan tertentu, belajar menggunakan sistem atau produk dengan efisien, efektif dan kebebasan dari resiko serta dalam hal kepuasan pada konteks tertentu.

d. *User* *interface* *aesthetics*, sejauh mana tampilan antarmuka pengguna dari sistem memungkinkan interaksi yang menyenangkan dan memuaskan pengguna.

e. *User* *error* *protection*, sejauh mana produk atau sistem melindungi pengguna terhadap suatu kesalahan.

f. *Accessibility*, sejauh mana produk atau sistem dapat digunakan oleh semua kalangan untuk mencapai suatu tujuan tertentu sesuai dengan konteks

1. *Compatibility*

Pada tahap ini menjalankan fungsi lain yang diperlukan secara bersamaan ketika berbagi perangkat keras dan environment perangkat lunak yang sama. Aspek yang diuji pada tahap *compatibility* meliputi *Co*-*existence*, Operating System, Device (Jayanto & Jati, 2017). Berikut tahapan yang dilakukan pada uji *compatibility*:

1. *Co-existence*

Pada tahap ini aplikasi akan diuji apakah dapat berjalan berdampingan dengan aplikasi lainya dengan menggunakan *smartphone* Pocophone f1. Kerangka pengujian *Co****-****existence* dapat dilihat pada Tabel 3.7.

Tabel 3.7 Kerangka Pengujian *Co-existence*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NO** | **Aplikasi yang dijalankan** | | **Berhasil** | **Gagal** |
| 1 | Aplikasi pendataan BC | Play Store |  |  |
| 2 | Aplikasi pendataan BC | Google Chrome |  |  |
| 3 | Aplikasi pendataan BC | Whatsapp |  |  |
| 4 | Aplikasi pendataan BC | Facebook |  |  |
| 5 | Aplikasi pendataan BC | Maps |  |  |
| 6 | Aplikasi pendataan BC | Apex Legends |  |  |
| **Total** | | |  |  |

1. *Operating System dan device*

Pada tahap ini akan dilakukan pengujian di berbagai sistem operasi android dan beberapa perangkat, pengujian dibagi menjadi tiga bagian yaitu *setup* *suite* mewakili instalasi aplikasi di perangkat, *built-in explorer* yaitu menjalankan aplikasi pada perangkat pengujian, dan terakhir *teardown* *suite* yaitu melakukan penghapusan pada perangkat. Perangkat - perangkat yang digunakan untuk pengujian dapat dilihat pada Tabel 3.8.

Tabel 3.8 Kerangka Pengujian *Operating System Dan Device*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***No*** | ***Perangkat*** | ***Setup Suite*** | ***Built-in Explorer Suite*** | ***Teardown Suite*** | ***Jumlah*** |
| 1 | Xiaomi Pochophone F1  (android 10) |  |  |  |  |
| 2 | Xiaomi Pocophone X3 nft  (android 11) |  |  |  |  |
| 3 | Oppo f1s  (android 5.1) |  |  |  |  |
| 4 | Redmi 4x  (android 7) |  |  |  |  |
| Total | |  |  |  |  |

1. *Performance* *Efficiency*

Pengujian pada tahap ini berguna untuk mengetahui sejauh mana kinerja *relative* aplikasi dengan perangkat yang digunakan pada kondisi tertentu. Pengujian dilakukan menggunakan tools *Firebase* test lab secara *automation* menggunakan lima perangkat dan hasil pengujian dibagi menjadi tiga bagian. Aspek yang akan di uji pada tahap *performance Efficiency* sebagai berikut:

1. *Time Behaviour* pada pengujian time behaviour nilai didapatkan berdasarkan persentase thread yang di dapat dari pengujian pada firebase test lab.
2. CPU *Utilization* pada tahap ini melakukan pengujian terhadap CPU ketika menjalankan aplikasi dengan rentang waktu yang ditentukan dan akan didapatkan persentase dari penggunaan CPU
3. *Memory Utilization* pada pengujian ini melakukan pengujian pada *memory* Ketika menjalankan aplikasi dan akan didapatkan nilai rata-rata *memory* yang digunakan. Setelah pembuatan kriteria penerimaan tahapan selanjutnya dalam penelitian ini yaitu melakukan pembuatan rencana iterasi (*iteration plan).*

### 3.3.4 *Iteration Plan*

*Iteration Plan* merupakan proses komunikasi dan perencanaan fase iterasi pengembangan sistem yang akan di bangun, yang bertujuan untuk menentukan berapa lama pembuatan aplikasi yang akan dibangun, berdasarkan prioritas dan *value* pada *user stories.* rencana iterasi yang telah dibuat dapat dilihat pada Tabel 3.9.

Tabel 3.9 Rencana Iterasi

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Kode *Stories*** | ***Time***  **(*days*)** | ***Iteration*** | | | | |
| **1** | **2** | **3** | **4** | **5** |
| Story-01 | 4 | ✓ |  |  |  |  |
| Story-02 | 4 | ✓ |  |  |  |  |
| Story-12 | 4 | ✓ |  |  |  |  |
| Story-07 | 4 |  | ✓ |  |  |  |
| Story-09 | 3 |  | ✓ |  |  |  |
| Story-03 | 3 |  | ✓ |  |  |  |
| Story-04 | 3 |  |  | ✓ |  |  |
| Story-05 | 3 |  |  | ✓ |  |  |
| Story-10 | 3 |  |  | ✓ |  |  |
| Story-13 | 3 |  |  |  | ✓ |  |
| Story-14 | 3 |  |  |  | ✓ |  |
| Story-06 | 3 |  |  |  | ✓ |  |
| Story-08 | 3 |  |  |  |  | ✓ |
| Story-11 | 3 |  |  |  |  | ✓ |

Tabel *iteration* plan diatas menunjukan terdapat lima iterasi yang akan dilakukan, iterasi pertama membahas *story*-01, *story*-02 dan *story*-12, pada iterasi kedua membahas *story*-07, *story*-09 dan *story*-03, dan seterusnya sampai tahap iterasi selesai.

#### 3.3.4.1 *Iterations*

Pada tahap ini melakukan pengujian terhadap hasil implementasi dari setiap *user* stories untuk mengetahui bahawa *user* *stories* sudah diimplementasikan sesuai dengan kebutuhan *user*. Tetapi jika hasil implementasi belum memenuhi keinginan dan kebutuhan *user*, maka akan dilakukan perubahan listing kode program. Pada metode *Extreme Programming*, jika implementasi *user stories* belum sesuai keinginan *user* maka dilarang untuk melanjutkan ke iterasi selanjutnya. Namun jika pengembang sudah mencapai iterasi tiga ataupun diatasnya, pengembang boleh kembali ke iterasi sebelumnya jika ada perubahan kebutuhan *user*. Contoh tahapan iterasi dapat dilihat pada Tabel 3.10.

Tabel 3.10 Contoh Tahapan Iterasi

|  |  |
| --- | --- |
| **Iterasi** | **Status** |
| **Iterasi 1** | OK |
| **Iterasi 2** | OK |
| **Iterasi 3** | NOT OK (boleh Kembali ke iterasi sebelumnya untuk menyesuaikan kebutuhan *user*) |
| **Iterasi 4** |  |
| **Iterasi 5** |  |

#### 3.3.4.2. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk memudahkan penelitian ini dalam pembuatan rancangan sistem. analisis dibagi menjadi dua kategori yaitu kebutuhan fungsional dan kebutuhan non fungsional, hasil analisa yang didapatkan dapat dilihat sebagai berikut:

1. Kebutuhan Fungsional

Berikut ini merupakan kebutuhan fungsional berupa fungsi atau proses yang nantinya akan dibutuhkan sistem sesuai kebutuhan pengguna:

a. *User* Pimpinan

*User* pimpinan adalah aktor yang dapat melihat semua informasi yang ada pada aplikasi. Kegiatan yang dapat dilakukan oleh *user* pimpinan adalah sebagai berikut:

1. Mengelola data transaksi
2. Menampilkan data transaksi
3. Memfilter data transaksi
4. Mencari data transaksi
5. Menampilkan data kendaraan dan pegawai
6. Mengelola data kendaraan dan pegawai
7. Melakukan transaksi penyeberangan dan kreditur
8. Menampilkan rekap harian
9. Menampilkan dan mengelola stok
10. Menampilkan dan mengelola harga pelabuhan
11. Menambahkan pengeluaran
12. *User* Pegawai

*User* pegawai adalah aktor yang dapat semua informasi yang ada pada aplikasi tanpa bisa melakukan perubahan data. Kegiatan yang dapat dilakukan oleh *user* pimpinan adalah sebagai berikut:

1. Menampilkan data transaksi
2. Memfilter data transaksi
3. Mencari data transaksi
4. Menampilkan data kendaraan dan pegawai
5. Melakukan transaksi penyeberangan dan kreditur
6. Menampilkan rekap harian
7. Menampilkan stok
8. Menampilkan harga pelabuhan
9. Menambah pengeluaran

2. Kebutuhan Non Fungsional

Berikut ini merupakan kebutuhan non fungsional atau kebutuhan berupa perangkat-perangkat untuk mendukung sistem yang akan dikembangkan:

* 1. Aplikasi dapat dijalankan di beberapa versi android diantaranya android 10, 11, 5 dan 7.
  2. Aplikasi memiliki tampilan *user* interface yang friendly serta responsif.
  3. Aplikasi dapat menyimpan data-data kedalam *database*.

3. Spesifikasi Perangkat

Berikut merupakan kebutuhan berupa perangkat keras dan perangkat lunak yang digunakan dalam penelitian:

1. Perangkat keras (*Hardware*)

Perangkat keras yang digunakan pengembang dalam penelitian ini yaitu sebuah komputer dengan spesifikasi sebagai berikut:

1. *Processor* Ryzen 5 3500X 6-Core ~3.6GHz
2. *Memory* RAM 16 GB
3. *Memory* *internal* 1.25 TB
4. VGA *card* Radeon RX5500XT
5. Perangkat lunak (*Software*)

Perangkat lunak yang digunakan pengembang dalam penelitian ini yaitu sebagai berikut:

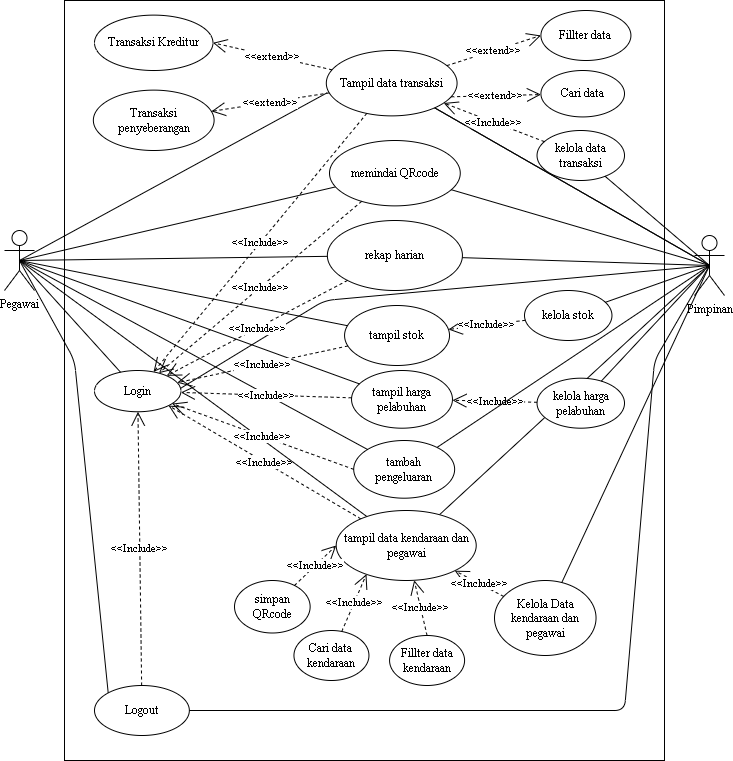
1. Sistem operasi windows 10 (64-bit)
2. Sistem operasi android
3. Visual Code
4. Figma

## 3.4. *Design*

Perancangan ini bertujuan untuk merancang atau mendesain sistem dalam rangka memenuhi kebutuhan pengguna sistem dan untuk memberikan gambaran yang jelas. Pada tahap perancangan sistem ini terdiri dari *use case*, *activity diagram*, CRC dan *prototype* *design*.

### 3.4.1 *Use Case* *Diagram*

Berikut ini merupakan sebuah proses kegiatan dan proses bisnis yang dilakukan oleh aktor yang berinteraksi dengan *use case*. Pada penelitian ini penulis menggunakan dua aktor yaitu pegawai dan pengurus. *Use case* dapat dilihat pada Gambar 3.2.

******

Gambar 3.2 *Use Case* Diagram Aplikasi Pendataan

Penjelasan dari proses kegiatan pada *use case* yang telah dibuat dapa dapat dilihat padaTabel 3.11.

Tabel 3.11 Penjelasan *Use Case*

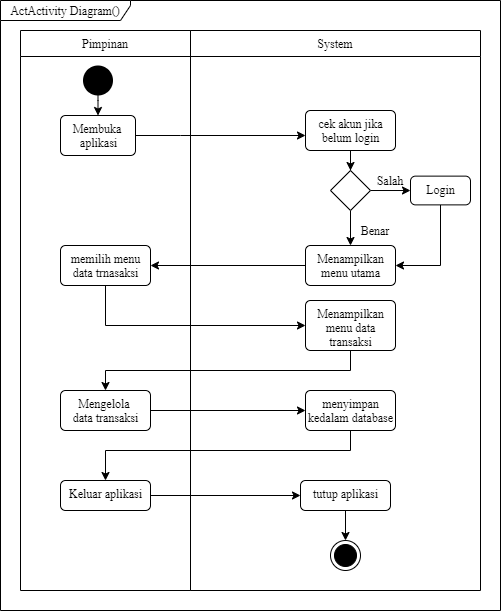
|  |  |
| --- | --- |
| **Aktor** | **Kegiatan** |
| Pimpinan | Kegiatan yang dilakukan pimpinan yaitu login terlebih dahulu untuk masuk ke halaman utama aplikasi, kemudian pimpinan dapat Mengelola data transaksi, menampilkan data transaksi, memfilter data transaksi, mencari data transaksi, menampilkan data kendaraan dan pegawai, mengelola data kendaraan dan pegawai, melakukan transaksi penyeberangan dan kreditur, menampilkan rekap harian, menampilkan dan mengelola stok, menampilkan dan mengelola harga pelabuhan, menambahkan pengeluaran |
| Pegawai | Kegiatan yang di lakukan pegawai yaitu login terlebih dahulu untuk masuk ke halaman utama aplikasi, kemudian pegawai dapat menampilkan data transaksi, memfilter data transaksi, mencari data transaksi, menampilkan data kendaraan dan pegawai, melakukan transaksi penyeberangan dan kreditur, menampilkan rekap harian, menampilkan stok, menampilkan harga pelabuhan, menambah pengeluaran |

### 3.4.2 *Activity Diagram*

*Activity diagram* pada rancangan aplikasi pendataan transaksi dapat dilihat pada gambar dibawah ini:

1. *Activity diagram* mengelola data transaksi

Hasil pembuatan *activity diagram* mengelola data transaksi yang dilakukan oleh pimpinan dapat dilihat pada Gambar 3.3.



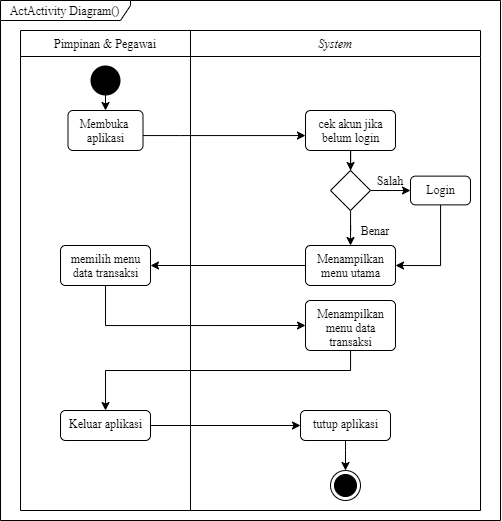
Gambar 3.3 *Activity Diagram* Mengelola Data Transaksi

Penjelasan:

Diawali dengan pimpinan membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pimpinan memilih menu data transaksi dan sistem menampilkan, kemudian pimpinan melakukan pengelolaan data transaksi dan kemudian sistem melakukan penyimpanan data ke *database*, setelah melakukan perubahan jika pimpinan menutup aplikasi sistem akan menutup.

1. *Activity diagram* menampilkan data transaksi

Hasil pembuatan *activity diagram* menampilkan data transaksi yang dilakukan oleh pimpinan dan pegawai dapat dilihat pada Gambar 3.4.



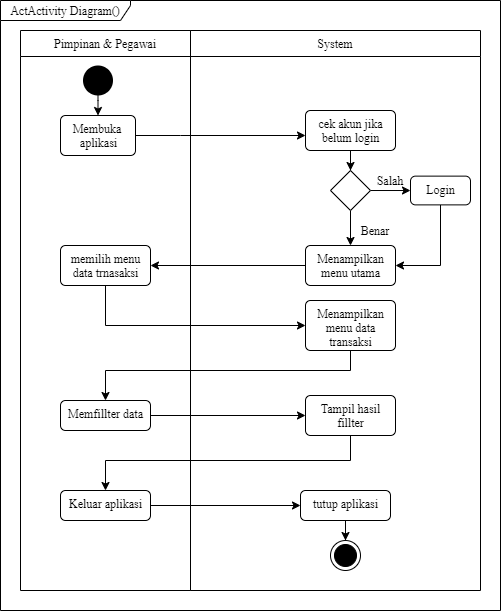
Gambar 3.4 *Activity Diagram* Menampilkan Data Transaksi

Penjelasan:

Diawali dengan pimpinan membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pimpinan atau pegawai memilih menu data transaksi kemudian sistem menampilkan data transaksi.

1. *Activity Diagram* memfilter data transaksi

Hasil pembuatan *activity diagram* memfilter data transaksi yang dilakukan oleh pimpinan dan pegawai dapat dilihat pada Gambar 3.5.



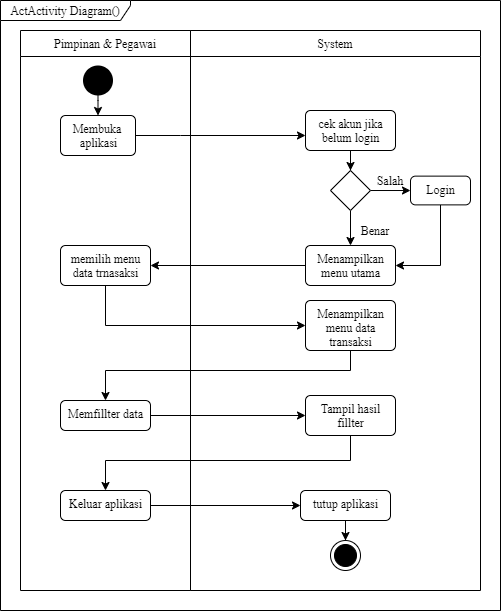
Gambar 3.5 *Activity Diagram* Memfilter Data Transaksi

Penjelasan:

Diawali dengan pimpinan atau pegawai membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pegawai atau pimpinan memilih menu data transaksi kemudian sistem menampilkan data transaksi selanjutnya menekan fungsi filter data yang disediakan dan sistem akan menampilkan hasil filter, setelah melakukan filter data jika pengguna menutup aplikasi sistem akan menutup.

1. *Activity diagram* pencarian data transaksi

Hasil pembuatan *activity diagram* pencarian data transaksi yang dilakukan oleh pimpinan dan pegawai dapat dilihat pada Gambar 3.6.



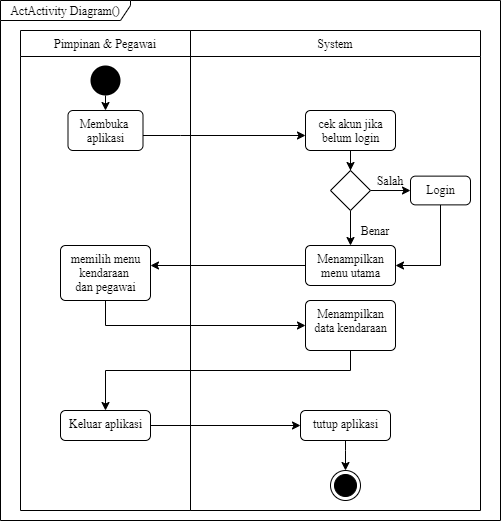
Gambar 3.6 *Activity Diagram* Pencarian Data Transaksi

Penjelasan:

Diawali dengan pimpinan atau pegawai membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pengguna atau pimpinan memilih menu data transaksi kemudian sistem menampilkan data transaksi selanjutnya menekan fungsi pencarian data yang disediakan dan sistem akan menampilkan hasil pencarian, setelah melakukan pencarian data jika pengguna menutup aplikasi sistem akan menutup.

1. *Activity diagram* menampilkan data kendaraan dan pegawai

Hasil pembuatan *activity diagram* menampilkan data kendaraan dan pegawai yang dilakukan pimpinan atau pegawai dapat dilihat pada Gambar 3.7.



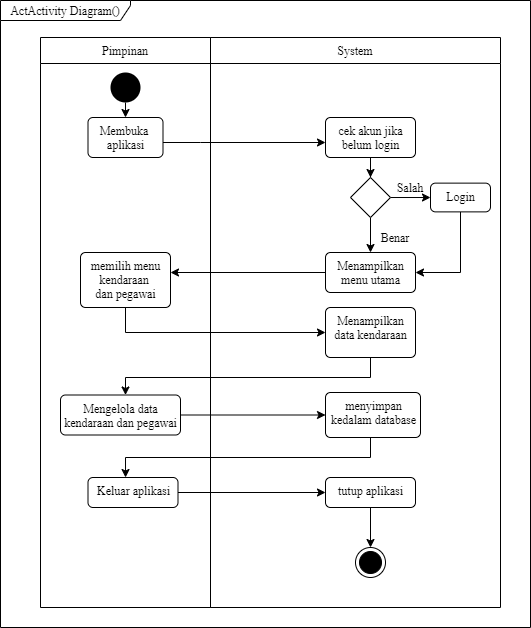
Gambar 3.7 *Activity Diagram* Menampilkan Data Kendaraan dan Pegawai

Penjelasan:

Diawali dengan pimpinan atau pegawai membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pegawai atau pimpinan memilih menu kendaraan dan pegawai, kemudian sistem menampilkan data kendaraan dan pegawai, jika pengguna menutup aplikasi sistem akan menutup.

1. *Activity diagram* mengelola data kendaraan dan pegawai

Hasil pembuatan *activity diagram* menglola data kendaraan dan pegawai dapat dilihat pada Gambar 3.8.



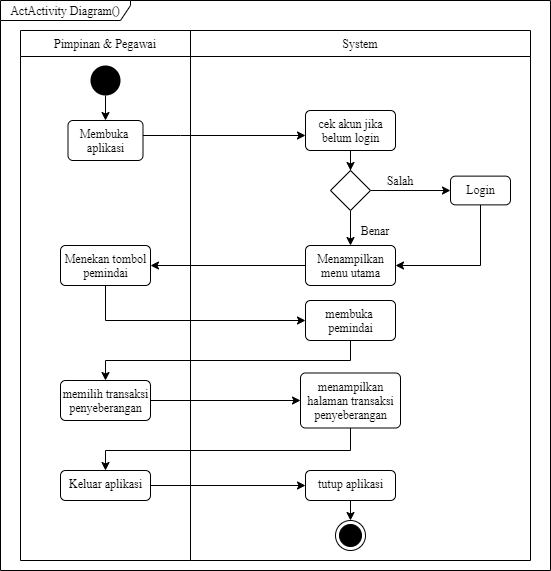
Gambar 3.8 *Activity Diagram* Mengelola Data Kendaraan dan Pegawai

Penjelasan:

Diawali dengan pimpinan membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pimpinan memilih menu kendaraan dan pegawai, kemudian sistem menampilkan data kendaraan dan pegawai , kemudian pimpinan melakukan pengelolaan data pegawai atau data kendaraan dan kemudian sistem melakukan penyimpanan data ke *database*, jika pimpinan menutup aplikasi sistem akan menutup.

1. *Activity diagram* melakukan transaksi penyeberangan

Hasil pembuatan *Activity diagram* melakukan transaksi penyeberangan yang dilakukan oleh pimpinan dan pegawai dapat dilihat pada Gambar 3.9.



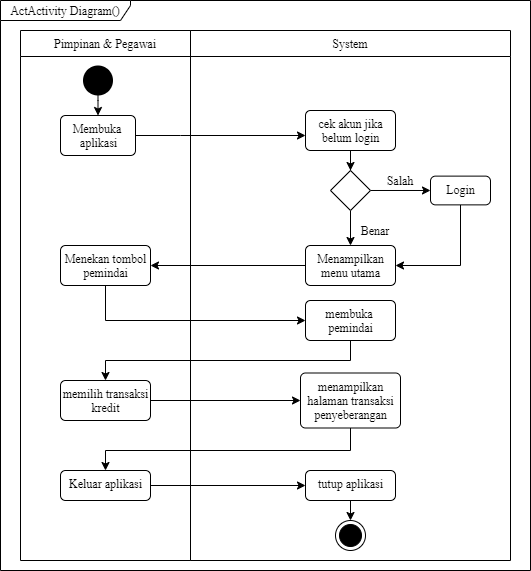
Gambar 3.9 *Activity Diagram* Melakukan Transaksi Penyeberangan

Penjelasan:

Diawali dengan pimpinan atau pegawai membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pegawai atau pimpinan memilih menu pemindai dan dilanjutkan memindai *QR-code* untuk memasukan data-data kendaraan yang akan melakukan transaksi, kemudian sistem menampilkan data kendaraan, kemudian pimpinan atau pegawai memilih transaksi penyeberangan dan sistem akan menampilkan halaman transaksi penyeberangan, telah melakukan transaksi sistem akan menyimpan data ke *database*, jika pimpinan atau pegawai menutup aplikasi sistem akan menutup.

1. *Activity diagram* melakukan transaksi kredit

Hasil pembuatan *activity diagram* melakukan transaksi kredit yang dilakukan oleh pimpinan dan pegawai dapat dilihat pada Gambar 3.10.



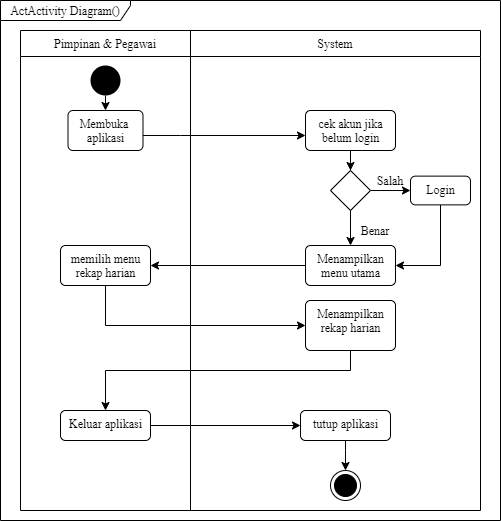
Gambar 3.10 *Activity Diagram* Melakukan Transaksi Kredit

Penjelasan:

Diawali dengan pimpinan atau pegawai membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pegawai atau pimpinan memilih menu pemindai dan dilanjutkan memindai *QR-code* untuk memasukan data-data kendaraan yang akan melakukan transaksi, kemudian sistem menampilkan data kendaraan, kemudian pimpinan atau pegawai memilih transaksi kredit dan sistem akan menampilkan halaman transaksi kredit, telah melakukan transaksi sistem akan menyimpan data ke *database*, jika pimpinan atau pegawai menutup aplikasi sistem akan menutup.

1. *Activity diagram* Menampilkan rekap harian

Hasil pembuatan *activity diagram* Menampilkan rekap harian yang dilakukan oleh pimpinan dan pegawai dapat dilihat pada Gambar 3.11.



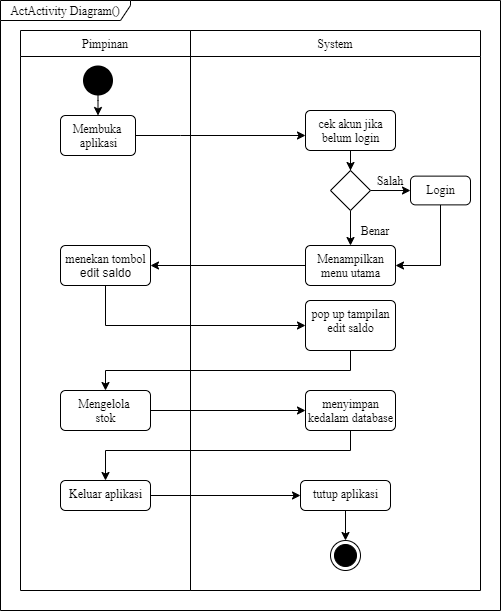
Gambar 3.11 *Activity Diagram* Menampilkan Rekap Harian

Penjelasan:

Diawali dengan pimpinan atau pegawai membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pegawai atau pimpinan memilih rekap harian, kemudian sistem menampilkan hasil rekap, jika pengguna menutup aplikasi sistem akan menutup.

1. *Activity diagram* Menampilkan dan mengelola stok

Hasil pembuatan *Activity diagram* Menampilkan dan mengelola stok yang dilakukan oleh pimpinan yang dapat dilihat pada Gambar 3.12.



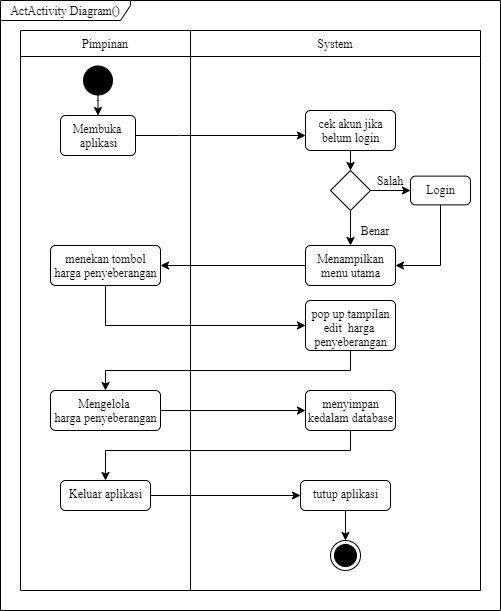
Gambar 3.12 *Activity diagram* Menampilkan dan mengelola stok

Penjelasan:

Diawali dengan pimpinan membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pimpinan menekan tombol kelola stok yang ada pada menu utama, kemudian sistem menampilkan tampilan untuk mengubah stok, kemudian pimpinan melakukan perubahan stok dan sistem melakukan penyimpanan data ke *database*, jika pimpinan menutup aplikasi sistem akan menutup.

1. *Activity diagram* Menampilkan dan mengelola harga pelabuhan

Hasil pembuatan *Activity diagram* Menampilkan dan mengelola harga pelabuhan yang dilakukan oleh pimpinan yang dapat dilihat pada Gambar 3.13.



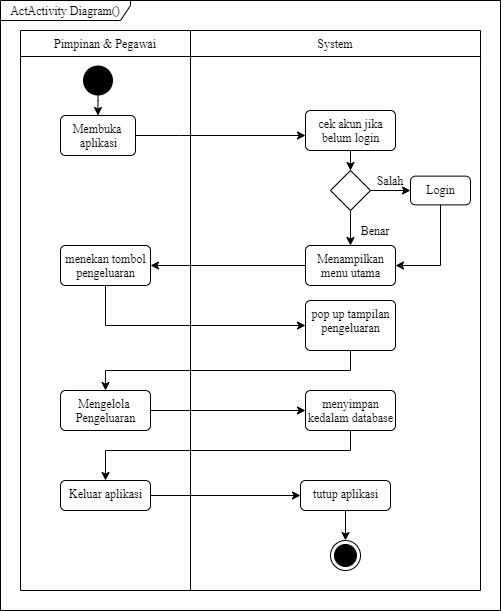
Gambar 3.13 *Activity Diagram* Menampilkan dan Mengelola Harga Pelabuhan

Penjelasan:

Diawali dengan pimpinan membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pimpinan menekan tombol harga pelabuhan yang ada pada menu utama, kemudian sistem menampilkan tampilan harga pelabuhan dan pimpinan dapat langsung mengubah data penyeberangan dan sistem melakukan penyimpanan data ke *database*, jika pimpinan menutup aplikasi sistem akan menutup.

1. *Activity diagram* Menambahkan pengeluaran

Hasil pembuatan *Activity diagram* menambahkan pengeluaran yang dilakukan oleh pimpinan dan pegawai yang dapat dilihat pada Gambar 3.14.



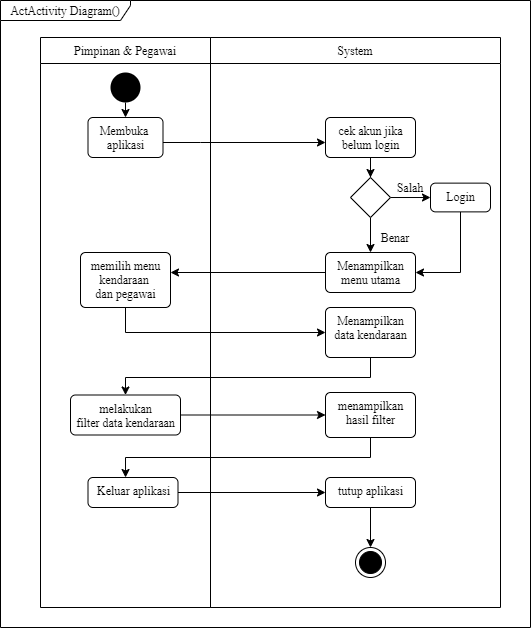
Gambar 3.14 *Activity Diagram* Menambahkan Pengeluaran

Penjelasan:

Diawali dengan pimpinan atau pegawai membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pimpinan atau pegawai menekan tombol pengeluaran yang ada pada menu utama, kemudian sistem menampilkan tampilan pengeluaran dan pimpinan pegawai dapat langsung menambah pengeluaran dan sistem melakukan penyimpanan data ke *database*, jika pimpinan menutup aplikasi sistem akan menutup.

1. *Activity diagram* filter data kendaraan

Hasil pembuatan *Activity diagram* filter data kendaraan yang dilakukan oleh pimpinan dan pegawai yang dapat dilihat pada Gambar 3.15.



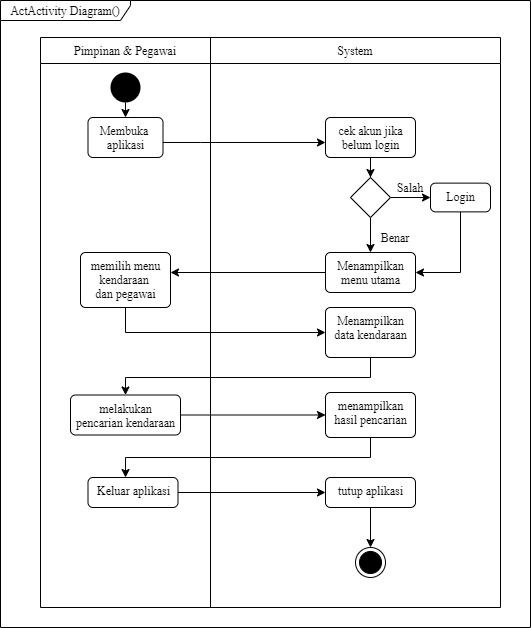
Gambar 3.15 *Activity Diagram* Filter Data Kendaraan

Penjelasan:

Diawali dengan pimpinan atau pegawai membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pegawai atau pimpinan memilih menu data kendaraan dan pegawai kemudian sistem menampilkan halaman data kendaraan selanjutnya menekan fungsi filter data yang disediakan dan sistem akan menampilkan hasil filter, setelah melakukan filter data jika pengguna menutup aplikasi sistem akan menutup.

1. *Activity diagram* pencarian data kendaraan

Hasil pembuatan *activity diagram* pencarian data kendaraan yang dilakukan oleh pimpinan dan pegawai dapat dilihat pada Gambar 3.16.



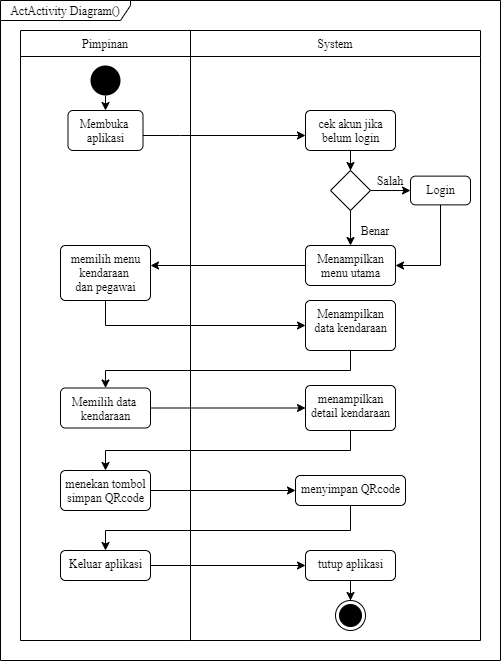
Gambar 3.16 *Activity Diagram* Pencarian Data Kendaraan

Penjelasan:

Diawali dengan pimpinan atau pegawai membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pengguna atau pimpinan memilih menu data kendaraan dan pegawai kemudian sistem menampilkan data pegawai selanjutnya menekan fungsi pencarian data yang disediakan dan sistem akan menampilkan hasil pencarian, setelah melakukan pencarian data jika pengguna menutup aplikasi sistem akan menutup.

1. *Activity diagram* simpan *QR-code*

Hasil pembuatan *activity diagram* simpan *QR-code* yang dilakukan oleh pimpinan dan pegawai dapat dilihat pada Gambar 3.17.



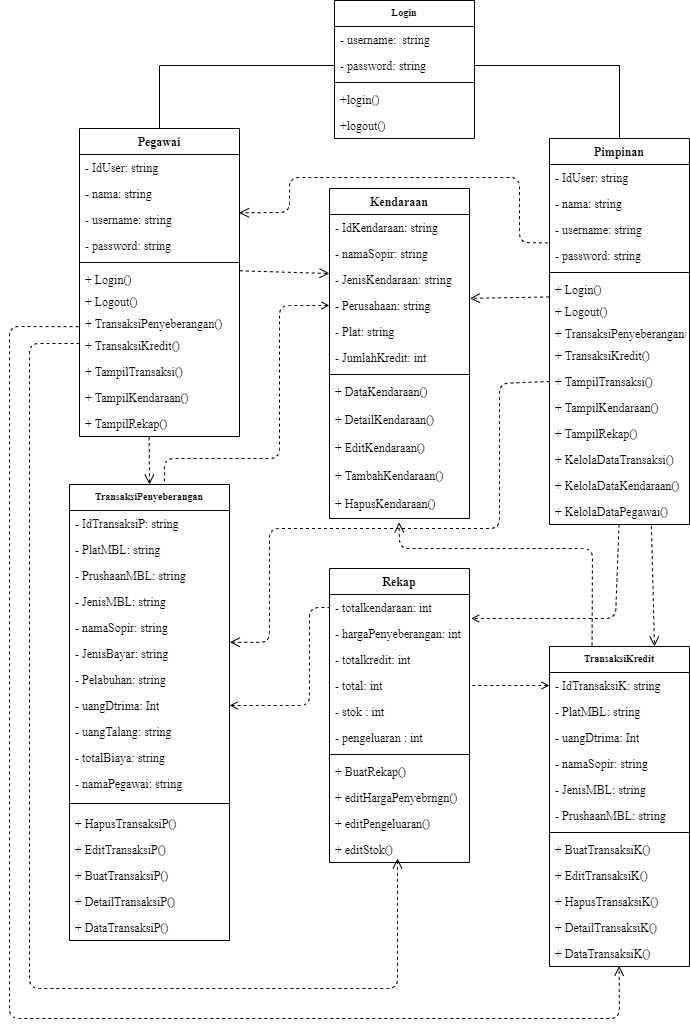
Gambar 3.17 Activity Diagram Simpan QR-Code

Penjelasan:

Diawali dengan pimpinan atau pegawai membuka aplikasi selanjutnya sistem melakukan pengecekan apakah sudah sudah login atau belum, jika terdeteksi belum melakukan login, sistem menampilkan halaman login jika sudah, sistem menampilkan halaman utama, lalu pengguna atau pimpinan memilih menu pegawai dan kendaraan kemudian sistem menampilkan halaman pegawai dan kendaraan selanjutnya memilih data kendaraaan yang ingin disimpan *QR-code* nya selanjutnya sistem akan menampilkan detail kendaraan dan pegawai atau pimpinan, kemudian menekan tombol simpan *QR-code* sistem akan langsung menyimpan *QR-code* ke folder, setelah melakukan simpan *QR-code* jika pengguna menutup aplikasi sistem akan menutup.

### 3.4.3 *Class Diagram*

*Class diagram* yang dibuat untuk menggambarkan suatu struktur dari kelas-kelas yang akan digunakan dalam mengembangkan sistem pada penelitian ini dapat dilihat pada Gambar 3.18.

******

Gambar 3.18 *Class diagram* aplikasi pendataan transaksi

### 3.4.1 *Class Responsibility Collaborator Card* (*CRC Card*).

Berikut ini merupakan *CRC Card* yang terdiri dari *class*, *responsibility*, dan *collaborator* dari sistem yang akan dibuat yang di mana pegawai dan pimpinan sudah melakukan login terlebih dahulu:

Tabel 3.12 *CRC Card* AUTH

|  |  |
| --- | --- |
| **Nama *Class*: *Auth*** | |
| ***Responsibility*** | ***Collaborator*** |
| Dapat login ke sistem | Pegawai |
| Pimpinan |

Tabel 3.13 *CRC Card* Pegawai

|  |  |
| --- | --- |
| **Nama *Class*: Pegawai** | |
| ***Responsibility*** | ***Collaborator*** |
| Melakukan transaksi penyeberangan | TransaksiPenyeberangan |
| Menampilkan data transaksi |
| Menampilkan data kreditur | Kreditur |
| Melakukan transaksi kreditur |
| Menampilkan data kendaraan | Kendaraan |
| Menampilkan data rekap harian | Rekap |

Tabel 3.14 *CRC Card* Pimpinan

|  |  |
| --- | --- |
| **Nama *Class*: Pimpinan** | |
| ***Responsibility*** | ***Collaborator*** |
| Melakukan transaksi penyeberangan | TransaksiPenyeberangan |
| Menampilkan dan mengelola data transaksi |
| Melakukan transaksi kreditur | TransaksiKreditur |
| Menampilkan dan mengelola data kreditur |
| Menampilkan dan mengelola data kendaraan | Kendaraan |
| Menampilkan dan engelola data pegawai | Pegawai |
| Menampilkan data rekap harian | Rekap |

Tabel 3.15 *CRC Card* Transaksi Penyeberangan

|  |  |
| --- | --- |
| **Nama *Class*: TransaksiPenyeberangan** | |
| ***Responsibility*** | ***Collaborator*** |
| Menghapus data transaksi penyeberangan |  |
| Mengubah data transaksi penyeberangan |
| Membuat transaksi penyeberangan | Kendaraan |
| Membuat detail transaksi |

Tabel 3.16 *CRC Card* Kendaraan

|  |  |
| --- | --- |
| **Nama *class*: Kendaraan** | |
| ***Responsibility*** | ***Collaborator*** |
| Mengubah data kendaraan |  |
| Menambah data kendaraan |
| Menghapus data kendaraan |

Tabel 3.17 *CRC Card* rekap

|  |  |
| --- | --- |
| **Nama *class*: Rekap** | |
| ***Responsibility*** | ***Collaborator*** |
| Membuat rekap | TransaksiPenyeberangan |
| TransaksiKreditur |

Tabel 3.18 *CRC Card* Transaksi Kreditur

|  |  |
| --- | --- |
| **Nama *class*: TransaksiKreditur** | |
| ***Responsibility*** | ***Collaborator*** |
| Menampilkan data transaksi kreditur | Kendaraan |
| Membuat detail transaksi kreditur |

### 3.4.2 *Prototype* *Design*

Pada tahap ini pembuatan *prototype* *user interface* sesuai dengan kebutuhan menggunakan Figma. Bertujuan untuk memberi gambaran aplikasi yang akan dibuat kepada klien. Berikut ini merupakan tampilan yang telah penulis buat:

1. Halaman *login*

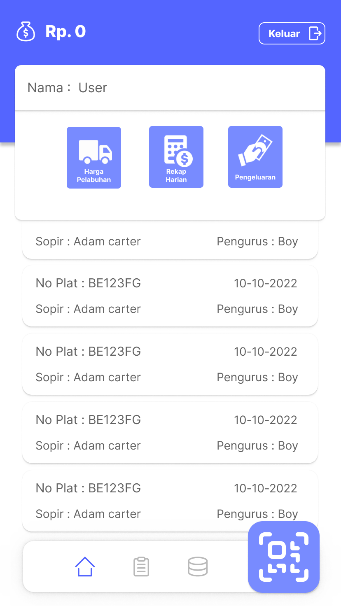
Tampilan login merupakan tampilan awal ketika mengakses aplikasi. Pada tampilan ini sebagai pemisah antara *user* pegawai dan pimpinan, kegiatan harus dilakukan pengguna yaitu memasukkan *user*name dan password, tampilan login dapat dilihat pada Gambar 3.19.



Gambar 3.19 Tampilan *Login*

1. Halaman home pimpinan dan karyawan

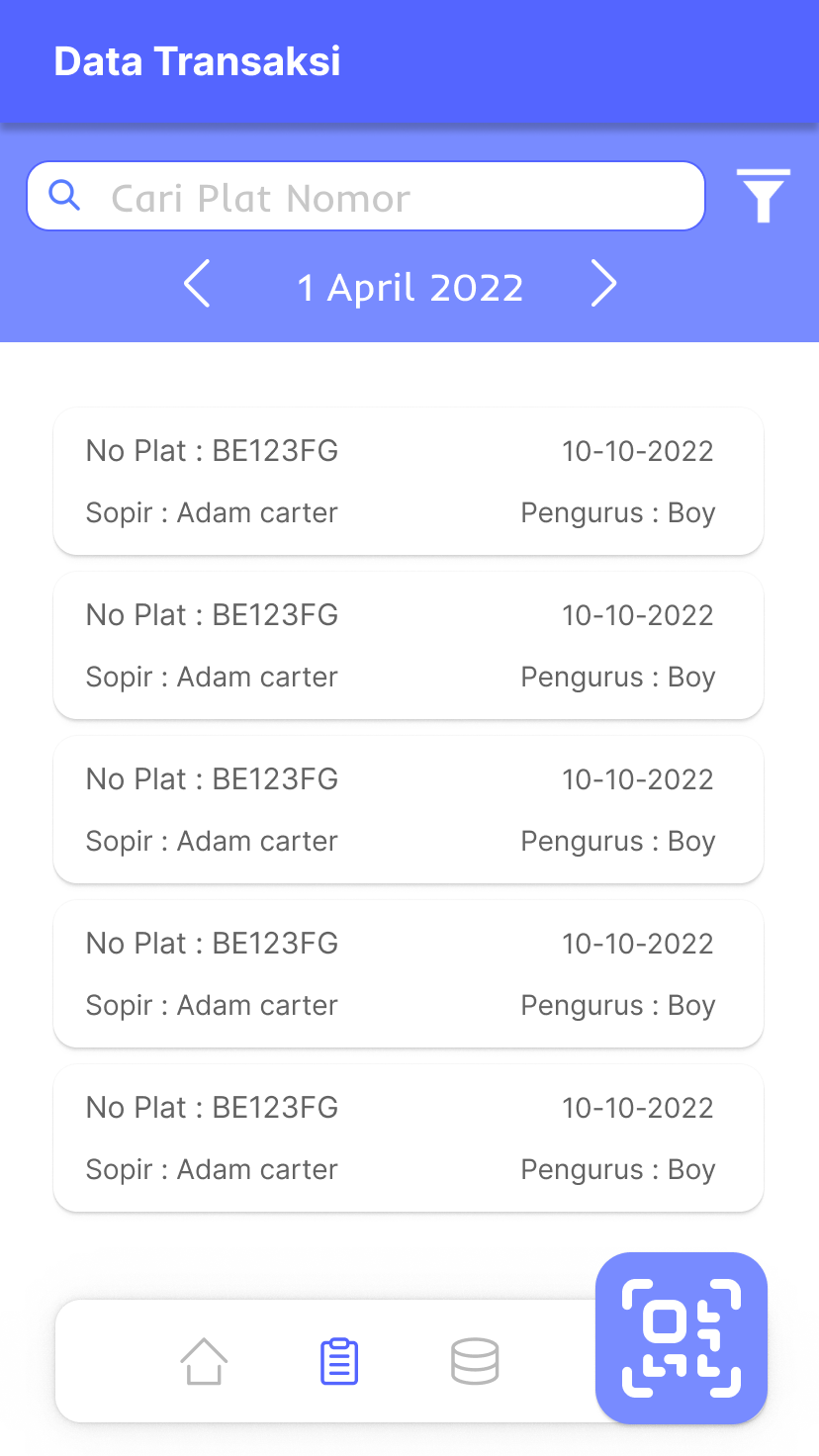
Ketika pegawai maupun pimpinan berhasil *login* maka akan muncul halaman *home*, pada tampilan awal terdapat jumlah saldo, list transaksi secara *real time*, dan tombol yang berguna untuk melakukan pemindan. halaman *home* pengguna dapat dilihat pada Gambar 3.20.



Gambar 3.20 Halaman Home

1. Halaman data transaksi

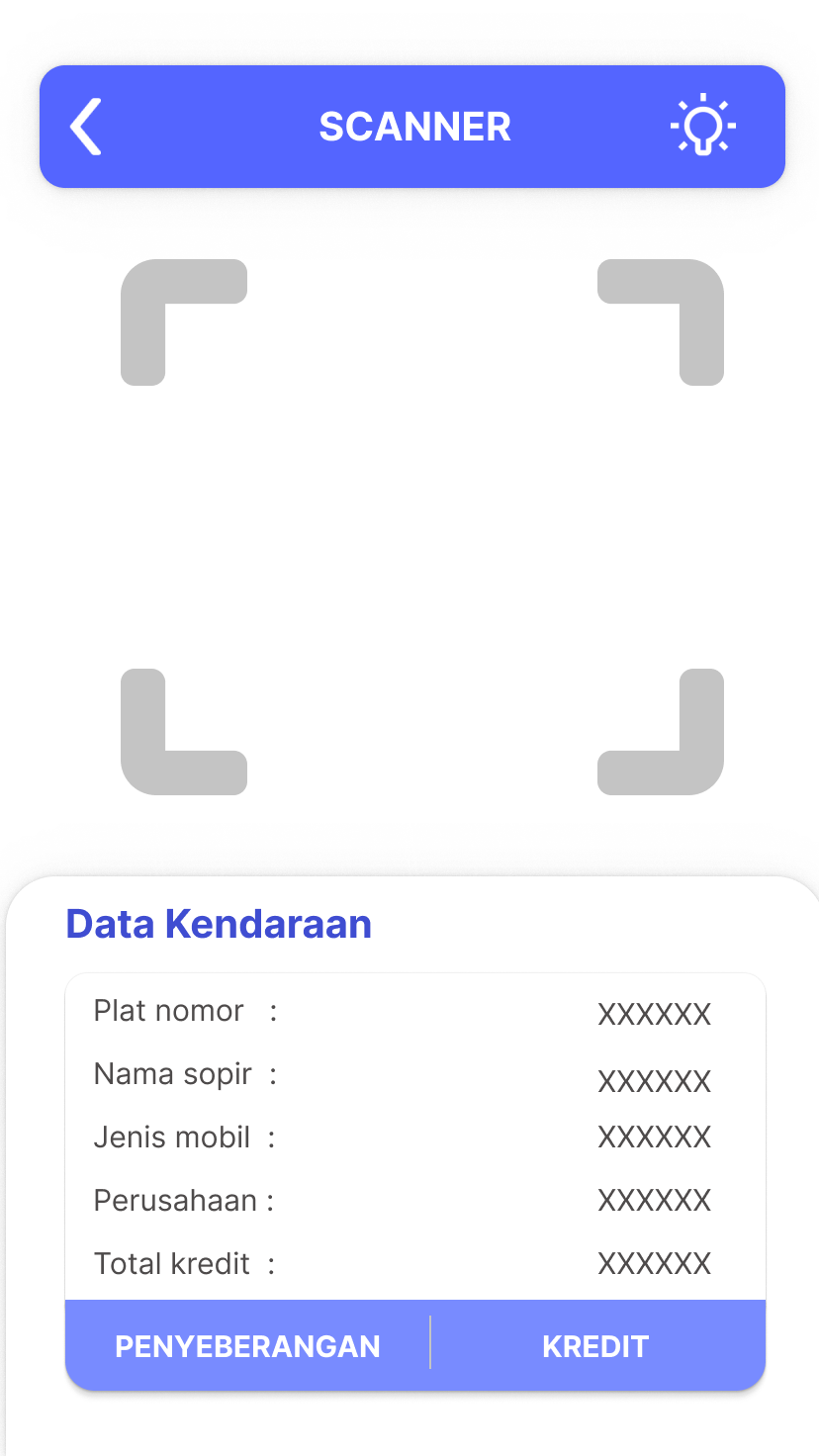
Pada halaman transaksi pengguna dapat melakukan pencariaan data transaksi dan melihat detail transaksi, pada halaman ini pimpinan dapat merubah detail transaksi jika ada kesalahan, tampilan data transaksi dapat dilihat pada gambar 3.21.



Gambar 3.21 Halaman Data Transaksi

1. Halaman *Scanner*

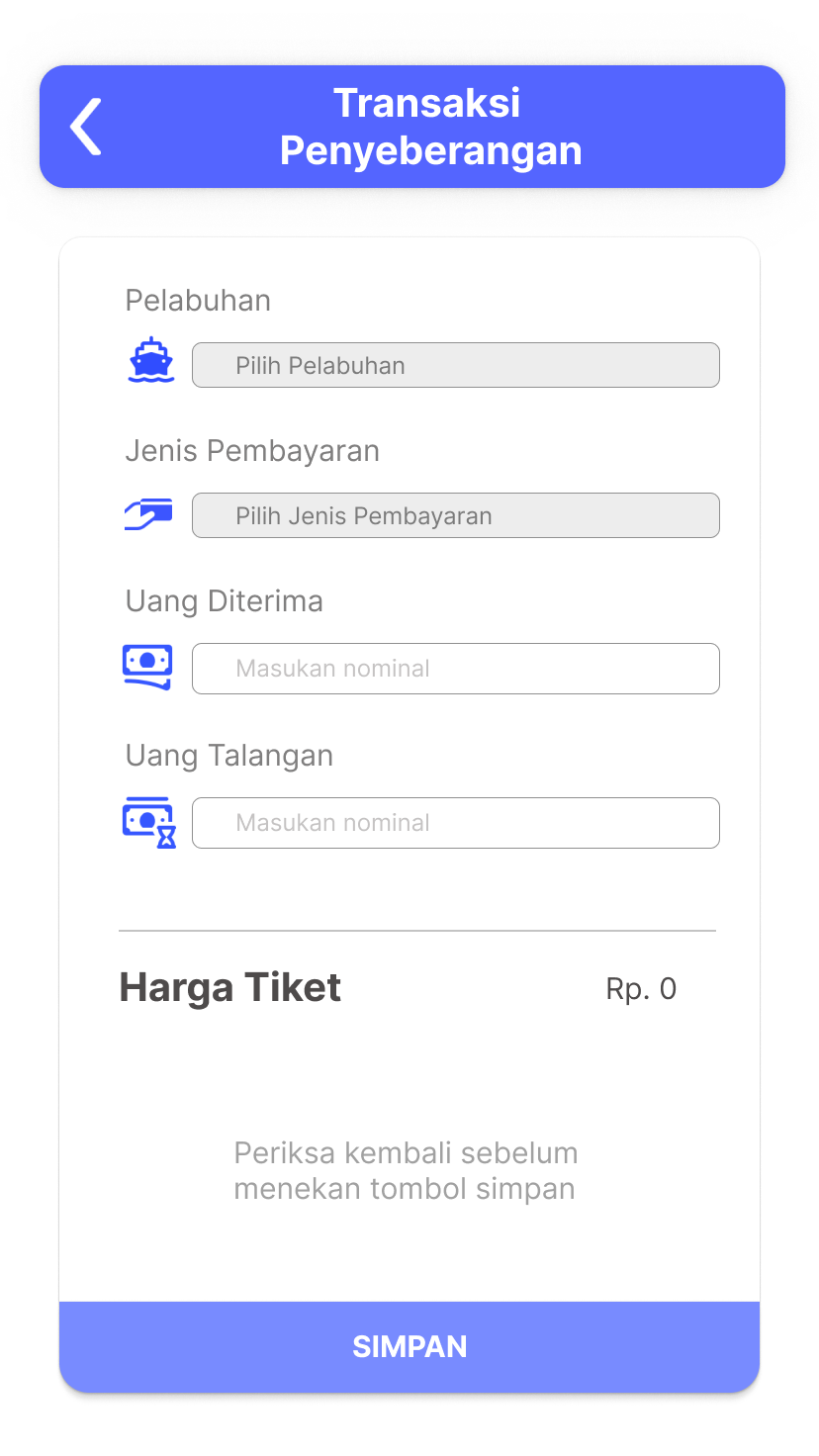
Halaman *scanner* berguna untuk pemindaian *QR-code* kendaraan sebelum melakukan transaksi, pada tampilan ini dapat menampilkan data kendaraan setelah melakukan pemindaian untuk memastikan kendaraan yang melakukan transaksi sesuai, tampilan *scanner* dapat dilihat pada gambar 3.22.



Gambar 3.22 Halaman *Scanner*

1. Halaman proses transaksi penyeberangan

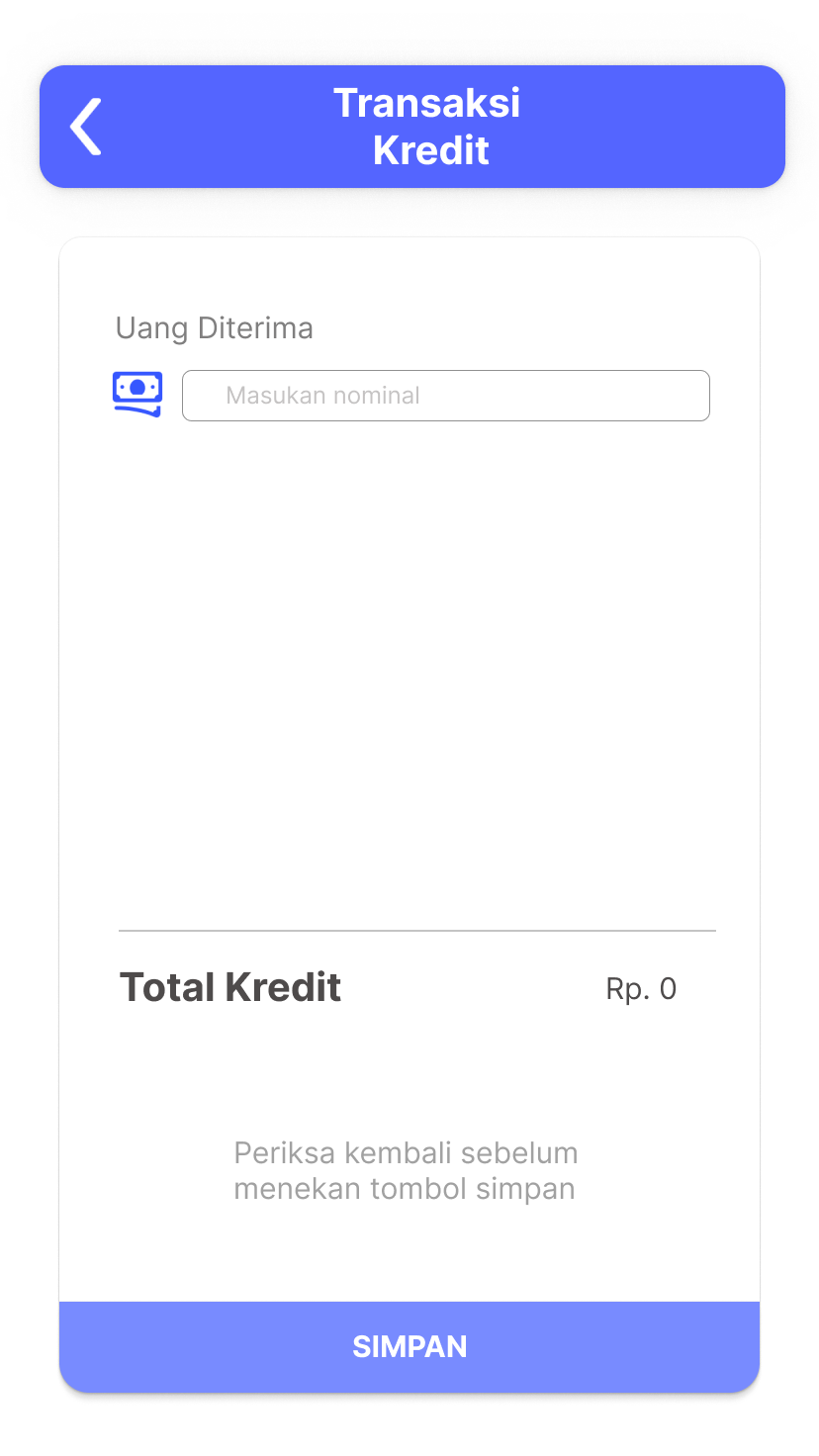
Halaman proses transaksi berguna untuk menginputkan data transaksi penyeberangan seperti jenis Pelabuhan, jenis pembayaran dan uang diterima dan uang talangan, halaman proses transaksi penyeberangan dapat dilihat pada gambar 3.23.



Gambar 3.23 Halaman Transaksi Penyeberangan

1. Halaman proses transaksi kredit

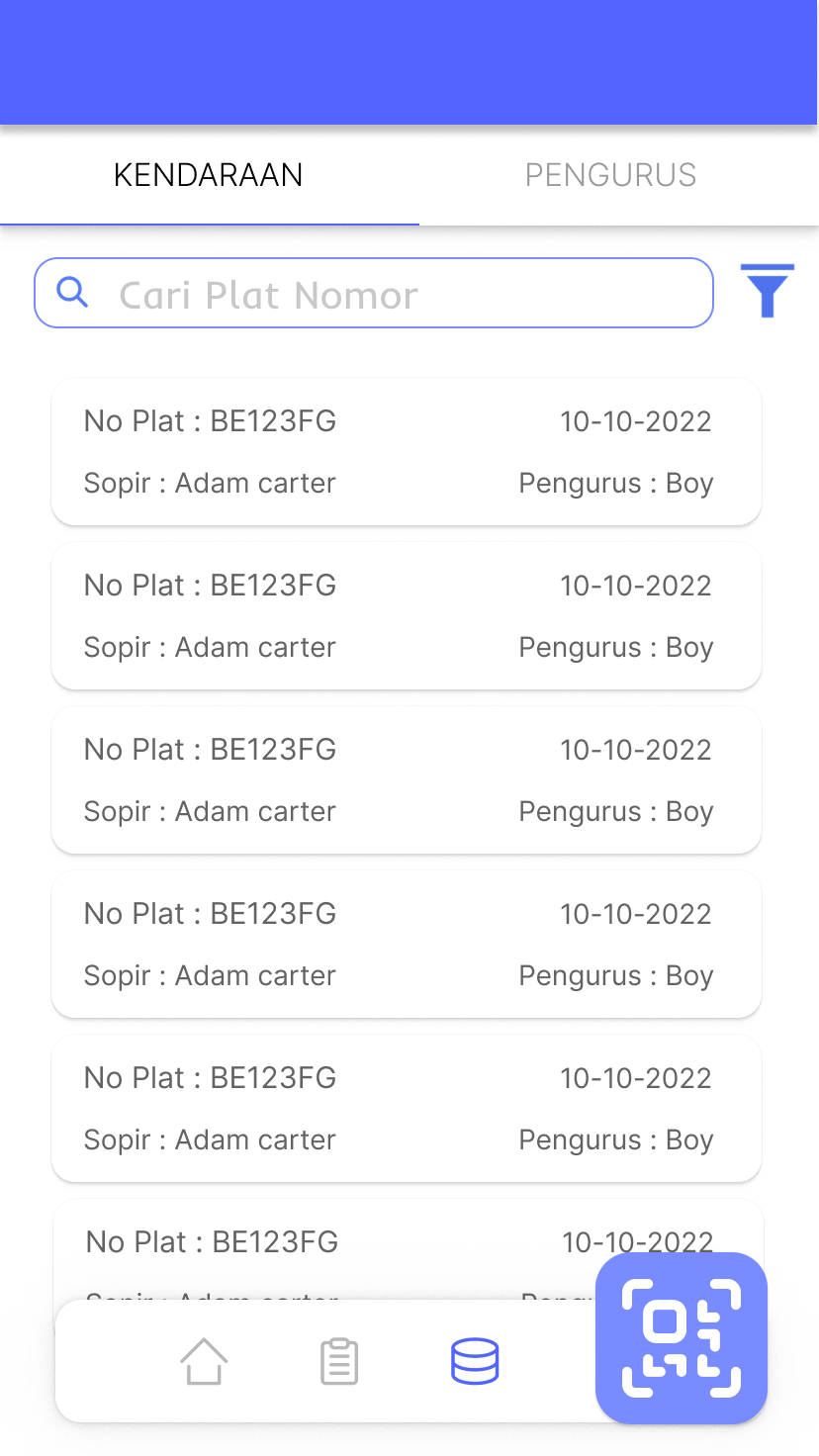
Halaman proses transaksi kreditur berguna untuk melakukan transaksi pembayaran kredit dan sebelum melakukan transaksi harus melakukan pemindaian *QR-code* terlebih dahulu, halaman proses transaksi kreditur dapat di lihat pada gambar 3.24.



Gambar 3.24 Halaman Transaksi *Kredit*

1. Halaman kendaraan dan pegawai

Halaman data kendaraan dan pegawai berguna untuk menampilkan data kendaraan dan data pegawai, pada halaman ini pegawai hanya bisa melihat detail data kendaraaan dan pegawai sedangkan pimpinan dapat melakukan edit, hapus dan menambah data, halaman data dapat dilihat pada gambar 3.25.



Gambar 3.25 Halaman Data Kendaraan Dan Pegawai

1. Halaman harga pelabuhan.

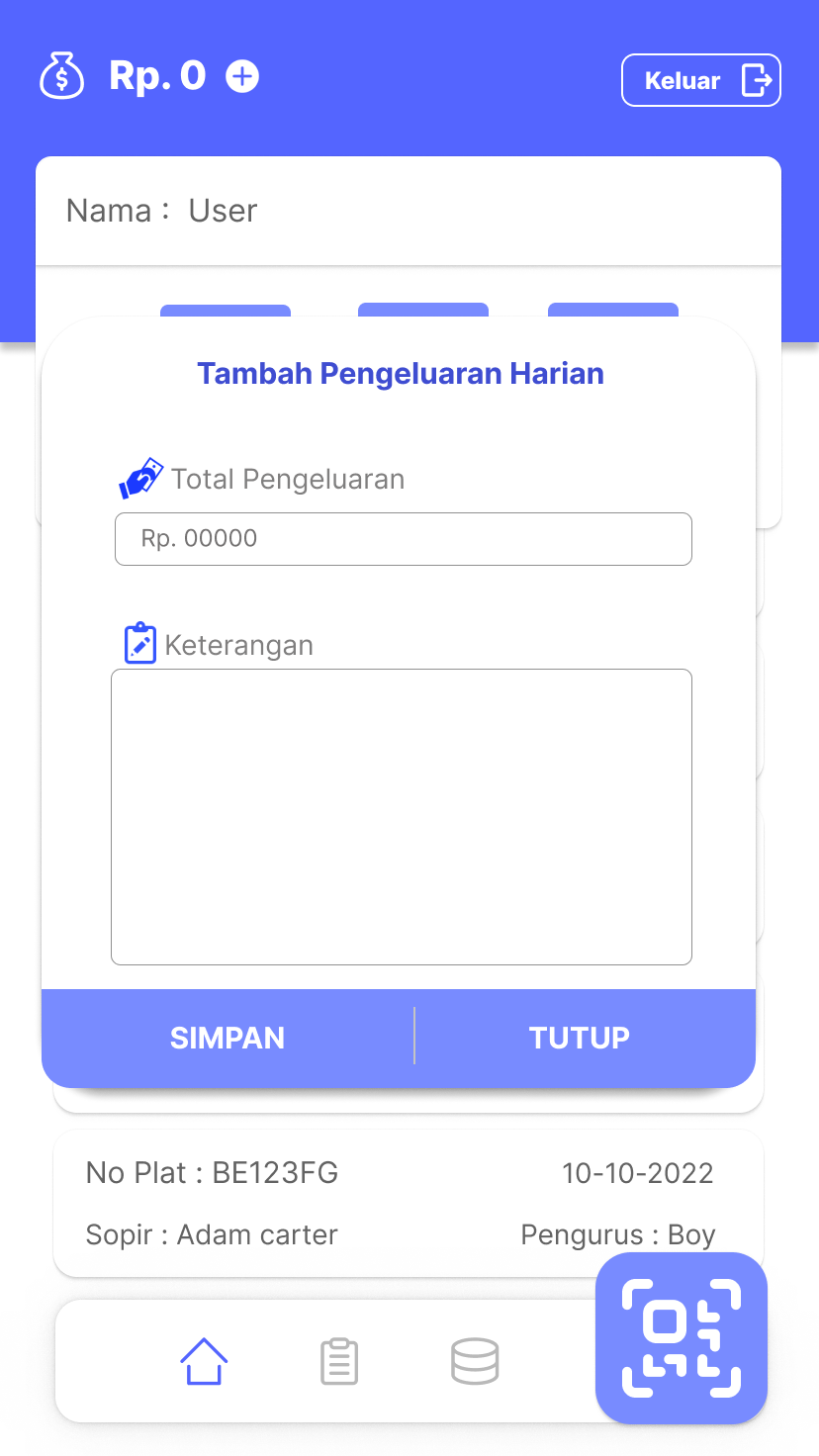
Pada halaman ini terdapat fitur untuk menampilkan harga-harga pelabuhan sesuai dengan jenis kendaraan dan jenis pelabuhan, fitur ini dapat digunakan oleh pimpinan dan pegawai, selanjutnya di dalam halaman dapat mengubah harga pelabuhan yang hanya bisa dilakukan oleh, halaman dapat dilihat pada gambar 3.26.



Gambar 3.26 Halaman Harga Pelabuhan

1. Halaman tambah pengeluaran harian

Pada halaman ini memiliki fungsi menambah pengeluaran harian yang berisi text input untuk memasukan nilai pengeluaran dan keterangan pengeluaran. Halaman dapat dilihat pada gambar 3.267.



Gambar 3.27 Halaman Tambah Pengeluaran Harian

1. Halaman rekap harian

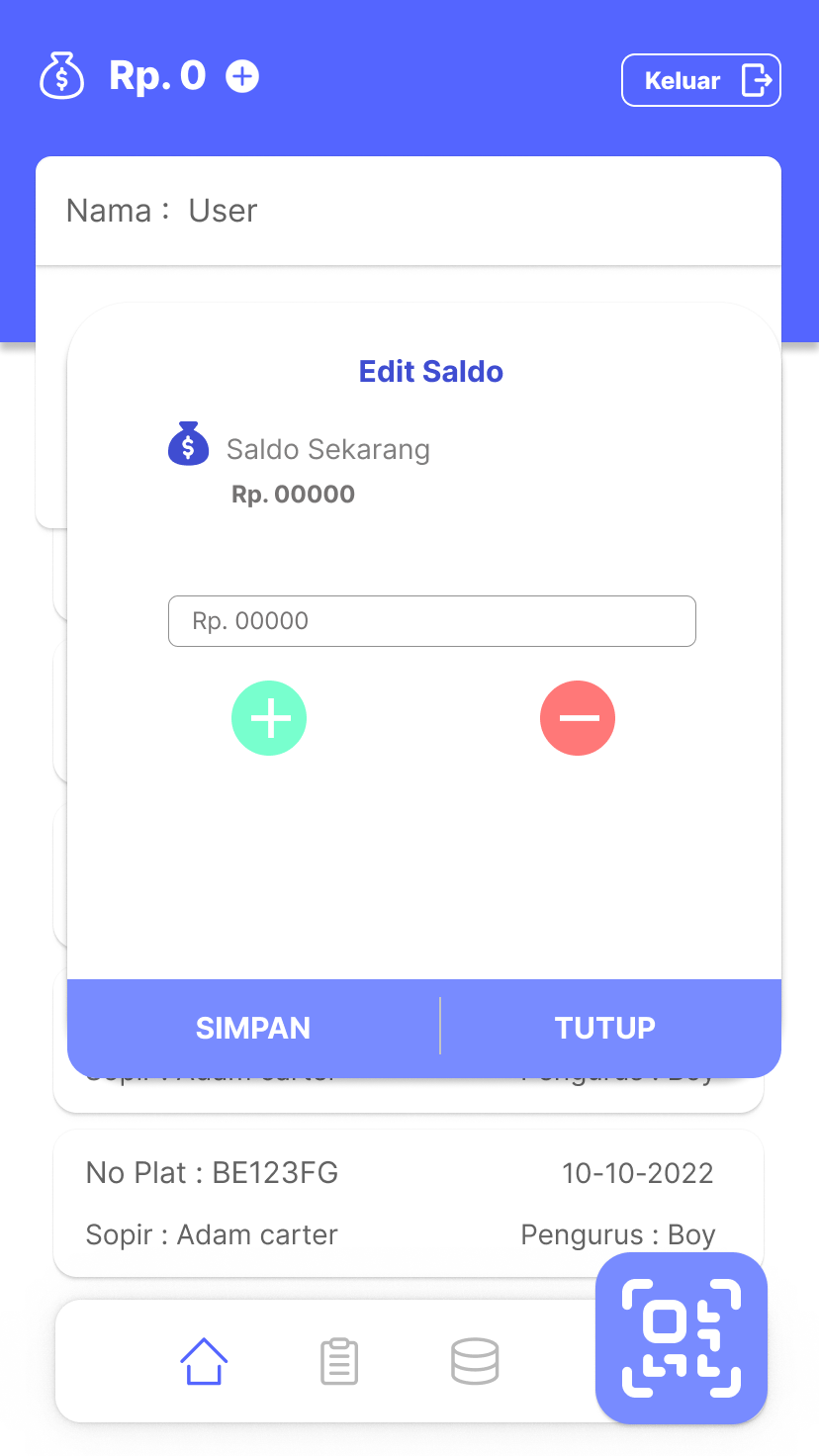
Pada halaman ini berfungsi menghitung dan menampilkan hasil rekap setiap harinya, perhitungan rekap dilakukan oleh sistem agar mempermudah jasa dalam perekapan harian, halaman ini dapat diakses oleh pimpinan dan pegawai. Halaman dapat dilihat pada gambar 3.28.



Gambar 3.28 Halaman Rekap Harian

1. Halaman edit saldo

Fungsi utama dari halam ini yaitu mengubah saldo yang hanya dapat di akses oleh pimpinan, pimpinan dapat mengurangi dan menambahkan saldo yang ada jika dibutuhkan. Halaman dapat dilihat pada gambar 2.29.



Gambar 3.29 Halaman Edit Saldo

## 3.5 Jadwal Penelitian

Dibawah ini adalah jadwal penelitian proposal skripsi yang telah dibuat oleh penulis:

**Tabel 3.18** Jadwal Penelitian

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Kegiatan Penelitian** | **Tahun 2022** | | | | | | | | | | | | | | | |
| **Maret** | | | | **April** | | | | **Mei** | | | | **Juni** | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1. | Wawancara |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2. | Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3. | Perencanaan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4. | Perancangan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5. | Penyusunan Proposal Skripsi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6. | Seminar Proposal Skripsi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7. | Pengkodean |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8. | Pengujian |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9. | Sidang Skripsi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Keterangan:

= Telah Dilaksanakan

= Belum Dilaksanakan

# DAFTAR PUSTAKA

Alham Romadony, Willi, Muhammad Ardianto, Wisnu Kartiko Arie Pangestu, Didih Rizki Chandranegara, Wildan Suharso, Universitas Muhammadiyah Malang, Malang Kontak Person, and Wildan Suharso Jalan Raya Tlogomas No. 2019. “Sistem Pencatatan Keuangan Toko Berbasis Android.” *Seminar Nasional Teknologi Dan Rekayasa (SENTRA)2019* 2527–6050.

Ariyanti, Lisa, Muhammad Najib, Dwi Satria, and Debby Alita. 2020. *Sistem Informasi Akademik Dan Administrasi Dengan Metode Extreme Programming Pada Lembaga Kursus Dan Pelatihan*. Vol. 1.

Artina, Nyimas. 2006. “Penerapan Analisis Kebutuhan Metode *Use Case* Pada Metode Pengembangan Terstruktur.” *Jurnal Ilmiah STMIK G1 MDP* 2(3):1–6.

Ayu, Vittalis. 2017. *Pemodelan Proses Pemilihan Rute Pada Protokol Babel Dengan Activity Diagram Dan Transition System*. Vol. 12.

Cahyati, Yuliana, and Hari Murti. 2018. *Sistem E-Surat Pada Government Resource Management System Provinsi Jawa Tengah Berbasis Framework Codeigniter*.

Fatoni, Ahmad, and Dhany Dwi. 2016. “Rancang Bangun Sistem Extreme Programming Sebagai Metodologi Pengembangan Sistem.” 3(1).

Gultom, Dedek K., Paham Ginting, and Beby Kf Sembiring. 2014. *Pengaruh Bauran Pemasaran Jasa Dan Kualitas Pelayanan Terhadap Kepuasan Mahasiswa Program Studi Manajemen Fakultas Ekonomi Universitas Muhammadiyah Sumatera Utara*. Vol. 14.

Hansson, Niclas, and Tomas Vidhall. 2016. *Institutionen För Datavetenskap Effects on Performance and Usability for Cross-Platform Application Development Using React Native*.

Hendini, Ade. 2016. *Pemodelan Uml Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak)*. Vol. IV.

Iskandar, and Umar Tsani Abdurrahman. 2020. “Perancangan Aplikasi Kasir *Point Of Sales* Berbasis Android Menggunakan Metode Rapid Application Development Untuk Usaha Retail.” *INFOTECH: Jurnal Informatika & Teknologi* 1(2):67–77. doi: 10.37373/infotech.v1i2.62.

Ismail, Muh, Al Ghazali Syam, Riwayat Artikel, and Kata kunci. 2021. *Aplikasi Qr Code Sebagai Sarana Penyampaian Informasi Pohon Dikebun Raya Jompie Informasi Artikel*. Vol. 1.

Juhardi, Ujang, and Khairullah. 2019. “Sistem Pencatatan Dan Pengolahan Keuangan Pada Aplikasi Manajemen Keuangan E-Dompet Berbasis Android.” *JTIS* 2(1):2614–3089.

Loveri, Tomi Tm. 2018. “Sistem Informasi Aplikasi Pengelolaan Transaksi Keuangan Dan Pendataan Konsumen Pada Cv. Puplas.” *Jurnal Sains Dan Informatika* 4(2):139. doi: 10.22216/jsi.v4i2.3584.

Mubarak, Abdul, Jl Jati Metro, and Kota Ternate Selatan. 2019. *Rancang Bangun Aplikasi Web Sekolah Menggunakan Uml (Unified Modeling Language) Dan Bahasa Pemrograman Php (Php Hypertext Preprocessor) Berorientasi Objek*. Vol. 02.

Nugraha, Made Muliana, Nur Muhammad Luthf, and Hetti Hidayati. 2020. “Aplikasi Pengelolaan Sport ClubUniversitas TelkomBerbasis Website.” *E-Proceeding of Applied Science* 6(2):2562.

Saputra, Agung Dwi, Agi Putra Kharisma, and Lutfi Fanani. 2021. *Pengembangan Aplikasi Klinik Kecantikan Sebagai Pengelola Transaksi Berbasis Android Menggunakan Metode Prototype (Studi Kasus :Klinik Kecantikan CV Nana Beautyskin)*. Vol. 5.

Solikin, Imam. 2014. “Perancangan Sistem Informasi Penjualan Berbasis Framework MVC Pada PT Thamrin Brother Perancangan Sistem Infomasi Penjualan Berbasis Framework Model View Controller (Mvc) Pada Pt Thamrin Brother Cabang Oki.” *Jurnal Media Informatika Dan Komputer* 4(1).

Sovia Maranti, Ovi, Lis Saumi Ramdhani, Rusli Nugraha, Khairul Rizal, Stmik Nusa, and Mandiri Sukabumi. 2018. “Rancang Bangun Aplikasipengelolaanpinjaman Koperasi Berbasis *Mobil*epada Koperasi Pkk Sejahtera Sukabumi.” *Jurnal Swabumi* 6(1).

Widodo. 2008. *Extreme Programming : Pengembangan Perangkat Lunak Semi Formal*.

# LAMPIRAN

|  |
| --- |
| **DOKUMEN SURAT PENELITIAN** |
| \ |

Lampiran 1. Dokumen Penelitian

Lampiran 2. Dokumen hasil kegiatan wawancara

|  |
| --- |
| **DOKUMEN HASIL WAWANCARA** |
|  |

Lampiran 3. Foto kegiatan wawancara

|  |
| --- |
| **FOTO WAWANCARA** |
|  |