

**E-COMMERCE KOMODITAS BERBASIS PWA PADA PT.BEJO
AGRO LESTARI MENGGUNAKAN FRAMEWORK LARAVEL**

SKRIPSI

**Diajukan Untuk Memenuhi Persyaratan Gelar Sarjana Komputer
Pada Program Studi Sistem Informasi Fakultas Teknik dan Komputer
Universitas Harapan Medan**



**DAVID GLADIO SITUMORANG
182370080**

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK DAN KOMPUTER
UNIVERSITAS HARAPAN MEDAN
MEDAN
2022**

**E-COMMERCE KOMODITAS BERBASIS PWA PADA PT.BEJO
AGRO LESTARI MENGGUNAKAN FRAMEWORK LARAVEL**

**DAVID GLADIO SITUMORANG
182370080**

SKRIPSI

**Diajukan Untuk Memenuhi Persyaratan Gelar Sarjana Komputer
Pada Program Studi Sistem Informasi Fakultas Teknik dan Komputer
Universitas Harapan Medan**

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK DAN KOMPUTER
UNIVERSITAS HARAPAN MEDAN
MEDAN
AGUSUTUS, 2022**

PERNYATAAN PEMBIMBING

Saya/Kami dengan ini menyatakan bahwa saya/kami telah memeriksa Skripsi Mahasiswa ini dan menurut pendapat saya/kami, Skripsi Mahasiswa ini telah mencukupi untuk ruang lingkup dan kualitas untuk dianugerahkan gelar Sarjana Komputer dalam bidang Sistem Informasi.

Medan, Agustus 2022

Pembimbing 1

Pembimbing 2

(Yulia Agustina Dalimunthe, S.Si., M.Kom) (Septiana Dewi Andriana, S.Kom., M.Kom)

PERNYATAAN MAHASISWA

Saya yang bertanda tangan di bawah ini:

Nama : David Gladio Situmorang
NPM : 182370080
Nama Orang Tua : Asner Situmorang
Fakultas : Teknik dan Komputer
Program Studi : Sistem Informasi
Jenjang Studi : Strata-1

menyatakan bahwa :

1. Skripsi ini merupakan gagasan, rumusan dan ide saya sendiri, tanpa bantuan dari pihak lain kecuali arahan dari tim Dosen Pembimbing.
2. Skripsi ini belum pernah diajukan untuk mendapatkan gelar sarjana, baik di Program Studi Sistem Informasi Fakultas Teknik dan Komputer Universitas Harapan Medan maupun di Perguruan Tinggi lain.
3. Dalam Skripsi ini tidak terdapat karya atau pendapat yang ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan mencantumkan sebagai acuan dalam naskah dengan menyebut nama pengarang dan dicantumkan dalam daftar pustaka.

Demikianlah pernyataan ini saya perbuat dengan sesungguhnya dan apabila dikemudian hari terbukti pernyataan ini tidak benar, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diberikan melalui karya tulis ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi ini.

Medan, Agustus 2022
Yang Menyatakan

David Gladio Situmorang
182370080

PERSETUJUAN

JUDUL : E-COMMERCE KOMODITAS BERBASIS PWA
PADA PT. BEJO AGRO LESTARI
MENGUNAKAN FRAMEWORK LARAVEL

KATEGORI : SKRIPSI

NAMA : DAVID GLADIO SITUMORANG

NOMOR POKOK MAHASISWA : 182370080

FAKULTAS : TEKNIK DAN KOMPUTER

PROGRAM STUDI : SISTEM INFORMASI

TAHUN TAMAT : 2022

DISETUJUI OLEH

Komisi Pembimbing

Pembimbing 1

Pembimbing 2

Yulia Agustina Dalimunthe, S.Si., M.Kom Septiana Dewi Andriana, S.Kom., M.Kom

Mengetahui,

Ketua Program Studi Sistem Informasi

Edy Rahman Syahputra, S.T., M.Kom

KATA PENGANTAR

Puji dan syukur penulis ucapkan kepada tuhan yang maha kuasa atas limpahan berkat, rahmat, serta kemudahan yang telah diberikan sehingga penulis dapat menyelesaikan Skripsi ini yang merupakan syarat untuk mendapatkan gelar Sarjana Komputer pada Program studi Sistem Informasi, Fakultas Teknik dan Komputer, Universitas Harapan Medan.

Dalam kurun waktu pengerjaan Skripsi ini penulis menyadari bahwa sangat banyak pihak yang berjasa turut membantu penulisan dalam penyelesaian Skripsi ini. Dalam kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Drs. Sriadhi, ST., M.Pd., M.Kom., Ph.D selaku Rektor Universitas Harapan Medan.
2. Bapak Abdul Jabar Lubis, ST., M.Kom selaku Dekan Fakultas Teknik dan Komputer Universitas Harapan Medan.
3. Bapak Edy Rahman Syahputra, S.T., M.Kom selaku Ketua Program Studi Sistem Informasi Fakultas Teknik dan Komputer Universitas Harapan Medan.
4. Bapak Ahmad Zakir, S.T., M.Kom selaku Sekretaris Program Studi Sistem Informasi Fakultas Teknik dan Komputer Universitas Harapan Medan.
5. Ibu Yulia Agustina Dalimunthe, S.Si., M.Kom selaku pembimbing I dan Ibu Septiana Dewi Andriana, S.kom., M.Kom selaku pembimbing II yang telah meluangkan waktu membimbing penulis selama pengerjaan Skripsi ini.
6. Ayahanda Asner Situmorang serta Ibunda Rotua Simamora atas doa dan kasih sayangnya yang tulus dan tak terhingga kepada penulis.
7. Teman-teman seperjuangan mahasiswa Program Studi Sistem Informasi, Fakultas Teknik dan Komputer, Universitas Harapan Medan angkatan 2018 yang telah memberikan motivasi dan perhatiannya.
8. Semua pihak yang terlibat baik langsung maupun tidak langsung dalam pengerjaan Skripsi ini yang tidak bisa penulis sebutkan satu persatu diucapkan terimakasih.

Penulis menyadari bahwa Skripsi ini masih belum sempurna. Oleh karena itu untuk menyempurnakan Skripsi ini, kritik dan saran yang membangun sangat penulis harapkan. Akhir kata penulis berharap semoga Skripsi ini dapat bermanfaat.

Medan, Agustus 2022

Penulis

David Gladio Situmorang

182370080

ABSTRAK

PT. Bejo Agro Lestari merupakan suatu usaha perdagangan komoditi yang berdiri sekitar awal tahun 2019 dan beroperasi di kota Medan. Perdagangan komoditi pada kopi masih bersifat B2C atau *Business to Customer*, dengan kategori *green bean* dan *roast bean*. Masalah yang ditemukan pada perusahaan saat ini yaitu penggunaan sistem masih menggunakan cara konvensional dalam melakukan penjualan terhadap produknya. Hal ini kurang efektif karena penjualan masih bersifat *door to door* dan menggunakan pesan *instant* via *WhatsApp*, *Gmail* atau konsumen datang langsung ke tempat usaha. Cara penjualan pada produk yang masih konvensional, tidak mungkin meningkatkan penjualan secara signifikan, sehingga dari permasalahan tersebut penulis mendapatkan solusi untuk menyelesaikannya yaitu dengan membangun sistem *E-Commerce* pada produk B2C atau *Business to Customer* menggunakan metode *SDLC Waterfall*, yang diharapkan bisa menyediakan informasi produk komoditi seperti biji kopi *green bean* dan *roast bean* yang berguna membantu pelanggan dalam mengetahui detail informasi produk seperti harga dan kualitasnya, serta dapat dengan mudah melakukan *update* data produk kopi secara *realtime* menggunakan *REST API web service*. Penerapan *e-commerce* berbasis *PWA* ini diharapkan dapat membantu meningkatkan penjualan produk kopi pada perusahaan.

Kata Kunci : *E-Commerce, Laravel, PWA, Roast Bean, Green Bean*

ABSTRACT

PT. Bejo Agro Lestari is a commodity trading business that was founded in early 2019 and operates in the city of Medan. Initially, trade was still limited to coffee commodities, with the terms green beans and roasted coffee, where sales of coffee commodities were B2C or Business to Customer. The problem with the current company is that the use of system still uses conventional methods in making sales to sales. This is less effective because sales are still door to door and use instant messages via WhatsApp, Gmail or consumers come directly to the place of business. How to sell products that are still conventional, it is impossible to increase sales significantly, so from this problem the authors get a solution to solve it, namely by building a PWA-based E-Commerce system on B2C or Business to Customer products with the SDLC Waterfall method, which is expected to provide information commodity products such as green beans and roast beans, which are useful for helping customers find detailed product information such as price and quality, and can easily update coffee product data in real-time using the REST API web service. The application of PWA-based e-commerce is expected to help increase the company's coffee product sales.

Keywords : E-Commerce, Laravel, PWA, Roast Bean, Green Bean

DAFTAR ISI

| | Halaman |
|---|---------|
| PERNYATAAN PEMBIMBING | i |
| PERNYATAAN MAHASISWA | iv |
| PERSETUJUAN | v |
| KATA PENGANTAR | v |
| ABSTRAK | vi |
| ABSTRACT | vii |
| DAFTAR ISI | viii |
| DAFTAR TABEL | xi |
| DAFTAR GAMBAR | xii |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 3 |
| 1.3 Batasan Masalah | 3 |
| 1.4 Tujuan Penelitian | 3 |
| 1.5 Manfaat Penelitian | 3 |
| BAB II LANDASAN TEORI | 4 |
| 2.1 Pengertian Sistem | 4 |
| 2.2 Pengertian Informasi | 4 |
| 2.3 Pengertian Sistem Informasi | 5 |
| 2.4 <i>E-Commerce</i> | 5 |
| 2.5 B2C (<i>Business-to-Customer</i>) | 6 |
| 2.6 Komoditas | 7 |
| 2.7 <i>Website</i> | 7 |
| 2.8 <i>PHP</i> | 8 |
| 2.8.1 <i>Framework Laravel</i> | 8 |
| 2.9 <i>JavaScript</i> | 9 |
| 2.9.1 <i>Framework Nuxt.Js</i> | 9 |
| 2.9.2 <i>PWA</i> | 10 |
| 2.10 <i>Database</i> | 10 |
| 2.10.1 <i>MySQL</i> | 11 |
| 2.10.2 <i>Xampp</i> | 12 |

| | |
|--|-----------|
| 2.11 <i>Web Service</i> | 12 |
| 2.11.1 <i>REST API</i> | 13 |
| 2.11.2 <i>JWT (Json Web Token)</i> | 13 |
| 2.12 <i>Waterfall</i> | 14 |
| 2.13 <i>UML (Unified Modelling Language)</i> | 15 |
| BAB III METODE PENELITIAN | 20 |
| 3.1 <i>Bahan dan Alat Penelitian</i> | 20 |
| 3.2 <i>Prosedur Penelitian</i> | 21 |
| 3.2.1 <i>Metode Pengumpulan Data</i> | 21 |
| 3.2.2 <i>Metode Pengembangan Perangkat Lunak</i> | 22 |
| 3.3 <i>Analisis Kebutuhan</i> | 23 |
| 3.3.1 <i>Analisis Masalah</i> | 23 |
| 3.3.2 <i>Analisis Fungsional</i> | 24 |
| 3.4 <i>Perancangan Sistem</i> | 25 |
| 3.4.1 <i>Use Case Diagram</i> | 25 |
| 3.4.2 <i>Activity Diagram</i> | 26 |
| 3.4.3 <i>Sequence Diagram</i> | 38 |
| 3.4.4 <i>Class Diagram</i> | 41 |
| 3.5 <i>Perancangan Database</i> | 42 |
| 3.6 <i>Desain Antar Muka</i> | 46 |
| BAB IV HASIL DAN PEMBAHASAN | 54 |
| 4.1 <i>Hasil Penelitian</i> | 54 |
| 4.2 <i>Hasil Pada Tampilan Admin dan Customer</i> | 54 |
| 4.2.1 <i>Halaman Register Customer PWA</i> | 54 |
| 4.2.2 <i>Halaman Login Customer PWA</i> | 55 |
| 4.2.3 <i>Halaman Dashboard Customer PWA</i> | 56 |
| 4.2.4 <i>Halaman Utama Customer PWA</i> | 56 |
| 4.2.5 <i>Halaman Kategori Produk Customer PWA</i> | 57 |
| 4.2.6 <i>Halaman Cart Customer PWA</i> | 58 |
| 4.2.7 <i>Halaman Detail Order Customer PWA</i> | 58 |
| 4.2.8 <i>Halaman Snap Payment Gateway Midtrans</i> | 59 |
| 4.2.9 <i>Halaman Login Admin</i> | 59 |
| 4.2.10 <i>Halaman Menu Dashboard Admin</i> | 60 |
| 4.2.11 <i>Halaman Data Kategori Produk Admin</i> | 60 |

| | |
|---|-----------|
| 4.2.12 Halaman Tambah Data Kategori Produk <i>Admin</i> | 61 |
| 4.2.13 Halaman Edit Data Kategori Produk <i>Admin</i> | 61 |
| 4.2.14 Halaman Data Produk <i>Admin</i> | 62 |
| 4.2.15 Halaman Tambah Data Produk <i>Admin</i> | 62 |
| 4.2.16 Halaman Edit Data Produk <i>Admin</i> | 63 |
| 4.2.17 Halaman Data <i>Invoices Admin</i> | 63 |
| 4.2.18 Halaman Data <i>Customers Admin</i> | 64 |
| 4.2.19 Halaman Edit Data <i>Customers Admin</i> | 64 |
| 4.2.20 Halaman Data <i>Sliders Admin</i> | 65 |
| 4.2.21 Halaman Tambah Data <i>Sliders Admin</i> | 65 |
| 4.2.22 Halaman Data <i>Admin</i> | 66 |
| 4.2.23 Halaman Edit Data <i>Admin</i> | 66 |
| 4.2.24 Halaman Utama <i>Customer</i> atau Pengunjung | 67 |
| 4.2.25 Halaman Berdasarkan Kategori | 67 |
| 4.2.26 Halaman Semua Produk | 68 |
| 4.2.27 Halaman Detail Produk | 68 |
| 4.2.28 Halaman <i>Cart Customer</i> | 69 |
| 4.2.29 Halaman <i>Invoice Customer</i> | 69 |
| 4.2.30 Tampilan Transaksi <i>Payment Gateway</i> | 70 |
| 4.2.31 Tampilan Ulasan <i>Customer</i> | 70 |
| 4.3 Pembahasan | 71 |
| 4.3.1 <i>Testing</i> Sistem | 71 |
| BAB V KESIMPULAN DAN SARAN | 74 |
| 5.1 Kesimpulan | 74 |
| 5.2 Saran | 74 |
| DAFTAR PUSTAKA | 75 |

DAFTAR TABEL

| | Halaman |
|--|---------|
| Tabel 2.1 <i>Use Case Diagram</i> | 16 |
| Tabel 2.2 <i>Activity Diagram</i> | 17 |
| Tabel 2.3 <i>Sequence Diagram</i> | 18 |
| Tabel 2.4 <i>Class Diagram</i> | 19 |
| Tabel 3.1 Kebutuhan Perangkat Keras | 20 |
| Tabel 3.2 Kebutuhan Perangkat Lunak | 21 |
| Tabel 3.3 Tabel <i>users</i> | 42 |
| Tabel 3.4 Tabel <i>customers</i> | 42 |
| Tabel 3.5 Tabel <i>categories</i> | 43 |
| Tabel 3.6 Tabel <i>products</i> | 43 |
| Tabel 3.7 Tabel <i>provinces</i> | 43 |
| Tabel 3.8 Tabel <i>cities</i> | 44 |
| Tabel 3.9 Tabel <i>carts</i> | 44 |
| Tabel 3.10 Tabel <i>orders</i> | 44 |
| Tabel 3.11 Tabel <i>invoices</i> | 45 |
| Tabel 3.12 Tabel <i>reviews</i> | 45 |
| Tabel 3.13 Tabel <i>sliders</i> | 46 |
| Tabel 4.1 Pengujian <i>Black Box Login</i> | 72 |
| Tabel 4.2 Pengujian <i>Black Box Register</i> | 72 |
| Tabel 4.3 Pengujian <i>Black Box Menu Admin</i> | 73 |
| Tabel 4.4 Pengujian <i>Black Box Menu Customer</i> | 73 |

DAFTAR GAMBAR

| | Halaman |
|--|---------|
| Gambar 2.1 Model <i>Waterfall</i> | 14 |
| Gambar 2.2 Contoh <i>Use Case Diagram</i> | 16 |
| Gambar 2.3 Contoh <i>Activity Diagram</i> | 17 |
| Gambar 2.4 Contoh <i>Sequence Diagram</i> | 18 |
| Gambar 2.5 Contoh <i>Class Diagram</i> | 19 |
| Gambar 3.1 <i>Use Case Diagram</i> | 25 |
| Gambar 3.2 <i>Activity Diagram Login Admin</i> | 26 |
| Gambar 3.3 <i>Activity Diagram Kelola Kategori</i> | 27 |
| Gambar 3.4 <i>Activity Diagram Kelola Produk</i> | 28 |
| Gambar 3.5 <i>Activity Diagram Invoices Admin</i> | 29 |
| Gambar 3.6 <i>Activity Diagram Kelola Customers</i> | 30 |
| Gambar 3.7 <i>Activity Diagram Kelola Sliders</i> | 31 |
| Gambar 3.8 <i>Activity Diagram Kelola Admin</i> | 32 |
| Gambar 3.9 <i>Activity Diagram Register Customer</i> | 33 |
| Gambar 3.10 <i>Activity Diagram Login Customer</i> | 34 |
| Gambar 3.11 <i>Activity Diagram View Produk</i> | 35 |
| Gambar 3.12 <i>Activity Diagram Order Produk</i> | 36 |
| Gambar 3.13 <i>Activity Diagram Transaksi Customer</i> | 37 |
| Gambar 3.14 <i>Sequence Diagram Login</i> | 38 |
| Gambar 3.15 <i>Sequence Diagram Register</i> | 38 |
| Gambar 3.16 <i>Sequence Diagram View Kategori</i> | 39 |
| Gambar 3.17 <i>Sequence Diagram View Produk</i> | 39 |
| Gambar 3.18 <i>Sequence Diagram Cart Order</i> | 40 |
| Gambar 3.19 <i>Sequence Diagram Transaksi</i> | 40 |
| Gambar 3.20 <i>Class Diagram</i> | 41 |
| Gambar 3.21 Desain Halaman <i>Register PWA Mobile</i> | 46 |
| Gambar 3.22 Desain Halaman <i>Login PWA Mobile</i> | 47 |
| Gambar 3.23 Desain Halaman <i>Dashboard PWA Mobile</i> | 47 |
| Gambar 3.24 Desain Halaman <i>Utama PWA Mobile</i> | 48 |
| Gambar 3.25 Desain Halaman <i>Detail Produk PWA Mobile</i> | 48 |
| Gambar 3.26 Desain Halaman <i>Cart PWA Mobile</i> | 49 |

| | |
|--|----|
| Gambar 3.27 Desain Tampilan <i>Snap pay</i> Midtrans PWA Mobile | 49 |
| Gambar 3.28 Desain Halaman Utama | 50 |
| Gambar 3.29 Desain Halaman <i>Login Admin</i> | 50 |
| Gambar 3.30 Desain Halaman Kategori <i>Admin</i> | 51 |
| Gambar 3.31 Desain Halaman Produk <i>Admin</i> | 51 |
| Gambar 3.32 Desain Halaman <i>Invoices Admin</i> | 52 |
| Gambar 3.33 Desain Halaman Data <i>Customers Admin</i> | 52 |
| Gambar 3.34 Desain Halaman <i>Sliders Admin</i> | 53 |
| Gambar 3.35 Desain Halaman Data <i>Admin</i> | 53 |
| Gambar 4.1 Halaman <i>Register Customer PWA</i> | 54 |
| Gambar 4.2 Halaman <i>Login Customer PWA</i> | 55 |
| Gambar 4.3 Halaman <i>Dashboard Customer PWA</i> | 56 |
| Gambar 4.4 Halaman Utama <i>Customer PWA</i> | 56 |
| Gambar 4.5 Halaman Kategori Produk <i>Customer PWA</i> | 57 |
| Gambar 4.6 Halaman <i>Cart Customer PWA</i> | 58 |
| Gambar 4.7 Halaman Detail <i>Order Customer PWA</i> | 58 |
| Gambar 4.8 Halaman <i>Snap Payment Gateway</i> Midtrans | 59 |
| Gambar 4.9 Halaman <i>Login Admin</i> | 59 |
| Gambar 4.10 Halaman Menu <i>Dashboard Admin</i> | 60 |
| Gambar 4.11 Halaman Data Kategori Produk (<i>Admin</i>) | 60 |
| Gambar 4.12 Halaman Tambah Data Kategori Produk (<i>Admin</i>) | 61 |
| Gambar 4.13 Halaman Edit Data Kategori Produk (<i>Admin</i>) | 61 |
| Gambar 4.14 Halaman Data Produk (<i>Admin</i>) | 62 |
| Gambar 4.15 Halaman Tambah Data Produk (<i>Admin</i>) | 62 |
| Gambar 4.16 Halaman Edit Data Produk (<i>Admin</i>) | 63 |
| Gambar 4.17 Halaman Data <i>Invoices (Admin)</i> | 63 |
| Gambar 4.18 Halaman Data <i>Customers (Admin)</i> | 64 |
| Gambar 4.19 Halaman Edit Data <i>Customers (Admin)</i> | 64 |
| Gambar 4.20 Halaman Data <i>Sliders (Admin)</i> | 65 |
| Gambar 4.21 Halaman Tambah Data <i>Sliders (Admin)</i> | 65 |
| Gambar 4.22 Halaman Data <i>Admin</i> | 66 |
| Gambar 4.23 Halaman Edit Data <i>Admin</i> | 66 |
| Gambar 4.24 Halaman Utama <i>Customer</i> atau Pengunjung | 67 |
| Gambar 4.25 Halaman Berdasarkan Kategori | 67 |

| | |
|---|----|
| Gambar 4.26 Halaman Semua Produk | 68 |
| Gambar 4.27 Halaman Detail Produk | 68 |
| Gambar 4.28 Halaman <i>Cart Customer</i> | 69 |
| Gambar 4.29 Halaman <i>Invoice Customer</i> | 69 |
| Gambar 4.30 Tampilan Transaksi <i>Payment Gateway</i> | 70 |
| Gambar 4.31 Tampilan Ulasan <i>Customer</i> | 70 |

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi saat ini memberikan dampak yang signifikan terhadap persaingan antar perusahaan di seluruh dunia, terutama dalam penggunaan sistem informasi penjualan *online*. Saat ini, sistem informasi berkembang sangat pesat baik dari segi teknologi maupun manajemen operasionalnya. Kemajuan teknologi informasi di dunia ini juga berdampak besar pada rancangan dan implementasi *agribisnis* (Shodiq, A. R. A. R., Priatna, W. B., & Kusnadi, N., 2018).

Perusahaan ini merupakan suatu usaha perdagangan komoditi yang berdiri sekitar awal tahun 2019 dan beroperasi di kota Medan. Pada awalnya perdagangan masih terbatas pada komoditi kopi, dengan istilah kopi *green bean* dan *roast bean*. Seiring dengan perkembangan usaha, maka perusahaan mengembangkan pemasaran sampai luar kota Medan, khususnya di Jakarta, Bandung, Semarang, Surabaya dan Jogjakarta. Bidang usaha juga mengalami perkembangan dengan melakukan perdagangan komoditi yang lain selain kopi. Komoditi vanili dan cengkeh merupakan produk yang dipasarkan oleh perusahaan dengan tujuan pasar masih di pasar lokal, di mana penjualan komoditi vanili dan cengkeh masih bersifat B2B atau *Business to Business* dan penjualan komoditi kopi bersifat B2C atau *Business to Customer* (PT. Bejo Agro Lestari, 2021).

E-commerce merupakan cara bagi konsumen untuk membeli apa yang mereka inginkan secara *online*, *e-commerce* juga memudahkan konsumen untuk melakukan pemesanan, karena konsumen tidak harus bertemu langsung dengan penjual untuk memilih produk yang diinginkan, sehingga semua orang dapat dengan mudah mengakses informasi produk kapan saja dan di mana saja (Gunawan, I. P., Sukarsa, I. M., & Putra, I. M. S., 2020). *E-Commerce* dapat dilakukan oleh siapa saja dengan mitra bisnis mereka, tanpa dibatasi oleh ruang dan waktu. Dalam aktivitas *e-commerce* sebenarnya berkaitan dengan hubungan antara penjual dan pembeli, transaksi antar pelaku bisnis dan proses *internal* yang mendukung transaksi dengan perusahaan. Saat ini kegiatan *e-commerce* dilakukan menggunakan media berbasis *website* maupun *mobile* seperti *smartphone* atau *tablet* (Muchtar, A. Z., & Munir, S., 2019).

Penulis menggunakan *framework PHP* yaitu *Laravel* dalam implementasi pada *sistem e-commerce* ini. *Laravel* merupakan sebuah *MVC web development framework* untuk sebuah pengembangan aplikasi yang diharapkan dapat meningkatkan kualitas aplikasi yang

dihasilkan, dengan mengurangi biaya pengembangan dan perbaikan serta menghasilkan source code yang rapih, fungsional dan dapat digunakan secara efisien (Muchtari, A. Z., & Munir, S., 2019).

Laravel merupakan sebuah *Framework PHP* dirilis dibawah *lisensi MIT* dengan kode yang sudah disediakan oleh *Github*, sama seperti *framework* yang lain. *Laravel* dibangun dengan konsep MVC (*Model-Controller-View*), kemudian *Laravel* dilengkapi juga dengan *command line tool* yang bernama “*Artisan*” yang bisa digunakan untuk *packaging bundle* dan instalasi *bundle* melalui *command prompt* (Ramdiansyah, F. D., & Anubhakti, D., 2020).

Masalah yang ditemukan pada perusahaan saat ini yaitu penggunaan sistem masih menggunakan cara konvensional dalam melakukan penjualan terhadap produknya. Hal ini kurang efektif karena penjualan masih bersifat *door to door* dan menggunakan pesan *instant* via *WhatsApp*, *Gmail* atau konsumen datang langsung ke tempat usaha. Dari permasalahan tersebut penulis mendapatkan solusi untuk menyelesaikannya, yaitu membangun sistem *E-Commerce* pada produk B2C atau *Business to Customer*, yang diharapkan bisa menyediakan informasi produk komoditi seperti biji kopi *green bean* dan *roast bean* yang berguna membantu pelanggan dalam mengetahui detail informasi produk seperti harga dan kualitasnya, serta dapat dengan mudah melakukan *update* data produk kopi secara *realtime* menggunakan *API web service*. Sehingga penerapan *e-commerce* berbasis *PWA* ini dapat membantu meningkatkan penjualan produk kopi pada perusahaan.

Metode pengembangan yang digunakan penulis adalah *SDLC* model *Waterfall*. Metode ini juga memiliki keunggulan bahwa hasil perangkat lunak yang dikembangkan biasanya berkualitas dan dokumentasi pengembangan sistem sangat terorganisir. Namun, meskipun memiliki metode air yang mengalir layaknya air terjun, diwajibkan adanya umpan balik pada setiap fase. Umumnya, fase yang lain akan mulai berjalan ketika fase sebelumnya sudah selesai dikerjakan serta perlu adanya batas waktu dalam setiap fase pengerjaan untuk menghasilkan sistem yang baik dan sesuai dengan kebutuhan (Dewi, F. S., Poerbo, H., & Susilowati, M., 2020).

Oleh karena itu, berdasarkan permasalahan diatas penulis akan mengangkat penelitian skripsi berjudul “**E-Commerce Komoditas Berbasis PWA Pada PT. Bejo Agro Lestari Menggunakan Framework Laravel**”. Sehingga dapat mempermudah perusahaan dalam melakukan proses jual beli dan transaksi online pada produk yang dijual.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, dapat dirumuskan permasalahan dalam penelitian ini yaitu bagaimana membangun *E-commerce* komoditas berbasis *PWA* Pada PT. Bejo Agro Lestari Menggunakan *Framework Laravel*, yang diharapkan dapat mempermudah dalam transaksi penjualan, memperluas wilayah penjualan, dan memberikan informasi data tentang produk yang akan dijual kepada masyarakat luas.

1.3 Batasan Masalah

Adapun pada penelitan ini, penulis membatasi masalah sebagai berikut :

1. Sistem ini menjadi sarana penjualan pada produk B2C atau *Business to Customer* pada PT. Bejo Agro Lestari.
2. Sistem yang dibangun menggunakan *web service Rest API*.
3. Sistem diintegrasikan dengan layanan *Payment Gateway* (Midtrans) dan Raja Ongkir.
4. Metode pengembangan sistem yang digunakan yaitu *SDLC Waterfall*.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai pada penelitian ini sebagai berikut :

1. Membangun sistem informasi *E-Commerce* komoditas dalam membantu perusahaan sebagai media penjualan *online* pada produk yang di jual.
2. Membantu perusahaan melayani transaksi melalui *payment gateway* tanpa menggunakan cara konvensional.
3. Membantu meningkatkan penjualan produk kopi pada perusahaan.

1.5 Manfaat Penelitian

Adapun manfaat yang dicapai pada penelitian ini sebagai berikut :

1. Mempermudah perusahaan dalam melakukan penjualan produk dalam skala luas dan *online*.
2. Mempermudah *customer* dalam melakukan pembayaran tanpa harus melakukan *input* bukti pembayaran secara konvensional, karena sistem akan terintegrasi melalui *payment gateway* (Midtrans).
3. Mempermudah *customer* dalam memilih layanan jasa pengiriman karena sudah terintegrasi Raja Ongkir.

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Menurut Wahyuni, D., & Rezky, S. F (2019) pada dasarnya, sistem merupakan kerangka kerja prosedur yang saling terkait, diatur menurut skema yang komprehensif untuk melakukan aktivitas perusahaan dan menyediakan informasi yang digunakan sebagai pendukung keputusan manajemen operasional. Sistem dapat diartikan sebagai sekumpulan elemen, komponen, atau variabel. Variabel-variabel ini terorganisir, terintegrasi, saling bergantung, dan terintegrasi.

Sistem merupakan sekumpulan elemen yang saling berinteraksi atau bekerja sama untuk mencapai tujuan yang telah ditentukan (Dedi, D., Sutarman, S., & Septiyani, N., 2020). Menurut Itmaana, E (2021) “Sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama bagian terpadu sistem yang bersangkutan. Unsur-unsur sistem berhubungan erat satu dengan yang lain dan sifat serta kerjasama antar unsur sistem tersebut mempunyai bentuk tertentu”.

2.2 Pengertian Informasi

Informasi adalah data yang telah diproses atau diolah untuk digunakan dalam pengambilan suatu keputusan. Teori informasi lebih tepat disebut sebagai teori komunikasi yang memberikan pandangan dan berguna bagi sistem informasi (Yanuardi, Y., & Permana, A. A., 2019). menurut Itmaana, E (2021) “Informasi dapat didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya, kualitas dari suatu informasi bergantung dari tiga hal yaitu : Akurat (*Accurate*), Tepat waktu (*Timelines*) , Relevan (*Relevance*)”.

Menurut Hermanto, B (2019) “Informasi merupakan hasil pengolahan data yang telah diolah menjadi lebih bermanfaat dan bermakna bagi yang menerima”. Sumber informasi merupakan data dan beberapa informasi dapat dikatakan layak jika manfaatnya lebih efektif daripada biaya mendapatkannya. Penggunaan informasi dimaksudkan untuk mengurangi ketidakpastian proses pengambilan keputusan tentang situasi keadaan. Informasi yang digunakan dalam sistem informasi biasanya digunakan untuk berbagai tujuan.

2.3 Pengertian Sistem Informasi

Sistem informasi merupakan kumpulan dari orang-orang yang bekerja sama untuk mencapai suatu tujuan tertentu, dalam hal lain, sistem informasi juga dapat melakukan pendukung keputusan. Dalam arti lain, adalah kombinasi terorganisir dari individu (*brainware*), perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komputer, komunikasi data, dan basis data dalam mengumpulkan, mendistribusikan, dan memodifikasi informasi dalam suatu bentuk organisasi (Ariyanti, L., Satria, M. N. D., & Alita, D., 2020).

Dalam istilah bahasa, sistem informasi terdiri dari sistem yang dapat diartikan sebagai kumpulan orang atau beberapa orang yang saling bekerja sama dan secara terstruktur untuk memenuhi tujuan-tujuan tertentu (Hermawan, 2019). Menurut Itmaana, E (2021) “Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi yang menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan”.

2.4 E-Commerce

Menurut Hasanudin, M (2019) Menyatakan “*E-Commerce* atau perdagangan elektronik merupakan suatu aktivitas yang berkaitan dengan penjualan, pembelian, pemasaran barang, dan sistem pengumpulan data secara otomatis dengan menggunakan teknologi telekomunikasi dan informasi”. Sedangkan menurut Gunawan, I. P., Sukarsa, I. M., & Putra, I. M. S (2020) “*Electronic Commerce (e-commerce)* adalah proses pembelian, penjualan atau pertukaran produk, jasa dan informasi melalui jaringan komputer. *E-commerce* merupakan bagian dari *e-business*, di mana cakupan *e-business* lebih luas, tidak hanya sekedar perniagaan tetapi mencakup juga pengkolaborasi mitra bisnis, pelayanan nasabah, lowongan pekerjaan”.

(Sari, E. P., Pudjiarti, E., & Susanti, H 2020) *E-commerce* merupakan transaksi penghubung antara penjual dan pembeli melalui media internet, keuntungan yang diperoleh dengan menggunakan transaksi melalui *e-commerce* adalah untuk meningkatkan pendapatan dengan menggunakan penjualan *online* yang biayanya lebih murah dan juga sekaligus biaya-biaya operasional lainnya. *E-commerce* memiliki beberapa komponen terkait. Berikut komponen-komponen yang membentuk mekanisme pasar *e-commerce* :

1. Pembeli

Konsumen merupakan target pasar potensial yang diberikan penawaran berupa produk, jasa dan informasi dari penjual.

2. Penjual

Penjual individu maupun organisasi, adalah pihak yang menyediakan produk, layanan, atau informasi kepada konsumen. Proses penjualan dilakukan dari media *website* penjual.

3. Produk

Dalam *e-commerce*, produk yang ditawarkan sebenarnya adalah produk *digital*. Karena pembeli tidak melihat produk yang ditawarkan secara langsung dalam bentuk fisik, tetapi hanya sebagai gambar yang divisualisasikan dalam bentuk katalog produk pada halaman *website*.

4. *FrontEnd*

Front end merupakan aplikasi *web* yang memungkinkan pembeli berinteraksi langsung dengan penjual. Bisnis yang termasuk dalam *front end* ini meliputi katalog, *shopping cart*, dan *search engine*.

5. *BackEnd*

Backend merupakan layanan aplikasi yang secara tidak langsung berperan sebagai pendukung dari aplikasi *frontend*. Proses aktivitas pada *backend* mencakup manajemen produk, pemrosesan pembayaran, dan pengiriman produk.

6. Infrastruktur

Komponen infrastruktur yang akan digunakan sebagai media elektronik yaitu berupa penggunaan perangkat keras, perangkat lunak, serta sistem jaringan komunikasi internet.

7. Pihak Ketiga

Pihak ketiga merupakan pihak yang dapat melakukan kerjasama dengan para penjual. Contoh pihak ketiga adalah seperti *bank* yang bisa memudahkan proses pembayaran (*payment gateway*) yang dilakukan oleh para konsumen baik *via bank transfer*, *credit card*, *e-wallet*, maupun *mobile banking*.

2.5 B2C (*Business-to-Customer*)

B2C (*Business to Consumer*) merupakan proses bisnis antara perusahaan dan pelanggannya, termasuk pengumpulan data pelanggan, informasi tentang produk, atau transaksi produk melalui media elektronik (Fatah, K., & Andayani, T. D., 2021). *Business to customer* juga bisa dikatakan bisnis yang dilakukan orang-orang atau pihak-pihak yang memiliki kepentingan bisnis yang sama antar perusahaan, yang artinya produsen hanya

menjual dan memasarkan produk maupun jasa sehingga pihak konsumen hanya sebagai pengguna atau pembeli (Darojat, T. A., 2019).

Banyak perusahaan B2C menggunakan internet dan teknologi informasi untuk menjual dan memasarkan produk mereka, implementasi konsep B2C biasanya dilakukan dengan membuat *website e-commerce* yang akan digunakan untuk transaksi langsung dengan konsumen (Yulianto, E., & Mauluddin, A., 2019). B2C merupakan salah satu jenis usaha yang dilakukan pengusaha terhadap konsumen. Sebagai konsumen, perusahaan hanya menjual produknya kepada konsumen. Saat ini, jenis *e-commerce* ini terus berkembang pesat dengan munculnya situs *website* (Wahyuni, F. D., & Sari, S. N., 2021).

2.6 Komoditas

Salah satu komoditas primer dari sektor perkebunan yang menjadi andalan di Indonesia adalah kopi. Indonesia berada di urutan keempat sebagai negara penghasil kopi terbaik di dunia pada tahun 2017 dan kopi Indonesia termasuk berdaya saing tinggi hingga dapat bersaing dengan negara pengekspor kopi terbesar lainnya seperti Brazil, Kolombia, dan Vietnam (Alexander & Nadapdap, 2019). Menurut Parnadi, F., & Loisa, R (2018) “Indonesia adalah salah satu negara produsen dan eksportir kopi paling besar di dunia. Berkaitan dengan komoditi-komoditi agrikultur, kopi adalah penghasil devisa terbesar keempat untuk Indonesia setelah minyak sawit, karet dan kakao. Dengan keunikan cita rasa dan aroma kopi asal Indonesia, Indonesia memiliki peluang besar untuk meningkatkan perdagangan kopinya di dunia”.

Menurut Rivaldi, M. R (2022) “Komoditas kopi adalah komoditas bernilai jual tinggi di mana penikmatnya tersebar di seluruh penjuru dunia dan Indonesia dikenal sebagai salah satu negara eksportir kopi terbesar. Olahan dari biji kopi itu sendiri selain dijadikan bubuk untuk diseduh menjadi minuman kopi, juga dapat diolah menjadi berbagai macam produk, seperti pengharum ruangan, aksesoris, maupun menjadi produk olahan kreatif lainnya”. Pertimbangan penting dalam meningkatkan kualitas bahan kopi dan pengembangan usaha adalah pengelolaan pengelolaannya, khususnya pengelolaan rantai pasok. Manajemen yang baik dalam manajemen rantai pasokan mendukung keberhasilan suatu perusahaan.

2.7 Website

Website merupakan halaman yang menampilkan informasi dalam format teks, gambar, audio, atau video yang terletak pada *server / hosting* yang membutuhkan jaringan internet untuk mengaksesnya, untuk dapat mengakses sebuah *website* di internet, harus ada dua

komponen yaitu *domain* dan *hosting*. Menjual produk dan jasa dengan menggunakan *website* dalam dunia bisnis dapat menjadi pilihan terbaik sebagai keunggulan persaingan bisnis, *website* juga dapat digunakan sebagai sarana untuk meningkatkan pelayanan kepada konsumen (Lesnanda, M., & Raharjo, Y. A. B., 2019).

Definisi lain dari *website* adalah kumpulan *hyperlink* yang menggunakan bahasa *HTML* untuk berpindah dari satu alamat ke alamat lain, dan merupakan layanan yang banyak digunakan di internet (Wijayakusuma, N. H. P., Saintika, Y., & Susanto, I., 2021). *Website* adalah kumpulan informasi / halaman, yang biasanya diakses melalui internet. Siapa pun dapat menggunakannya di mana saja kapan saja selama terhubung ke internet. Secara teknis, *website* adalah kumpulan dari *page* yang terkait dengan *domain* atau *subdomain* tertentu (Romadhon, M. H., Yudhistira, Y., & Mukrodin, M., 2021).

2.8 PHP

PHP adalah bahasa pemrograman khusus yang digunakan oleh pengguna *web* untuk membangun sebuah sistem, dan juga *php* merupakan *tools* yang sangat penting ketika ingin membuat sebuah sistem di dalam sebuah *website*. *PHP* merupakan singkatan dari *Hypertext Preprocessor* serta bahasa pemrograman *open source* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *web* yang disisipkan pada dokumen *HTML* (Naofal, N., Ulhaq, M. R. D., & Prianto, C., 2022).

Bahasa *PHP* bersifat dinamis serta dapat dijalankan pada berbagai macam sistem operasi seperti *Windows*, *Linux*, dan *Mac Os*. Selain *Apache*, *PHP* juga mendukung beberapa *web server* lain, seperti *Microsoft ISS*, *Caudium*, dan *PWS*. *PHP* dapat memanfaatkan *database* untuk menghasilkan halaman *web* yang dinamis, adapun sistem manajemen *database* yang sering digunakan menggunakan *PHP* adalah *MYSQL* (Novendri, M. S., Saputra, A., & Firman, C. E., 2019).

2.8.1 Framework Laravel

Laravel merupakan sebuah *framework PHP* yang dirilis di bawah lisensi *MIT*, dibangun dengan konsep *MVC (model view controller)*, *laravel* didesain untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan perbaikan serta meningkatkan produktifitas pekerjaan dengan sintak yang bersih dan fungsional yang dapat mengurangi banyak waktu untuk implementasi (Lesnanda, M., & Raharjo, Y. A. B., 2019). Keunggulan dari *laravel* sendiri adalah :

1. Memiliki komunitas yang besar, sehingga dapat dipertimbangkan jika ingin merancang sebuah *web* yang didukung oleh komunitas besar, maka dapat mudah belajar dan mencari solusi.
2. Faktor keamanan tentu dalam *web* keamanan data pengguna sangat penting, *ORM* pada *laravel* menggunakan *PDO* yang dapat mencegah serangan *SQL injection*.
3. *Laravel* memiliki fitur *migration*, *migration* merupakan fitur utama *laravel* di mana memungkinkan mempertahankan struktur *database* tanpa harus membuat kembali dengan cara menulis *code PHP* untuk mengatur *database*.

Laravel merupakan sebuah *framework* aplikasi *web* dengan ekspresif, sintaks yang elegan. *Laravel* bertujuan untuk mempermudah bagi pengembang tanpa mengorbankan fungsionalitas sebuah aplikasi (Rudi dan Prehanto 2020). *Laravel* merupakan salah satu *framework php* berbasis *website* yang kini banyak digunakan oleh beberapa *developer*. *Laravel* merupakan sebuah *framework* yang digunakan untuk memberikan informasi dan mengolah data (Jorgie, R. A., Anggraeny, F. T., & Via, Y. V., 2021).

2.9 JavaScript

Javascript merupakan bahasa pemrograman untuk *client side*. *Javascript* mudah dipelajari karena merupakan bahasa pemrograman yang mendekati dengan bahasa manusia atau bisa disebut bahasa tingkat tinggi. Tujuan dari *Javascript* itu sendiri adalah untuk meningkatkan fungsionalitas pada *website* agar lebih dinamis, seperti untuk menampilkan dan menghilangkan objek pada *website* kemudian dengan fungsi *javascript* dapat memanggil kembali objek yang dihilangkan tersebut (Marlina, M., & Masnur, M., 2021).

Javascript adalah bahasa yang berbentuk kumpulan skrip di mana fungsi digunakan untuk menambahkan interaksi antara situs *web* dan pengguna situs *web*. *Javascript* berjalan pada sisi klien dan menyediakan fungsionalitas pada halaman *web* yang lebih unggul daripada fungsionalitas yang terdapat pada *HTML* (Febrianto, A., Achmadi, S., & Sasmito, A. P., 2021). *Javascript* adalah bahasa pemrograman yang berjalan pada *engine* yang disediakan oleh *browser*. *Javascript* banyak digunakan di *browser* untuk memasukkan data ke dalam konten situs *web*, melakukan animasi, dan sebagainya. Sehingga bahasa ini sangat menarik bagi para pengembang *website* (Saputra, M. H. K., 2021).

2.9.1 Framework Nuxt.Js

Nuxt.Js adalah salah satu kerangka kerja *JavaScript* populer yang memudahkan proses pembuatan aplikasi *web*, *nuxt* merupakan kerangka kerja tingkat tinggi yang dibangun

di atas kerangka kerja lain yang disebut *vue.js*. *Nuxt.js* mengabstraksi detail distribusi kode *server* dan klien sehingga dapat fokus pada pengembangan aplikasi, secara *internal* membuat *instance* koneksi untuk dapat menambahkan kustom *middleware*, di mana sebuah *middleware* menyediakan fungsionalitas seperti *server* ke aplikasi berbasis klien sepenuhnya. File konfigurasi *nuxt* dikonfigurasi dengan modul yang diperlukan, *plugin*, dan otentikasi *middleware* diatur di *router*. Selain itu, aplikasi dapat dimuat dan di *cache* di penyimpanan lokal *browser*, sehingga secara signifikan dapat meningkatkan waktu dalam membuka aplikasi (Dhanush, G. A., 2021).

Nuxt.js membuat penyebaran situs *web* berbasis komponen menjadi lebih mudah secara signifikan. Misalnya, aplikasi yang dibuat dengan *Nuxt.js* dapat memenuhi persyaratan *PWA* (Magruder, D. S., 2021). *Framework Nuxt* adalah sebuah *framework* yang digunakan untuk membuat *Universal Vue.js application*. Di dalam *Nuxt* berisi beberapa *dependency* seperti *vue.js*, *node.js*, *webpack*, dan *babel.js*. Manfaat dari penggunaan *framework* ini adalah dapat mengurangi waktu interaktivitas dan meningkatkan (*SEO*) *Search Engine Optimization* (Susrama, I. G., Sampurno, I. A. W., & Sugiarto, S., 2020).

2.9.2 PWA

Progressive Web Apps adalah aplikasi *web* yang memuat seperti halaman *web* tetapi dapat menawarkan fungsionalitas pengguna seperti bekerja *offline*, notifikasi, dan akses perangkat keras *mobile web*. Secara sederhana, teknologi *PWA* merupakan *mobile site* yang dapat dimuat seperti layaknya *mobile Apps*. Jadi disaat pengguna mengakses *website*, maka akan muncul notifikasi *Add to home Screen* untuk memberitahu pengguna adanya fasilitas menambahkan akses cepat aplikasi pada *desktop* dan *smartphone* (Dewi et al., 2020)

Progressive Web Apps merupakan aplikasi yang dibangun dengan melakukan optimasi pada sebuah *website*. Memanfaatkan fitur *web* yang *modern* agar terlihat seperti aplikasi *mobile*. *PWA* sendiri merupakan kumpulan teknologi, konsep desain, dan *Web API* (*Application Programming Interface*) yang bekerja sama untuk memberikan sentuhan aplikasi pada *web mobile* (Muddin, S., Tehuayo, H., & Iksan, F., 2021).

2.10 Database

Database adalah kumpulan data yang bersifat mekanis, didefinisikan secara formal, dan dapat dikendalikan. Fungsi *database* itu sendiri adalah untuk mengelompokkan data dan informasi agar lebih mudah dipahami, mencegah duplikasi dan inkonsistensi data, serta mempermudah proses penyimpanan, pengaksesan, pemutakhiran, dan penghapusan data.

Manfaat *database* adalah mencegah redundansi data, menjaga integritas data, memfasilitasi berbagi data, dan menjaga keamanan data. Ada beberapa jenis-jenis *database* seperti *Microsoft Access*, *Oracle*, dan *MySql* (Rahardja, U., Febriyanto, E., Handayani, I., & Ningsih, H. W., 2019).

Menurut Simanullang (2021) dalam jurnal Aswiputri, M (E-ISSN: 2686-5238) terdapat tiga komponen pada sistem informasi yang akan saling berkeja sama dalam membuat atau menginput data, yaitu :

1. Proses *input* data yang memungkinkan kegiatan perpindahan atau memindahkan data untuk masuk ke dalam sistem yang ada pada komputer, agar data bisa digunakan dan disimpan melalui penyimpanan yang akan dimuat dalam bentuk file.
2. Pengolahan data merupakan proses suatu pengolahan yang akan dimuat dalam bentuk informasi.
3. Proses menghasilkan *output* berupa produk informasi dan menghasilkan sekumpulan data yang dikelola menjadi nilai yang menghasilkan informasi tersebut.

2.10.1 MySQL

Pada perkembangannya, *MYSQL* disebut juga *SQL* yang merupakan kependekan dari *Structured Query Language*. *SQL* adalah bahasa terstruktur yang khusus digunakan untuk memproses *database*. Pertama kali *SQL* ditemukan oleh *American National Standards Institute* (ANSI) pada tahun 1986. *MYSQL* adalah sebuah sistem manajemen *database* yang bersifat *relational* dan *open source*. Artinya, data yang dikelola pada *database* akan dimasukkan kedalam beberapa tabel yang terpisah, sehingga manipulasi data akan jauh lebih cepat. *SQL* adalah bahasa terstruktur yang khusus digunakan untuk memproses *database*. *SQL* memungkinkan pengguna untuk mengetahui di mana *SQL* berada atau bagaimana informasi diatur. *SQL* juga merupakan bahasa pemrograman yang dirancang khusus untuk mengirim perintah *query* (mengakses data berdasarkan alamat tertentu) ke *database* (Novendri, M. S., Saputra, A., & Firman, C. E., 2019).

MySQL adalah implementasi dari sistem manajemen basis data relasional (*RDBMS*) yang didistribusikan secara gratis. Semua pengguna bebas menggunakan *MySQL*, tetapi dengan batasan, serta perangkat lunak tidak boleh digunakan sebagai turunan komersial. *SQL* adalah konsep pengoperasian *database*, terutama untuk memilih, seleksi, dan memasukkan data, yang membuat pengoperasian data menjadi mudah dan otomatis (Gunawan, I. P., Sukarsa, I. M., & Putra, I. M. S., 2020).

2.10.2 Xampp

Menurut Itmaana, E (2021) “*Xampp* merupakan salah satu paket dari *PHP* yang berbasis *open source* yang digunakan sebagai *web server* untuk menjalankan program *PHP*”. Menurut Novendri, M. S., Saputra, A., & Firman, C. E (2019) *Xampp* adalah perangkat lunak gratis yang mendukung banyak sistem operasi dan merupakan kumpulan dari beberapa program, seperti *Apache*, *MYSQL*, *PHP*, dan *Perl*. *Xampp* merupakan alat yang menyediakan paket perangkat lunak dalam satu paket. Paket *Xampp* sudah termasuk *Apache* (*web server*), *Mysql* (*database*), *PHP* (*server-side script*), *Perl*, *FTP server*, *PhpMyAdmin*, dan berbagai *helper library* lainnya. Bagian penting dari *XAMPP* yang biasa digunakan :

1. *Htdocs* adalah folder yang ditempatkan untuk menjalankan file *PHP*, *HTML*, dan file skrip lainnya.
2. *PhpMyAdmin* adalah bagian dari pengelolaan *database MySQL* di komputer. Untuk melihatnya, buka *web browser* dan masukkan alamat `http://localhost/ phpmyadmin` maka akan diteruskan ke halaman *PhpMyAdmin*.
3. Kontrol panel berfungsi untuk mengelola layanan *XAMPP*. Seperti menghentikan (*stop*) layanan, ataupun memulai (*start*).

2.11 Web Service

Web service merupakan salah satu bentuk perkembangan teknologi saat ini yang dapat digunakan untuk integrasi sistem. *Web service* dapat menjembatani antara berbagai sistem yang ada. *Web service* memiliki antarmuka yang ditulis dalam format yang dapat dibaca oleh mesin. Arsitektur *web service* dapat dibuat menggunakan (*SOAP*) *Simple Object Access Protocol* atau (*REST*) *Representational State Transfer* (Wijaya, F., Jacobus, A., & Sambul, A., 2021). *Web service* merupakan sebuah teknologi yang dapat mengubah kemampuan internet dengan cara menambahkan teknologi *transactional web*, yaitu teknologi *web* yang dapat berkomunikasi menggunakan pola *program-to-program* (P2P). *Web service* bekerja ketika pengguna meminta fungsi pada *server* dan *server* menjalankan perintah sesuai dengan layanan yang diinginkan oleh pengguna (Mebiyantara, F., Faisal, A., & Ariwibisono, F. X., 2021).

Menurut Wardhany, V. A., & Hidayat, A (2021) “*Web service* merupakan sebuah potongan informasi dari suatu *website* tanpa harus mengunjungi *website* tersebut, cukup dengan mengetahui fungsi / *method web service* yang disediakan oleh *web* tersebut”. *Web service* merupakan sistem pertukaran informasi berbasis *XML* yang menggunakan internet untuk berinteraksi antar aplikasi. Teknologi ini merupakan standar yang diadopsi oleh

banyak vendor perangkat lunak, karena standar yang terbuka. Standar dalam membuat aplikasi *web service* dapat diimplementasikan oleh vendor yang berbeda, sehingga dapat berkomunikasi satu sama lain. Ektensi ini, membuat *web service* berkembang dengan cepat dan luas, karena fungsi dari *web service* bisa digunakan di mana saja (Nugraha, E. S., Padri, A. R., Nurdiawan, O., Faqih, A., & Anwar, S., 2021).

2.11.1 REST API

Application programming interface (API) adalah dokumen pengembangan perangkat lunak yang mendeskripsikan layanan yang disediakan oleh aplikasi atau sistem. Penggunaan *API* ini dimaksudkan untuk memudahkan pengembang mempelajari sistem yang sedang berjalan dan sebagai panduan untuk menggunakan fungsi-fungsi yang disediakan. Dapat disimpulkan bahwa *REST API* adalah sistem terdistribusi yang menggunakan *web service* untuk mengekspos fungsionalitas di internet dengan menggeneralisasi desain antarmuka (Sutrisno, S., Asyidiq, M., & Santoso, S., 2018). Cara kerja *REST API* adalah *server* memperlakukannya sebagai objek yang dapat dibuat, diedit, dihapus, dan dibaca oleh klien. Dengan kata lain *create* sebagai *request POST*, *update* sebagai *request PUT* atau *PATCH*, hapus sebagai *request DELETE*, dan baca sebagai *request GET* (Mebiyantara, F., Faisol, A., & Ariwibisono, F. X., 2021).

Application Programming Interface atau *API* merupakan integrasi dari dua bagian dari sistem aplikasi. Terdiri dari elemen *function*, *protocols*, dan *tools* lainnya digunakan pengembang membuat sebuah aplikasi. *Web API* terdapat beberapa hal mendukung fungsi *CRUD* yang bekerja melalui *HTTP protocol* dengan *method GET*, *POST*, *PUT* dan *DELETE* (Yanti, S. N., & Rihyanti, E., 2021). *REST API* mendefinisikan seperangkat prinsip arsitektur yang dapat digunakan untuk merancang *web service* yang berfokus pada sumber daya sistem, di mana tujuannya adalah untuk meningkatkan kinerja sistem, cepat, dan mudah untuk dikembangkan hal komunikasi data dan pertukaran data (Edy, E., Ferdiansyah, F., Pramusinto, W., & Waluyo, S., 2019).

2.11.2 JWT (Json Web Token)

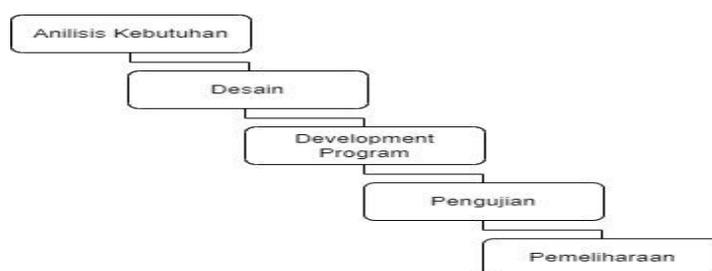
Menggunakan *JSON Web Token authentication* dengan *REST API* membuat aplikasi lebih aman karena tidak dapat mengakses aplikasi tanpa menggunakan *token*. *Token JWT* dapat dikirim melalui *URL*, parameter *HTTP POST*, atau *header HTTP*. Karena ukurannya yang kecil, maka dapat ditransmisikan dengan lebih cepat. *JWT* bekerja di mana *token* ini seperti *password*. Saat *user* berhasil melakukan *login* maka *server* akan memberikan sebuah

token. Kemudian *token* disimpan oleh *user* di penyimpanan lokal atau dalam *cookie* di *browser* dan harus berisi *token* jika *user* mengunjungi halaman tertentu. Oleh karena itu, *user* akan mengirim kembali *token* yang ditentukan di awal sebagai bukti bahwa *user* telah *login*. Struktur *JWT* terdiri dari tiga bagian yang dipisahkan oleh titik (*.*), yaitu *header*, *payload*, dan *signature* (Edy, E., Ferdiansyah, F., Pramusinto, W., & Waluyo, S. (2019).

JSON Web Token adalah *token* yang biasa digunakan sebagai media autentikasi antara aplikasi *front-end* dan penyedia layanan. *Token* ini menggunakan tipe data *string* dan berisi data penting yang digunakan aplikasi *front-end* untuk berkomunikasi dengan penyedia layanan. Isi dari *JSON Web Token* disebut *payload* dan dapat dilihat dengan mudah, tetapi tidak dapat dimodifikasi karena menggunakan kunci yang mengidentifikasi apakah *token* yang dikirim adalah *token* yang sama. Sehingga membatasi pengiriman dan perubahan informasi yang dimiliki oleh *token* (Saputra, M. H. K., 2021).

2.12 Waterfall

Menurut Novitasari (2018) Metode *waterfall* adalah metode yang menggambarkan pendekatan yang sistematis dan berurutan dalam pengembangan perangkat lunak. Tahapan dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan yaitu perencanaan, permodelan, konstruksi, sebuah system dan penyerahan sistem kepada pengguna, dukungan pada perangkat lunak lengkap yang dihasilkan. Adapun fase pada *waterfall* adalah seperti gambar berikut.



Gambar 2.1 Model Waterfall

Sumber : (Simatupang, J., & Sianturi, S., 2018)

Gambar di atas merupakan tahapan-tahapan pada model *waterfall*, adapun pada model ini terbagi atas lima tahap, yaitu :

1. Analisis Kebutuhan

Tahap ini merupakan analisis kebutuhan sistem yang diperlukan untuk menyederhanakan proses desain dan pengembangan sistem. Fase ini meliputi analisis proses bisnis yang sedang berjalan, analisis masalah dan analisis kebutuhan fungsional.

2. Desain

Tahap ini merupakan proses perancangan tampilan sistem yang sesuai dengan hasil analisis kebutuhan yang dilakukan selama tahap analisis.

3. *Development Program*

Tahap ini merupakan tahapan pengkodean sistem yang dihasilkan dari integrasi desain sistem, di mana dibuat dalam bahasa pemrograman dan *database* di desain pada *MySQL*.

4. Pengujian

Merupakan tahap pengujian dari sistem yang dibangun, yang bertujuan untuk mengetahui kelayakan dan kekurangan dari sistem tersebut. Hasil dari tahapan ini akan digunakan sebagai bahan evaluasi untuk perbaikan dan penambahan pada sistem yang telah dibangun.

5. Pemeliharaan

Merupakan tahap akhir dari pembuatan sistem perangkat lunak dan sistem yang telah dibangun dapat dimodifikasi atau ditambahkan sesuai dengan kebutuhan pengguna.

2.13 *UML (Unified Modelling Language)*

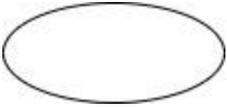
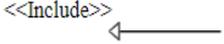
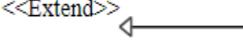
Menurut Heriyanto, Y (2018) “*UML* merupakan bahasa pemodelan yang berbentuk grafis yang digunakan untuk memvisualisasi, menspesifikasikan suatu sistem perangkat lunak”. Sedangkan menurut Nunu Nurdiana (2018:29) dalam jurnal Aman, M (2021) menyatakan bahwa “*Unified Modeling Language (UML)* adalah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggambarkan diagram dan teks-teks pendukung”.

Berikut beberapa diagram yang termasuk dalam *UML* :

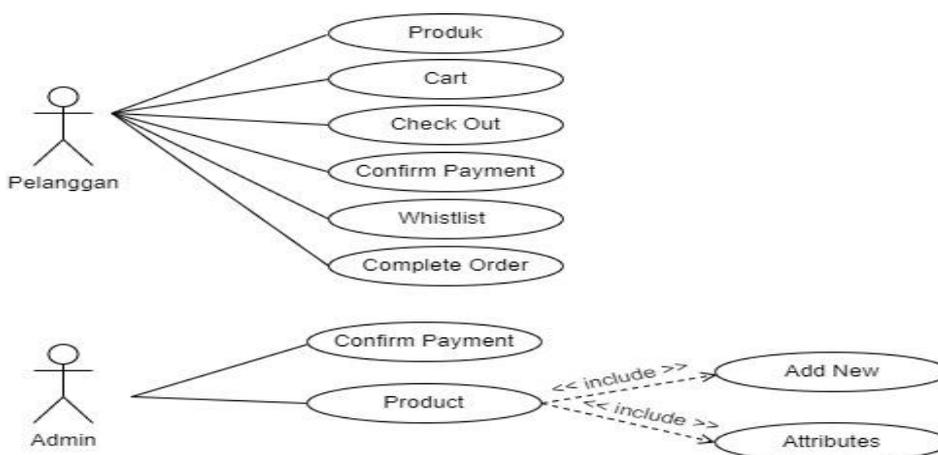
1. *Use Case Diagram*

Menurut Pratama (2019), “*Use case diagram* adalah gambaran grafis dari beberapa atau semua *actor*, *use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case diagram* tidak menjelaskan secara detil tentang penggunaan *use case*, tetapi hanya memberi gambaran singkat hubungan antara *use case*, aktor, dan sistem”. Berikut fungsi simbol-simbol pada diagram *use case* :

Tabel 2.1 Use Case Diagram

| Simbol | Keterangan |
|---|--|
|  | <i>Use case</i> menggambarkan fungsionalitas yang disediakan oleh sistem sebagai entitas di mana entitas secara aktif bertukar pesan antar entitas yang dinyatakan menggunakan kata kerja. |
|  | <i>Actor</i> atau aktor adalah abstraksi dari orang atau sistem lain yang mengaktifkan fungsi dari target sistem. |
|  | Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data. |
|  | Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem. |
|  | <i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan sebuah fungsi program. |
|  | <i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi. |
|  | <i>Dependency</i> , kelas antar kelas dengan makna kebergantungan antar kelas. |

Sumber : (Sukrianto, D. dan Agustina, S 2018)

**Gambar 2.2 Contoh Use Case Diagram**

Sumber : (Alda, M., 2020)

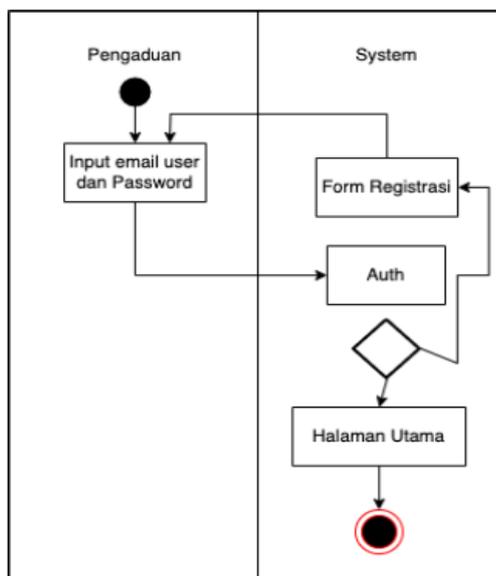
2. Activity Diagram

Diagram aktivitas menggambarkan alur kerja atau aktivitas dari sistem dan proses bisnis, atau menu yang ada pada perangkat lunak. Penekanan pada diagram aktivitas bukan pada apa yang dilakukan aktor, tetapi pada deskripsi aktivitas sistem atau aktivitas yang dapat dilakukan sistem (Simatupang, J., & Sianturi, S., 2019). Berikut adalah keterangan dan simbol-simbol pada *activity diagram* :

Tabel 2.2 Activity Diagram

| Simbol | Keterangan |
|---|--|
|  | <i>Initial Node</i> , Bagaimana objek dibentuk atau diawal. |
|  | <i>Activity</i> , Memperlihatkan bagaimana masing-masing kelas antar muka saling berinteraksi satu sama lain |
|  | <i>Decision</i> , Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari Satu. |
|  | <i>Activity Final Node</i> , akhir aktivitas |
|  | <i>Fork Node</i> , Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

Sumber : (Suppa, R., & Sulaeman, B., 2022)



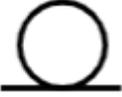
Gambar 2.3 Contoh Activity Diagram

Sumber : (Nugraha, E. S., Padri, A. R., Nurdyawan, O., Faqih, A., & Anwar, S., 2021)

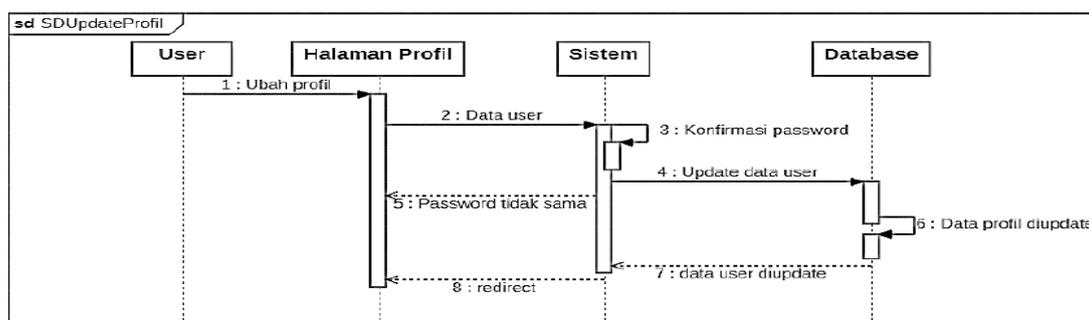
3. Sequence Diagram

Menurut Sukrianto, D. dan Agustina, S (2018) “Interaksi dari objek yang disusun dalam suatu urutan waktu atau kejadian tertentu dalam suatu proses, dapat digambarkan dengan *sequence diagram*”. Berikut adalah keterangan dan simbol-simbol pada *Sequency diagram*.

Tabel 2.3 Sequence Diagram

| Simbol | Keterangan |
|---|---|
|  | <i>Entity Class</i> , merupakan bagian dari system yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem. |
|  | <i>Boundary Class</i> , berisi kumpulan kelas yang menjadi interaksi antar satu atau lebih <i>actor</i> dengan sistem. |
|  | <i>Control Class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas. |
|  | <i>Message</i> , symbol mengirim pesan antar class. |
|  | <i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri. |
|  | <i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi. |
|  | <i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> . |

Sumber : (Sukrianto. D dan Agustina.S., 2018)



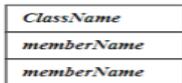
Gambar 2.4 Contoh Sequence Diagram

Sumber : (Naofal, N., Ulhaq, M. R. D., & Prianto, C., 2022)

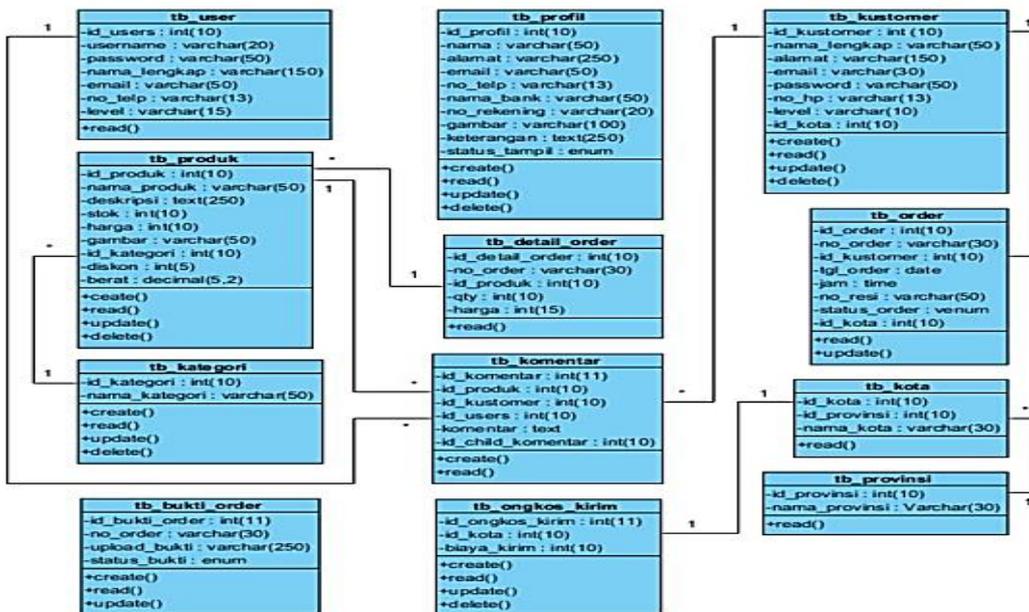
4. Class Diagram

Class Diagram adalah diagram yang menggambarkan struktur suatu sistem dalam hal mendefinisikan kelas yang dibuat untuk membuat sistem. Kelas ini berisi kumpulan objek yang mempunyai struktur, perilaku umum, hubungan, dan semantik / kata yang umum (Suppa, R., & Abduh, H. 2022). Berikut adalah simbol-simbol Class diagram :

Tabel 2.4 Class Diagram

| Simbol | Keterangan |
|---|---|
|  | Class, kelas pada struktur system. |
|  | Interface, sama dengan konsep antar muka dalam pemrograman berorientasi objek. |
|  | Directed association, Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain. |
|  | Generalisasi, relasi antar kelas dengan makna generalisasi spesialisasi (umum-khusus). |
|  | Dependency, relasi antar kelas dengan makna kebergantungan antar kelas. |
|  | Aggregation, relasi antar kelas dengan makna semua-bagian (whole-part). |

Sumber : (Sukrianto. D dan Agustina.S., 2018)



Gambar 2.5 Contoh Class Diagram

Sumber : (Hasanudin, M., 2019)

BAB III

METODE PENELITIAN

3.1 Bahan dan Alat Penelitian

Dalam penelitian ini, terdapat beberapa bahan dan alat penelitian yang akan digunakan, yaitu sebagai berikut :

1. Bahan Penelitian

Bahan ini bertujuan untuk mendukung proses penelitian, di mana sistem ini akan memudahkan proses penjualan dan pembelian produk kopi pada PT. Bejo Agro Lestari. Dalam membangun sistem ini, bahan penelitian yang diperlukan adalah data produk seperti gambar produk kopi, harga, dan detail informasi produk.

2. Alat Penelitian

Untuk mencapai tujuan dari dibangunnya sistem ini, diperlukan dukungan berupa alat penelitian dan pengembangan untuk sistem tersebut. Berikut alat pada penelitian ini.

1. Kebutuhan Perangkat Keras

Berikut alat penelitian berupa perangkat keras yang dibutuhkan untuk membangun sistem *E-Commerce* Komoditas Berbasis *PWA* Pada PT. Bejo Agro Lestari.

Tabel 3.1 Kebutuhan Perangkat Keras

| No | Jenis Perangkat Keras | Spesifikasi |
|-----------|------------------------------|--------------------|
| 1 | Processor | Intel Core I3 |
| 2 | Memori (Ram) | 8 GB |
| 3 | Hardisk | 500GB |

2. Kebutuhan Perangkat Lunak

Berikut alat penelitian berupa perangkat lunak yang dibutuhkan.

Tabel 3.2 Kebutuhan Perangkat Lunak

| No | Jenis Perangkat Lunak | Spesifikasi |
|----|-----------------------|---------------------|
| 1 | Sistem Operasi | Windows 10 |
| 2 | Text Editor | Visual Studio Code |
| 3 | Framework | Laravel 8 & Nuxt.Js |
| 4 | API Testing | Postman |
| 5 | Database | MySql |
| 6 | Web Server | Composer & Xampp |
| 7 | Desain Sistem | Draw.io |

3.2 Prosedur Penelitian

Proses penelitian dilakukan untuk mendapatkan penjelasan langkah-langkah dalam proses pembangunan sistem *E-Commerce* Komoditas Berbasis *PWA* Pada PT. Bejo Agro Lestari. Pada tahap ini akan diuraikan rangkaian kegiatan yang dilakukan penulis secara teratur dan sistematis untuk mencapai tujuan penelitian. Berikut prosedur pada penelitian ini yaitu :

3.2.1 Metode Pengumpulan Data

Metode yang digunakan penulis untuk melakukan proses pengumpulan data adalah dengan menggunakan metode kualitatif. Metode ini bertujuan untuk mengamati dan menganalisis data berdasarkan fakta yang ada di lapangan. Oleh karena itu, digunakan teknik pengumpulan data berupa observasi, wawancara, dan dokumentasi. Berikut langkah-langkahnya.

1. Observasi

Pada tahap ini, dilakukan observasi atau pengamatan lapangan dalam mengumpulkan data yang dibutuhkan, adapun lokasi yang dilakukan penulis bertempat di PT. Bejo Agro Lestari.

2. Wawancara

Pada tahap ini, penulis melakukan wawancara secara langsung kepada direktur perusahaan untuk mengetahui masalah yang sedang dihadapi, serta mencari informasi terkait data-data produk perusahaan berdasarkan fakta yang ada, sehingga dapat menghasilkan data dan informasi yang dibutuhkan.

3. Dokumentasi

Pada tahap ini dilakukan dengan melihat perusahaan secara langsung, yang kemudian didokumentasi seperti *text* dan gambar dari jenis dan kategori produk yang dipasarkan.

3.2.2 Metode Pengembangan Perangkat Lunak

Metode pengembangan yang digunakan oleh penulis dalam pengembangan sistem yaitu metode pengembangan *SDLC* (*System Development Life Cycle*) dengan model *waterfall*. Adapun tahapan pada model *waterfall* seperti analisis kebutuhan, desain, pembuatan kode program, pengujian (*testing*), dan pemeliharaan (*maintenance*) dapat dijelaskan sebagai berikut :

1. Analisis Kebutuhan

Merupakan tahap awal, di mana penulis melakukan proses pengumpulan data yang dilakukan untuk menentukan kebutuhan sistem. Proses pengumpulan data yang dibutuhkan adalah jenis produk, kategori produk, harga produk, dan jumlah stok produk. Sehingga sistem mudah dipahami sesuai dengan kebutuhan *admin* dan *customer*.

2. Desain

Tahap selanjutnya yaitu desain, pada tahap ini berfokus pada desain pembuatan program perangkat lunak termasuk struktur *database*, arsitektur perangkat lunak, rancangan *user interface & experience* sistem. Sehingga tahap selanjutnya dapat diimplementasikan menjadi kode program.

3. Pembuatan Kode Program

Pada tahap ini arsitektur perangkat lunak dan rancangan *interface* akan dibuat kedalam bentuk kode program. Adapun arsitektur perangkat lunak menggunakan *framework laravel* dan *nuxt.js*, penggunaan *testing rest api* dengan *postman*, struktur *database* menggunakan *MySQL*, serta prosedur pengodean berorientasi *api service oriented* di mana *frontend* dan *backend* dibuat terpisah dan dihubungkan dengan *web service rest api*. Pada proses pembuatan kode program penulis menggunakan *software open source visual studio code* sebagai *text editor*.

4. Pengujian

Selanjutnya, pengujian dilakukan pada sistem yang telah dirancang untuk memastikan bahwa semua komponen telah diuji. Hal ini dilakukan untuk meminimalkan kesalahan dan memastikan bahwa *output* yang dihasilkan sesuai dengan yang diinginkan. Pada

tahap ini penulis menggunakan *software postman* sebagai pengujian *rest api* dan *blackbox* sebagai metode pengujian sistem.

5. Pemeliharaan

Tahap terakhir di mana dilakukan perawatan atau pemeliharaan sistem dan tidak menutup kemungkinan sistem dapat diperbaiki setelah dikirimkan ke pengguna. Perubahan bisa terjadi karena *bug* atau *error* yang muncul dan tidak terdeteksi selama pengujian, tetapi tidak membuat perangkat lunak yang baru.

3.3 Analisis Kebutuhan

Analisis kebutuhan sistem dilakukan untuk mengidentifikasi dan mengumpulkan informasi tentang kebutuhan yang diperlukan untuk membangun sistem yang sedang diteliti. Analisis kebutuhan akan dijelaskan dalam beberapa sub bahasa antara lain analisis masalah dan analisis fungsional.

3.3.1 Analisis Masalah

Sistem yang sedang berjalan pada bisnis B2C atau *Business to Customer* di PT. Bejo Agro Lestari masih secara konvensional, berdasarkan hasil observasi dan wawancara yang telah dilakukan. Adapun penjualan yang dijalankan masih bersifat *door to door* dan penyebaran informasi penjualan produk masih menggunakan media sosial atau konsumen datang secara langsung. Hal ini tentu menjadi kendala yang dihadapi perusahaan, karena penjualan media sosial terbatas hanya menampilkan foto produk dan harga jual produk tersebut. Penyebaran informasi juga masih mengandalkan media brosur dan media sosial. Proses penjualan produk hanya dapat dilakukan dengan kontak informasi yang berupa nomor telepon dan *email* yang tertera pada media brosur dan *website company profile* pada PT. Bejo Agro Lestari. Sehingga langkah pemecahan masalah yang akan dilakukan pada penelitian ini yaitu akan dibuat sebuah sistem *E-Commerce* Komoditas Berbasis *PWA*. Di mana sistem yang akan dikembangkan nantinya mampu menjawab segala permasalahan diatas, diantaranya :

1. Transaksi penjualan tidak harus dilakukan secara langsung atau *door to door*.
2. Sistem ini nantinya dapat meningkatkan penjualan produk kopi pada perusahaan.
3. Metode pembayaran dapat dilakukan dengan metode *payment gateway*.

3.3.2 Analisis Fungsional

Analisis fungsional dimaksudkan untuk mengidentifikasi spesifikasi atau fitur yang akan dimiliki oleh sistem. Terdapat 2 pengguna yaitu *admin* dan *customer*.

1. *Admin*

Admin akan memiliki beberapa spesifikasi atau fitur yang terdapat pada sistem sebagai berikut :

- a. Dapat melakukan *login*.
- b. Dapat melihat informasi data *dashboard*.
- c. Dapat mengelola data kategori.
- d. Dapat mengelola data produk.
- e. Dapat melihat data *invoices*.
- f. Dapat mengelola data *customers*.
- g. Dapat mengelola data *sliders* (iklan).
- h. Dapat mengelola data *admin*.
- i. *Logout*.

2. *Customer* (pengunjung)

Customer dapat mengunjungi sistem *E-Commerce* Komoditas Berbasis *PWA* dengan spesifikasi sistem sebagai berikut :

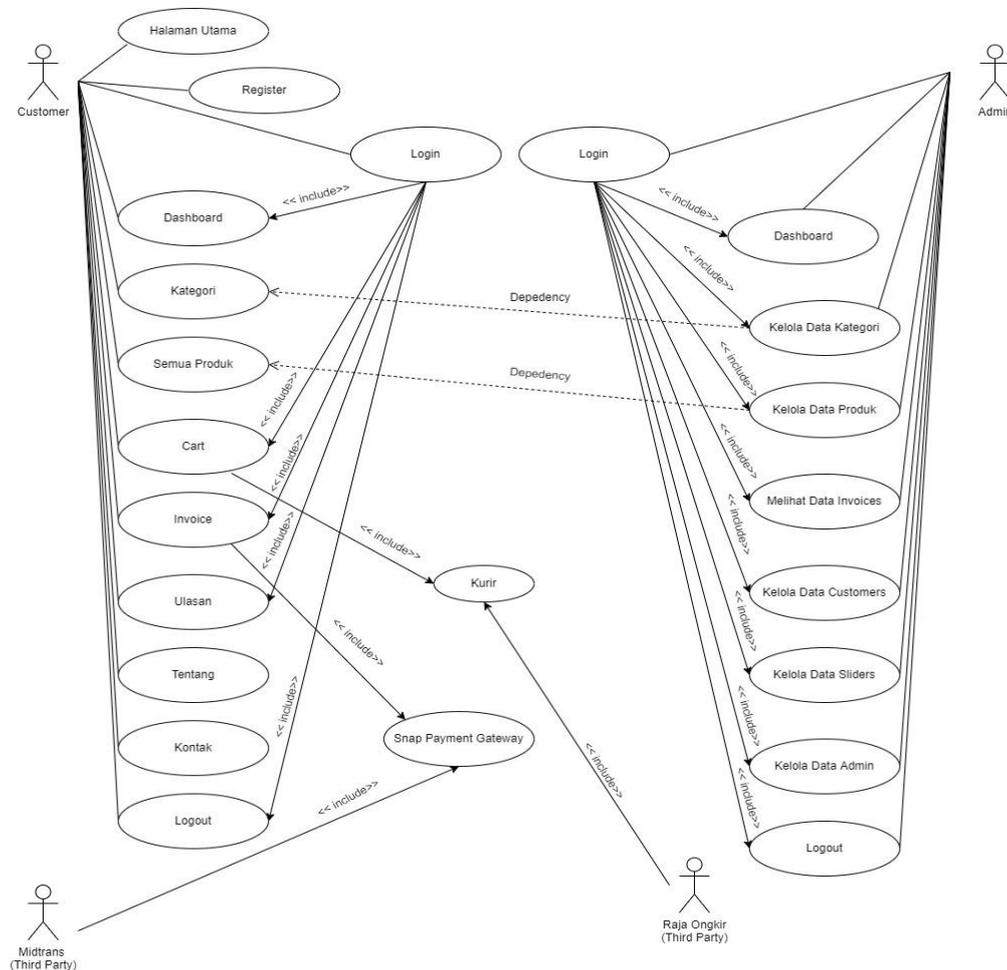
- a. Dapat mengunjungi halaman *home*.
- b. Dapat melakukan *login*.
- c. Dapat melakukan *register*.
- d. Dapat melihat kategori.
- e. Dapat melihat produk.
- f. Dapat melihat tentang dan kontak perusahaan.
- g. Dapat melakukan *order* atau transaksi melalui *payment gateway*.
- h. Dapat melihat *dashboard customer*, jika sudah melakukan *login*
- i. Dapat memilih kurir jasa pengiriman.
- j. Dapat melihat detail *order (invoice)*.
- k. Dapat melihat dan input ulasan produk.
- l. *Logout*

3.4 Perancangan Sistem

Perancangan yang dipakai untuk membangun sistem ini menggunakan *UML (Unified Modelling Language)* yaitu metode pemodelan visual yang digunakan sebagai sarana untuk merancang sistem berorientasi objek. Adapun *UML* yang digunakan pada perancangan ini diantaranya *Use Case Diagram, Activity Diagram, Sequence Diagram* dan *Class Diagram*.

3.4.1 Use Case Diagram

Use Case Diagram merupakan suatu diagram yang memodelkan atau menggambarkan hubungan interaksi antara sistem dan aktor. Diagram ini digunakan untuk menjelaskan semua fungsi yang ada didalam sistem dan siapa saja yang berhak menggunakan fungsi tersebut. Berikut adalah gambar rancangan *Use Case Diagram*.



Gambar 3.1 Use Case Diagram

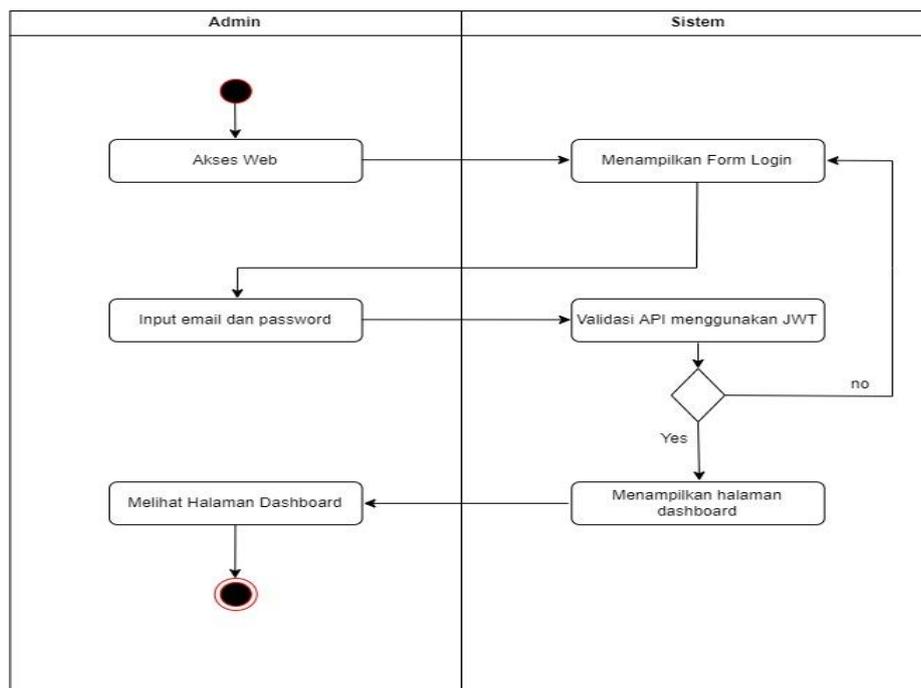
Keterangan gambar :

Pada gambar 3.1 Menjelaskan pada *use case diagram* terdapat 2 aktor utama yaitu *admin* dan *customer* (pengunjung), serta 2 aktor *third party* yaitu *midtrans* dan *Raja Ongkir*.

Adapun *admin* dapat melakukan *login*, melihat *dashboard*, kelola data kategori, kelola data produk, melihat data *invoices*, kelola data *customers*, kelola data *sliders*, kelola data *admin*, dan *logout*. Sedangkan *customer* dapat melihat halaman *home*, melakukan *register*, melakukan *login*, melihat *dashboard*, melihat kategori, melihat semua produk, memasukkan produk kedalam *cart* (*checkout*), melakukan transaksi pembayaran *invoice* (*payment gateway*), melihat / *input* ulasan produk (*optional*) dan *logout*. Sedangkan *midtrans* dan *raja ongkir* berfungsi sebagai aktor *third party* atau penyedia layanan *payment gateway* serta layanan kurir dan biaya ongkos pengiriman.

3.4.2 Activity Diagram

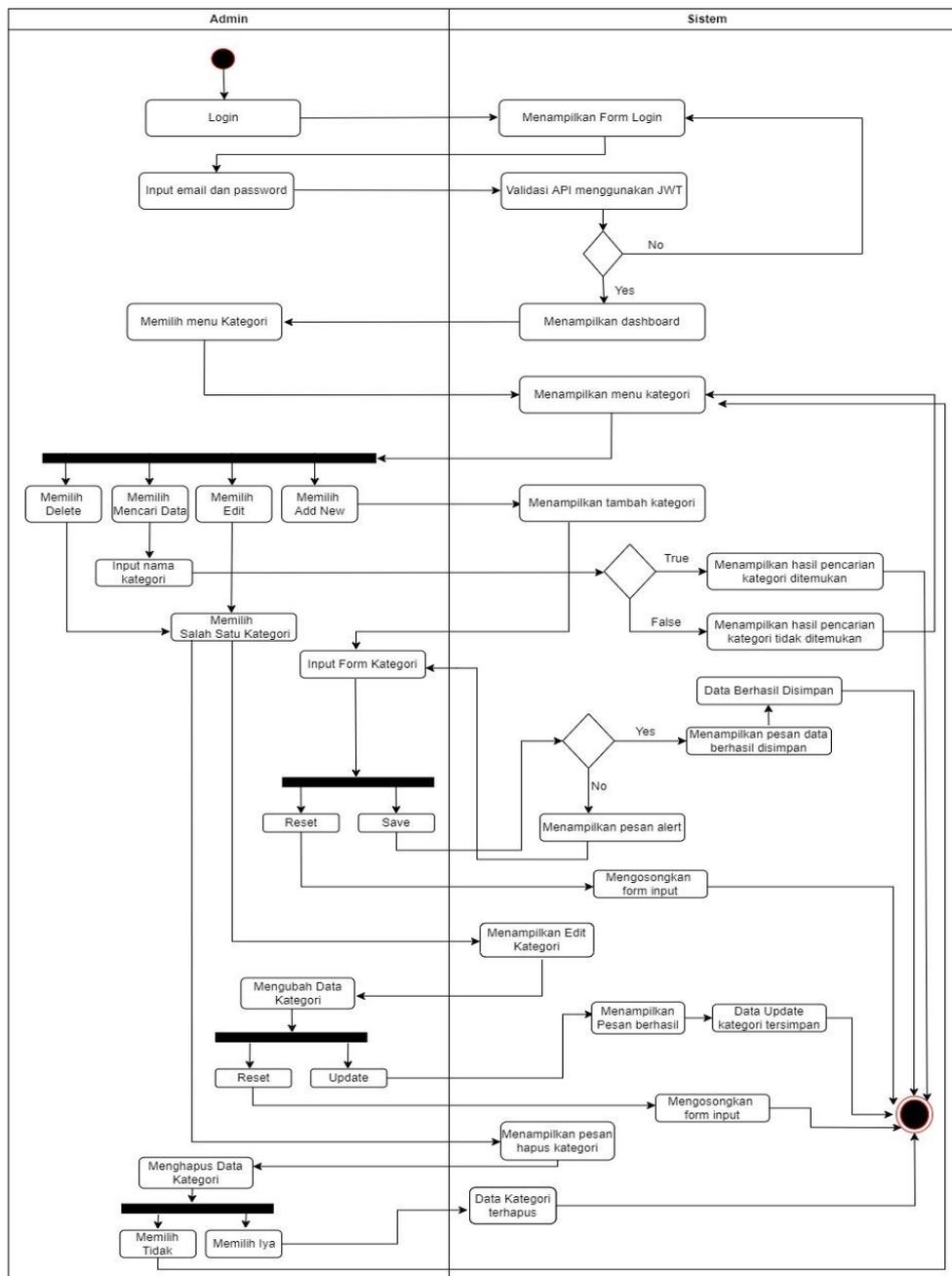
Activity Diagram merupakan sebuah diagram yang memodelkan atau menggambarkan aliran dari awal hingga akhir dari suatu proses pada sistem. Diagram ini juga digunakan untuk mendeskripsikan aktivitas sistem yang akan dijalankan dan mendefinisikan tampilan dari sistem. Berikut adalah gambar rancangan *Activity Diagram*.



Gambar 3.2 Activity Diagram Login Admin

Keterangan gambar :

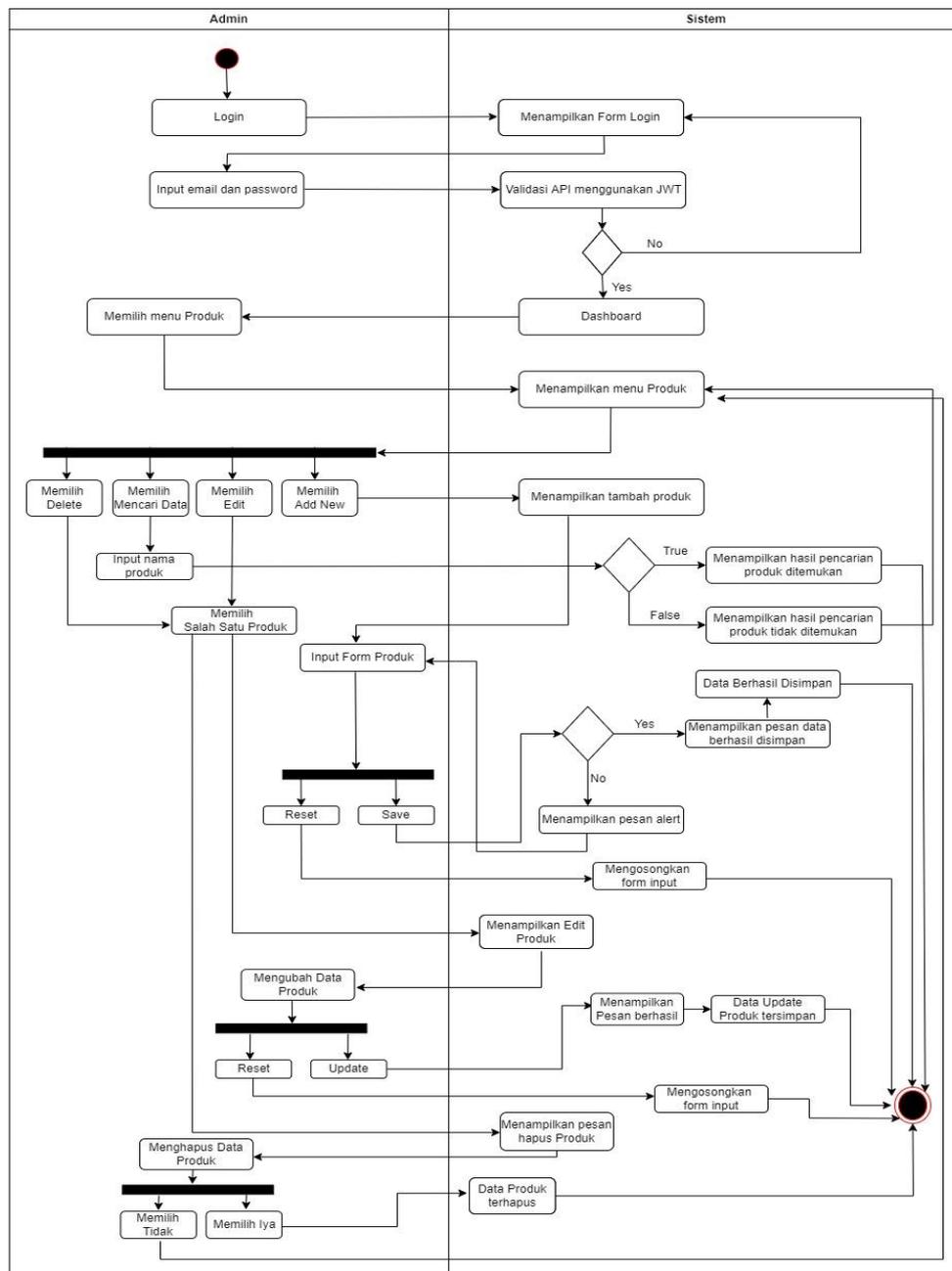
Pada gambar 3.2 menjelaskan *activity diagram login*, bahwa *admin* harus melakukan *input email* dan *password* dengan benar, kemudian proses *login* akan divalidasi menggunakan *JWT* dan pengecekan pada *database* apakah *email* atau *password* benar. Jika gagal maka akan dikembalikan ke *form login* kembali.



Gambar 3.3 Activity Diagram Kelola Kategori

Keterangan gambar :

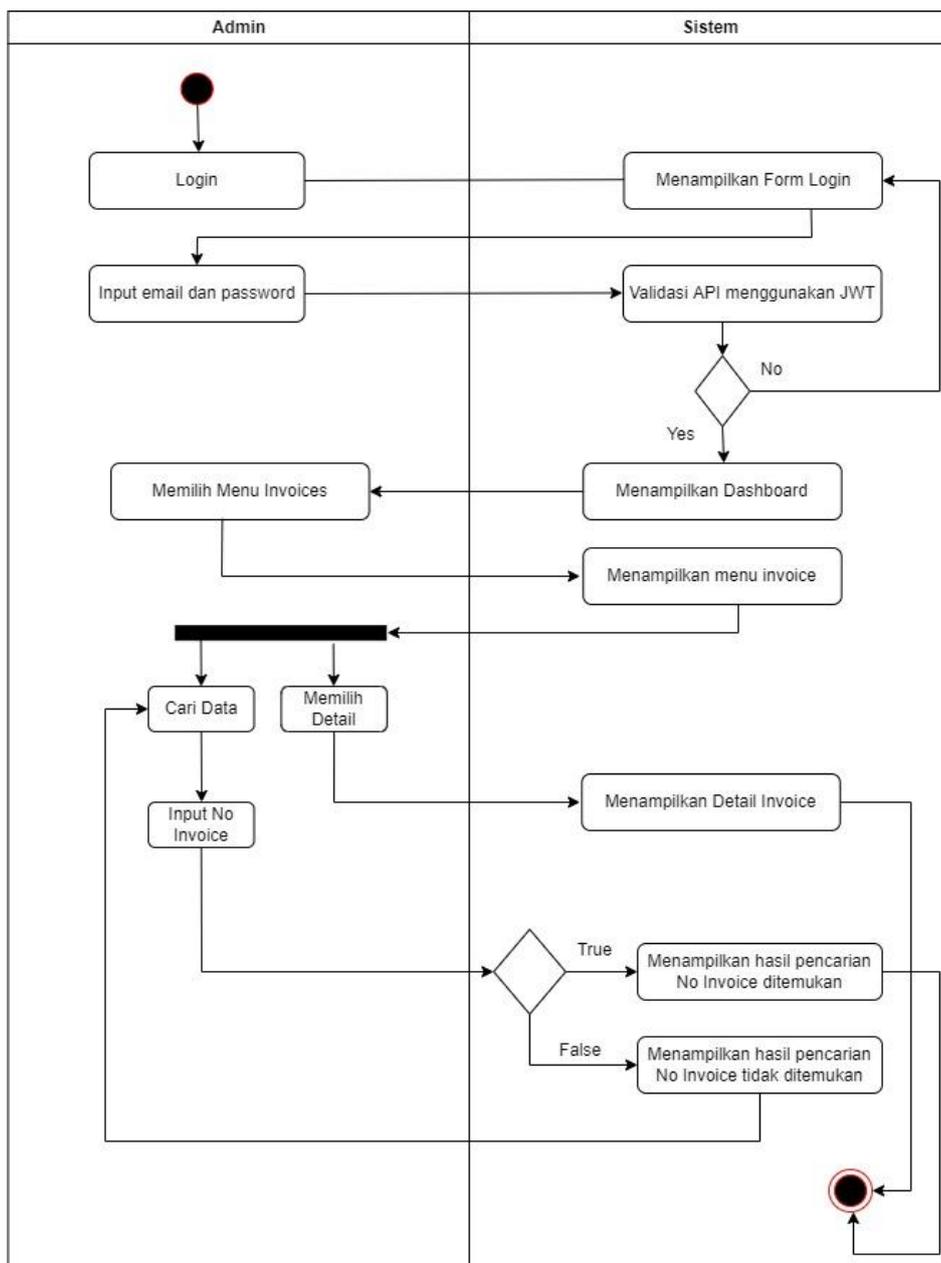
Pada gambar 3.3 menjelaskan *activity diagram* kelola kategori, di mana *admin* dapat melakukan aktivitas seperti *add new*, edit, mencari data dan *delete* data. Ketika *admin* melakukan *add new* maka sistem akan menampilkan tambah kategori, ketika *admin* memilih edit salah satu kategori maka sistem akan menampilkan edit kategori, ketika *admin* mencari data kategori maka *admin* harus *input* nama kategori dan sistem akan menampilkan hasil pencarian, ketika *admin* memilih hapus salah satu data kategori maka sistem akan menampilkan pesan hapus kategori.



Gambar 3.4 Activity Diagram Kelola Produk

Keterangan gambar :

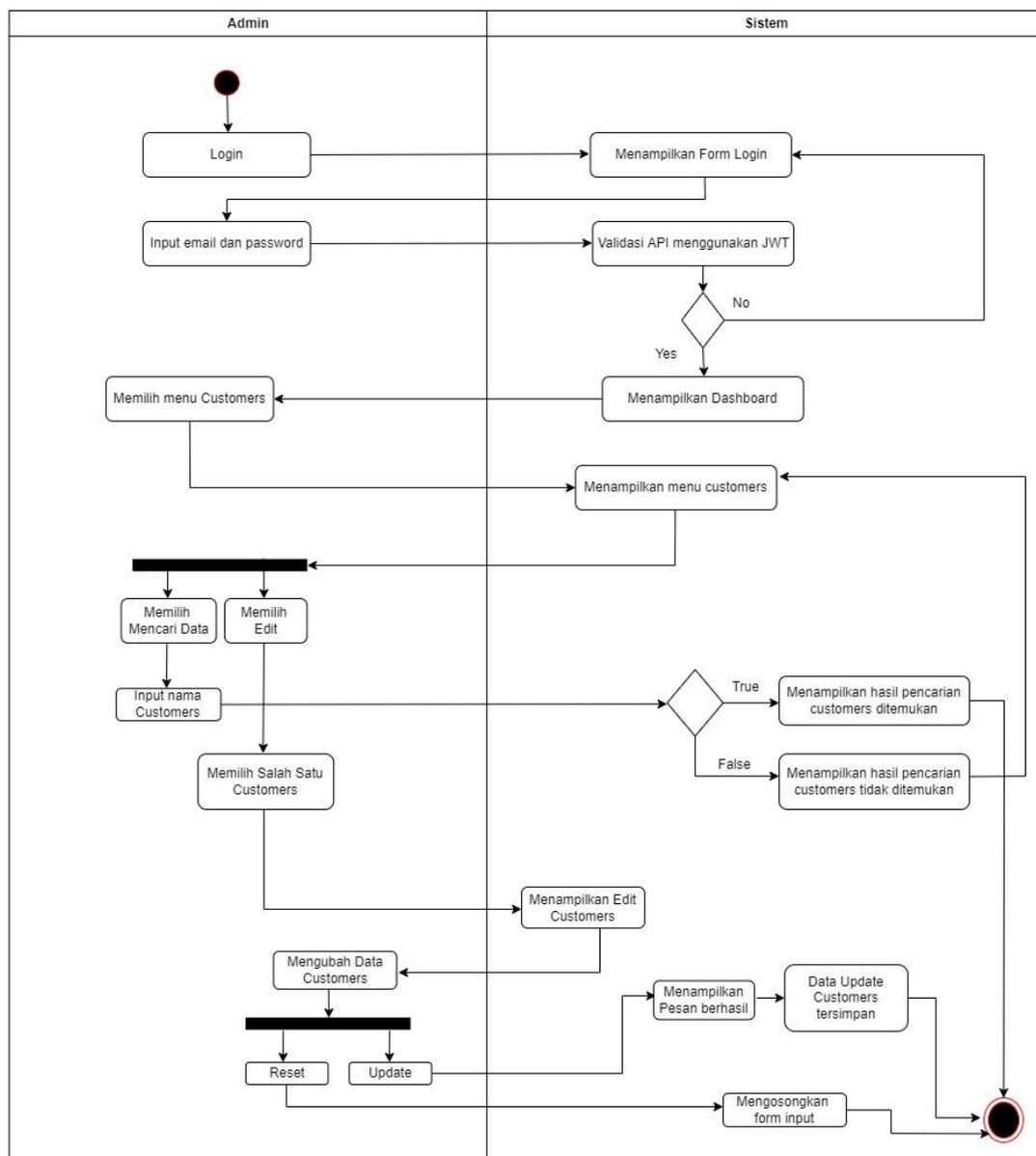
Pada gambar 3.4 menjelaskan *activity diagram* kelola produk, di mana *admin* dapat melakukan aktivitas seperti *add new*, edit, mencari data dan *delete* data. Ketika *admin* melakukan *add new* maka sistem akan menampilkan tambah produk, ketika *admin* memilih edit salah satu produk maka sistem akan menampilkan edit produk, ketika *admin* mencari data produk maka *admin* harus *input* nama produk dan sistem akan menampilkan hasil pencarian, ketika *admin* memilih hapus salah satu data produk maka sistem akan menampilkan pesan hapus produk.



Gambar 3.5 Activity Diagram Invoices Admin

Keterangan gambar :

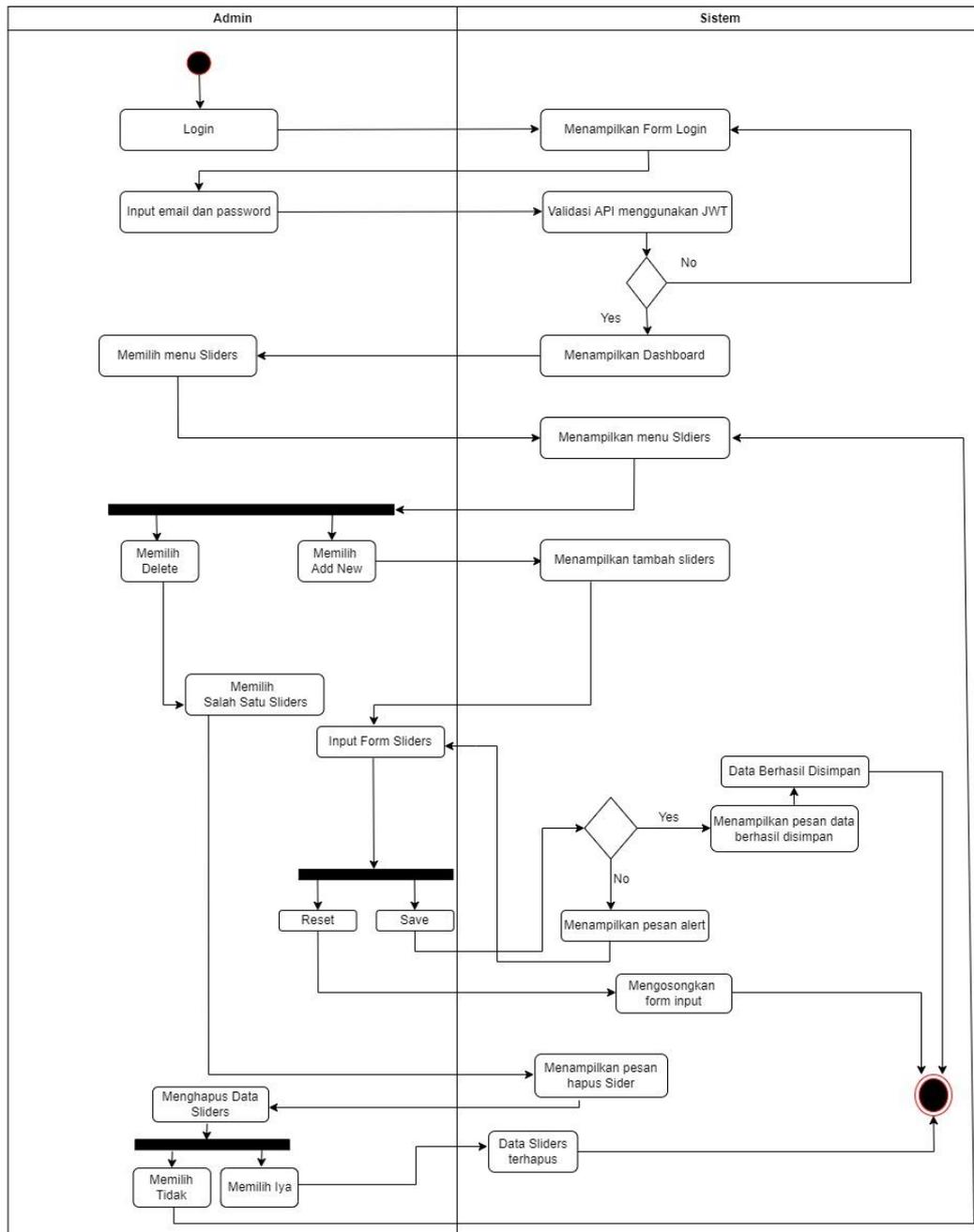
Pada gambar 3.5 menjelaskan *activity diagram invoices*, di mana *admin* dapat memilih detail *invoice* dan mencari data *invoice* dengan melakukan *input* nomor *invoice*.



Gambar 3.6 Activity Diagram Kelola Customers

Keterangan gambar :

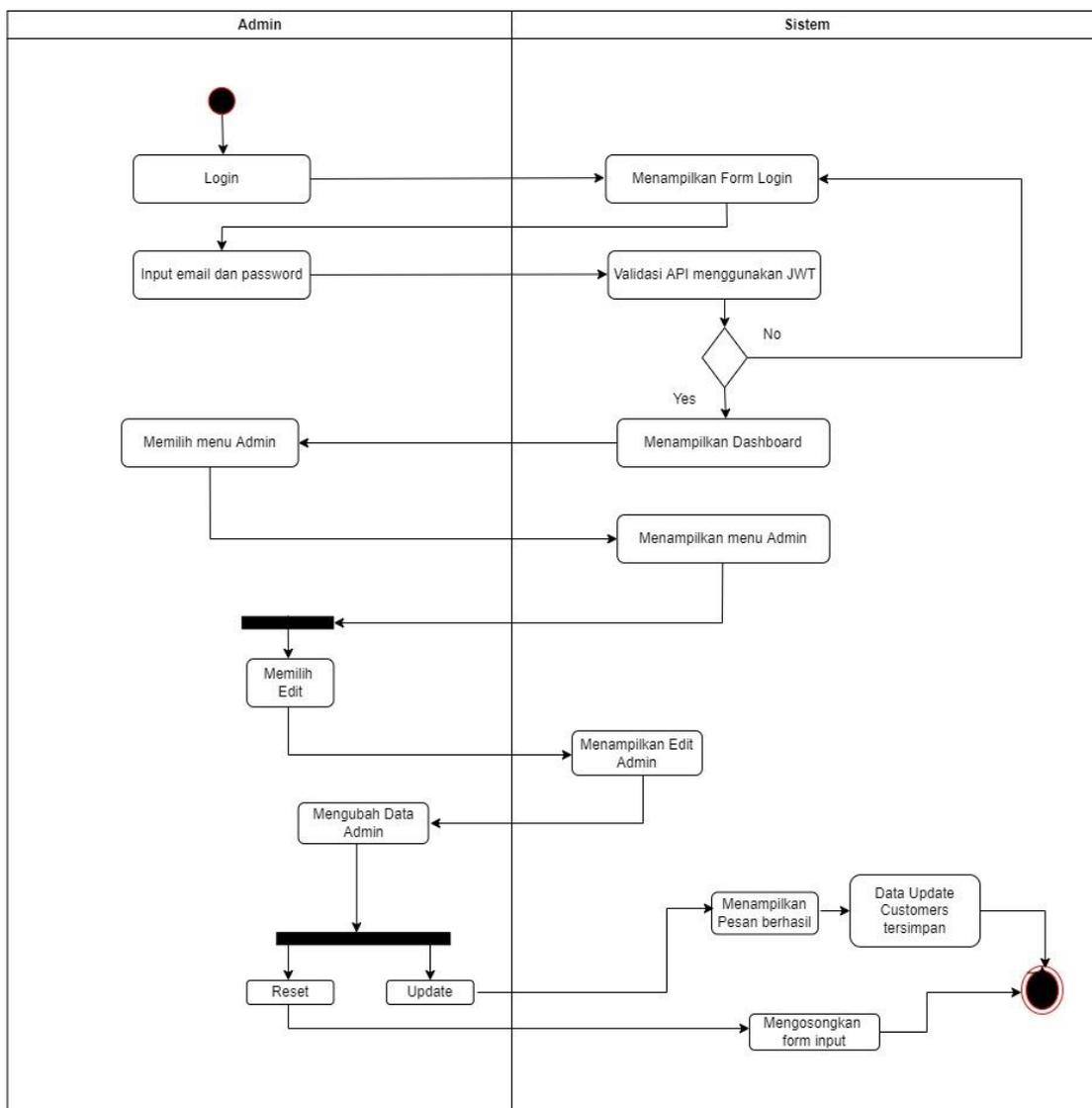
Pada gambar 3.6 menjelaskan *activity diagram* kelola *customers*, di mana *admin* dapat melakukan aktivitas *seperti* edit, dan mencari data. Ketika *admin* memilih edit salah satu *customers* maka sistem akan menampilkan edit *customers*, ketika *admin* mencari data *customers* maka *admin* harus *input* berdasarkan nama *customers* dan sistem akan menampilkan hasil pencarian.



Gambar 3.7 Activity Diagram Kelola Sliders

Keterangan gambar :

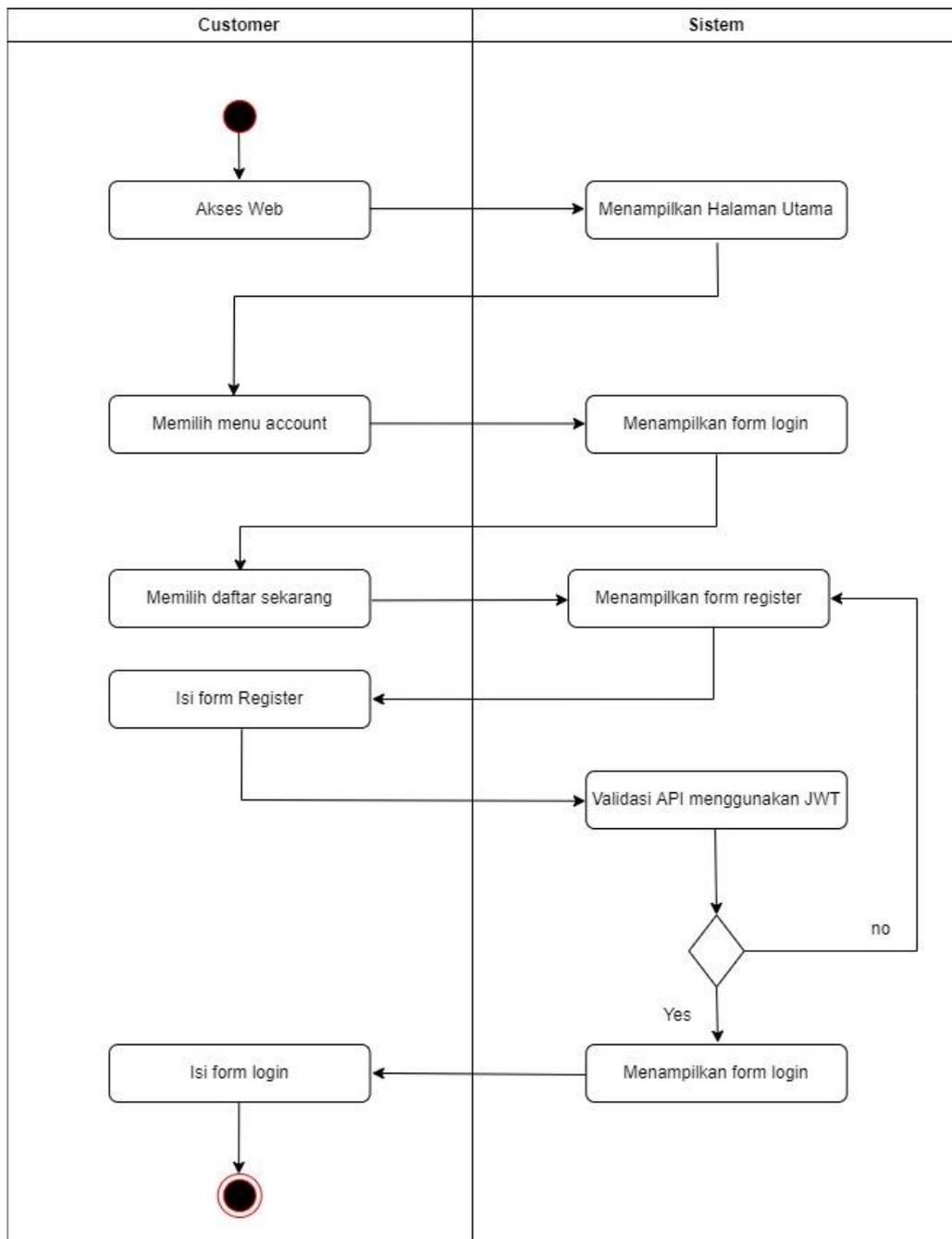
Pada gambar 3.7 menjelaskan *activity diagram* kelola sliders, di mana admin dapat melakukan aktivitas seperti *add new* dan *delete* data. Ketika admin melakukan *add new* maka sistem akan menampilkan tambah sliders, dan ketika admin memilih hapus salah satu data sliders maka sistem akan menampilkan pesan hapus data sliders.



Gambar 3.8 Activity Diagram Kelola Admin

Keterangan gambar :

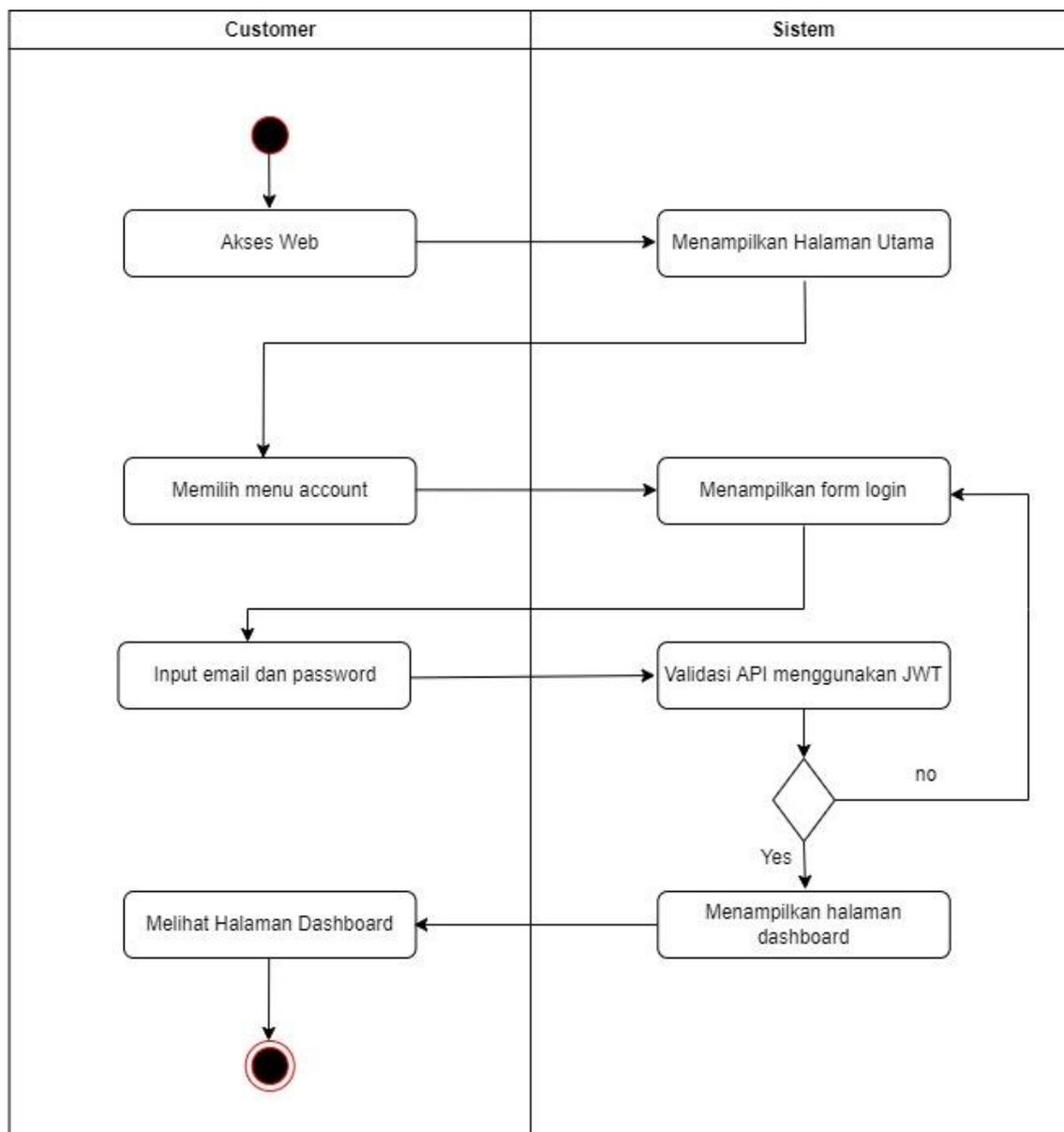
Pada gambar 3.8 menjelaskan *activity diagram* kelola *admin*, dimana dapat melakukan aktivitas edit data. Ketika *admin* melakukan edit data *admin* maka sistem akan menampilkan edit *admin*.



Gambar 3.9 Activity Diagram Register Customer

Keterangan gambar :

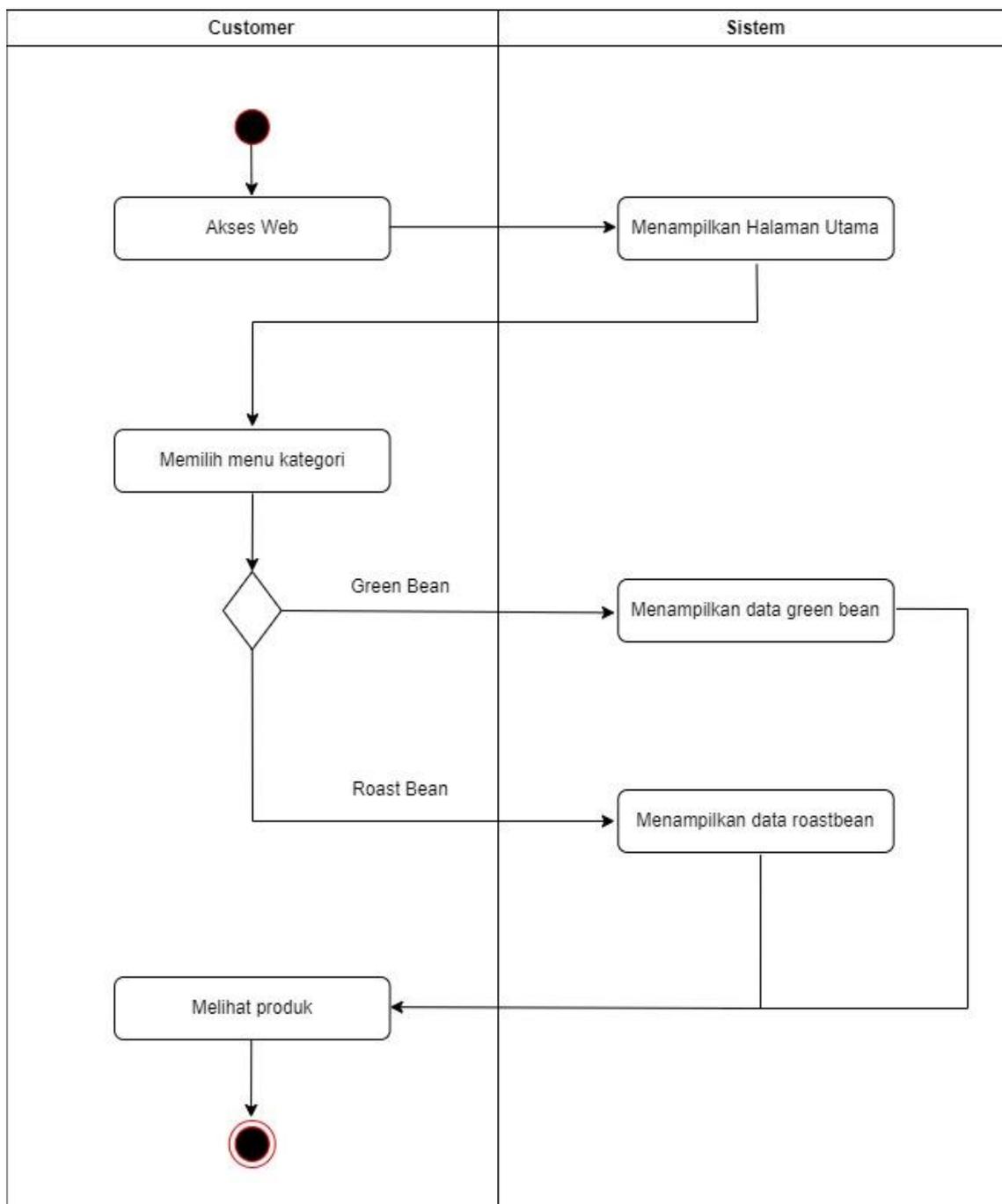
Pada gambar 3.9 menjelaskan *diagram register customer*, bahwa *customer* pada saat melakukan aktivitas mengakses *web*, maka sistem akan menampilkan halaman utama, yang kemudian memilih menu *account*, dan sistem akan menampilkan *form login*, selanjutnya *customer* melakukan klik *daftar sekarang* lalu sistem akan menampilkan *form register*, di mana pada *form* ini *customer* harus mengisi data dengan benar, jika *register* berhasil maka sistem akan menampilkan kembali *form login* dan *customer* dapat melakukan *login*.



Gambar 3.10 Activity Diagram Login Customer

Keterangan gambar :

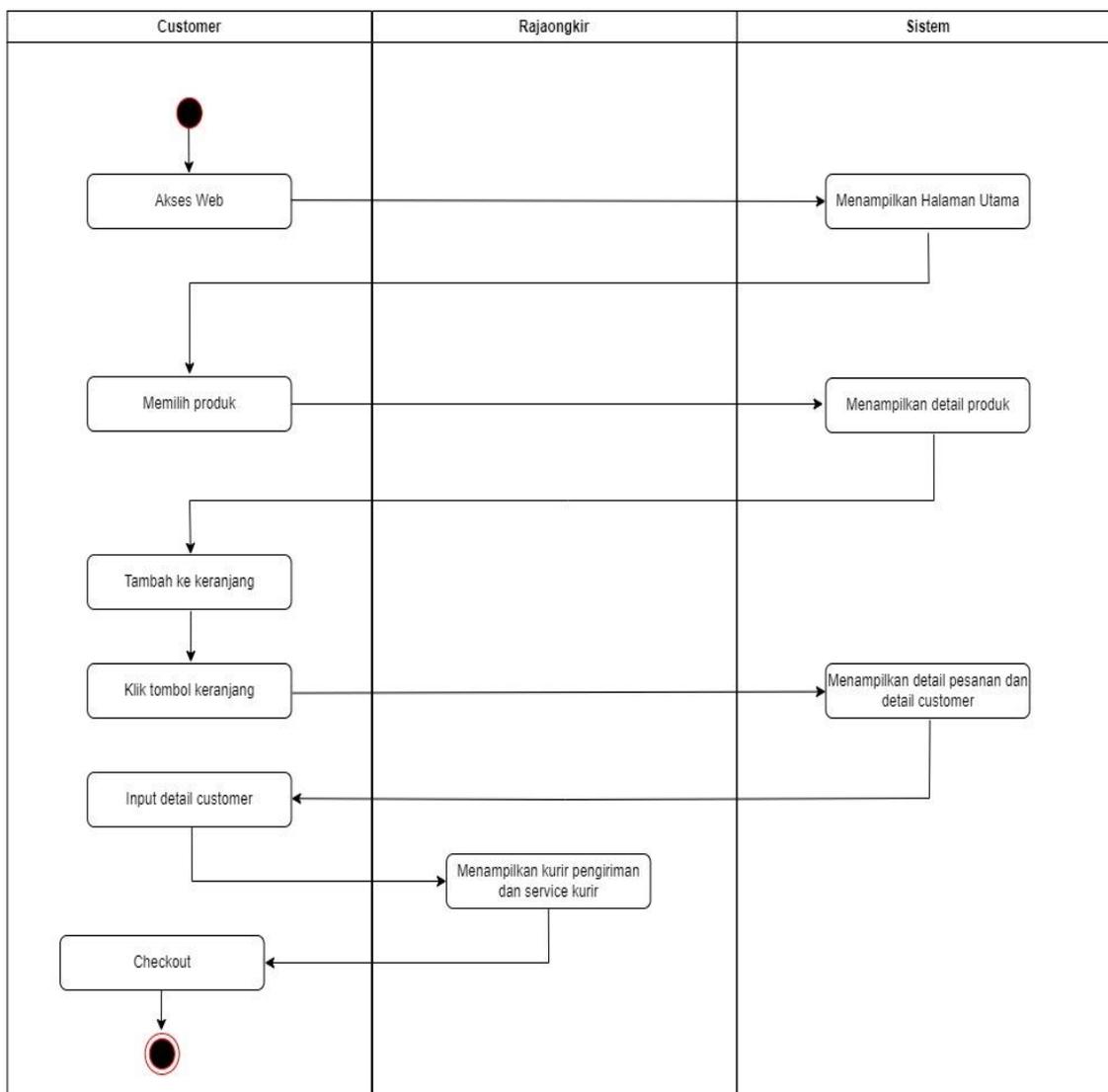
Pada gambar 3.10 menjelaskan *activity diagram login*, bahwa customer harus melakukan *input email* dan *password* dengan benar, kemudian proses *login* akan divalidasi menggunakan *JWT* dan pengecekan pada *database* apakah *email* atau *password* benar. Jika gagal maka akan dikembalikan ke *form login* kembali.



Gambar 3.11 Activity Diagram View Produk

Keterangan gambar :

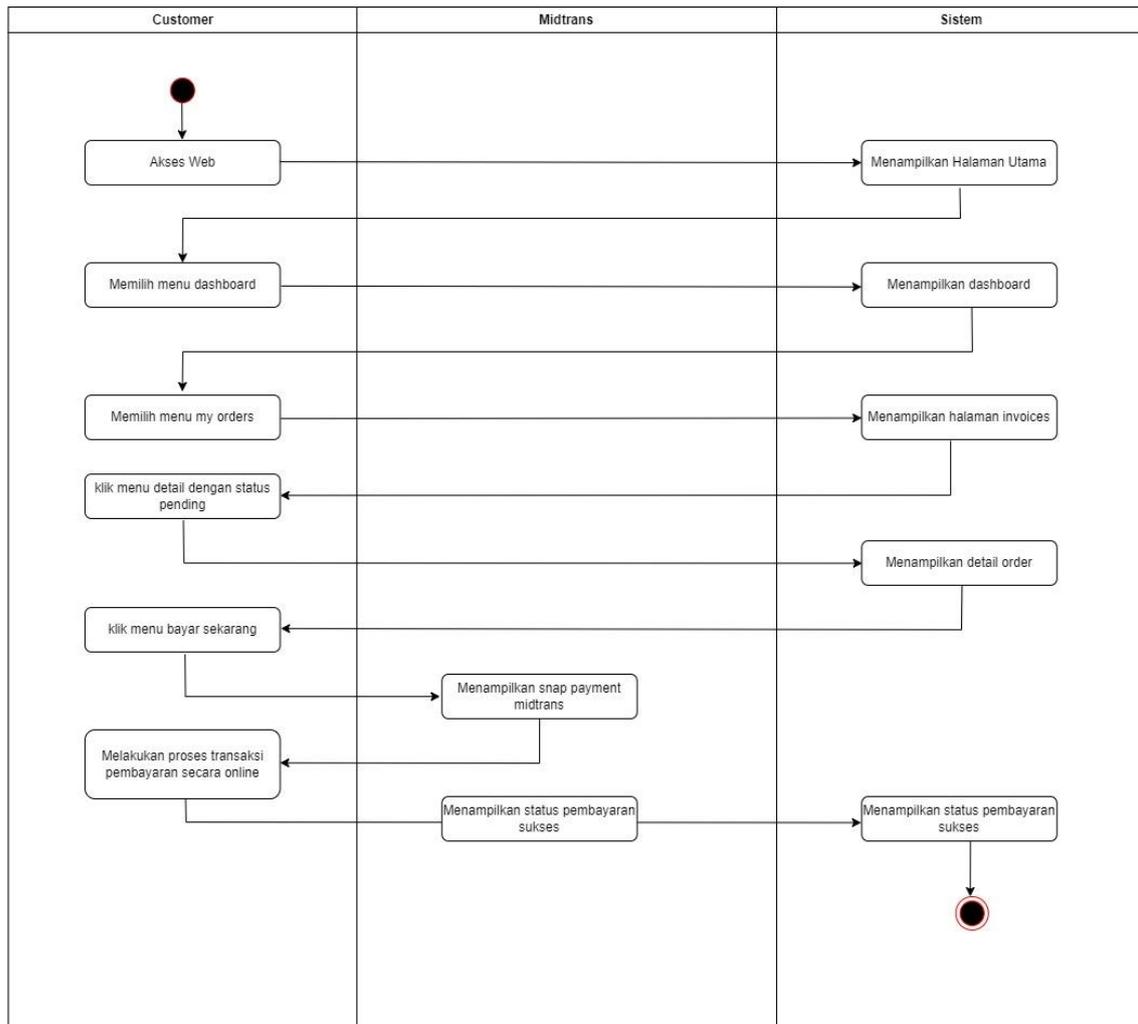
Pada gambar 3.11 menjelaskan diagram *view* produk, bahwa *customer* pada saat melakukan aktivitas mengakses *web*, *customer* dapat memilih menu kategori, di mana terdapat dua kategori yaitu *green bean* dan *roast bean*, ketika *customer* memilih salah satu dari kategori tersebut maka sistem akan menampilkan seluruh data produk sesuai kategori yang dipilih.



Gambar 3.12 Activity Diagram Order Produk

Keterangan gambar :

Pada gambar 3.12 menjelaskan diagram *order* produk, bahwa *customer* pada saat melakukan aktivitas mengakses *web*, *customer* memilih menu produk dan sistem akan menampilkan produk, selanjutnya *customer* menambahkan produk kedalam keranjang, jika sudah maka *customer* klik tombol keranjang, kemudian sistem akan menampilkan detail pesanan dan melakukan *input* detail *customer*, setelah melakukan *input* *customer* dapat klik *checkout*.



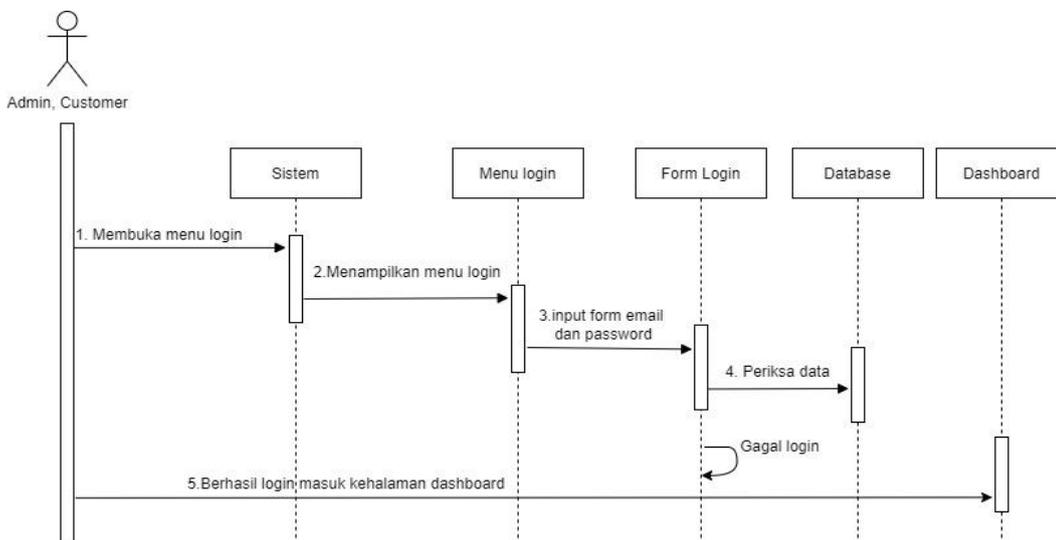
Gambar 3.13 Activity Diagram Transaksi Customer

Keterangan gambar :

Pada gambar 3.13 menjelaskan diagram transaksi *customer*, bahwa *customer* pada saat melakukan aktivitas mengakses *web*, *customer* memilih menu *dashboard*, selanjutnya *customer* memilih menu *my orders* dan sistem akan menampilkan halaman *invoice*, kemudian klik menu detail dengan status *pending* maka sistem akan menampilkan detail *order*, lalu klik bayar sekarang dan sistem menampilkan *snap pay* midtrans, jika *customer* sudah selesai melakukan transaksi maka status pembayaran akan berubah menjadi sukses.

3.4.3 Sequence Diagram

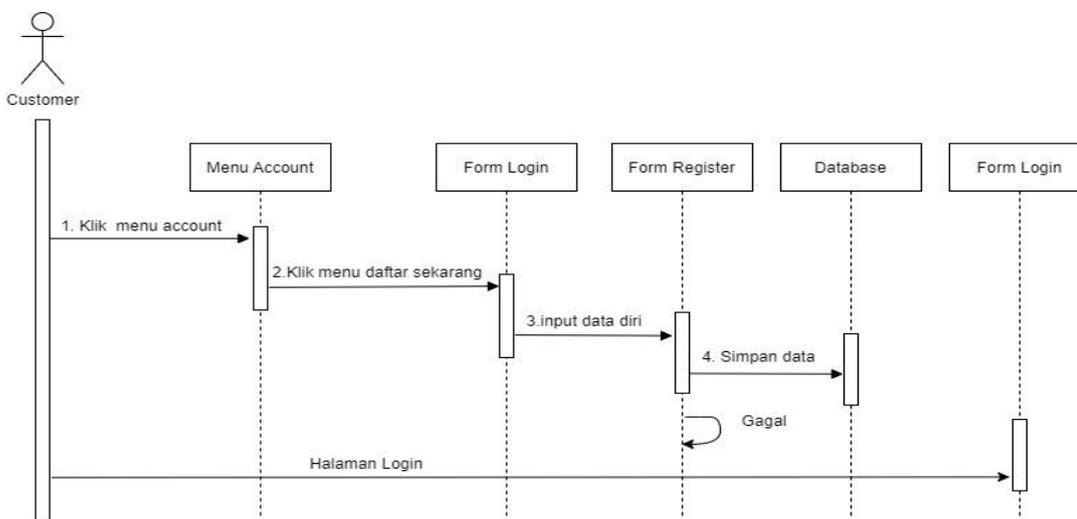
Sequence Diagram merupakan diagram yang menggambarkan langkah-langkah interaksi antar objek berdasarkan urutan waktu dan mendeskripsikan bagaimana sebuah proses pada sistem dapat berjalan bersamaan. Berikut adalah gambar rancangan *Sequence Diagram*.



Gambar 3.14 Sequence Diagram Login

Keterangan gambar :

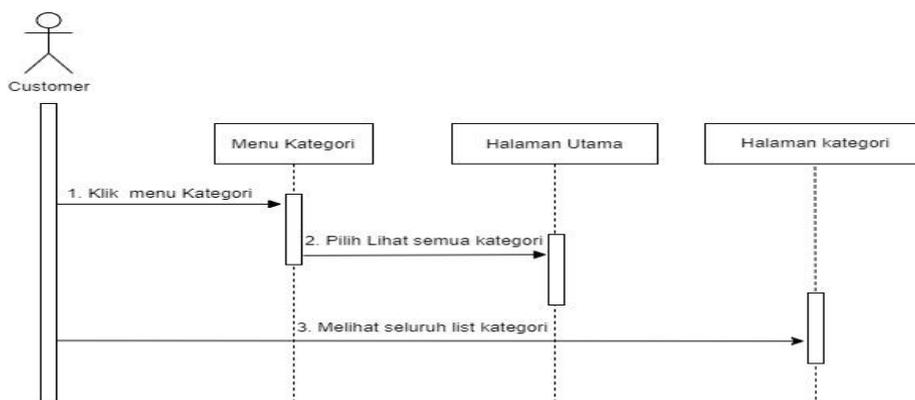
Pada gambar 3.14 menjelaskan bahwa *admin* dan *customer* ketika ingin melakukan *login*, harus memasukkan *email* dan *password* terlebih dahulu yang kemudian data akan di periksa pada *database*, jika data benar maka akan lanjut ke halaman *dashboard* dan jika gagal maka kembali ke *form login*.



Gambar 3.15 Sequence Diagram Register

Keterangan gambar :

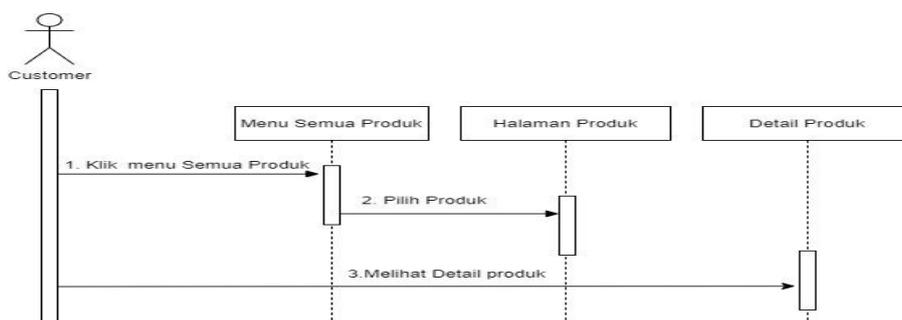
pada gambar 3.15 menjelaskan bahwa ketika *customer* ingin melakukan *registrasi* maka harus klik menu *account* yang kemudian akan diarahkan ke *form login* dan selanjutnya melakukan klik menu daftar sekarang pada *form login* tersebut, kemudian *customer* akan diarahkan ke *form register* serta melakukan *input* data diri dan klik tombol *register*, jika data yang di *input* benar maka akan diteruskan kehalaman *form login*, jika gagal maka kembali ke *form register*.



Gambar 3.16 Sequence Diagram View Kategori

Keterangan gambar :

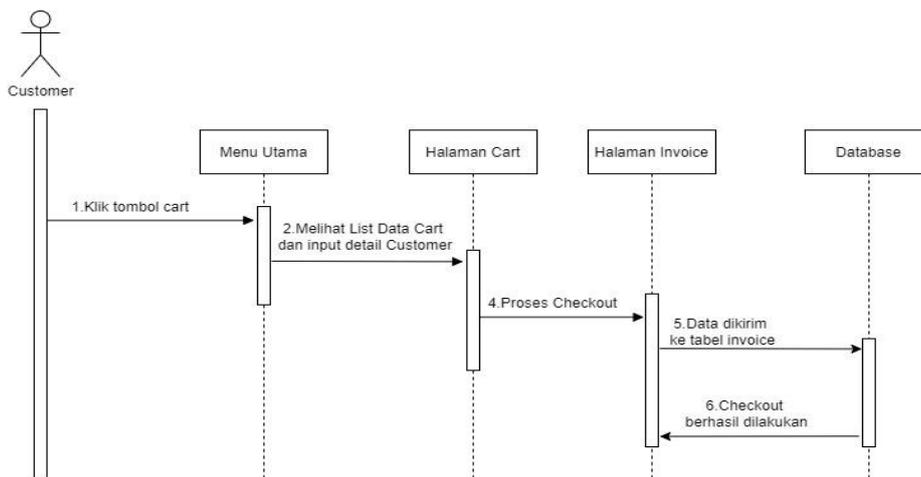
Pada gambar 3.16 menjelaskan, *customer* melakukan klik menu kategori pada halaman utama, dan klik tombol lihat semua kategori, makan akan ditampilkan seluruh *list* kategori.



Gambar 3.17 Sequence Diagram View Produk

Keterangan gambar:

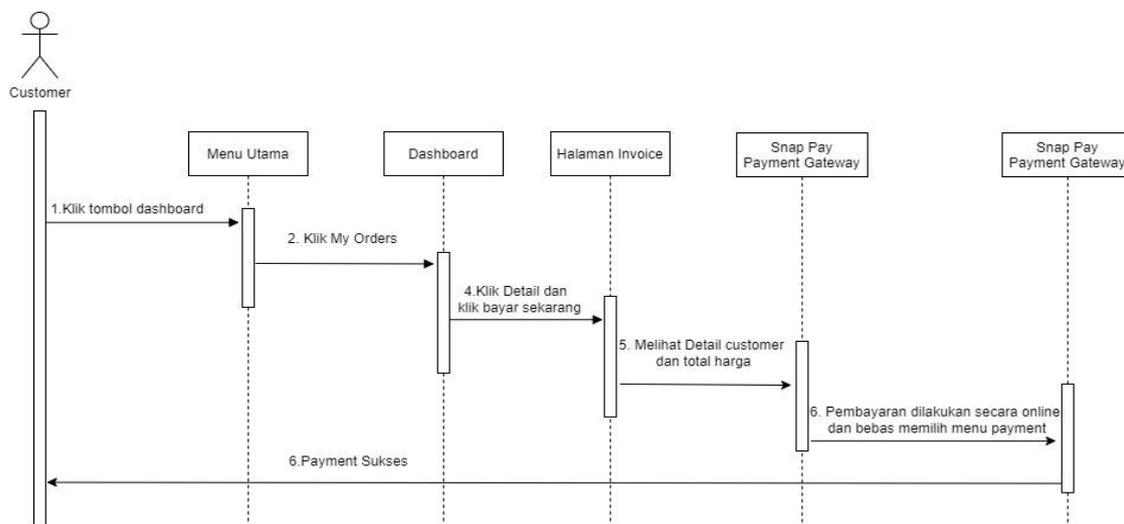
Pada gambar 3.17 menjelaskan, *customer* melakukan klik menu semua produk pada halaman utama, kemudian semua produk akan di tampilkan dihalaman produk, selanjutnya pilih produk untuk melihat detail produk tersebut.



Gambar 3.18 Sequence Diagram Cart Order

Keterangan gambar :

Pada gambar 3.18 menjelaskan, *customer* dapat melihat data pembelian produk dengan melakukan klik tombol *cart* di halaman utama, kemudian *customer* dapat melihat *list* data *order* di halaman *cart* untuk dapat dilihat dan dihapus, di halaman ini *customer* bisa melakukan *input* data detail *customer* dan jasa layanan kurir, selanjutnya melakukan proses *checkout* dan data akan tersimpan ke dalam *database*, setelah itu maka *checkout* berhasil dilakukan dan diarahkan ke halaman *invoice*.



Gambar 3.19 Sequence Diagram Transaksi

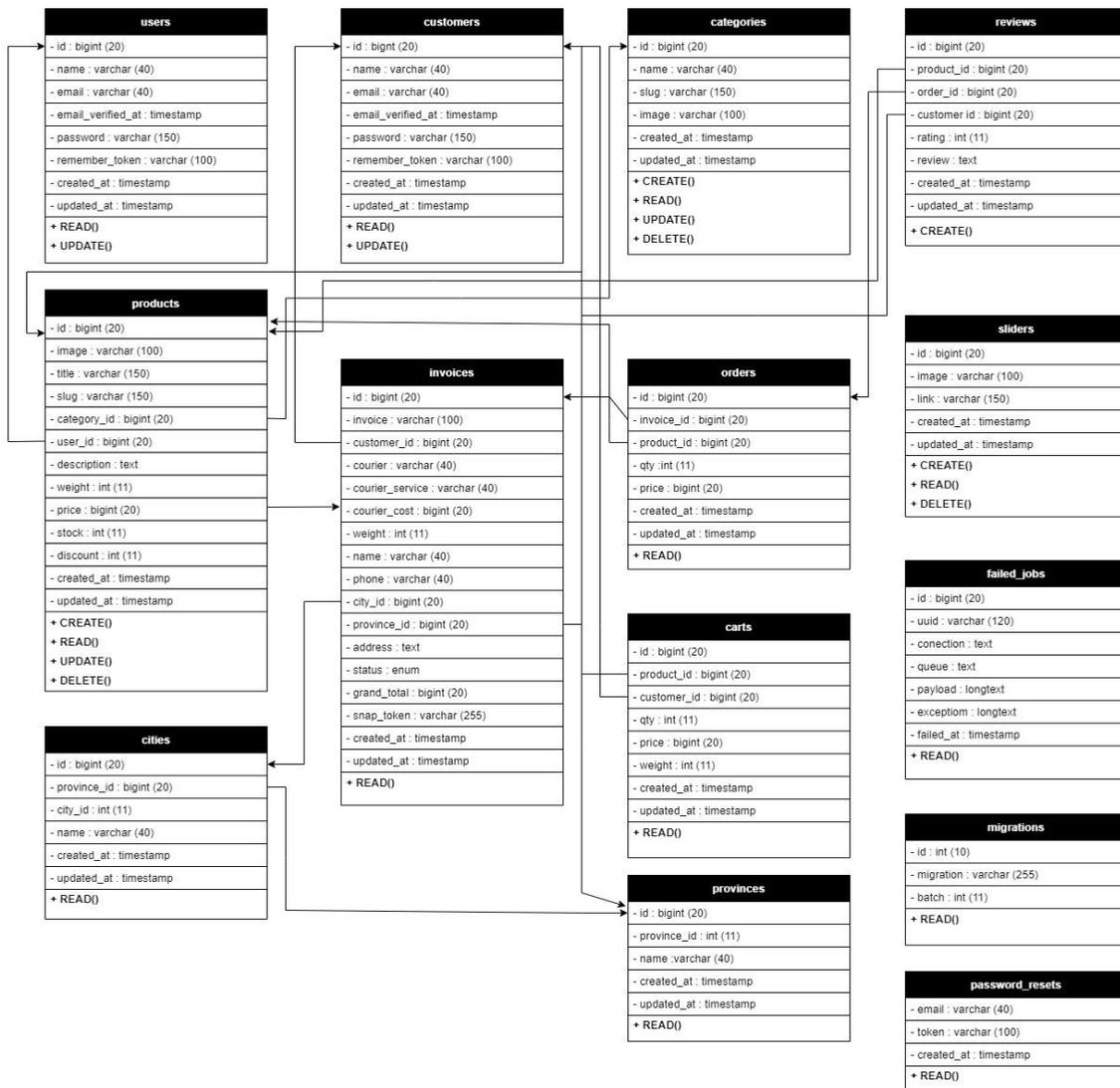
Keterangan gambar:

Pada gambar 3.19 menjelaskan, *customer* klik tombol *dashboard*, lalu klik *my orders*, setelah itu *customer* akan diarahkan ke halaman *invoice*, klik detail dan klik bayar sekarang,

selanjutnya sistem akan *direct* ke *snap pay* mindtrans dan terdapat info detail *customer* dan total harga, lalu *customer* dapat melakukan pembayaran secara *online*, jika *customer* sudah melakukan pembayaran maka status pembayaran akan otomatis menjadi sukses.

3.4.4 Class Diagram

Class Diagram merupakan sebuah *diagram* yang digunakan untuk menggambarkan struktur tabel-tabel dalam *database* sistem serta hubungannya antara *class*. Berikut adalah gambar rancangan pada *Class Diagram*.



Gambar 3.20 Class Diagram

Keterangan gambar:

Pada gambar 3.20 menjelaskan bahwa pada *database* memiliki banyak tabel seperti *users*, *customers*, *categories*, *reviews*, *products*, *invoices*, *orders*, *sliders*, *cart*, *provinces*,

cities, *failed jobs*, *migrations*, dan *password resets* yang saling berelasi. Adapun tabel yang tidak digunakan oleh sistem ini yaitu tabel *failed jobs*, *migrations*, dan *password resets* karena merupakan tabel bawaan dari *framework laravel*.

3.5 Perancangan Database

Perancangan *database* digunakan untuk membuat suatu sistem penyimpanan yang berisi banyak tabel-tabel yang saling berelasi dan menentukan nilai atau *type* data dari sebuah atribut yang terdapat di *database*. Berikut adalah desain *database* yang dirancang.

Tabel 3.3 Tabel users

| Nama Field | Tipe Data | Size | Status |
|-------------------|-----------|------|-------------|
| id | bigint | 20 | Primary Key |
| name | varchar | 40 | - |
| email | varchar | 40 | Index |
| email_verified_at | timestamp | - | - |
| password | varchar | 150 | - |
| remember_token | varchar | 100 | - |
| created_at | timestamp | - | - |
| updated_at | timestamp | - | - |

Tabel 3.4 Tabel customers

| Nama Field | Tipe Data | Size | Status |
|-------------------|-----------|------|-------------|
| id | bigint | 20 | Primary Key |
| name | varchar | 40 | - |
| email | varchar | 40 | Index |
| email_verified_at | timestamp | - | - |
| password | varchar | 150 | - |
| remember_token | varchar | 100 | - |
| created_at | timestamp | - | - |
| updated_at | timestamp | - | - |

Tabel 3.5 Tabel *categories*

| Nama Field | Tipe Data | Size | Status |
|------------|-----------|------|-------------|
| id | bigint | 20 | Primary Key |
| name | varchar | 40 | - |
| slug | varchar | 150 | - |
| image | varchar | 100 | - |
| created_at | timestamp | - | - |
| updated_at | timestamp | - | - |

Tabel 3.6 Tabel *products*

| Nama Field | Tipe Data | Size | Status |
|-------------|-----------|------|-------------|
| id | bigint | 20 | Primary Key |
| image | varchar | 100 | - |
| title | varchar | 150 | - |
| slug | varchar | 150 | - |
| category_id | bigint | 20 | Index |
| user_id | bigint | 20 | Index |
| description | text | - | - |
| weight | int | 11 | - |
| price | bigint | 20 | - |
| stock | int | 11 | - |
| discount | int | 11 | - |
| created_at | timestamp | - | - |
| updated_at | timestamp | - | - |

Tabel 3.7 Tabel *provinces*

| Nama Field | Tipe Data | Size | Status |
|-------------|-----------|------|-------------|
| id | bigint | 20 | Primary Key |
| province_id | int | 11 | - |
| name | varchar | 40 | - |
| created_at | timestamp | - | - |
| updated_at | timestamp | - | - |

Tabel 3.8 Tabel *cities*

| Nama Field | Tipe Data | Size | Status |
|-------------|-----------|------|-------------|
| id | bigint | 20 | Primary Key |
| province_id | bigint | 20 | Index |
| city_id | int | 11 | - |
| name | varchar | 40 | - |
| created_at | timestamp | - | - |
| updated_at | timestamp | - | - |

Tabel 3.9 Tabel *carts*

| Nama Field | Tipe Data | Size | Status |
|-------------|-----------|------|-------------|
| id | bigint | 20 | Primary Key |
| product_id | bigint | 20 | Index |
| customer_id | bigint | 20 | Index |
| qty | int | 11 | - |
| price | bigint | 20 | - |
| weight | int | 11 | - |
| created_at | timestamp | - | - |
| updated_at | timestamp | - | - |

Tabel 3.10 Tabel *orders*

| Nama Field | Tipe Data | Size | Status |
|------------|-----------|------|-------------|
| id | Bigint | 20 | Primary Key |
| invoice_id | Bigint | 20 | Index |
| product_id | Bigint | 20 | Index |
| qty | Int | 11 | - |
| price | bigint | 20 | - |
| created_at | timestamp | - | - |
| updated_at | timestamp | - | - |

Tabel 3.11 Tabel *invoices*

| Nama Field | Tipe Data | Size | Status |
|-----------------|--|------|-------------|
| id | bigint | 20 | Primary Key |
| invoices | varchar | 100 | - |
| customer_id | bigint | 20 | Index |
| courier | varchar | 40 | - |
| courier_service | varchar | 40 | - |
| courier_cost | bigint | 20 | - |
| weight | int | 11 | - |
| name | varchar | 40 | - |
| phone | varchar | 40 | - |
| city_id | bigint | 20 | Index |
| province_id | bigint | 20 | Index |
| address | text | - | - |
| status | enum ('pending', 'success', 'expired', 'failed') | - | - |
| grand_total | Bigint | 20 | - |
| snap_token | varchar | 255 | - |
| created_at | timestamp | - | - |
| updated_at | timestamp | - | - |

Tabel 3.12 Tabel *reviews*

| Nama Field | Tipe Data | Size | Status |
|-------------|-----------|------|-------------|
| id | bigint | 20 | Primary Key |
| product_id | bigint | 20 | Index |
| order_id | bigint | 20 | Index |
| customer_id | Bigint | 20 | Index |
| rating | Int | 11 | - |
| review | text | - | - |
| created_at | timestamp | - | - |
| updated_at | timestamp | - | - |

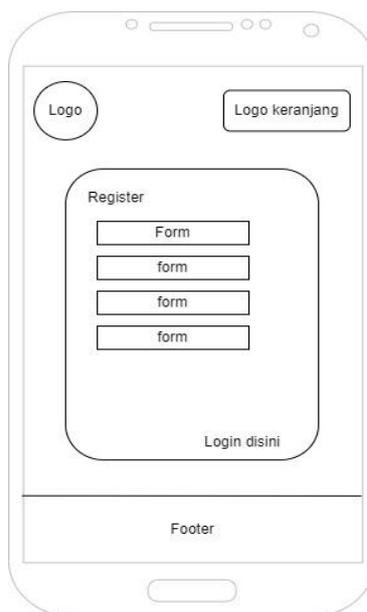
Tabel 3.13 Tabel sliders

| Nama Field | Tipe Data | Size | Status |
|------------|-----------|------|-------------|
| id | bigint | 20 | Primary Key |
| image | varchar | 100 | - |
| link | varchar | 150 | - |
| created_at | timestamp | - | - |
| updated_at | timestamp | - | - |

3.6 Desain Antar Muka

Desain *layout* merupakan salah satu tahap yang sangat penting dalam mengembangkan sebuah *website* atau aplikasi sebelum masuk ke tahap *development*. Desain *layout* juga berfungsi untuk membuat tata letak suatu desain *website* atau aplikasi agar sesuai dengan yang akan di kerjakan. berikut ini beberapa desain *layout* yang akan diimplementasikan ke dalam *PWA mobile* dan *website*:

1. Desain Halaman *Register PWA Mobile*

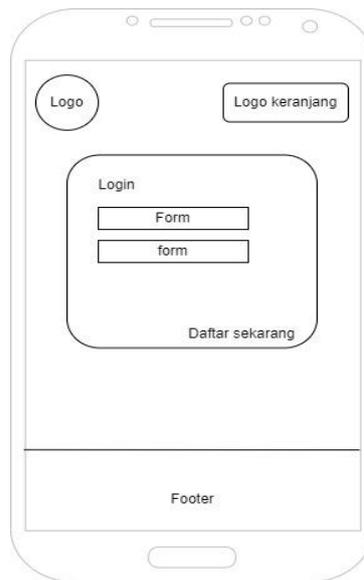


Gambar 3.21 Desain Halaman Register PWA Mobile

Keterangan gambar :

Pada gambar 3.21 menjelaskan desain halaman *register* pada *PWA mobile* yang memuat *form register*. *Customer* yang ingin melakukan transaksi atau pembelian produk diharuskan melakukan *input* data diri seperti nama, *email*, dan *password* pada *form register* yang telah dirancang.

2. Desain Halaman *Login PWA Mobile*

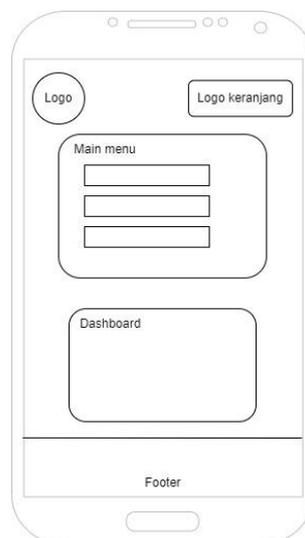


Gambar 3.22 Desain Halaman *Login PWA Mobile*

Keterangan gambar :

Pada gambar 3.22 menjelaskan desain halaman *login* pada *PWA mobile* yang memuat *form login*. *Customer* harus melakukan *input email* dan *password* dengan benar.

3. Desain halaman *Dashboard PWA Mobile*

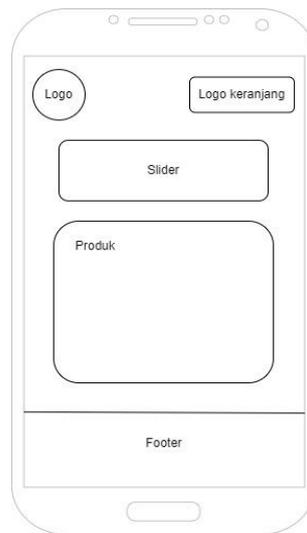


Gambar 3.23 Desain Halaman *Dashboard PWA Mobile*

Keterangan gambar :

Pada gambar 3.23 menjelaskan desain halaman *dashboard* pada *PWA mobile* yang memuat *main menu* dan *dashboard*.

4. Desain Halaman Utama *PWA Mobile*



Gambar 3.24 Desain Halaman Utama *PWA Mobile*

Keterangan gambar :

Pada gambar 3.24 menjelaskan desain halaman utama pada *PWA Mobile*, memuat logo dan menu, *slider*, serta katalog produk.

5. Desain Halaman Detail Produk *PWA Mobile*

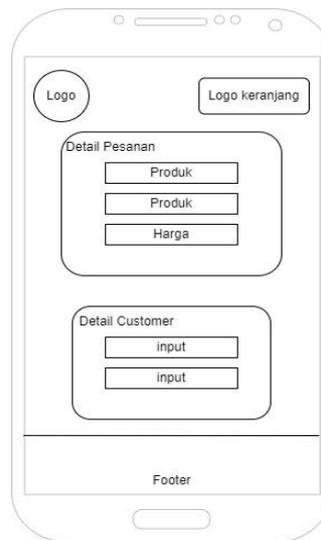


Gambar 3.25 Desain Halaman Detail Produk *PWA Mobile*

Keterangan gambar :

Pada gambar 3.25 menjelaskan desain halaman detail produk pada *PWA mobile*, memuat foto produk, detail, dan ulasan produk.

6. Desain Halaman *Cart PWA Mobile*

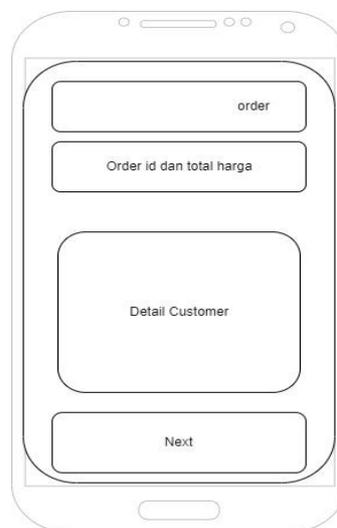


Gambar 3.26 Desain Halaman *Cart PWA Mobile*

Keterangan gambar :

Pada gambar 3.26 menjelaskan desain halaman *cart* pada *PWA mobile*, memuat informasi detail pesanan, dan *input* detail *customer*.

7. Desain Tampilan *Snap pay Midtrans PWA Mobile*

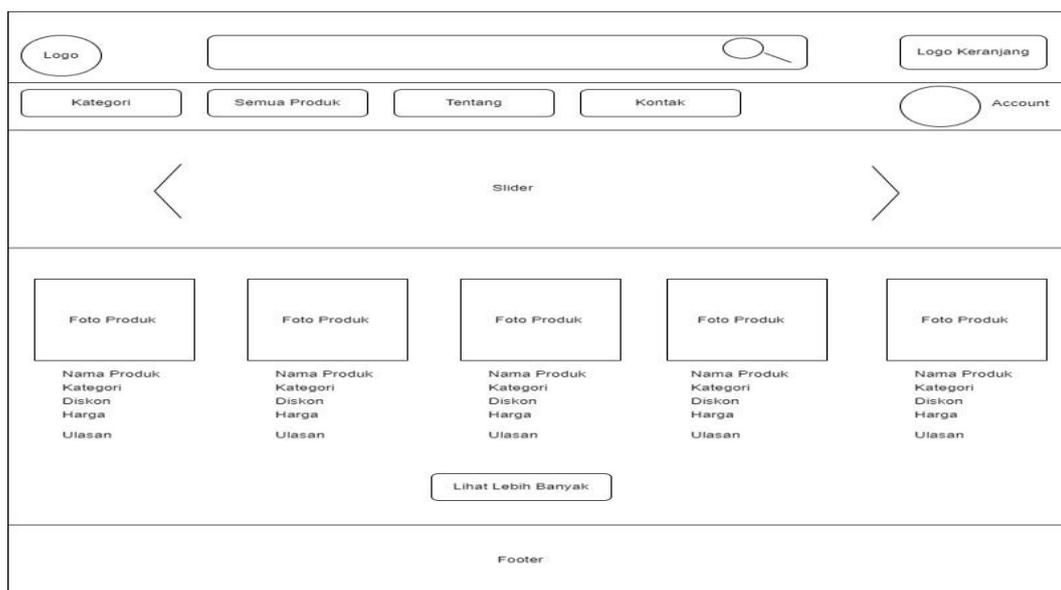


Gambar 3.27 Desain Tampilan *Snap pay Midtrans PWA Mobile*

Keterangan gambar :

Pada gambar 3.27 menjelaskan desain tampilan antar muka *snap pay* midtrans pada *PWA mobile* memuat informasi detail pesanan, dan melakukan *input* detail *customer*.

8. Desain Halaman Utama *Website*

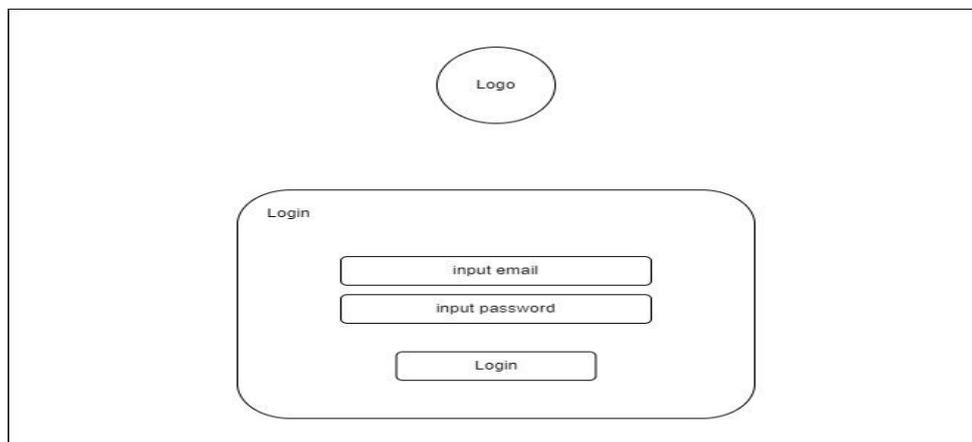


Gambar 3.28 Desain Halaman Utama

Keterangan gambar :

Pada gambar 3.28 menjelaskan desain halaman utama *website* memuat logo dan menu, *slider*, serta katalog produk.

9. Desain Halaman *Login (Admin)*

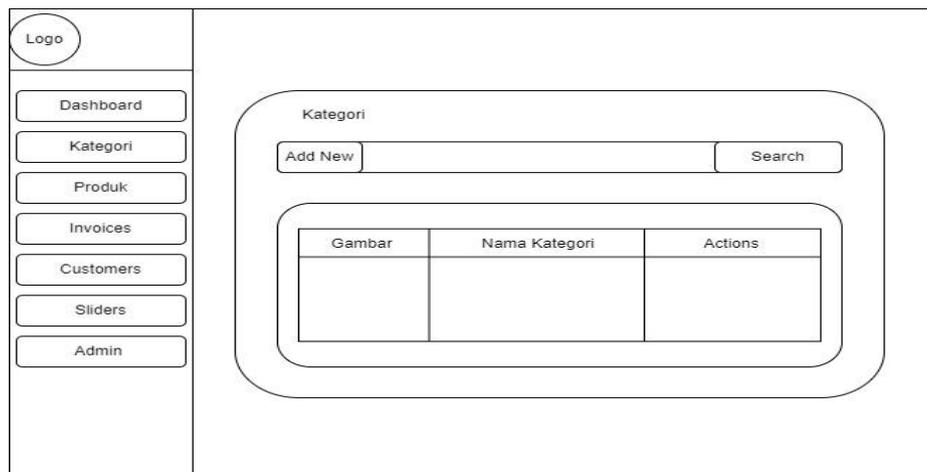


Gambar 3.29 Desain Halaman *Login Admin*

Keterangan gambar :

Pada gambar 3.29 menjelaskan desain halaman *login admin* pada *website* memuat logo, *form input email* dan *password*.

10. Desain Halaman Kategori (*Admin*)

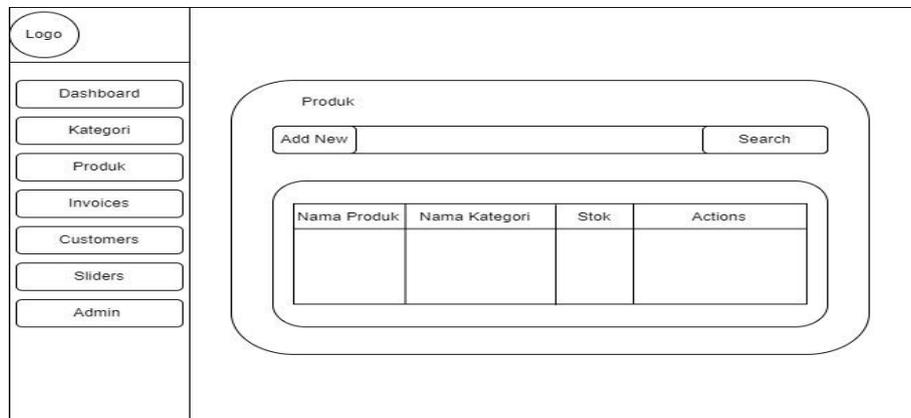


Gambar 3.30 Desain Halaman Kategori *Admin*

Keterangan gambar :

Pada gambar 3.30 menjelaskan desain halaman kategori *admin* pada *website* memuat tabel kategori *admin*, di mana *admin* bisa menambah kategori.

11. Desain Halaman Produk (*Admin*)

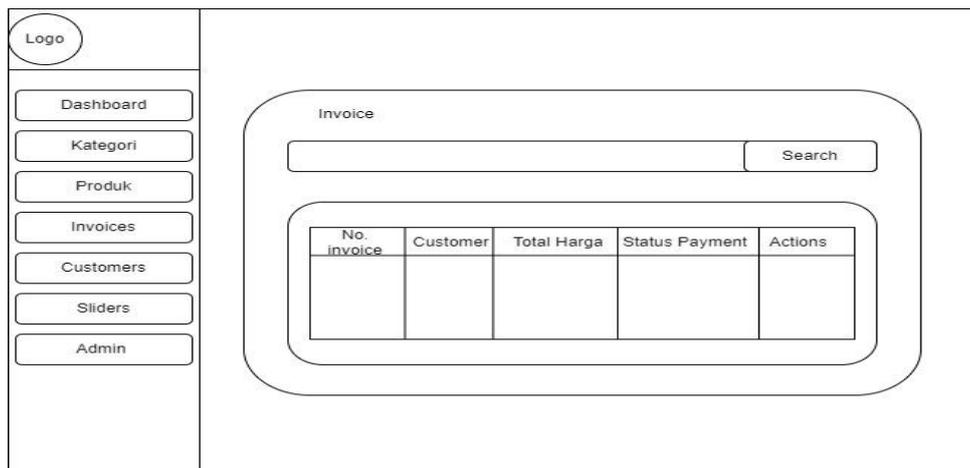


Gambar 3.31 Desain Halaman Produk *Admin*

Keterangan gambar :

Pada gambar 3.31 menjelaskan desain halaman produk *admin* pada *website* memuat tabel produk *admin*, di mana *admin* bisa menambah produk.

12. Desain Halaman *Invoices (Admin)*

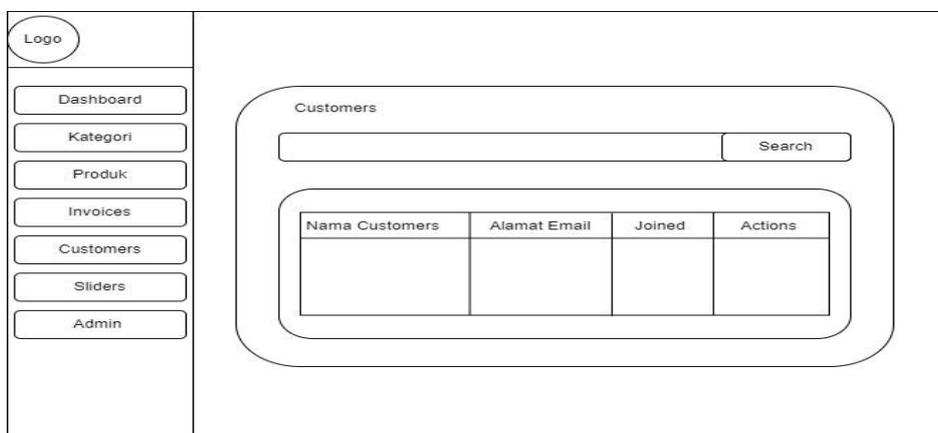


Gambar 3.32 Desain Halaman *Invoices Admin*

Keterangan gambar :

Pada gambar 3.32 menjelaskan desain halaman kategori *admin* pada *website* memuat tabel *invoices*, di mana *admin* bisa melihat data *invoice customer*.

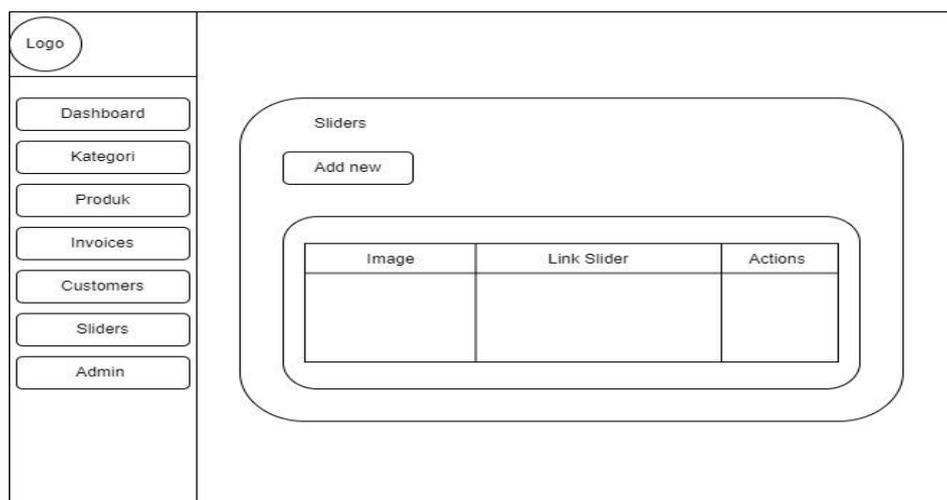
13. Desain Halaman Data *Customers (Admin)*



Gambar 3.33 Desain Halaman Data *Customers Admin*

Keterangan gambar :

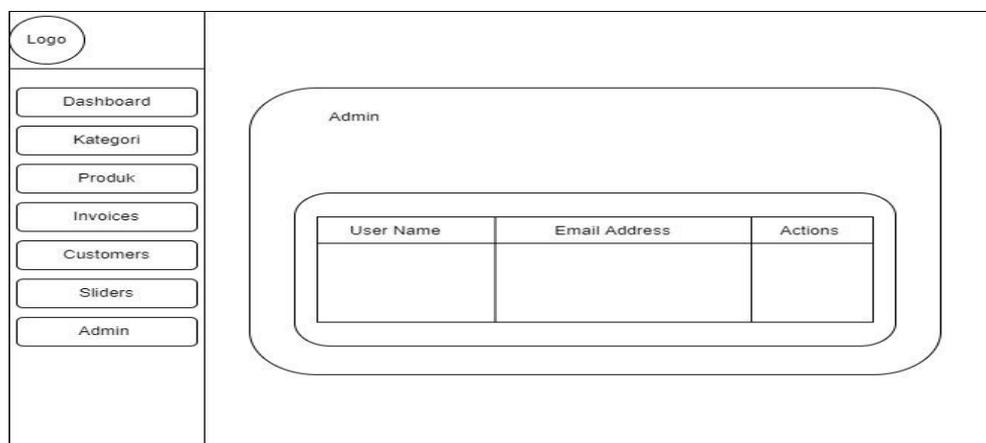
Pada gambar 3.33 menjelaskan desain halaman data *customers* pada *website* memuat nama *customers*, alamat *email*, dan tanggal *joined*.

14. Desain Halaman Data *Sliders* (Admin)Gambar 3.34 Desain Halaman *Sliders Admin*

Keterangan gambar :

Pada gambar 3.34 menjelaskan desain halaman data *sliders admin* pada *website* memuat gambar dan *link slider*.

15. Desain Halaman Data Admin



Gambar 3.35 Desain Halaman Data Admin

Keterangan gambar :

Pada gambar 3.35 menjelaskan desain halaman data *admin* pada *website* memuat *user name* dan *email address*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Setelah dianalisis pada bab sebelumnya, maka selanjutnya adalah tahap implementasi sistem, di mana akan menghasilkan sebuah sistem yang bisa digunakan oleh pengguna, sehingga pada bab ini akan dijelaskan hasil dari keseluruhan sistem yang dibangun. Berikut hasil dari tampilan pada sistem beserta penjelasannya agar mudah dipahami.

4.2 Hasil Pada Tampilan *Admin* dan *Customer*

Tampilan ini terdiri dari beberapa halaman-halaman dan menu. Pada sistem ini terdapat dua aktor utama yaitu *admin* dan *customer*. Adapun fitur pada *admin* seperti, halaman *login admin*, *dashboard admin*, halaman data kategori, halaman data produk, halaman data *invoices*, halaman data *customers*, halaman data *sliders*, dan halaman data *admin*. Fitur pada *customer* atau pengunjung seperti halaman *login*, *register*, *dashboard*, kategori, produk, detail produk, *cart*, *invoices*, *payment gateway*, dan *reviews*.

4.2.1 Halaman *Register Customer PWA*

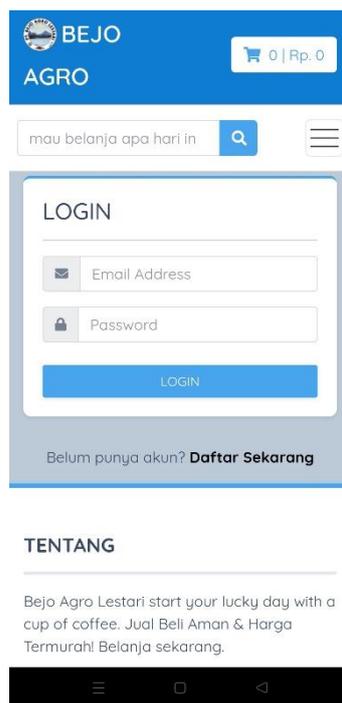
The image shows a mobile application interface for a PWA. At the top, there is a blue header with the logo 'BEJO AGRO' and a shopping cart icon indicating '0 | Rp. 0'. Below the header is a search bar with the placeholder text 'mau belanja apa hari ini'. The main content area is titled 'REGISTER' and contains four input fields: 'Nama Lengkap', 'Email Address', 'Password', and 'Password Confirmation'. There are two buttons: 'REGISTER' and 'RESET'. At the bottom of the form, there is a link: 'Sudah punya akun? Login Disini'. The page is displayed on a mobile device, as indicated by the 'TENTANG' label and the Android navigation bar at the bottom.

Gambar 4.1 Halaman *Register Customer PWA*

Keterangan gambar :

Pada gambar 4.1 menjelaskan, bahwa halaman ini berfungsi untuk *customer* melakukan registrasi. Ketika *customer* ingin melakukan transaksi atau membeli produk, maka *customer* diharuskan melakukan registrasi terlebih dahulu dengan melakukan *input* data diri seperti nama lengkap, *email*, *password* dan *password confirmation* secara benar dan lengkap, yang nantinya *customer* dapat melakukan *login* kedalam sistem.

4.2.2 Halaman *Login Customer PWA*

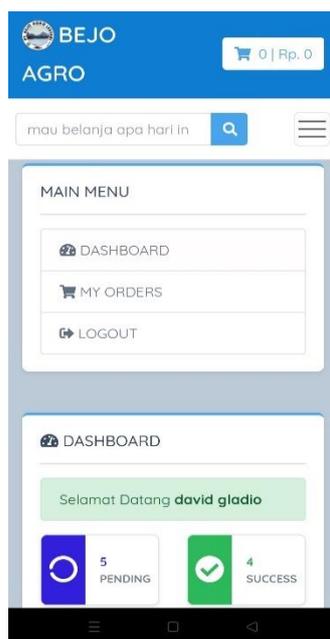


Gambar 4.2 Halaman *Login Customer PWA*

Keterangan gambar :

Pada gambar 4.2 menjelaskan, bahwa halaman ini digunakan *customer* untuk melakukan *login*. *Customer* harus melakukan *input* data *email* dan *password* dengan benar sesuai data yang telah di registrasi sebelumnya, apabila gagal maka sistem akan memberikan *message* atau keterangan *Email or Password is incorrect*.

4.2.3 Halaman *Dashboard Customer PWA*



Gambar 4.3 Halaman *Dashboard Customer PWA*

Keterangan gambar :

Pada gambar 4.3 menjelaskan, bahwa halaman ini terdapat fitur *dashboard*, dan *my orders* untuk melihat detail transaksi atau melakukan proses transaksi yang belum dilanjutkan, dihalaman ini juga *customer* bisa melakukan proses *logout*.

4.2.4 Halaman *Utama Customer PWA*

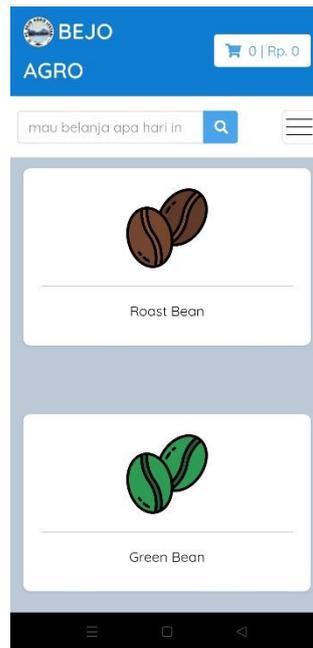


Gambar 4.4 Halaman *Utama Customer PWA*

Keterangan gambar :

Pada gambar 4.4 menjelaskan, bahwa halaman ini merupakan halaman utama *customer* atau pengunjung, di mana *customer* atau pengunjung dapat melihat tampilan utama dari *website*. Pada halaman ini terdapat menu kategori, semua produk, tentang, kontak, dan *dashboard*.

4.2.5 Halaman Kategori Produk *Customer PWA*



Gambar 4.5 Halaman Kategori Produk *Customer PWA*

Keterangan gambar :

Pada gambar 4.5 menjelaskan, bahwa halaman ini merupakan halaman utama *customer* atau pengunjung, di mana *customer* dapat melihat tampilan utama *PWA*. Pada halaman ini terdapat menu kategori, semua produk, tentang, kontak, dan *dashboard*.

4.2.6 Halaman *Cart Customer PWA*

Gambar 4.6 Halaman *Cart Customer PWA*

Keterangan gambar :

Pada gambar 4.6 menjelaskan, bahwa halaman ini bertujuan untuk menampilkan detail pesanan dan mengisi *form* detail *customer* seperti mengisi nama lengkap, mengisi nomor telepon, memilih provinsi, memilih kota, memilih jasa kurir menggunakan raja ongkir, dan *input* alamat lengkap.

4.2.7 Halaman *Detail Order Customer PWA*

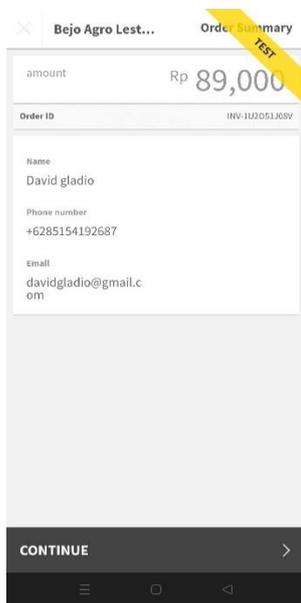
| DETAIL ORDER | |
|--------------------------|------------------------------|
| NO. INVOICE | : INV-1U2O51J08V |
| Nama Lengkap | : David gladia |
| NO HANDPHONE | : 085154192687 |
| COURIER / SERVICE / COST | : jne / OKE / Rp. 42.000 |
| KOTA | : Deli Serdang - (Kabupaten) |
| PROVINSI | : Sumatera Utara |

Gambar 4.7 Halaman *Detail Order Customer PWA*

Keterangan gambar :

Pada gambar 4.7 menjelaskan, bahwa halaman ini *customer* dapat melakukan pengecekan detail *order* sebelum melakukan proses transaksi pembayaran.

4.2.8 Halaman *Snap Payment Gateway* Midtrans

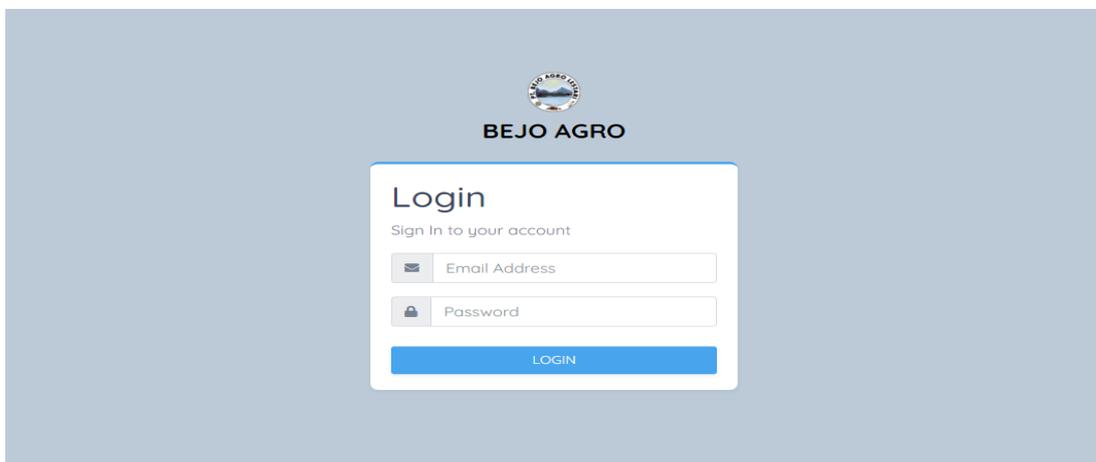


Gambar 4.8 Halaman *Snap Payment Gateway* Midtrans

Keterangan gambar :

Pada gambar 4.8 menjelaskan, bahwa tampilan ini merupakan langkah selanjutnya dari halaman *invoices*, disini *customer* akan melakukan transaksi pembayaran secara otomatis menggunakan *payment gateway* midtrans dengan berbagai metode pembayaran *online*.

4.2.9 Halaman *Login Admin*

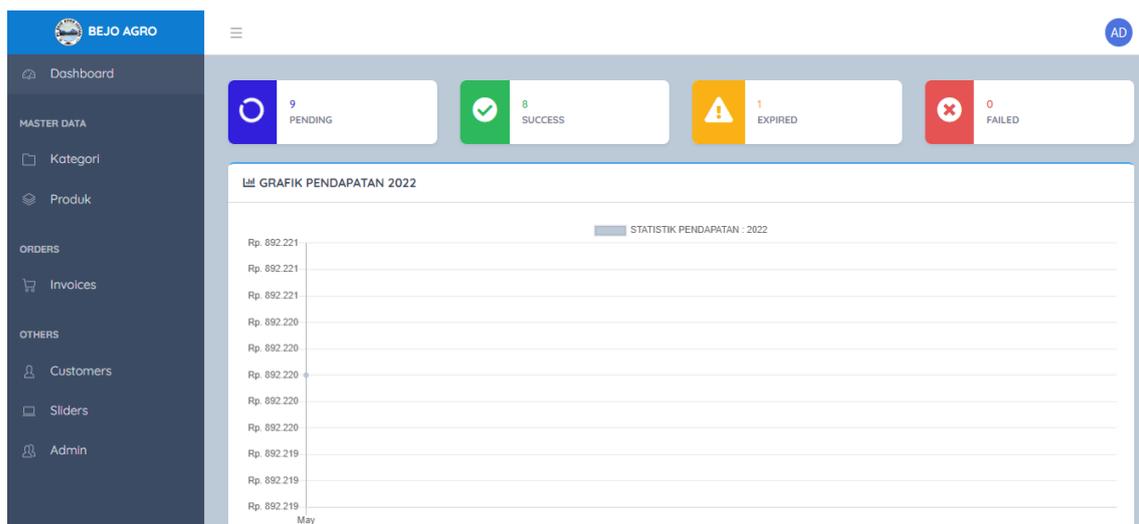


Gambar 4.9 Halaman *Login Admin*

Keterangan gambar :

Pada gambar 4.9 menjelaskan, bahwa halaman ini berfungsi untuk *admin* melakukan *login*. Sebelum masuk kedalam sistem *admin* wajib mengisi data *email* dan *password*. Apabila berhasil *login* pada sistem maka dapat mengelola data, dan apabila *login* gagal maka sistem akan memberikan keterangan atau *message*.

4.2.10 Halaman Menu *Dashboard Admin*



Gambar 4.10 Halaman Menu *Dashboard Admin*

Keterangan gambar :

Pada gambar 4.10 menjelaskan, bahwa halaman ini merupakan tampilan *dashboard* ketika *admin* berhasil melakukan *login*.

4.2.11 Halaman Data Kategori Produk *Admin*

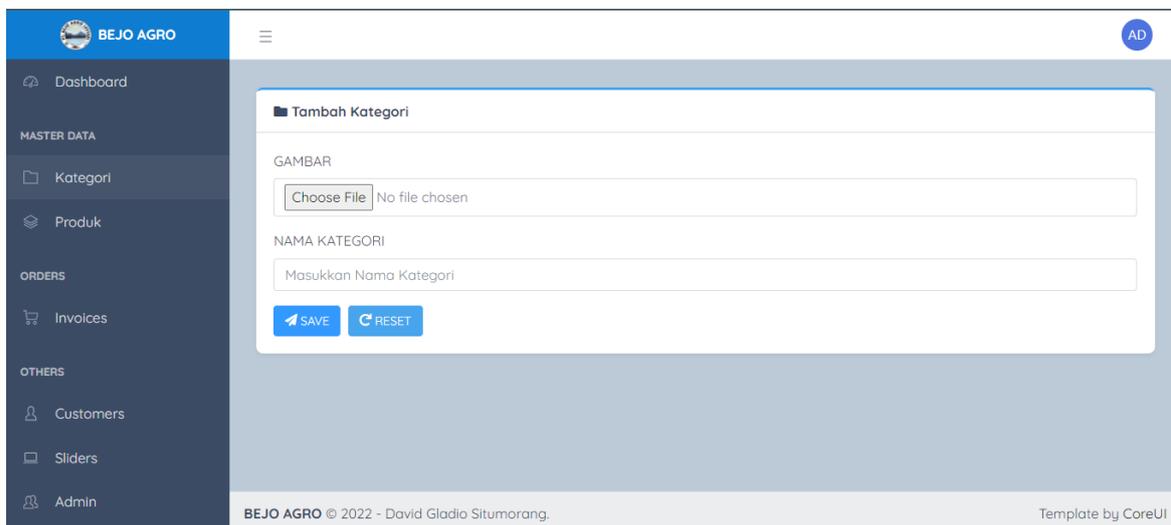
| Gambar | Nama Kategori | Actions |
|---|---------------|---|
|  | Roast Bean | EDIT DELETE |
|  | Green Bean | EDIT DELETE |

Gambar 4.11 Halaman Data Kategori Produk (*Admin*)

Keterangan gambar :

Pada gambar 4.11 menjelaskan, bahwa halaman ini memuat data kategori produk, di mana *admin* dapat melihat dan mengelola informasi data seperti *create*, *update*, dan *delete*.

4.2.12 Halaman Tambah Data Kategori Produk *Admin*



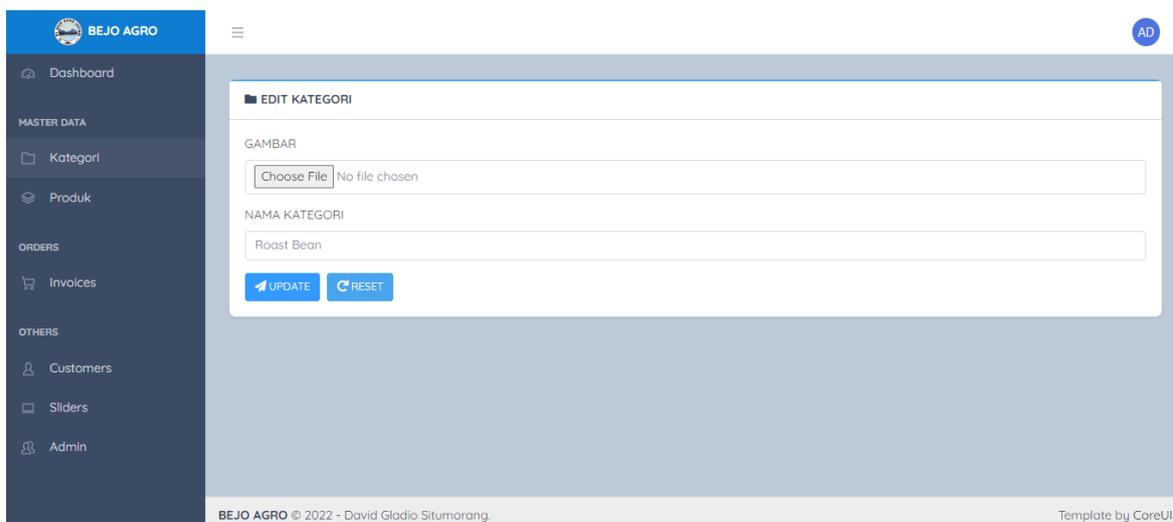
The screenshot shows the 'Tambah Kategori' form in the BEJO AGRO admin dashboard. The form is titled 'Tambah Kategori' and contains two input fields: 'GAMBAR' (Image) with a 'Choose File' button and 'No file chosen' text, and 'NAMA KATEGORI' (Category Name) with the placeholder text 'Masukkan Nama Kategori'. Below the input fields are two buttons: 'SAVE' and 'RESET'. The dashboard includes a sidebar with navigation options like Dashboard, MASTER DATA (Kategori, Produk), ORDERS (Invoices), and OTHERS (Customers, Sliders, Admin). The footer contains the text 'BEJO AGRO © 2022 - David Gladio Situmorang.' and 'Template by CoreUI'.

Gambar 4.12 Halaman Tambah Data Kategori Produk (*Admin*)

Keterangan gambar :

Pada gambar 4.12 menjelaskan, bahwa admin dapat menambah atau *input* data yang baru.

4.2.13 Halaman Edit Data Kategori Produk *Admin*



The screenshot shows the 'EDIT KATEGORI' form in the BEJO AGRO admin dashboard. The form is titled 'EDIT KATEGORI' and contains two input fields: 'GAMBAR' (Image) with a 'Choose File' button and 'No file chosen' text, and 'NAMA KATEGORI' (Category Name) with the text 'Roast Bean'. Below the input fields are two buttons: 'UPDATE' and 'RESET'. The dashboard includes a sidebar with navigation options like Dashboard, MASTER DATA (Kategori, Produk), ORDERS (Invoices), and OTHERS (Customers, Sliders, Admin). The footer contains the text 'BEJO AGRO © 2022 - David Gladio Situmorang.' and 'Template by CoreUI'.

Gambar 4.13 Halaman Edit Data Kategori Produk (*Admin*)

Keterangan gambar :

Pada gambar 4.13 menjelaskan, *admin* dapat melakukan edit atau *update* data kategori seperti gambar dan nama kategori.

4.2.14 Halaman Data Produk *Admin*

| Nama Produk | Nama Kategori | Stok | Actions |
|-----------------------|---------------|------|---|
| Kopi mandheling | Roast Bean | 1000 | EDIT DELETE |
| Kopi dance mandheling | Roast Bean | 998 | EDIT DELETE |
| Kopi dance stdikalang | Roast Bean | 998 | EDIT DELETE |
| kopi dance gayo | Roast Bean | 998 | EDIT DELETE |
| Green Bean Grade 5 | Green Bean | 1000 | EDIT DELETE |
| Kopi Gayo | Roast Bean | 999 | EDIT DELETE |
| Kopi Lintong | Roast Bean | 999 | EDIT DELETE |
| Green Bean Grade 3 | Green Bean | 999 | EDIT DELETE |

Gambar 4.14 Halaman Data Produk (*Admin*)

Keterangan gambar :

Pada gambar 4.14 menjelaskan, halaman ini berisi data-data produk, di mana *admin* dapat melihat dan mengelola informasi data seperti *create*, *update*, dan *delete*.

4.2.15 Halaman Tambah Data Produk *Admin*

TAMBAH PRODUK

GAMBAR
 No file chosen

NAMA PRODUK:

KATEGORI:

BERAT (Gram):

STOK:

DESKRIPSI

HARGA:

DISKON (%):

Gambar 4.15 Halaman Tambah Data Produk (*Admin*)

Keterangan gambar :

Pada gambar 4.15 menjelaskan, bahwa *admin* dapat menambah atau *input* data produk yang baru.

4.2.16 Halaman Edit Data Produk *Admin*

Gambar 4.16 Halaman Edit Data Produk (*Admin*)

Keterangan gambar :

Pada gambar 4.16 menjelaskan, bahwa *admin* dapat melakukan edit atau *update* data produk.

4.2.17 Halaman Data *Invoices Admin*

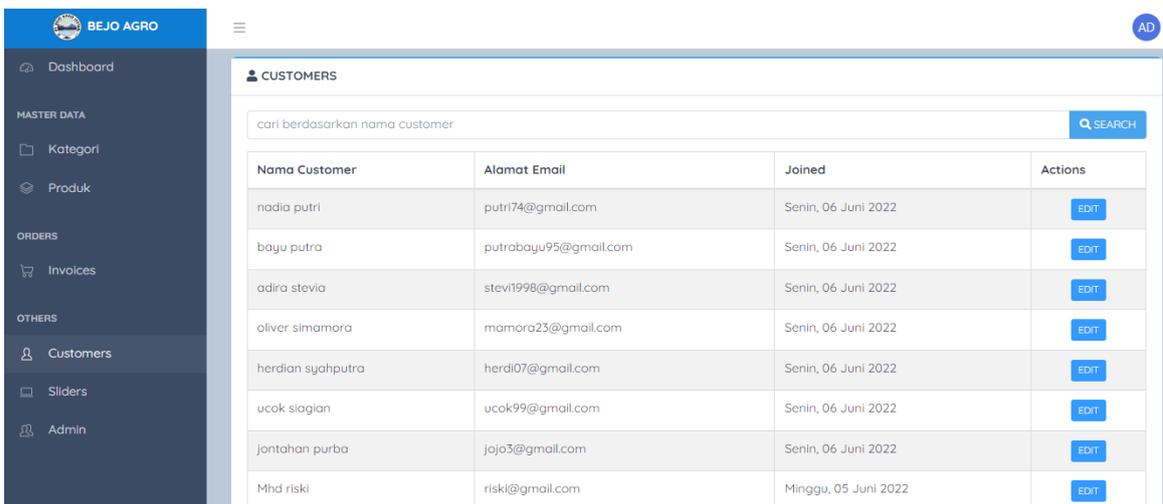
| No. Invoice | Customer | Total Harga | Status Payment | Actions |
|----------------|-----------------|-------------|----------------|---------|
| INV-MDYFUV5JU5 | adira stevia | Rp. 89.000 | pending | DETAIL |
| INV-30F16UFUSG | bayu putra | Rp. 373.460 | success | DETAIL |
| INV-1U2O51J08V | david gladio | Rp. 89.000 | pending | DETAIL |
| INV-53MOZY1YGD | arhinza rivaldi | Rp. 82.000 | expired | DETAIL |
| INV-50T4CN9T2Y | david gladio | Rp. 55.000 | pending | DETAIL |
| INV-T6EZID56EL | david gladio | Rp. 54.000 | pending | DETAIL |
| INV-G0BDZ6S2YQ | david gladio | Rp. 84.000 | success | DETAIL |
| INV-1924B5SDM3 | david gladio | Rp. 84.000 | success | DETAIL |

Gambar 4.17 Halaman Data *Invoices Admin*

Keterangan gambar :

Pada gambar 4.17 menjelaskan, bahwa *admin* hanya melihat data *invoices* atau transaksi dari *customer* seperti data nomor *invoice*, *customer*, total harga dan status *payment* atau status pembayaran.

4.2.18 Halaman Data Customers Admin



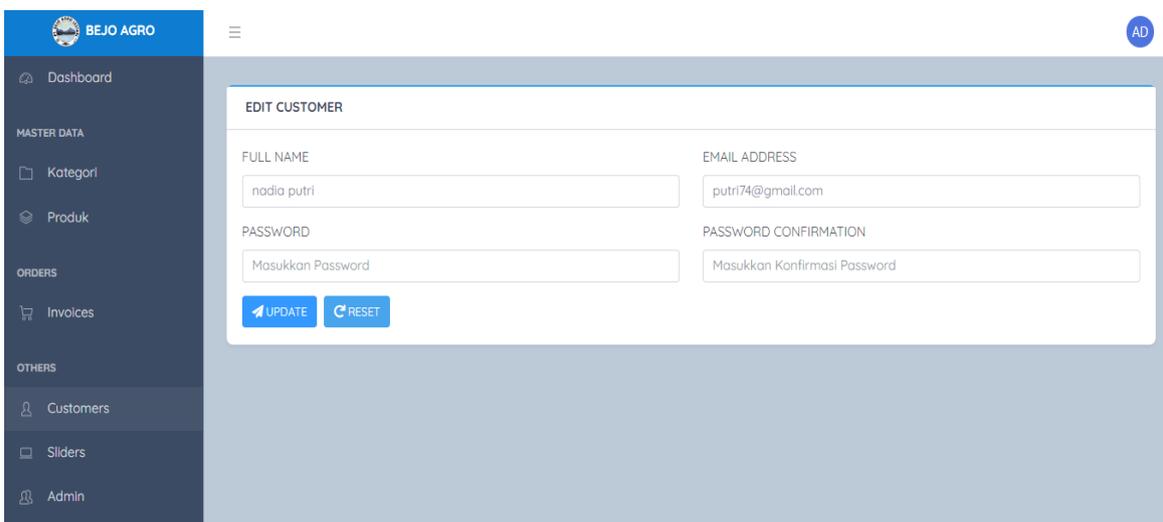
| Nama Customer | Alamat Email | Joined | Actions |
|-------------------|-----------------------|----------------------|----------------------|
| nadia putri | putri74@gmail.com | Senin, 06 Juni 2022 | EDIT |
| bagu putra | putrabagu95@gmail.com | Senin, 06 Juni 2022 | EDIT |
| adira stevia | stev1998@gmail.com | Senin, 06 Juni 2022 | EDIT |
| oliver simamora | mamora23@gmail.com | Senin, 06 Juni 2022 | EDIT |
| herdian syahputra | herdi07@gmail.com | Senin, 06 Juni 2022 | EDIT |
| ucok siagian | ucok99@gmail.com | Senin, 06 Juni 2022 | EDIT |
| jontahan purba | jojo3@gmail.com | Senin, 06 Juni 2022 | EDIT |
| Mhd riski | riski@gmail.com | Minggu, 05 Juni 2022 | EDIT |

Gambar 4.18 Halaman Data Customers (Admin)

Keterangan gambar :

Pada gambar 4.18 menjelaskan, bahwa *admin* dapat melihat dan mengelola informasi data seperti edit atau *update*.

4.2.19 Halaman Edit Data Customers Admin



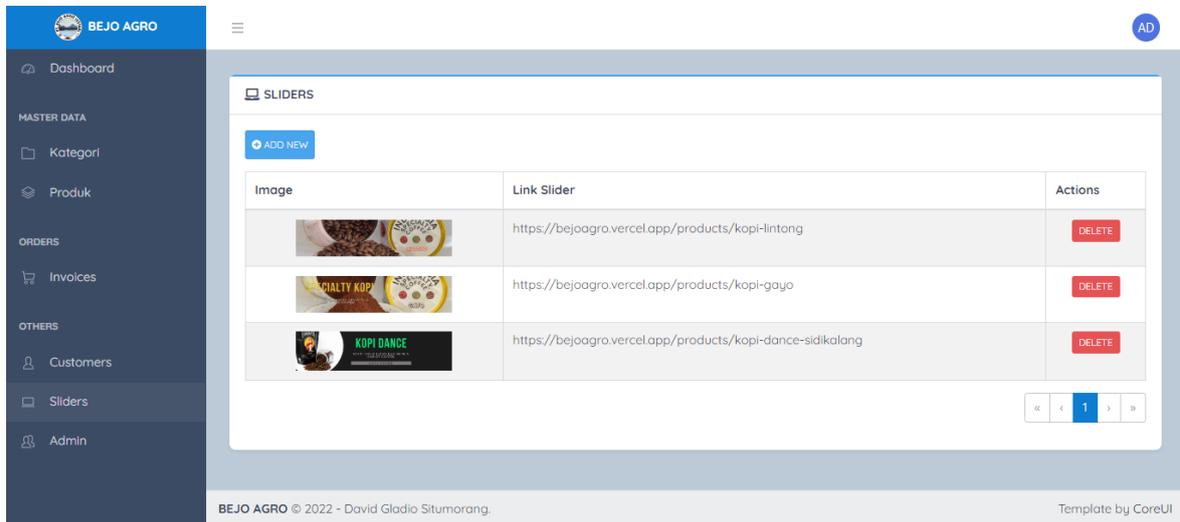
| EDIT CUSTOMER | |
|--|---|
| FULL NAME | EMAIL ADDRESS |
| <input type="text" value="nadia putri"/> | <input type="text" value="putri74@gmail.com"/> |
| PASSWORD | PASSWORD CONFIRMATION |
| <input type="text" value="Masukkan Password"/> | <input type="text" value="Masukkan Konfirmasi Password"/> |
| <input type="button" value="UPDATE"/> <input type="button" value="RESET"/> | |

Gambar 4.19 Halaman Edit Data Customers (Admin)

Keterangan gambar :

Pada gambar 4.19 menjelaskan, bahwa *admin* dapat melakukan edit atau *update* data *customer*.

4.2.20 Halaman Data Sliders Admin

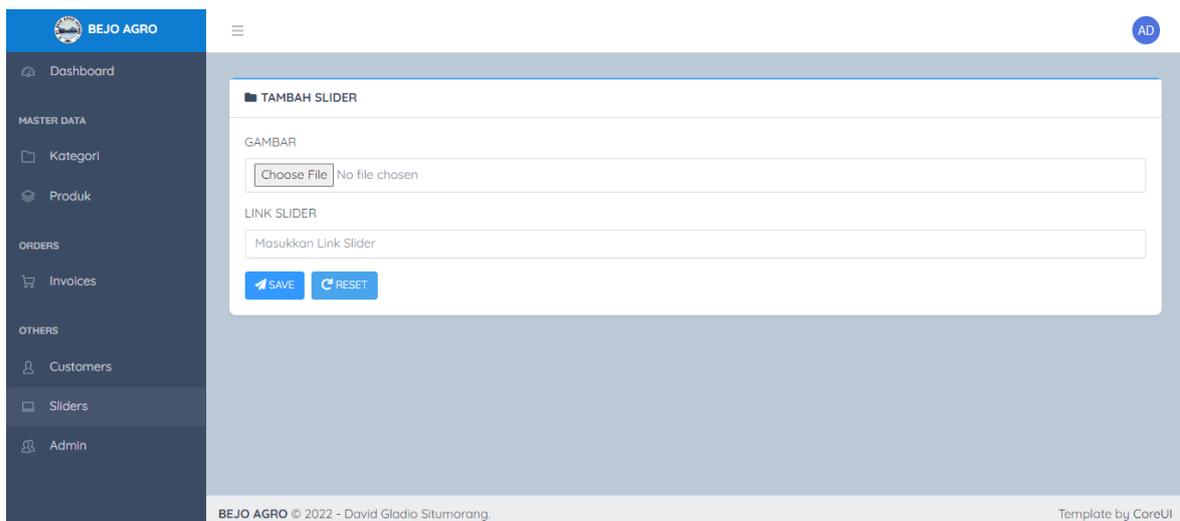


Gambar 4.20 Halaman Data Sliders (Admin)

Keterangan gambar :

Pada gambar 4.20 menjelaskan, bahwa *admin* dapat melihat dan mengelola informasi data *sliders* seperti *create* dan *delete*.

4.2.21 Halaman Tambah Data Sliders Admin

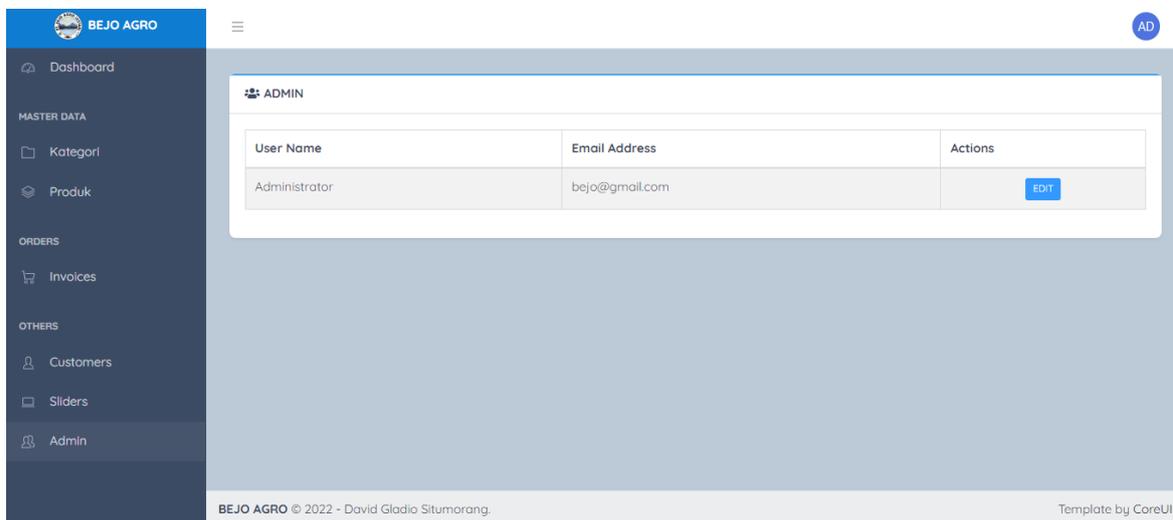


Gambar 4.21 Halaman Tambah Data Sliders (Admin)

Keterangan gambar :

Pada gambar 4.21 menjelaskan, bahwa *admin* dapat menambah atau *input* data yang baru seperti *upload* gambar dan input *link slider*.

4.2.22 Halaman Data Admin

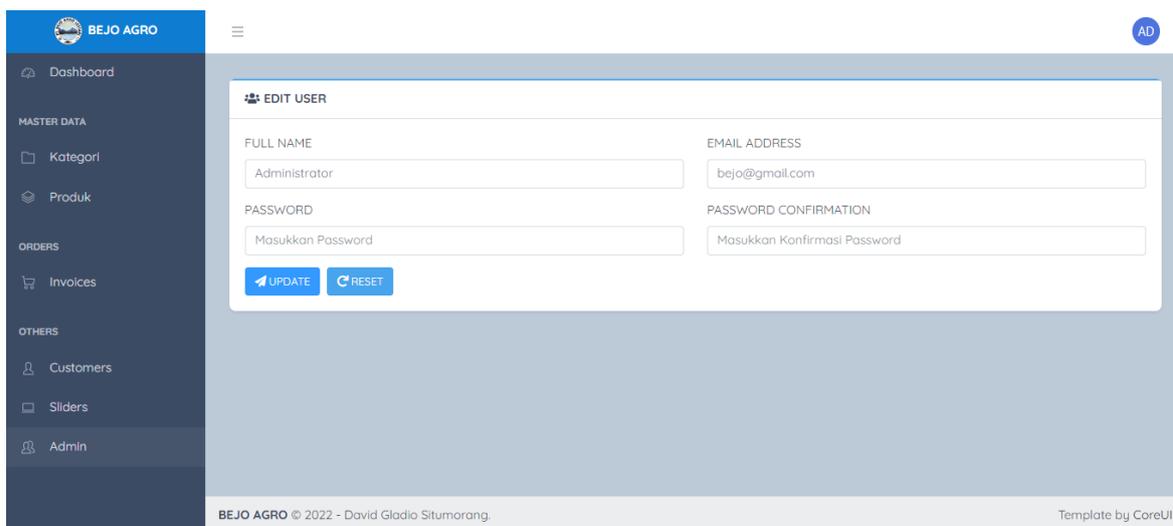


Gambar 4.22 Halaman Data Admin

Keterangan gambar :

Pada gambar 4.22 menjelaskan, bahwa *admin* dapat melihat dan mengelola informasi data seperti *update*.

4.2.23 Halaman Edit Data Admin

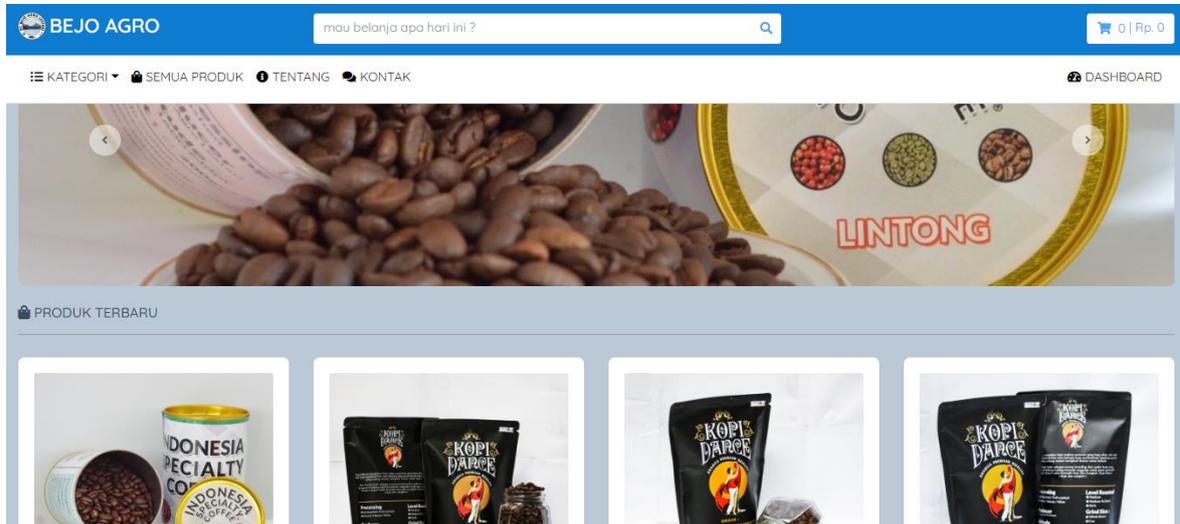


Gambar 4.23 Halaman Edit Data Admin

Keterangan gambar :

Pada gambar 4.23 menjelaskan, bahwa *admin* dapat melakukan edit atau *update* data.

4.2.24 Halaman Utama *Customer* atau Pengunjung

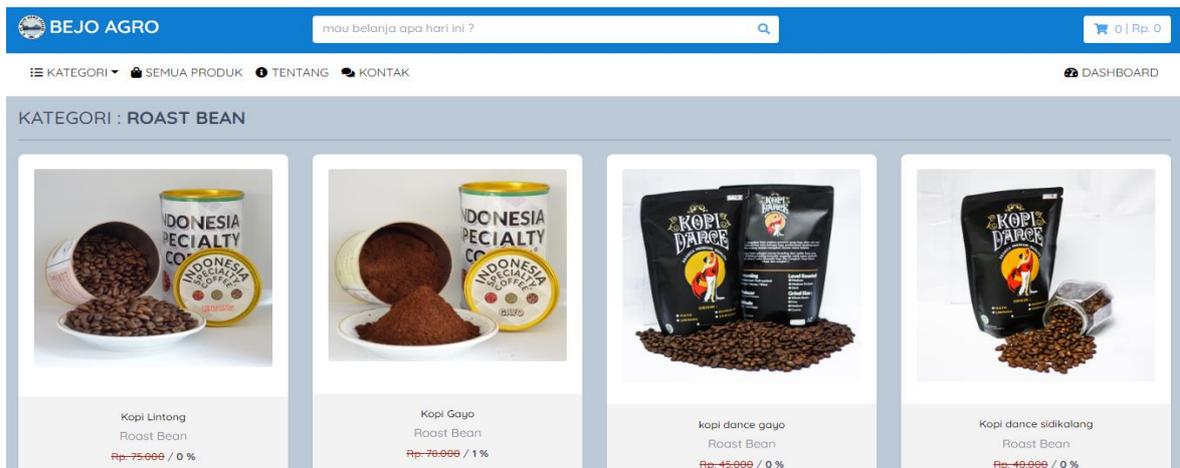


Gambar 4.24 Halaman Utama *Customer* atau Pengunjung

Keterangan gambar :

Pada gambar 4.24 menjelaskan, bahwa pengunjung maupun *customer* yang sudah melakukan *login* dapat melihat tampilan utama dari *website*, pada halaman ini terdapat menu kategori, semua produk, tentang, kontak, dan *dashboard*.

4.2.25 Halaman Berdasarkan Kategori

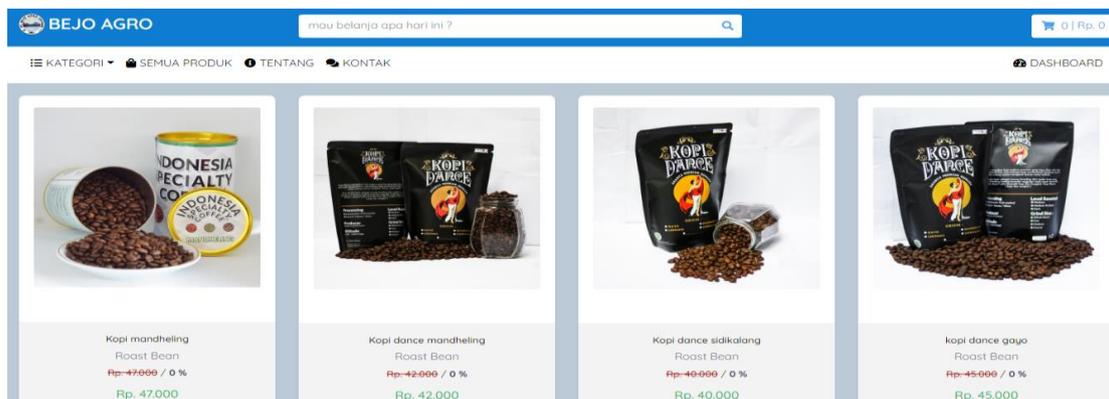


Gambar 4.25 Halaman Berdasarkan Kategori

Keterangan gambar :

Pada gambar 4.25 menjelaskan, bahwa halaman ini berdasarkan kategori, di mana pengunjung maupun *customer* dapat melihat dan memilih produk berdasarkan kategori yang diinginkan.

4.2.26 Halaman Semua Produk

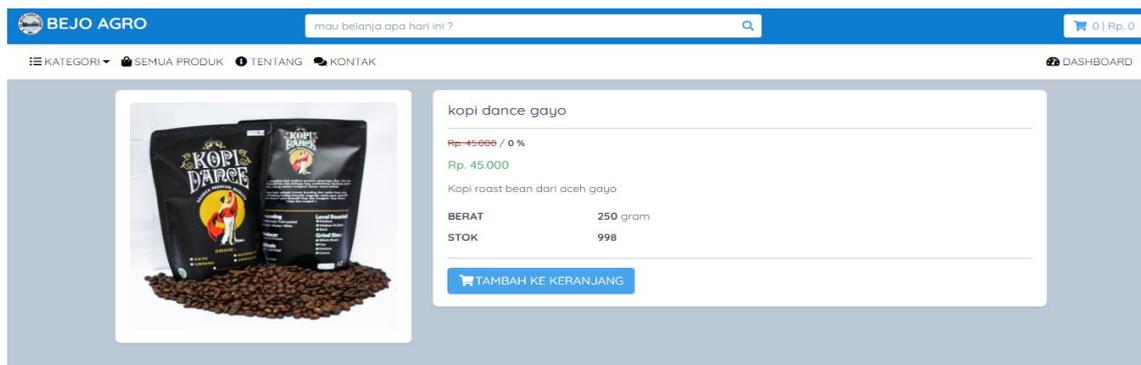


Gambar 4.26 Halaman Semua Produk

Keterangan gambar :

Pada gambar 4.26 menjelaskan, bahwa pada halaman ini *customer* atau pengunjung dapat melihat semua produk yang tersedia pada *website*.

4.2.27 Halaman Detail Produk



Gambar 4.27 Halaman Detail Produk

Keterangan gambar :

Pada gambar 4.27 menjelaskan, bahwa pada halaman ini menampilkan detail produk yang ingin dilihat atau diorder oleh *customer* maupun pengguna, di mana jika *customer* ingin melakukan pembelian dapat melakukan klik tombol tambah ke keranjang.

4.2.28 Halaman *Cart Customer*

Gambar 4.28 Halaman *Cart Customer*

Keterangan gambar :

Pada gambar 4.28 menjelaskan, bahwa halaman ini bertujuan untuk menampilkan detail pesanan dan mengisi *form detail customer* seperti mengisi nama lengkap, mengisi nomor telepon, memilih provinsi, memilih kota, memilih jasa kurir menggunakan raja ongkir, dan *input* alamat lengkap.

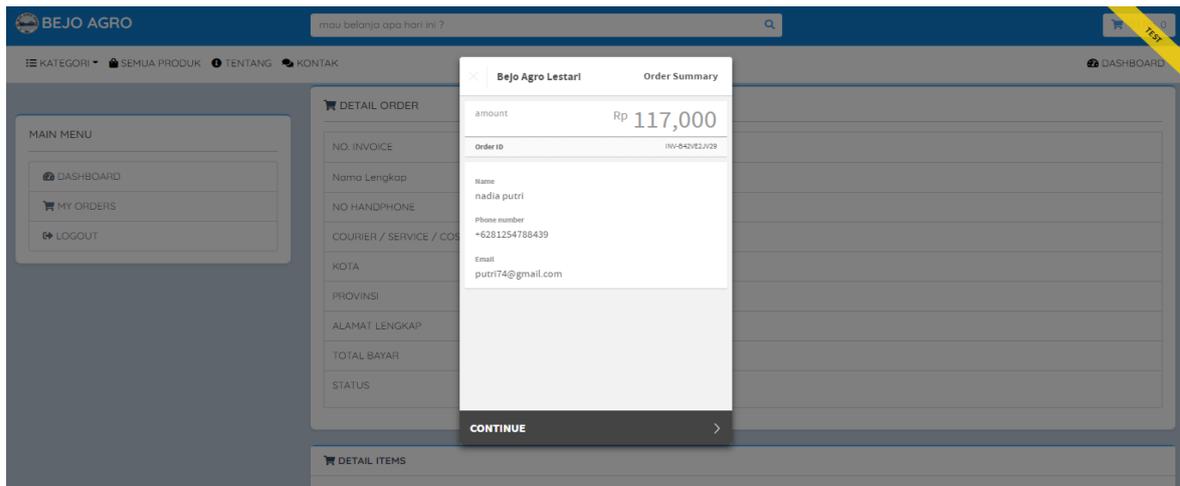
4.2.29 Halaman *Invoice Customer*

Gambar 4.29 Halaman *Invoice Customer*

Keterangan gambar :

Pada gambar 4.29 menjelaskan, bahwa pada halaman ini *customer* dapat melihat detail *order* seperti nomor *invoice*, total pembayaran dan status *payment*.

4.2.30 Tampilan Transaksi *Payment Gateway*

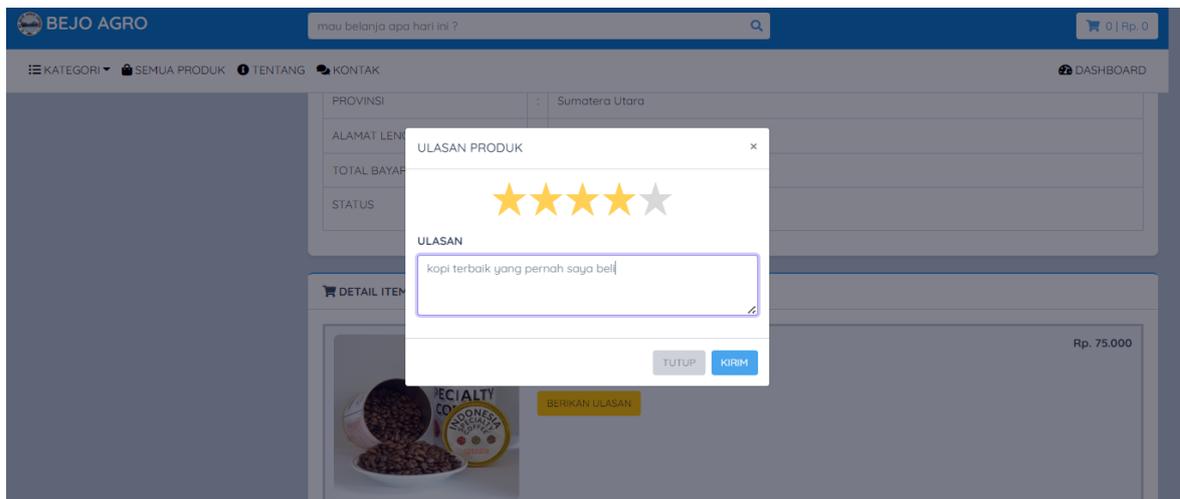


Gambar 4.30 Tampilan Transaksi *Payment Gateway*

Keterangan gambar :

Pada gambar 4.30 menjelaskan, bahwa pada tampilan ini merupakan langkah selanjutnya dari halaman *invoices*, disini *customer* akan melakukan transaksi pembayaran secara otomatis menggunakan *payment gateway* midtrans dengan berbagai metode pembayaran *online*.

4.2.31 Tampilan Ulasan *Customer*



Gambar 4.31 Tampilan Ulasan *Customer*

Keterangan gambar :

Pada gambar 4.31 menjelaskan, bahwa pada tampilan ini dapat digunakan *customer* ketika sudah berhasil melakukan proses pembayaran secara *online*, tampilan ini bersifat

optional. *Customer* secara bebas dapat melakukan ulasan atau tidak dan *customers* hanya bisa melakukan sekali ulasan di tiap transaksi yang telah berhasil.

4.3 Pembahasan

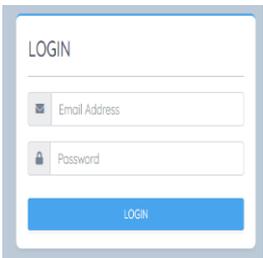
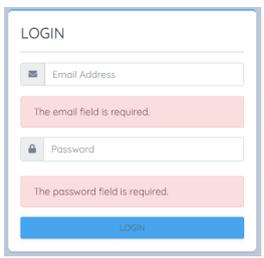
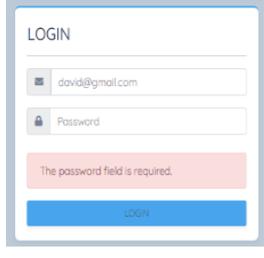
Perancangan pada sistem ini di bangun bertujuan untuk menghasilkan sistem yang dapat membantu penjualan dan pemasaran pada perusahaan. Sistem ini terdapat 2 aktor utama yaitu *admin* dan *customer* serta 2 aktor *third party* yaitu midtrans dan Raja Ongkir. Sistem ini juga dapat mempermudah dalam transaksi penjualan, karena menggunakan *payment gateway* dalam proses transaksi pembayarannya. Dengan adanya sistem ini dapat membantu *customer* dalam mengetahui informasi-informasi produk yang tersedia dan tidak lagi harus datang ke toko *merchant* perusahaan. Bahasa pemrograman yang digunakan oleh penulis adalah *PHP* dan *JavaScript* dengan menggunakan *framework Laravel* sebagai *API web service (backend)* dan *Nuxt.js* sebagai *Frontend*. Sistem ini juga dibangun berdasarkan *API Service Oriented* atau *Web Service* di mana folder pada *frontend* dan *backend* dibangun secara terpisah, *frontend* hanya menggunakan data *API* atau *web service* yang telah dibangun pada sisi *backend*. Sehingga sistem ini dapat menghasilkan *output PWA* atau *Progressive Web App* yang mampu memberikan pelayanan bagi pengguna layaknya menggunakan aplikasi *mobile*.

4.3.1 Testing Sistem

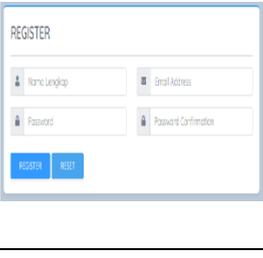
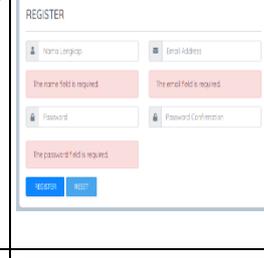
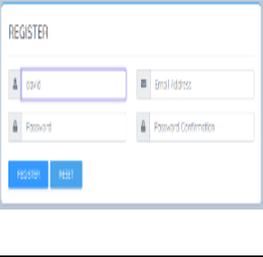
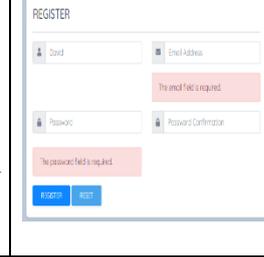
1. Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui apakah sistem yang dibangun sudah berjalan atau berfungsi dengan baik atau tidak, serta dapat mengetahui kesalahan pada sistem. Adapun tahap pengujian pada sistem ini dilakukan menggunakan metode *blackbox* atau pengujian yang dilaksanakan untuk mengetahui hasil program melalui uji fungsional dari suatu perangkat lunak.

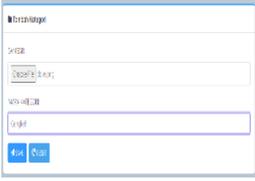
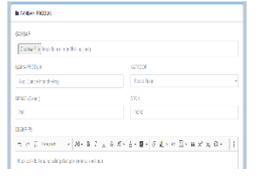
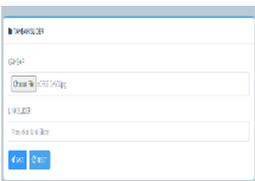
Tabel 4.1 Pengujian *Black Box Login*

| Pengujian | Case | Hasil Yang Diharapkan | Hasil Pengujian | Status |
|--|---|--|--|--------|
| Mengosongkan <i>email</i> dan <i>password</i> pada saat <i>login</i> |  | Sistem akan memberikan keterangan atau pesan, serta gagal <i>login</i> |  | Sukses |
| Hanya mengisi salah satu <i>field</i> saat <i>login</i> |  | Sistem menampilkan pesan “ <i>The password field is required.</i> ” |  | Sukses |

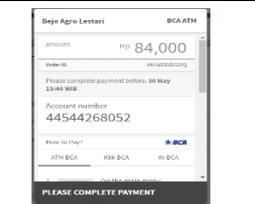
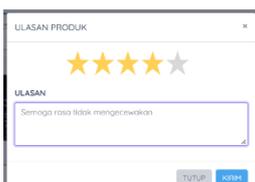
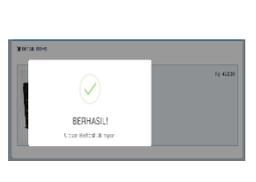
Tabel 4.2 Pengujian *Black Box Register*

| Pengujian | Case | Hasil Yang Diharapkan | Hasil Pengujian | Status |
|---|---|---|--|--------|
| Mengosongkan keseluruhan <i>input data register</i> |  | Sistem akan memberikan keterangan atau pesan, sehingga gagal registrasi |  | Sukses |
| Hanya mengisi salah satu <i>field</i> saat registrasi |  | Sistem menampilkan pesan pada tiap <i>field</i> yang terisi kosong |  | Sukses |

Tabel 4.3 Pengujian *Black Box Menu Admin*

| Pengujian | Case | Hasil Yang Diharapkan | Hasil Pengujian | Status |
|-----------------------------------|--|---|--|--------|
| tambah data kategori dengan benar |  | Sistem memberi notifikasi berhasil dan menyimpan data kategori baru |  | Sukses |
| Tambah data produk dengan benar |  | Sistem memberi notifikasi berhasil dan menyimpan data produk baru |  | Sukses |
| tambah data slider dengan benar |  | Sistem memberi notifikasi berhasil dan menyimpan data produk baru |  | Sukses |

Tabel 4.4 Pengujian *Black Box Menu Customer*

| Pengujian | Case | Hasil Yang Diharapkan | Hasil Pengujian | Status |
|---|---|--|---|--------|
| Mengisi detail <i>customer</i> dengan benar |  | Sistem memberi notifikasi berhasil dan masuk ke halaman <i>invoice</i> |  | Sukses |
| <i>Customer</i> melakukan pembayaran dengan benar |  | Pembayaran berhasil |  | Sukses |
| <i>Customer</i> bisa <i>review</i> setelah transaksi sukses |  | Sistem memberi notifikasi berhasil |  | Sukses |

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil yang diperoleh dari penelitian yang dilakukan oleh penulis, maka dapat diambil beberapa kesimpulan yaitu :

1. Sistem *E-commerce* Komoditas Berbasis *PWA* Pada PT. Bejo Agro Lestari, mempermudah proses transaksi jual beli karena sudah menggunakan *payment gateway* midtrans dan terintegrasi dengan layanan Raja Ongkir.
2. Sistem yang dibangun bisa diakses secara *realtime* karena sudah menggunakan *Rest API*.
3. Sistem *E-commerce* Komoditas Berbasis *PWA* Pada PT. Bejo Agro Lestari hanya menjual produk *Business to Consumer* (B2C).
4. Fitur registrasi pada sistem *E-commerce* Komoditas Berbasis *PWA* masih standar *API* belum terintegrasi dengan verifikasi *email*.

5.2 Saran

Berdasarkan kesimpulan diatas, ada beberapa saran yang dapat penulis sampaikan untuk pengembangan lebih lanjut pada *E-commerce* Komoditas Berbasis *PWA* dimasa yang akan datang yaitu :

1. Sistem *E-commerce* Komoditas Berbasis *PWA* Pada PT. Bejo Agro Lestari belum berbasis *android* sehingga kedepannya dapat dikembangkan.
2. Sistem yang sedang dikembangkan saat ini hanya menjual produk *Business to Consumer* (B2C), maka kedepannya akan dikembangkan dengan menjual produk (B2B) *Business to Business* nya juga.
3. Penerapan *E-commerce* ini sebaiknya selaras dengan sistem *inventory* secara komputerisasi untuk mengelola persediaan atau stok secara detail dan efektif.
4. Penelitian selanjutnya fitur registrasi pada sistem ini dapat dikembangkan menggunakan verifikasi *email*.

DAFTAR PUSTAKA

- Alexander, I., & Nadapdap, H. J. (2019). Analisis Daya Saing Ekspor Biji Kopi Indonesia di Pasar Global Tahun 2002-2017. *JSEP (Journal of Social and Agricultural Economics)*, 12(2), 1-16.
- Aman, M. (2021). Pengembangan Sistem Informasi Wedding Organizer Menggunakan Pendekatan Sistem Berorientasi Objek Pada CV Pesta. *Jurnal Janitra Informatika dan Sistem Informasi*, 1(1), 47-60.
- Ariyanti, L., Satria, M. N. D., & Alita, D. (2020). Sistem Informasi Akademik Dan Administrasi Dengan Metode Extreme Programming Pada Lembaga Kursus Dan Pelatihan. *Jurnal Teknologi Dan Sistem Informasi*, 1(1), 90-96.
- Aswiputri, M. (2022). LITERATURE REVIEW DETERMINASI SISTEM INFORMASI MANAJEMEN: DATABASE, CCTV DAN BRAINWARE. *Jurnal Ekonomi Manajemen Sistem Informasi*, 3(3), 312-322.
- Darojat, T. A. (2019). Aplikasi E-Commerce Business to Consumer Kartu Undangan. *JURIKOM (Jurnal Riset Komputer)*, 6(6), 559-562.
- Dedi, D., Sutarman, S., & Septiyani, N. (2020). Sistem Informasi E-Commerce Berbasis Web Pada Toko Indonesia Okubo Jepang. *Academic Journal of Computer Science Research*, 2(1).
- Dewi, F. S., Poerbo, H., & Susilowati, M. (2020). PERANCANGAN SISTEM INFORMASI PENJUALAN ONLINE DISTRO MD SHOES BERBASIS WEBSITE. *Kurawal-Jurnal Teknologi, Informasi dan Industri*, 3(1), 18-27.
- Dhanush, G. A. (2021). Develop a Scalable and Serverless Client-based Application using Agile Methodology. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(11), 5372-5379.
- Edy, E., Ferdiansyah, F., Pramusinto, W., & Waluyo, S. (2019). Pengamanan Restful API menggunakan JWT untuk Aplikasi Sales Order. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 3(2), 106-112.
- Fatah, K., & Andayani, T. D. (2021). Pengaruh Dimensi-Dimensi Kemudahan Penggunaan, Kegunaan, Dan Persepsi Risiko Terhadap Niatan Untuk Berbelanja Dari Aplikasi E-Commerce B2c Di Masa Pandemi Covid 19 (Studi Kasus Dikota Pekalongan). *Proceeding of The URECOL*, 1-20.

- Febrianto, A., Achmadi, S., & Sasmito, A. P. (2021). Penerapan Metode K-Means Untuk Clustering Pengunjung Perpustakaan Itn Malang. *Jurnal Mahasiswa Teknik Informatika*, 5(1), 61-70.
- Gunawan, I. P., Sukarsa, I. M., & Putra, I. M. S. Rancang Bangun Sistem Informasi Marketplace Agro (Sim a) Pertanian & Peternakan Berbasis Cloud. *Jurnal Ilmiah Teknologi dan Komputer*, 1(2), 302-310.
- Hasanudin, M. (2019). Aplikasi e-commerce sistem informasi penjualan rolling door berbasis rapid application development. *Petir*, 12(1), 64-71.
- Heriyanto, Y. (2018). Perancangan Sistem Informasi Rental Mobil Berbasis Web Pada PT. APM Rent Car. *Jurnal Intra Tech*, 2(2), 64-77.
- Hermanto, B. (2019). Sistem Informasi Manajemen Keuangan pada PT. Hulu Balang Mandiri Menggunakan Framework Laravel. *Jurnal Komputasi*, 7(1), 17-26.
- <https://bejoagrolestari.com>. Diakses pada 22 Juni 2022
- Itmaana, E. (2021). PEMANFAATAN WEBSITE DALAM SISTEM INFORMASI PERSEDIAAN BAHAN BAKU (STUDI KASUS: PT. ADI KARYA GEMILANG WAY KANAN). *Jurnal Alih Teknologi Sistem Informasi (JATSI)*, 1(2).
- Jorgie, R. A., Anggraeny, F. T., & Via, Y. V. (2021). Sistem Diagnosa Penyakit Pada Ayam Berbasis Web Laravel 8. *Jurnal Informatika dan Sistem Informasi (JIFoSI)*, 2(2), 457-466.
- Kurniawan, H., Apriliah, W., Kurniawan, I., & Firmansyah, D. (2020). Penerapan Metode Waterfall Dalam Perancangan Sistem Informasi Penggajian Pada Smk Bina Karya Karawang. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi Dan Komunikasi*, 14(4), 13-23.
- Lesnanda, M., & Raharjo, Y. A. B. (2019, August). Perancangan Website Penjualan Pada Online Shop Luxmoire Dengan Framework Laravel Dan Bootstrap. In *ENTER* (Vol. 2, No. 1, pp. 209-221).
- Magruder, D. S. (2021). Development of interactive software and AI-based algorithms for the analysis of biomedical data (Doctoral dissertation, Staats-und Universitätsbibliothek Hamburg Carl von Ossietzky).
- Marlina, M., & Masnur, M. (2021). Aplikasi E-Learning Siswa Smk Berbasis Web. *Jurnal Sintaks Logika*, 1(1), 8-17.

- Mebiyantara, F., Faisol, A., & Ariwibisono, F. X. (2021). PENGEMBANGAN APLIKASI MANAJEMEN REMOTE LABORATORY MENGGUNAKAN METODE RESTFUL WEB SERVICE BERBASIS MOBILE (Doctoral dissertation, Institut Teknologi Nasional).
- Muchtar, A. Z., & Munir, S. (2019). Perancangan Web E-Commerce Umkm Restoran Bakso Arema Menggunakan Framework Laravel. *Jurnal Teknologi Terpadu (JTT)*, 5(1).
- Muddin, S., Tehuayo, H., & Iksan, F. (2021). Penerapan Teknologi Progressive Web Apps (PWA) Pada Sistem Informasi Sma Negeri 7 Buru Selatan. *Jurnal Teknologi dan Komputer*, 1(1), 16-23.
- Naofal, N., Ulhaq, M. R. D., & Prianto, C. (2022). Development of E-Commerce Information System at Az-Zahra Shop Using Laravel Framework. *JOMLAI: Journal of Machine Learning and Artificial Intelligence*, 1(1), 95-106.
- Novendri, M. S., Saputra, A., & Firman, C. E. (2019). Aplikasi Inventaris Barang Pada Mts Nurul Islam Dumai Menggunakan Php Dan Mysql. *lentera dumai*, 10(2).
- Nugraha, E. S., Padri, A. R., Nurdiawan, O., Faqih, A., & Anwar, S. (2021). Implementasi Aplikasi Pengaduan Masyarakat Berbasis Android Pada Gedung DPRD. *JURIKOM (Jurnal Riset Komputer)*, 8(6), 360-366.
- Parnadi, F., & Loisa, R. (2018). Analisis Daya Saing Ekspor Kopi Indonesia di Pasar Internasional. *Jurnal Manajemen Bisnis dan Kewirausahaan*, 2(4).
- Rahardja, U., Febriyanto, E., Handayani, I., & Ningsih, H. W. (2019, December). Penerapan Teknologi Open Journal System Sebagai Media Publikasi Jurnal Ilmiah Elektronik Bagi Perguruan Tinggi Non It Di Tangerang. In *SNPMas: Seminar Nasional Pengabdian pada Masyarakat* (pp. 75-84).
- Ramdiansyah, F. D., & Anubhakti, D. (2020). SISTEM INFORMASI PENJUALAN BERBASIS E-COMMERCE UNTUK MENINGKATKAN PENJUALAN PADA BINTANG FASHION. *IDEALIS: InDonEsiA journal Information System*, 3(1), 457-461.
- Rivaldi, M. R. (2022). ANALISIS MANAJEMEN RANTAI PASOK KOMODITAS KOPI DI DESA LIBERIA KABUPATEN BOLAANG MONGONDOW TIMUR. *Jurnal EMBA: Jurnal Riset Ekonomi, Manajemen, Bisnis dan Akuntansi*, 10(2).
- Romadhon, M. H., Yudhistira, Y., & Mukrodin, M. (2021). Sistem Informasi Rental Mobil Berbasis Android Dan Website Menggunakan Framework Codeigniter 3 Studi Kasus: CV Kopja Mandiri. *Jurnal Sistem Informasi dan Teknologi Peradaban*, 2(1), 30-36.

- Rudi, R., & Prehanto, D. R. (2020). Pengembangan Aplikasi Sistem Pengelolaan Data Prestasi Mahasiswa Berdasarkan Standar ISO/IEC 25010. *Jurnal Manajemen Informatika*, 11(1).
- Saputra, M. H. K. (2021). PENERAPAN ARSITEKTUR MICROSERVICE PADA SISTEM TATA KELOLA MATAKULIAH PROYEK POLITEKNIK POS INDONESIA. *Jurnal Teknik Informatika*, 13(3), 22-28.
- Sari, E. P., Pudjiarti, E., & Susanti, H. (2020). Sistem Informasi Penjualan Pakaian Wanita Berbasis Web (E-Commerce) Pada Pt. Bunitop Indonesia. *Jurnal Teknologi Informasi Mura Juni*, 12(01), 1-13.
- Simatupang, J., & Sianturi, S. (2019). Perancangan Sistem Informasi Pemesanan Tiket Bus Pada Po. Handoyo Berbasis Online. *Jurnal Intra Tech*, 3(2), 11-25.
- Shodiq, A. R. A. R., Priatna, W. B., & Kusnadi, N. (2018). Perancangan Sistem Informasi Agribisnis Kedelai Berbasis Web di Kabupaten Grobogan, Jawa Tengah (Web-based Information System Design of Soybean Agribusiness in Grobogan, Central Jawa). *JURNAL IPTEKKOM (Jurnal Ilmu Pengetahuan & Teknologi Informasi)*, 20(2), 121-136.
- Sukrianto, D., & Agustina, S. (2018). PEMANFAATAN SMS GATEWAY PADA SISTEM INFORMASI ABSENSI SISWA DI SMAN 12 PEKANBARU BERBASIS WEB. *Jurnal Intra Tech*, 2(2), 78-90.
- Susrama, I. G., Sampurno, I. A. W., & Sugiarto, S. (2020). SISTEM TERINTEGRASI UJI KOMPETENSI LEMBAGA SERTIFIKASI PROFESI. *Jurnal Penelitian*, 5(3), 181-191.
- Sutrisno, S., Asyidiq, M., & Santoso, S. (2018). Perancangan Sistem Pemasangan Iklan Online Pada Aplikasi E-Commerce (E-Gemana) Menggunakan Metode Restful Api Dan Framework Laravel. *Jurnal Ilmiah Sains Dan Teknologi*, 2(2), 119-132.
- Suppa, R., & Abduh, H. (2022). APLIKASI PENGENALAN HURUF, ANGKA, BUAH DAN HEWAN BERBASIS ANDROID (STUDI KASUS TK ANANDA TANETE). *Indonesian Journal Of Education and Humanity*, 2(1), 1-12.
- Suppa, R., & Sulaeman, B. (2022). APLIKASI MEDIA PEMBELAJARAN SISTEM FOTOSINTESIS PADA TUMBUHAN BERBASIS ANDROID KELAS VIII SMP NEGRI 4 WOTU. *Indonesian Journal Of Education and Humanity*, 2(1), 20-33.
- Wahyuni, D., & Rezky, S. F. (2019). Perancangan Sistem Informasi Agribisnis Berbasis E-commerce (Studi Kasus Kelompok Tani Sri Rezki). *JITA (Journal of Information Technology and Accounting)*, 2(1), 29-41.

- Wahyuni, F. D., & Sari, S. N. (2021). ANALISIS PENERAPAN SISTEM E-COMMERCE PADA UMKM (Studi Kasus: Warung Nasi Goreng Mas No di Jalan Apron Jakarta Pusat). *JAMBIS: Jurnal Administrasi Bisnis*, 1(3), 155-168.
- Wardhany, V. A., & Hidayat, A. (2021). IoT system terpadu untuk pengelolaan sarang lebah. *JURNAL ELTEK*, 19(1), 9-17.
- Wijaya, F., Jacobus, A., & Sambul, A. (2021). Implementation Of Web Services On University Library Information Systems. *Jurnal Teknik Informatika*, 16(4), 421-428.
- Wijayakusuma, N. H. P., Saintika, Y., & Susanto, I. (2021). Perancangan Website E-commerce Produk Kopi Menggunakan Metode Prototyping (Studi Kasus: Kedai Kopi Kontekstual). *Journal of Information Systems and Informatics*, 3(3), 471-482.
- Yanuardi, Y., & Permana, A. A. (2019). Rancang Bangun Sistem Informasi Keuangan Pada Pt. Secret Discoveries Travel and Leisure Berbasis Web. *JIKA (Jurnal Informatika)*, 2(2).
- Yulianto, E., & Mauluddin, A. (2019). Penerapan E-Commerce Berbasis Business to Customer (B2C) Pada Industri Garment. *Informatic, Science, Entrepreneur, Applied Art, Research, Humanism*, 18(1), 124-13

