

LAMPIRAN

Lampiran 1. Surat Kesepakatan Bimbingan Skripsi

SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Fadillah Surya Irawan
NIM : 21090030
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Hepatika Zidny Ilmadina, S.Pd., M.Kom.
Status : Dosen
NIDN : 0618119101
Jabatan Fungsional :
Pangkat/Golongan :

Pada hari ini Jum'at tanggal 23 Mei 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I/II Tugas Akhir Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Tugas Akhir minimal 8 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Tugas Akhir

Tegal, 23 Mei 2025

Pihak Pertama


Fadillah Surya Irawan

Pihak Kedua


Hepatika Zidny Ilmadina, S.Pd., M.Kom.
NIPY: 08.017.340

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik
Informatika

Dyah Apriliani, S.Pd., M.Kom.
NIPY: 09.015.225

SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Fadillah Surya Irawan
NIM : 21090030
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

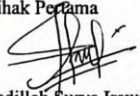
Nama : Arif Hidayah, S.Tr.Kom., M.Tr.Kom.
Status : Dosen
NIDN :
Jabatan Fungsional :
Pangkat/Golongan :

Pada hari ini Jum'at tanggal 23 Mei 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I/II Tugas Akhir Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Tugas Akhir minimal 1 bulan 3 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Tugas Akhir

Tegal, 23 Mei 2025

Pihak Pertama


Fadillah Surya Irawan

Pihak Kedua


Arif Hidayah, S.Tr.Kom., M.Tr.Kom.
NIPY: 10.024.548

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik
Informatika


Dyah Apriliani, S.T., M.Kom.
NIPY: 09.111.9.225

Lampiran 2. Surat Pernyataan Pengajuan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. N a m a : Fadillah Surya Irawan
Kewarganegaraan : Indonesia
Alamat : Ds. Sindang RT 02/05, Kecamatan Dukuwaru,
Kabupaten Tegal Provinsi Jawa Tengah
2. Nama : Hepatika Zidny Ilmadina, S.Pd., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jl. Kenanga Gang 1 Nomor 9, Kelurahan Mangkusuman,
Kecamatan Tegal Timur, Kota Tegal, Provinsi Jawa Tengah,
52123.
3. Nama : Arif Hidayah, S.Tr.Kom., M.Tr.Kom.
Kewarganegaraan : Indonesia
Alamat : Jl. Bulu Emas No.08, RT.001, RW.001, Kelurahan Lawatan,
Kecamatan Dukuhturi, Kabupaten Tegal, Jawa Tengah.

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : Sistem Manajemen Untuk Efisiensi Ketersediaan Stok Barang Pada Toko
Aby Snack Menggunakan *Laravel*
 - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
 - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
 - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
 - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
 - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
 - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.
4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
 - a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
 - b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.

- c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, Juni 2023



(Fadillah Surya Irawan)
Pemegang Hak Cipta *

(Hepatika Zidny Ilmadina, S.Pd., M.Kom.)
Pemegang Hak Cipta *

(Arif Hidayah, S.Tr.Kom., M.Tr.Kom.)
Pemegang Hak Cipta *

* Semua pemegang hak cipta agar menandatangani di atas materai.

Lampiran 3. Surat Pengalihan HKI

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1. **N a m a** : Fadillah Surya Irawan
 Kewarganegaraan : Indonesia
 Alamat : Ds. Sindang RT 02/05, Kecamatan Dukuwaru,
 Kabupaten Tegal Provinsi Jawa Tengah

2. **Nama** : Hepatika Zidny Ilmadina, S.Pd., M.Kom.
 Kewarganegaraan : Indonesia
 Alamat : Jl. Kenanga Gang 1 Nomor 9, Kelurahan Mangkusuman, Kecamatan
 Tegal Timur, Kota Tegal, Provinsi Jawa Tengah, 52123

3. **Nama** : Arif Hidayah, S.Tr.Kom., M.Tr.Kom.
 Kewarganegaraan : Indonesia
 Alamat : Jl. Bulu Emas No.08, RT.001, RW.001, Kelurahan Lawatan,
 Kecamatan Dukuhturi, Kabupaten Tegal, Jawa Tengah

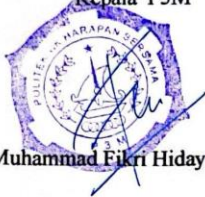
Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

N a m a : Pusat Penelitian dan Pengabdian Masyarakat (P3M)
 Politeknik Harapan Bersama
Alamat : Jl. Mataram No. 9, Pesurungan Lor, Kec. Margadana,
 Kota Tegal, Provinsi Jawa Tengah, 52417

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Sistem Manajemen Untuk Efisiensi Ketersediaan Stok Barang Pada Toko Aby Snack Menggunakan *Laravel*" untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Pemegang Hak Cipta
Kepala P3M



(Muhammad Fikri Hidayattullah, S.T., M.Kom)

Pencipta
Materai 10.000



(Fadillah Surya Irawan)

(Hepatika Zidny Ilmadina, S.Pd., M.Kom.)

(Arif Hidayah, S.Tr.Kom., M.Tr.Kom.)

MANUAL BOOK

**SISTEM INFORMASI EFISIENSI
KETERSEDIAAN BARANG PADA TOKO
ABY SNACK**

2024/2025

DISUSUN OLEH :

Fadillah Surya Irawan

Hepatika Zidny Ilmadina, S.Pd., M.Kom.

Arif Hidayah, S.Tr.Kom., M.Tr.Kom.

1. PENDAHULUAN

1.1. Tujuan Pembuatan Dokumen

Dokumen *manual book* Sistem Informasi manajemen pada toko aby snack berbasis *website* ini disusun dengan tujuan untuk memberikan gambaran umum serta panduan yang jelas mengenai cara penggunaan sistem. Dokumen ini ditujukan bagi seluruh pengguna, baik admin, supplier, customer, maupun mitra agar dapat memahami alur, fitur, dan fungsi dari sistem secara menyeluruh. Dengan adanya dokumen ini, diharapkan proses penggunaan sistem dapat berjalan lebih efektif, efisien, dan terstruktur.

1.2. Deskripsi Umum Sistem

Aplikasi Sistem Manajemen dan Pembelian Toko Aby Snack berbasis website ini dirancang untuk meningkatkan efisiensi dan transparansi dalam pengelolaan operasional toko. Aplikasi ini menyediakan berbagai fitur sesuai dengan peran pengguna, yaitu admin, supplier, mitra, dan customer. Admin dapat memantau seluruh aktivitas pembelian secara real-time, supplier dapat menambahkan produk serta memantau stok yang tersedia di toko, mitra dapat melakukan pembelian langsung melalui halaman utama, dan customer dapat mengakses produk secara lebih mudah dan terstruktur. Dengan sistem ini, proses manajemen menjadi lebih terintegrasi, akurat, dan mudah dipantau oleh semua pihak yang terlibat.

1.3. Deskripsi Dokumen

Dokumen ini dibuat untuk memberikan panduan pengguna “Sistem Informasi Manajemen Pada Toko Aby Snack”. Dokumen ini berisikan informasi sebagai berikut:

1. BAB I

Berisi informasi umum yang merupakan bagian pendahuluan, yang meliputi tujuan pembuatan dokumen, deskripsi umum sistem serta deskripsi dokumen.

2. BAB II

Berisi perangkat yang dibutuhkan untuk “Sistem Informasi Manajemen” meliputi perangkat lunak dan perangkat keras.

3. BAB III

Berisi panduan penggunaan Sistem Informasi Tugas Akhir

2. PERANGKAT YANG DIBUTUHKAN

2.1. Perangkat Lunak

Perangkat lunak yang dibutuhkan adalah sebagai berikut :

1. Sistem operasi seperti *windows*
2. *Web browser (Google Chrome, Microsoft Edge, dll)*

2.2. Perangkat Keras

Perangkat keras (*Hardware*) yang digunakan sebagai berikut :

1. Laptop atau komputer pc
2. Handphone
3. Mouse
4. Keyboard

3. FITUR DAN CARA PENGGUNAAN

3.1. Struktur fitur

Adapun struktur halaman pada Sistem Informasi ini berdasarkan peran pengguna adalah sebagai berikut :

1. Halaman Mitra

Pada halaman utama mitra, terdapat beberapa fitur yang dapat digunakan untuk mendukung seluruh proses, yaitu:

a. Halaman Utama

Pada halaman Utama berisikan navbar, menampilkan data profile user, logout dan menampilkan pesanan saya, menampilkan daftar gratis, menampilkan filter kategori, menampilkan produk, dan menampilkan tombol pembelian barang

b. Halaman Profile

Pada halaman profile mitra dapat untuk melihat profile dari user tersebut dan dapat edit data.

c. Pesanan Saya

Pada halaman pesanan saya dapat menambahkan produk yang sudah akan di beli, edit data pembelian pada keranjang, hapus data, dan menampilkan data yang sudah dibuat dan menampilkan status pembayaran ketika sudah melakukan bayar.

d. Logout

Pada halaman utama ketika user menekan tombol profile otomatis akan ada opsi logout yaitu fungsi untuk keluar dari akun.

e. Daftar

Pada halaman utama dapat dilihat tombol untuk pendaftaran akun secara gratis dan ketika pengguna mendaftar akun akan tertuju menuju halaman pendaftaran lalu isikan form yang terdapat di halaman pendaftaran.

f. Tampilan Penjualan Produk

Pada halaman utama dapat melihat berbagai produk user dapat memilih sesuai kategori lalu setelah itu dapat melakukan checkout untuk dilanjutkan ke halaman pembayaran.

2. Halaman Customer

Pada halaman utama customer, terdapat beberapa fitur yang dapat digunakan untuk mendukung seluruh proses, yaitu:

a. Halaman Utama

Pada halaman Utama berisikan navbar, menampilkan data profile user, logout dan menampilkan pesanan saya, menampilkan daftar gratis, menampilkan filter kategori, menampilkan produk, dan menampilkan tombol pembelian barang

b. Halaman Profile

Pada halaman profile mitra dapat untuk melihat profile dari user tersebut dan dapat edit data.

c. Pesanan Saya

Pada halaman pesanan saya dapat menambahkan produk yang sudah akan di beli, edit data pembelian pada keranjang, hapus data, dan

menampilkan data yang sudah dibuat dan menampilkan status pembayaran ketika sudah melakukan bayar.

d. Logout

Pada halaman utama ketika user menekan tombol profile otomatis akan ada opsi logout yaitu fungsi untuk keluar dari akun.

e. Daftar

Pada halaman utama dapat dilihat tombol untuk pendaftaran akun secara gratis dan ketika pengguna mendaftar akun akan tertuju menuju halaman pendaftaran lalu isikan form yang terdapat di halaman pendaftaran.

f. Tampilan Penjualan Produk

Pada halaman utama dapat melihat berbagai produk user dapat memilih sesuai kategori lalu setelah itu dapat melakukan checkout untuk dilanjutkan ke halaman pembayaran.

3. Halaman Admin

Halaman admin digunakan untuk pengelolaan sistem dan pengguna. Adapun fitur yang tersedia adalah:

a. Dashboard

Pada halaman dashboard berisikan navbar, menampilkan data user, data produk, total transaksi, dan stok barang.

b. Verifikasi Pengguna Baru

Pada halaman verifikasi pengguna baru dapat melihat list pengguna yang baru saja melakukan pendaftaran akun lalu sebagai admin dapat mengkonfirmasi sesuai kebutuhan admin.

c. Kelola Kategori

Pada halaman kelola kategori dapat melakukan tambah kategori produk ketika ada barang masuk dan juga edit kategori.

d. Kelola Produk Toko

Pada halaman Kelola Produk memungkinkan pengguna untuk mengatur data barang yang dijual di toko. Melalui fitur ini, pengguna dapat menambahkan produk baru, mengedit informasi produk seperti nama, harga, dan stok, serta menghapus produk yang sudah tidak digunakan.

e. Transaksi Penjualan

Halaman Transaksi Penjualan digunakan untuk mencatat setiap transaksi pembelian yang dilakukan oleh pelanggan. Pengguna dapat menambahkan transaksi baru dengan memilih produk yang akan dibeli, menentukan jumlahnya, lalu menyimpan transaksi tersebut. Setelah transaksi tersimpan, tersedia tombol Detail yang memungkinkan pengguna untuk melihat informasi lengkap dan juga mengubah status transaksi, misalnya dari "Pending" menjadi "Selesai" sesuai dengan perkembangan proses penjualan.

f. Pembayaran

Halaman Pembayaran digunakan untuk mengelola status pembayaran dari setiap transaksi. Jika terdapat transaksi yang belum lunas, pengguna dapat dengan mudah mengubah statusnya menjadi lunas dengan mengklik tombol Set Lunas.

g. Laporan Penjualan

Halaman Laporan Penjualan digunakan untuk menampilkan rekap data transaksi yang terjadi di toko dalam bentuk harian, bulanan, maupun tahunan. Pengguna dapat memilih rentang waktu tertentu untuk melihat jumlah transaksi dan total penjualan sesuai kebutuhan. Selain itu, halaman ini juga menyediakan fitur cetak laporan dalam format PDF, sehingga memudahkan proses dokumentasi dan pelaporan secara rapi dan profesional.

4. Halaman Supplier

Halaman supplier digunakan untuk pengelolaan sistem dan pengguna. Adapun fitur yang tersedia adalah:

a. Dashboard

Pada halaman dashboard berisikan navbar, menampilkan data profile, produk, dan notifikasi informasi terbaru.

b. Kelola Produk Saya

Halaman Supplier berfungsi untuk mengelola produk yang disediakan langsung oleh pihak supplier. Melalui halaman ini, supplier dapat menambahkan produk baru ke dalam sistem, serta melakukan pengeditan data produk seperti nama barang, harga, dan jumlah stok. Selain itu, supplier juga memiliki opsi untuk menghapus produk yang tidak lagi tersedia atau tidak dijual.

c. Lihat Produk Toko

Halaman Lihat Produk Toko digunakan untuk menampilkan daftar seluruh produk yang tersedia di toko berdasarkan peran pengguna sebagai pihak toko. Melalui halaman ini, pengguna dapat melihat informasi lengkap produk seperti nama barang, harga, dan jumlah stok yang tersedia. Selain itu, halaman ini juga menyediakan fitur bagi supplier untuk menyuplai barang ke toko, sehingga stok produk dapat diperbarui secara langsung.

3.2. Penggunaan

Pada bagian ini akan dijelaskan mengenai cara penggunaan dari fitur-fitur yang tersedia di Sistem Informasi Tugas Akhir.

1. Cara Membuka Sistem

Untuk menjalankan website sistem informasi tugas akhir sebagai berikut:

1. Bukalah *website* sistem melalui *web browser* yang tersedia seperti (*Google Chrome, Microsoft Edge, dll*).
2. Tekan tombol enter pada *keyboard*.
3. Setelah itu, akan muncul tampilan utama *website* yaitu fitur home, dapat mendaftar akun, login dan melihat produk yang tersedia di toko melalui website ini.



Gambar 1. Halaman Home

2. Halaman Customer dan Mitra

Pada sistem terdapat halaman yang hanya dapat di akses oleh user role customer dan mitra yang akan di arahkan setelah masuk kedalam website dan terdiri dari beberapa fitur sebagai berikut:

a. Fitur Halaman Utama

Fitur halaman utama mahasiswa untuk menampilkan navbar yang terdiri dari nama toko, login register dan juga dapat memilih produk, Adapun tampilan halaman utama dapat dilihat pada gambar 2.



Gambar 2. Halaman Utama

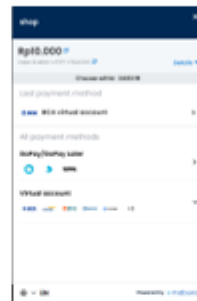
b. Fitur Keranjang

Fitur keranjang untuk melihat pesanan yang dimasukan keranjang, dan akan melakukan pembayaran ataupun hapus keranjang sesuai kebutuhan pengguna. Hapus pesanan ada pada fitur keranjang dapat dilihat halamannya pada gambar 3.



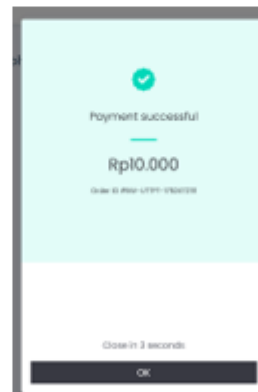
Gambar 3. Halaman Keranjang

Ketika sudah melakukan checkout dan bayar akan diarahkan menuju halaman pembayaran yang dimana anda bisa memilih metode pembayaran dan juga check status pada halaman tersebut. Dapat dilihat pada gambar 4.



Gambar 4. Halaman metode pembayaran

Jika anda sudah memilih metode pembayaran dan melakukan pembayaran sesuai dengan pesanan anda silahkan dapat melihat status pembayaran pada gambar 5 sebagai berikut, dan selesai:



Gambar 5. Halaman pembayaran

c. Fitur Register

Pada fitur register atau pendaftaran akun, pengguna wajib memakai data diri sendiri untuk melakukan register agar dapat mempermudah melakukan proses dari mulai transaksi hingga pembayaran. Caranya ketika baru masuk kedalam website arahkan dan tekan register lalu gambar 6 menampilkan halaman register.

Gambar 6. Halaman Registrasi

Setelah melakukan pendaftaran anda harus login untuk melakukan transaksi

d. Fitur Login

Fitur login untuk customer dan mitra dirancang agar pengguna dapat mengakses website secara lebih mendalam dan sesuai dengan kebutuhan masing-masing peran. Setelah melakukan login, customer dapat melihat dan membeli produk yang tersedia, memantau riwayat pembelian, serta mendapatkan layanan yang dipersonalisasi. Berikut gambar 7 akan menampilkan login untuk mempermudah :



Gambar 7. Halaman Login

3. Halaman Supplier

Pada sistem terdapat halaman yang hanya dapat di akses oleh user role supplier yang akan di arahkan setelah masuk kedalam website dan terdiri dari beberapa fitur sebagai berikut:

Ketika sudah masuk kedalam website akan menampilkan halaman utama



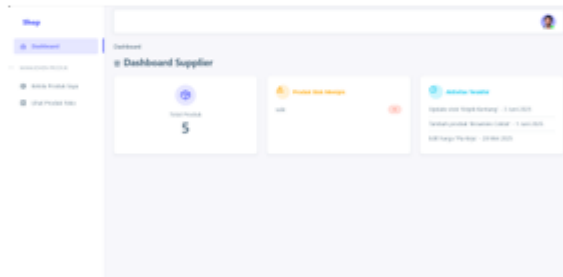
Gambar 8. Halaman Utama

Lalu setelah masuk kedalam halaman utama klik login untuk menampilkan halaman login sebagai berikut :



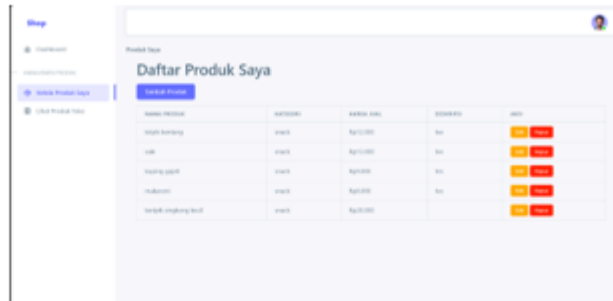
Gambar 9. Halaman Login

Setelah melakukan klik login anda akan disuguhkan ke dalam halaman dashboard yang dimana anda dapat melihat beberapa informasi mengenai role supplier tersebut.



Gambar 10. Halaman Dashboard Supplier

Disini anda dapat memilih selanjutnya untuk menuju halaman lihat produk toko dan kelola produk saya. Kita menuju ke kelola produk saya terlebih dahulu dapat dilihat pada gambar 11.



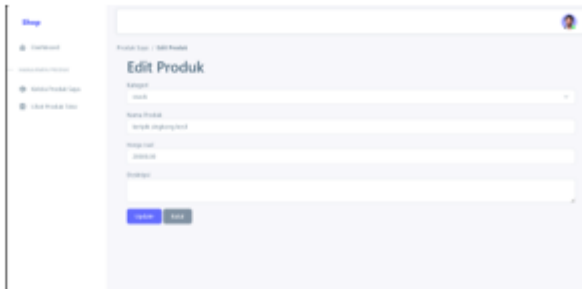
Gambar 11. Kelola Produk

Pada halaman kelola Produk saya, anda dapat menambahkan produk, edit dan hapus produk yang role supplier miliki. Ketika akan menambahkan produk klik tambah produk lalu akan muncul tampilan seperti ini yang dapat dilihat di gambar 12.



Gambar 12. Tambah Produk Supplier

Ketika sudah menambahkan produk toko dan klik [simpan sesuai](#) kebutuhan anda, dapat menampilkan halaman kelola produk, dan ketika ingin edit produk yang sudah ada di data anda dapat mengubahnya dengan klik edit lalu akan ditampilkan halaman 13 sebagai berikut :



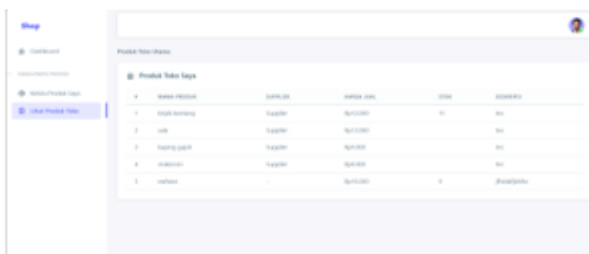
Gambar 13. Halaman Edit Produk Supplier

Setelah klik update pada halaman edit produk, anda dapat menghapus ketika barang yang anda masukan salah atau memang tidak digunakan dengan cara seperti pada gambar 14 sebagai berikut :



Gambar 14. Halaman Hapus Produk Supplier

Lalu ada halaman lihat produk toko yang dimana halaman ini ketika role supplier ingin melihat produk yang tersedia di toko dapat diakses melalui fitur ini yang dapat ditampilkan pada gambar 15.



Gambar 15. Halaman Lihat Produk Toko

4. Halaman Admin

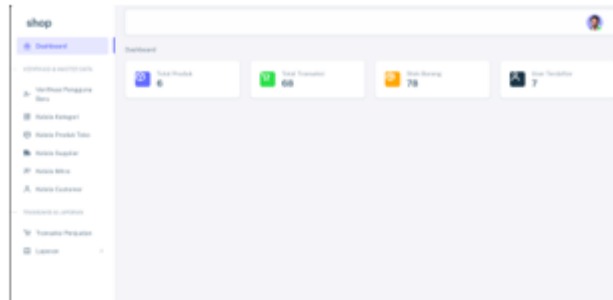
Pada halaman admin pusat kendali utama dalam aplikasi ini yang hanya dapat diakses oleh pengguna dengan peran sebagai admin setelah berhasil login. Melalui halaman ini, admin memiliki akses penuh untuk mengelola seluruh sistem, termasuk fitur-fitur seperti manajemen pengguna, pengaturan data produk, pemantauan transaksi, dan pengelolaan konten. Admin juga dapat meninjau aktivitas mitra dan customer, mengatur hak akses, serta memverifikasi data yang masuk.

Pertama login. Ketika sudah masuk ke halaman utama pengguna harus login ke dalam halaman [admin](#) berikut caranya. Pada halaman login isi username dan password lalu klik login berikut pada gambar 16.



Gambar 16. Halaman Login Admin

- a. Ketika sudah klik login lalu akan muncul halaman dashboard admin yang berisikan informasi dari akses yang dimiliki role admin berikut gambar 17 dashboard.



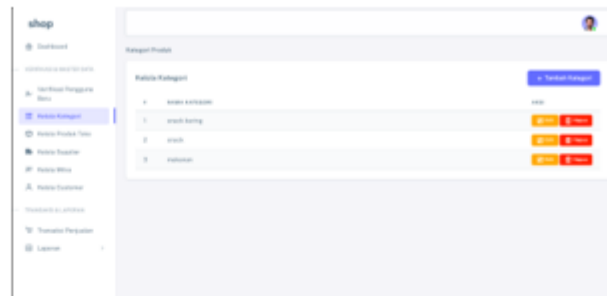
Gambar 17. Halaman Dashboard Admin

- b. Pada halaman admin terdapat berbagai navbar, disini akan kami beri tahu cara menggunakan fitur verifikasi pengguna baru. Verifikasi pengguna baru berfungsi untuk memberi akses para pembuat akun baru dari supplier, mitra atau customer. Berikut gambar 18 ditampilkan -



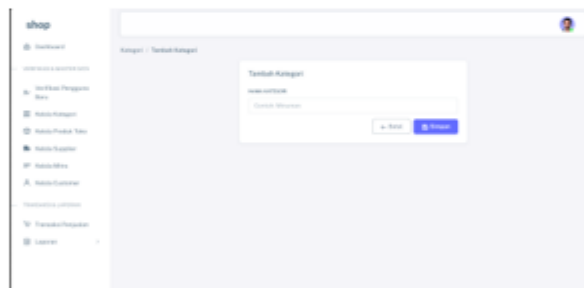
Gambar 18. Halaman Verifikasi Pengguna Baru

- c. Lalu ada fitur halaman kelola kategori, pada halaman ini anda dapat menambahkan kategori yang ingin anda tambahkan seperti kue,snack atau yang lainnya. Berikut gambar 19 tampilan kategori -



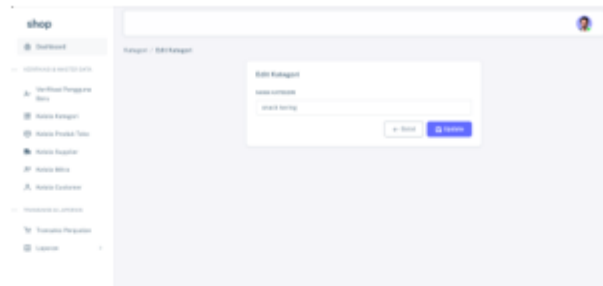
Gambar 19. Halaman Kelola Kategori

Pada halaman kategori anda bisa menambahkan edit dan juga hapus kategori yang ada di data. Berikut kami berikan cara untuk menambah kategori. Klik pada halaman kategori lalu klik tambah kategori, setelah itu akan ditampilkan halaman tambah kategori sebagai berikut :



Gambar 20. Halaman Tambah Kategori

Setelah klik tambah kategori anda bisa klik simpan lalu dapat edit kategori ketika ada penamaan yang salah. Berikut cara edit kategori :



Gambar 21. Halaman Edit Kategori

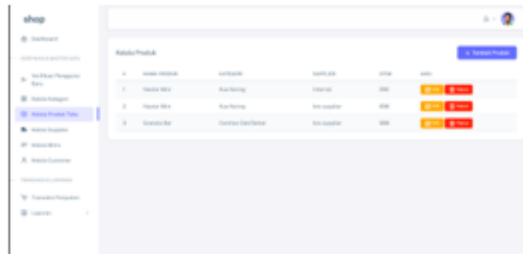
Klik update lalu edit kategori selesai. Lalu anda dapat menghapus kategori ketika tidak sesuai dengan kebutuhan anda dengan cara sebagai berikut gambar 22 akan menampilkan :



Gambar 22. Halaman Hapus Kategori

d. Halaman Kelola Produk Toko

Halaman produk toko menampilkan data yang berisi produk, cara menambah, edit dan juga hapus produk. Berikut kita mulai dari tambah produk. Pertama klik tombol tambah produk



Gambar 23. Halaman Kelola Produk

Setelah klik tambah produk akan muncul tampilan tambah produk sebagai berikut pada gambar [24](#) :

Gambar 24. Halaman Tambah Produk

Pada halaman tambah produk silahkan anda dapat mengisi kolom kosong dengan isian produk yang anda miliki satu per satu, lalu anda klik simpan jika sudah mengisi semua kolom dan kembali ke halaman kelola produk kembali.

Lalu kita ke cara edit produk bisa di lihat pada gambar [25](#) :



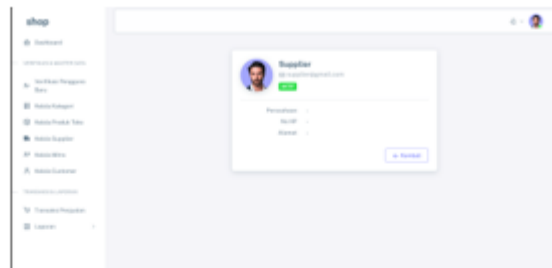
Gambar 27. Halaman Kelola Supplier

Pada halaman kelola supplier kita dapat melihat produk supplier, detail dan nonaktifkan akun. Kita langsung ke cara lihat produk. Pertama klik lihat produk pada halaman kelola supplier lalu akan muncul gambar seperti ini



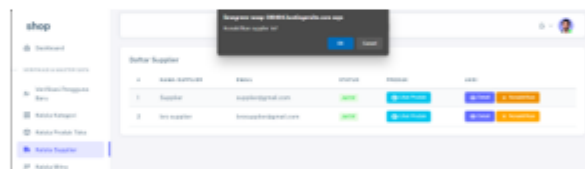
Gambar 28. Halaman Lihat Produk Supplier

Setelah melakukan lihat produk, kita lanjut ke klik detail dengan fungsi melihat profile supplier caranya akan ditampilkan pada gambar 29. Pada halaman kelola supplier klik detail.



Gambar 29. Halaman Detail Supplier

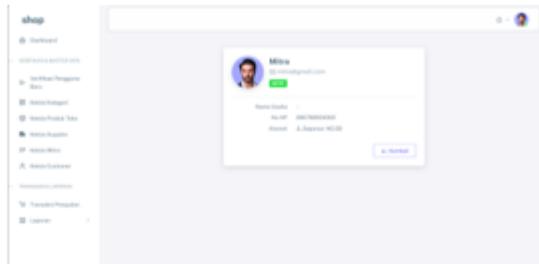
Jika sudah klik detail, lalu kita masuk ke halaman nonaktifkan akun supplier dengan cara klik nonaktif dapat ditampilkan pada gambar 30 sebagai berikut-:



Gambar 30. Halaman Nonaktif Supplier

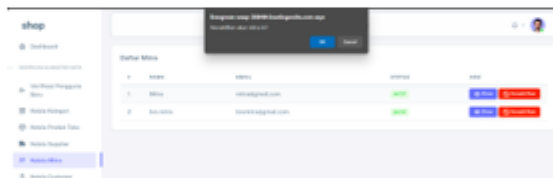
f. Halaman Kelola Mitra

Pada halaman kelola mitra disini dapat melihat mitra yang terdaftar dan dapat melihat profile lalu nonaktifkan akun melalui halaman ini. Berikut cara mengakses lihat profile akun mitra lalu klik Show akan muncul tampilan seperti ini-:



Gambar 31. Halaman Show Mitra

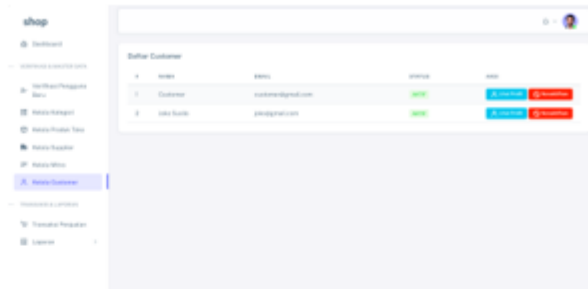
Lalu jika sudah lihat profile kita lanjut ke fungsi nonaktifkan akun dengan cara klik nonaktif pada halaman kelola mitra dan menampilkan jendela sebagai berikut :-



Gambar 32. Halaman Nonaktifkan Mitra

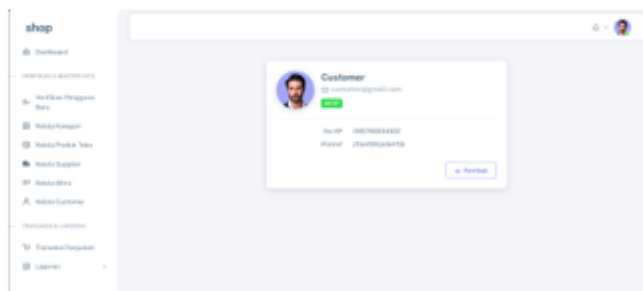
g. Halaman Kelola Customer

Pada halaman kelola customer disini terdapat beberapa fungsi yaitu show dan nonaktifkan. Seperti ini cara menggunakan melihat profil pada halaman tersebut :-



Gambar 33. Halaman Kelola Customer

Klik lihat profile lalu akan menampilkan halaman seperti [ini](#) :



Gambar 34. Halaman Lihat Profile Customer

Setelah kita mengakses profile customer lalu kita dapat menonaktifkan akun customer dengan cara klik nonaktifkan pada halaman kelola customer lalu muncul jendela pada gambar [34](#) :



Gambar 34. Halaman Nonaktifkan Akun Customer

h. Halaman Transaksi Penjualan

Terdapat history transaksi yang dilakukan user ketika pembelian barang lalu masuk kedalam halaman ini. Ada beberapa fungsi yaitu tambah transaksi dan juga detail. Berikut gambar 35 menampilkan halaman transaksi penjualan :

ID	Kode	Nama Produk	Status	Tanggal	dan Waktu	Aksi
1001	1001	Customer	Belum Dibayar	10/10/2020	10:00	Detail
1002	1002	Customer	Belum Dibayar	10/10/2020	10:00	Detail
1003	1003	Customer	Belum Dibayar	10/10/2020	10:00	Detail
1004	1004	Customer	Belum Dibayar	10/10/2020	10:00	Detail

Gambar 35. Halaman Transaksi Penjualan

Lalu klik tambah transaksi akan muncul halaman sebagai berikut.

Transaksi Penjualan Baru (Detail)

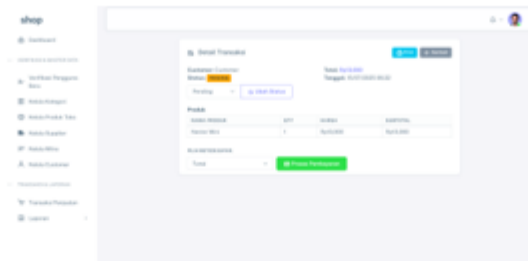
Nama Produk:

Status:

Tanggal:

Gambar 36. Halaman Tambah Transaksi

Ketika sudah memasukan kolom yang terdapat halaman diatas lalu klik simpan akan masuk ke halaman transaksi penjualan. Lalu ke fungsi detail yaitu dimana fungsi ini dapat mengubah status menjadi selesai dan menyelesaikan pembayaran dapat dilihat gambar 37 :



Gambar 37. Halaman Detail Transaksi

Lalu pada halaman detail ada tombol print untuk cetak invoice caranya klik print lalu akan muncul halaman pdf dan cetak dapat dilihat di gambar 38 :



Gambar 38. Halaman Print PDF

i. Halaman Laporan Penjualan

Pada halaman ini terdapat laporan yang dilihat sebagai admin sebagai berikut :



DOKUMEN TEKNIKAL

**SISTEM MANAJEMEN
UNTUK EFISIENSI
KETERSEDIAAN STOK
BARANG PADA TOKOABY
SNACK MENGGUNAKAN
LARAVEL**

Disusun Oleh :

Fadillah Surya Irawan

Hepatika Zidny Ilmadina. S.Pd., M.Kom.

Arif Hidayah, S.Tr.Kom., M.Tr.Kom



1. register controller

```
public function register(Request $r)
{
    // Validasi input
    $rules = [
        'nama' => 'required|string|max:255',
        'email' => 'required|email|unique:users,email',
        'password' => 'required|string|min:6',
        'role' => 'required|string|in:admin,partner,supplier,customer',
    ];
    $this->validate($r->all(), $rules);

    // Validasi apakah data role admin
    if ($r->input('role') === 'admin') {
        $this->validate($r->all(), [
            'nama' => 'required|string|max:255',
            'email' => 'required|email|unique:users,email',
            'password' => 'required|string|min:6',
            'role' => 'required|string|in:admin',
        ]);
    }

    // Validasi apakah data role partner
    if ($r->input('role') === 'partner') {
        $this->validate($r->all(), [
            'nama' => 'required|string|max:255',
            'email' => 'required|email|unique:users,email',
            'password' => 'required|string|min:6',
            'role' => 'required|string|in:partner',
        ]);
    }

    // Validasi apakah data role customer
    if ($r->input('role') === 'customer') {
        $this->validate($r->all(), [
            'nama' => 'required|string|max:255',
            'email' => 'required|email|unique:users,email',
            'password' => 'required|string|min:6',
            'role' => 'required|string|in:customer',
        ]);
    }

    // Simpan user ke db
    $data = [
        'nama' => $r->input('nama'),
        'email' => $r->input('email'),
        'password' => $r->input('password'),
        'role' => $r->input('role'),
    ];
    $data['password'] = bcrypt($data['password']);
    $user = User::create($data);
    return response($user, 201);
}
```

Fungsi [register\(\)](#) pada AuthController bertanggung jawab untuk memvalidasi input pendaftaran berdasarkan peran pengguna (admin, supplier, mitra, atau customer) dan menyimpan data pengguna ke dalam database.

Validasi dilakukan secara dinamis sesuai peran (*role*) yang dipilih, termasuk validasi tambahan untuk data spesifik seperti alamat, nomor HP, dan foto, sebelum data disimpan dengan password yang sudah di-hash.

2. login dan logout controllers

```
public function login(Request $r)
{
    $credentials = [
        'email' => $r->input('email'),
        'password' => $r->input('password'),
    ];
    if (!Auth::attempt($credentials)) {
        return response('Invalid email or password', 401);
    }
    $user = Auth::user();
    $token = $user->createToken($r->input('remember_me') ? 72 : 1);
    return response($token->accessToken, 200);
}

public function logout()
{
    Auth::logout();
    return response('Logout successfully', 200);
}

public function refreshToken()
{
    $token = $r->input('token');
    if (!Auth::validateToken($token)) {
        return response('Invalid token', 401);
    }
    $user = Auth::user();
    $newToken = $user->createToken(1);
    return response($newToken->accessToken, 200);
}
```


Login: Metode `login()` memverifikasi kredensial pengguna menggunakan `Auth::attempt()`, memeriksa status aktivasi akun, dan mengarahkan pengguna ke halaman dashboard sesuai peran mereka (admin, supplier, mitra, atau customer).

Logout: Metode `logout()` mengeluarkan pengguna dari sesi yang aktif menggunakan `Auth::logout()` dan mengarahkan kembali ke halaman login.

3. admin dashboard controllers

```

<?php
namespace App\Http\Controllers;

use App\Models/Product;
use App\Models\Order;
use App\Models\User;

class AdminDashboardController extends Controller
{
    public function index()
    {
        $produkCount = Product::count();
        $transaksiCount = Order::count();
        $stokTotal = Product::sum('stok');
        $userCount = User::count();

        return view('admin.dashboard', compact('produkCount', 'transaksiCount', 'stokTotal', 'userCount'));
    }
}

```

Metode `index()` pada `AdminDashboardController` menyiapkan data statistik berupa jumlah produk, total transaksi, akumulasi stok, dan jumlah pengguna, lalu mengirimkan data tersebut ke tampilan dashboard admin.

4. category controllers

```

<?php
namespace App\Http\Controllers;

use App\Models\Category;
use App\Models\Product;
use App\Models\User;

class CategoryController extends Controller
{
    public function index()
    {
        $categories = Category::all();
        return view('admin.categories.index', compact('categories'));
    }

    public function create()
    {
        if (auth()->user() != 'admin') abort(403);
        return view('admin.categories.create');
    }

    public function store(Request $request)
    {
        if (auth()->user() != 'admin') abort(403);
        $request->validate([
            'name' => 'required',
            'description' => 'required'
        ]);
        $category = Category::create($request->only('name', 'description'));
        return redirect()->route('categories.index');
    }

    public function edit($id)
    {
        if (auth()->user() != 'admin') abort(403);
        $category = Category::findOrFail($id);
        return view('admin.categories.edit', compact('category'));
    }

    public function update(Request $request, $id)
    {
        if (auth()->user() != 'admin') abort(403);
        $request->validate([
            'name' => 'required',
            'description' => 'required'
        ]);
        $category = Category::findOrFail($id);
        $category->update($request->only('name', 'description'));
        return redirect()->route('categories.index');
    }

    public function destroy($id)
    {
        if (auth()->user() != 'admin') abort(403);
        $category = Category::findOrFail($id);
        $category->delete();
        return redirect()->route('categories.index');
    }
}

```

5. customer controller

Controller *CustomerController* digunakan untuk mengelola data customer, termasuk menampilkan daftar, melihat detail, serta menonaktifkan akun customer dengan memverifikasi peran dan status aktif pengguna.

[illegible]

D-32

7. laporan controllers

```
class LaporanController extends Controller
{
    public function penjualan(Request $request)
    {
        $tglAwal = $request->get('tgl_awal', now())->format('Y-m-d');
        $tglAkhir = $request->get('tgl_akhir', now())->format('Y-m-d');

        $orders = Order::whereBetween('tanggal_order', [$tglAwal, $tglAkhir])
            ->where('status_order', 'selesai')
            ->with('user')
            ->orderBy('tanggal_order', 'desc')
            ->get();

        $total = $orders->sum('total_order');

        return view('admin.laporan.penjualan', compact('orders', 'tglAwal', 'tglAkhir', 'total'));
    }

    public function produkTerlaris(Request $request)
    {
        $tglAwal = $request->get('tgl_awal', now())->format('Y-m-d');
        $tglAkhir = $request->get('tgl_akhir', now())->format('Y-m-d');

        $terlaris = OrderItem::select('product_id', DB::raw('SUM(qty) as total_terjual'))
            ->whereHas('order', function ($q) use ($tglAwal, $tglAkhir) {
                $q->where('status_order', 'selesai')
                    ->whereBetween('tanggal_order', [$tglAwal, $tglAkhir]);
            })
            ->groupBy('product_id')
            ->orderByDesc('total_terjual')
            ->with('product')
            ->take(10)
            ->get();

        return view('admin.laporan.produk_terlaris', compact('terlaris', 'tglAwal', 'tglAkhir'));
    }
}
```

penjualan(Request \$request)

Menampilkan laporan penjualan berdasarkan rentang tanggal (tgl_awal dan tgl_akhir).

produkTerlaris(Request \$request)

Menampilkan 10 produk terlaris berdasarkan jumlah kuantitas (qty) yang terjual pada periode yang dipilih.

8. midtrans controllers

```

class MidtransController extends Controller {
  def init: Function[_root_.scala.concurrent.Future] = {
    // Set midtrans api key
    ConfigKeys.apiKey <- config("midtrans.api.key")
    ConfigKeys.notificationUrl <- config("midtrans.notificationUrl")
    ConfigKeys.notificationUrl <- config("midtrans.notificationUrl")
    ConfigKeys.notificationUrl <- config("midtrans.notificationUrl")
    ConfigKeys.notificationUrl <- config("midtrans.notificationUrl")
  }

  // Notification dari Midtrans
  def notificationHandler(request: Request) = {
    log {
      // Log request untuk debugging
      log.info(s"Midtrans notification received: ${request}")
      "status" <- RequestHandler.status
      "type" <- RequestHandler.type
      "raw_body" <- this.get("body").getOrElse("")
    }

    // Set request body
    request = request.withBody(request.body)

    // Log notification details
    log.info(s"Midtrans notification details: ${request}")
    "order_id" <- request("order_id")
    "transaction_status" <- request("transaction_status")
    "signature_key" <- request("signature_key")
    "signature_time" <- request("signature_time")
    "signature_type" <- request("signature_type")

    // Set order berdasarkan order_id (dalam order_id)
    order = Order.findById(order_id, Order.order_id)

    if (order) {
      log.info(s"Order not found for notification: ${order}")
      "order_id" <- Order.getId(order_id)
      "order_id" <- Order.getId(order_id)
    }

    return Response().with("message" <- "Order not found", 400)
  }
}

```

MidtransController digunakan untuk menangani konfigurasi awal dan notifikasi transaksi dari Midtrans, termasuk verifikasi status transaksi dan pencocokan `order_id` untuk update status pesanan. Data notifikasi yang diterima akan dicatat melalui log untuk keperluan debugging dan validasi keamanan.

```

class MidtransController extends Controller {
  // Update status order (dalam status)
  def updateStatusOrder(status: String) = {
    log.info(s"Updating midtrans transaction: ${status}")
    "order_id" <- Order.getId
    "transaction_status" <- Order.transaction_status
    "signature_key" <- Order.signature_key
    "signature_time" <- Order.signature_time
    "signature_type" <- Order.signature_type

    // Set request body
    request = request.withBody(request.body)

    // Log notification details
    log.info(s"Midtrans notification details: ${request}")
    "order_id" <- request("order_id")
    "transaction_status" <- request("transaction_status")
    "signature_key" <- request("signature_key")
    "signature_time" <- request("signature_time")
    "signature_type" <- request("signature_type")

    // Set order berdasarkan order_id (dalam order_id)
    order = Order.findById(order_id, Order.order_id)

    if (order) {
      log.info(s"Order not found for notification: ${order}")
      "order_id" <- Order.getId(order_id)
      "order_id" <- Order.getId(order_id)
    }

    return Response().with("message" <- "Order not found", 400)
  }

  // Set request body
  request = request.withBody(request.body)

  // Log notification details
  log.info(s"Midtrans notification details: ${request}")
  "order_id" <- request("order_id")
  "transaction_status" <- request("transaction_status")
  "signature_key" <- request("signature_key")
  "signature_time" <- request("signature_time")
  "signature_type" <- request("signature_type")

  // Set order berdasarkan order_id (dalam order_id)
  order = Order.findById(order_id, Order.order_id)

  if (order) {
    log.info(s"Order not found for notification: ${order}")
    "order_id" <- Order.getId(order_id)
    "order_id" <- Order.getId(order_id)
  }

  return Response().with("message" <- "Order not found", 400)
}

```

Kode ini menangani pembaruan status pesanan (`status_order`) berdasarkan notifikasi status transaksi dari Midtrans, seperti `settlement`, `expire`, `cancel`, dan `pending`, serta mencatat

log untuk setiap perubahan status. Proses ini mencakup verifikasi kondisi sebelum dan sesudah penyimpanan ke database, memastikan konsistensi data dan pencatatan informasi lengkap guna mendukung debugging dan audit transaksi.

9. mitra controller

```
class MitraController extends Controller
{
    // Lihat daftar mitra aktif & pending
    public function index()
    {
        $mitra = $this->show('role', 'mitra')->get();
        return view('admin.mitra.index', compact('mitra'));
    }

    // Edit mitra (tampilkan form)
    public function edit($id)
    {
        $mitra = $this->findOrFail($id);
        return view('admin.mitra.edit', compact('mitra'));
    }

    // Update mitra (nama, email, dll)
    public function update(Request $request, $id)
    {
        $mitra = $this->findOrFail($id);
        $request->validate([
            'name' => 'required|string|max:100',
            'email' => 'required|email'
        ]);
        $mitra->update($request->only('name', 'email'));
        return redirect()->route('mitra.index')->with('success', 'Data mitra diupdate!');
    }

    // Nonaktifkan mitra (set is_active = 0)
    public function disable($id)
    {
        $mitra = $this->findOrFail($id);
        $mitra->is_active = 0;
        $mitra->save();
        return back()->with('success', 'Akun mitra dinonaktifkan.');
```

Fungsi utama `MitraController` adalah untuk mengelola akun pengguna dengan peran mitra, termasuk menampilkan daftar mitra, menampilkan dan memproses form edit, mengupdate data mitra, menonaktifkan akun, dan menampilkan detail mitra tertentu.

Validasi input dilakukan pada proses update untuk memastikan integritas data, dan setiap perubahan data akan menghasilkan umpan balik melalui session flash message yang ditampilkan pada tampilan (view) terkait.

9. order controller

```

// List semua transaksi/order
public function index(Request $request)
{
    $query = Order::whereIn('user', 'payment');

    // Filter status jika ada
    if ($request->filled('status')) {
        $query->where('status_order', $request->status);
    }

    // Search by customer name jika ada
    if ($request->filled('q')) {
        $query->where('user', function ($q) use ($request) {
            $q->where('name', 'like', '%' . $request->q . '%');
        });
    }

    // Paginate, base query string
    $orders = $query->orderBy('created_at')->paginate(10)->withQueryString();
    $statuses = ['pending', 'lunas', 'selesai'];

    return view('admin.orders.index', compact('orders', 'statuses'));
}

// Form tambah transaksi baru (basic)
public function create()
{
    $products = Product::all();
    $customers = User::where('role', 'customer', 'with' ['api']);
    return view('admin.orders.create', compact('products', 'customers'));
}

public function store(Request $request)
{
    $request->validate([
        'user_id' => 'required|exists:users,id',
        'type_order' => 'required|in:offline,online,atm',
        'produk_id' => 'required|array',
        'produk_id.*' => 'required|exists:products,id',
        'qty' => 'required|array',
        'qty.*' => 'required|numeric|min:1',
    ]);

    // ... (rest of the code)
}

```

Dokumen ini menjelaskan proses perancangan, implementasi, dan pengujian sistem secara sistematis untuk memastikan fungsionalitas, keandalan, serta pemenuhan kebutuhan pengguna.

10. payment controllers

```

// List semua status pembayaran
public function index(Request $request)
{
    $query = Payment::where('order');

    // Filter optional
    if ($request->filled('status_bayar') && $request->status_bayar) {
        $query->where('status_bayar', $request->status_bayar);
    }
    if ($request->filled('metode_bayar') && $request->metode_bayar) {
        $query->where('metode_bayar', $request->metode_bayar);
    }

    $payments = $query->orderBy('id')->get();

    return view('admin.payments.index', compact('payments'));
}

// Tombol manual "lunas"
public function lunas($id)
{
    $payment = Payment::findOrFail($id);
    $payment->status_bayar = 'success';
    $payment->metode_bayar = null();
    $payment->save();

    // Update order status juga
    $payment->order();
    $payment->status = $payment->order->status_order;
    $payment->order->status_order = 'lunas';
    $payment->order->save();

    // Tampilkan message jika sukses, hanya update status ke lunas
    // Jika akan ditanggapi ketika ada status ke "selesai"

    // Hanya yang token dari session karena pembayaran sudah lunas secara manual
    session(['status_bayar' => $payment->order->id]);
    log('Info: [Manual Payment] User token cleared from session for manual payment', ['order_id' => $payment->order->id]);

    return back()->with('success', 'Pembayaran ditandai sebagai lunas.');
```

PaymentController mengelola proses pembayaran termasuk pembuatan transaksi, penandaan pembayaran manual, serta penanganan callback Midtrans untuk memperbarui status order dan pembayaran secara otomatis.

11. product controller

```
class ProductController extends Controller
{
    public function index()
    {
        // Tampilkan semua produk (internal & supplier)
        $products = Product::with('category', 'supplier')
            ->orderByDesc('created_at')
            ->paginate(20);

        $notif_products = Product::with('supplier')
            ->whereNotNull('supplier_id')
            ->orderByDesc('created_at')
            ->take(10) // atau limit sesuai kebutuhan
            ->get();

        return view('admin.products.index', compact('products', 'notif_products'));
    }

    public function show($id, $notificationId = null)
    {
        $product = Product::with(['category', 'supplier', 'images'])->findOrFail($id);

        // Mark as seen jika produk supplier & belum dilihat
        if ($product->supplier_id && !$product->notif_admin_seen) {
            $product->notif_admin_seen = 1;
            $product->save();
        }

        // Jika dari notifikasi Laravel (notificationId), tandai sudah dibaca juga
        if ($notificationId) {
            $notif = auth()->user()->notifications()->where('id', $notificationId)->first();
            if ($notif && !$notif->read_at) {
                $notif->markAsRead();
            }
        }

        return view('admin.products.show', compact('product'));
    }
}
```

ProductController bertanggung jawab untuk mengelola data produk pada sisi admin. Fungsi [index\(\)](#) digunakan untuk menampilkan daftar semua produk, baik internal maupun dari supplier, beserta notifikasi produk terbaru dari supplier. Fungsi [show\(\)](#) digunakan untuk menampilkan detail produk tertentu, sekaligus menandai produk dari supplier sebagai telah dilihat oleh admin, dan mengatur status notifikasi Laravel jika ditampilkan melalui sistem notifikasi.

```

public function create()
{
    $categories = Category::all();
    $suppliers = Supplier::where('role', 'supplier')->get();
    return view('admin.products.create', compact('categories', 'suppliers'));
}

public function store(Request $request)
{
    $user = auth()->user();

    $request->validate([
        'category_id' => 'required|exists:categories,id',
        'kode_produk' => 'required|string|max:50',
        'nama_produk' => 'required|string|max:100',
        'harga_beli' => 'nullable|numeric',
        'harga_jual' => 'required|numeric',
        'stok' => 'required|numeric',
        'deskripsi' => 'nullable|string',
        'images.*' => 'nullable|image|max:1000',
    ]);

    $data = $request->only('category_id', 'kode_produk', 'nama_produk', 'harga_beli', 'harga_jual', 'stok', 'deskripsi');
    $data['supplier_id'] = ($user->role == 'supplier') ? $user->id : $request->supplier_id;

    $product = Product::create($data);

    // Upload multi foto produk
    if ($request->hasFile('images')) {
        foreach ($request->file('images') as $img) {
            $product->images()->create([
                'file_path' => $img->store('product_images', 'public')
            ]);
        }
    }

    return redirect()->route('products.index')->with('success', 'Produk berhasil ditambahkan');
}

public function edit($id)
{
    $product = Product::findOrFail($id);
    $categories = Category::all();
    $suppliers = Supplier::where('role', 'supplier')->get();
}

```

create() Menampilkan halaman form tambah produk baru beserta daftar kategori dan supplier yang tersedia. **store(Request \$request)** Menyimpan data produk baru ke database, termasuk upload gambar produk jika tersedia. **edit(\$id)** Menampilkan halaman edit produk berdasarkan ID, lengkap dengan data kategori dan supplier. **update(Request \$request, \$id)** Memperbarui data produk yang ada berdasarkan ID, termasuk validasi dan pemrosesan data yang dikirim dari form.

12. profile controllers


```

class ProfileController extends Controller
{
    public function update(Request $request)
    {
        $user = Auth::user();
        $user->name = $request->name;
        $user->save();

        switch ($user->role) {
            case 'admin':
                $user->no_hp = $request->no_hp;
                $user->alamat = $request->alamat;
                $user->save();
                break;
            case 'customer':
                $customer = App\Models\Customer::where('user_id', $user->id)->first();
                if ($customer) {
                    $customer->nama_customer = $request->nama;
                    $customer->no_hp = $request->no_hp;
                    $customer->alamat = $request->alamat;
                    $customer->save();
                }
                break;
            case 'mitra':
                $mitra = App\Models\Mitra::where('user_id', $user->id)->first();
                if ($mitra) {
                    $mitra->nama_mitra = $request->nama;
                    $mitra->no_hp = $request->no_hp;
                    $mitra->alamat = $request->alamat;
                    $mitra->save();
                }
                break;
            case 'supplier':
                $supplier = App\Models\Supplier::where('user_id', $user->id)->first();
                if ($supplier) {
                    $supplier->nama_supplier = $request->nama;
                    $supplier->no_hp = $request->no_hp;
                    $supplier->alamat = $request->alamat;
                    $supplier->save();
                }
                break;
        }

        return back()->with('success', 'Profil berhasil diperbarui');
    }
}

```

ProfileController bertugas untuk menangani proses pembaruan data profil pengguna berdasarkan peran masing-masing (admin, customer, mitra, atau supplier). Fungsi utamanya adalah mengambil data pengguna yang sedang login, melakukan validasi input, lalu menyimpan perubahan informasi seperti nama, nomor telepon, dan alamat ke dalam database

13. supplier controllers

```

class SupplierController extends Controller
{
  // List semua supplier
  public function index()
  {
    $suppliers = User::where('role', 'supplier')->get();
    return view('admin.suppliers.index', compact('suppliers'));
  }

  // Menampilkan supplier berdasarkan id_supplier = $id
  public function product($supplier_id)
  {
    $supplier = User::findOrFail($supplier_id);
    $products = Product::where('supplier_id', $supplier_id)->get();
    return view('admin.suppliers.products', compact('supplier', 'products'));
  }

  // Debug version - List produk milik supplier dengan debug info
  public function productDebug($supplier_id)
  {
    $supplier = User::findOrFail($supplier_id);

    // Ambil semua produk dari supplier (termasuk yang belum di approve)
    $products = Product::where('supplier_id', $supplier_id)
      ->select('category')
      ->get();

    // Debug logging
    log('Data Supplier Products Debug: ', [
      'supplier_id' => $supplier_id,
      'supplier_name' => $supplier->name,
      'products_count' => $products->count(),
      'all_products_count' => Product::count(),
      'products_with_supplier_id' => Product::whereNotNull('supplier_id')->count()
    ]);

    return view('admin.suppliers.products-debug', compact('supplier', 'products'));
  }

  // Pesan produk dari supplier (user purchase dan close produk ke toko)
  public function orderProduct(Request $request, $supplier_id, $product_id)
  {

```

SupplierController mengelola data supplier, termasuk menampilkan daftar supplier, produk milik supplier, serta memfasilitasi pemesanan produk dari supplier ke toko internal. Controller ini juga menangani fitur debugging produk supplier, menyalin produk ke toko jika dilakukan pemesanan, serta menyediakan fungsi untuk menonaktifkan supplier dari sistem.

14. supplier dashboard controllers

```

<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Product;
use Illuminate\Support\Facades\Auth;

class SupplierDashboardController extends Controller
{
  public function index()
  {
    $supplierId = Auth::id();
    // Total produk milik supplier
    $totalProduk = Product::where('supplier_id', $supplierId)->count();

    // Produk sample (take 5)
    $produkSample = Product::where('supplier_id', $supplierId)
      ->where('stok', '>=', 5)
      ->orderBy('stok')
      ->get();

    // Contoh aktivitas (bisa ambil dari tabel log/produk, di sini hardcode contoh)
    $lastActivity = [
      'update_stok "Krispih Kentang" - 3 Juni 2025',
      'tambah produk "Brownies Coklat" - 1 Juni 2025',
      'edit harga "Pis Kaje" - 20 Mei 2025',
    ];

    return view('supplier.dashboard', compact('totalProduk', 'produkSample', 'lastActivity'));
  }
}

```

SupplierDashboardController bertugas menampilkan informasi penting pada **dashboard supplier**, termasuk total produk yang dimiliki, produk dengan stok menipis, dan riwayat aktivitas terakhir (sebagai informasi ringkas).

15. supplier product controllers

```
// class SupplierProductController extends Controller
// list produk supplier (dengan filter, search, paginasi)
public function index(Request $request)
{
    $query = Product::where('supplier_id', auth()->id()->id())->with('category', 'images');
    if ($request->filled('q')) {
        $query->where('nama_produk', 'like', '%' . $request->q . '%');
    }
    if ($request->filled('category_id')) {
        $query->where('category_id', $request->category_id);
    }
    $products = $query->orderByDesc('created_at')->paginate(10);
    $categories = Category::all();

    return view('supplier-products-product_list', compact('products', 'categories'));
}

// form tambah produk supplier
public function create()
{
    $categories = Category::all();
    return view('supplier-products-product_create', compact('categories'));
}

// Simpan produk supplier (tidak langsung tampil di admin)
public function store(Request $request)
{
    $request->validate([
        'nama_produk' => 'required|string|max:100',
        'harga_jual' => 'required|numeric|min:0',
        'stok' => 'required|integer|min:0',
        'category_id' => 'required|exists:categories,id',
        'deskripsi' => 'nullable|string|max:255',
        'foto.*' => 'nullable|image|max:2048',
    ]);

    $data = $request->only('nama_produk', 'harga_jual', 'stok', 'category_id', 'deskripsi');
    $data['supplier_id'] = auth()->id();
    $data['notif_admin_name'] = 1; // default tidak notif admin
    $data['is_approved'] = 0; // butuh approval admin

    $product = Product::create($data);
}
```

Controller ini mengelola seluruh proses CRUD produk oleh supplier, termasuk fitur tambahan seperti filter, pencarian, paginasi, serta penawaran produk ke toko/admin untuk ditinjau. Data produk yang ditambahkan tidak langsung muncul di toko utama karena membutuhkan persetujuan admin, dan mendukung unggahan multi-foto serta validasi input secara menyeluruh.

16. category models

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Category extends Model
{
    protected $fillable = [
        'nama_kategori',
        'deskripsi'
    ];

    public function products()
    {
        return $this->hasMany(Product::class, 'category_id');
    }
}
```

Model `Category` berfungsi untuk merepresentasikan data kategori produk dalam sistem. Model ini menyimpan informasi seperti nama kategori dan deskripsi, serta memiliki relasi one-to-many dengan model `Product`, artinya satu kategori bisa memiliki banyak produk. Melalui relasi [products\(\)](#), kita dapat mengakses seluruh produk yang termasuk dalam suatu kategori secara efisien menggunakan Eloquent ORM di Laravel.

17. customer models

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Customer extends Model
{
    protected $fillable = [
        'user_id',
        'nama_customer',
        'alamat',
        'no_hp',
        'foto'
    ];

    public function user()
    {
        return $this->belongsTo(User::class);
    }
}
```

Model `Customer` digunakan untuk merepresentasikan data pelanggan dalam sistem. Model ini menyimpan informasi seperti ID pengguna, nama customer, alamat, nomor HP, dan foto. Model ini juga memiliki relasi many-to-one dengan model `User`, yang berarti setiap customer terhubung ke satu akun user melalui relasi [user\(\)](#). Relasi ini memungkinkan pengambilan data pengguna terkait secara efisien dalam aplikasi Laravel.

18. mitra models

```

<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Mitra extends Model
{
    protected $fillable = [
        'user_id',
        'nama_mitra',
        'nama_usaha',
        'no_hp',
        'alamat',
        'foto'
    ];

    public function user()
    {
        return $this->belongsTo(User::class);
    }
}

```

Model **Mitra** berfungsi untuk merepresentasikan data mitra dalam sistem, termasuk identitas pribadi dan usaha, serta relasinya dengan pengguna yang terdaftar pada model **User**.

19. orders models

```

<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasMany;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Support\Facades\Log;

class Order extends Model
{
    protected $table = 'orders';

    // Nilai yang boleh diisi manual
    protected $fillable = [
        'user_id',
        'type_order',
        'tanggal_order',
        'total_order',
        'status_order',
        'alamat_kirim',
        'catatan',
        'alamat_order_id'
    ];

    // Relasi ke User
    public function user(): BelongsTo
    {
        return $this->belongsTo(User::class, 'user_id');
    }

    // Relasi ke Item pesanan
    public function item(): HasMany
    {
        return $this->hasMany(OrderItem::class, 'order_id', 'id');
    }

    // Relasi ke pembayaran
    public function payment(): HasOne
    {
        return $this->hasOne(Payment::class, 'order_id', 'id');
    }

    // Method untuk mengurangi stok produk ketika order selesai
    public function reduceProductStock()
    {
        // ...
    }
}

```

```

        $product->save();

        // Partition the stock items
        if ($product->stock < 0) {
            $product->stock = 0;
        }

        $product->save();

        // Log untuk tracking
        log('Info: [Stock Reduction] Product stock reduced [Order Completed]', [
            'product_id' => $product->id,
            'product_name' => $product->name_produk,
            'qty_sold' => $item->qty,
            'stock_remaining' => $product->stock,
            'order_id' => $item->id,
            'whitetrans_order_id' => $item->whitetrans_order_id,
            'trigger' => "Status changed to shipped"
        ]);
    }
}

// Method untuk mengembalikan stok produk (jika diperlukan untuk cancel/return)
public function restoreProductStock()
{
    foreach ($this->items as $item) {
        $product = $item->product;
        if ($product) {
            // Condition untuk cancel qty yang dibeli
            $product->stock += $item->qty;
            $product->save();

            // Log untuk tracking
            log('Info: [Stock Restoration] Product stock restored', [
                'product_id' => $product->id,
                'product_name' => $product->name_produk,
                'qty_restored' => $item->qty,
                'stock_after_restore' => $product->stock,
                'order_id' => $item->id,
                'whitetrans_order_id' => $item->whitetrans_order_id
            ]);
        }
    }
}

```

Model `Order` digunakan untuk merepresentasikan data pesanan, mencakup informasi pelanggan, status, dan detail pengiriman. Model ini juga memiliki relasi ke item pesanan, pembayaran, serta menyediakan metode untuk mengurangi atau mengembalikan stok produk sesuai perubahan status order.

20. order items models

```

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class OrderItem extends Model
{
    protected $table = 'order_items';

    protected $fillable = [
        'order_id',
        'product_id',
        'qty',
        'harga_jual',
        'subtotal',
    ];

    public function order(): BelongsTo
    {
        return $this->belongsTo(Order::class, 'order_id', 'id');
    }

    public function product(): BelongsTo
    {
        return $this->belongsTo(Product::class, 'product_id', 'id');
    }
}

```

Model `OrderItem` merepresentasikan detail dari setiap item dalam sebuah pesanan, mencakup produk yang dipesan, jumlah, harga jual, dan subtotal, serta membentuk relasi ke model `Order` dan `Product` untuk menghubungkan item dengan pesanan induk dan produk terkait.

21. payment models

```
<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Payment extends Model
{
    protected $table = 'payments';
    protected $fillable = [
        'order_id',
        'midtrans_order_id',
        'metode_bayar',
        'jumlah_bayar',
        'status_bayar',
        'waktu_bayar',
        'payload_midtrans'
    ];

    public function order()
    {
        return $this->belongsTo(Order::class, 'order_id');
    }
}
```

Model `Payment` merepresentasikan data pembayaran untuk setiap pesanan, termasuk metode dan jumlah pembayaran, status, waktu bayar, serta informasi dari Midtrans, dan terhubung dengan model `Order` untuk relasi ke pesanan terkait.

22. product models

```
<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    protected $fillable = [
        'supplier_id',
        'category_id',
        'kode_produk',
        'nama_produk',
        'harga_beli',
        'harga_jual',
        'stok',
        'deskripsi',
        'rental_admin_view',
        'is_approved'
    ];

    public function category()
    {
        return $this->belongsTo(App\Models\Category::class, 'category_id', 'id');
    }

    public function supplier()
    {
        return $this->belongsTo(Supplier::class, 'supplier_id');
    }

    public function images()
    {
        return $this->hasMany(ProductImage::class);
    }

    public function order_items()
    {
        return $this->hasMany(App\Models\OrderItem::class, 'product_id');
    }
}
```

Model `Product` merepresentasikan data produk yang ditawarkan oleh supplier, termasuk informasi seperti kategori, harga, stok, dan status persetujuan admin, serta memiliki relasi ke kategori, supplier, gambar produk, dan item pesanan melalui relasi `belongsTo` dan `hasMany`.

23. purchase models

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;

class Purchase extends Model
{
    protected $fillable = [
        'supplier_id',
        'tanggal_beli',
        'total_beli',
        'status_bayar',
        'bukti_bayar'
    ];

    public function supplier()
    {
        return $this->belongsTo(Supplier::class);
    }

    public function items()
    {
        return $this->hasMany(PurchaseItem::class);
    }
}
```

Model `Purchase` merepresentasikan transaksi pembelian produk dari supplier, menyimpan informasi seperti tanggal beli, total harga, status pembayaran, dan bukti pembayaran, serta memiliki relasi ke model `Supplier` (sebagai pemilik transaksi) dan `PurchaseItem` (sebagai daftar item yang dibeli).

24. purchase item models


```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class PurchaseItem extends Model
{
    protected $fillable = [
        'purchase_id',
        'product_id',
        'qty',
        'harga_satuan',
        'subtotal'
    ];

    public function purchase()
    {
        return $this->belongsTo(Purchase::class);
    }

    public function product()
    {
        return $this->belongsTo(Product::class);
    }
}

```

Model `PurchaseItem` digunakan untuk merepresentasikan detail dari setiap item yang dibeli dalam transaksi pembelian, mencatat informasi seperti produk yang dibeli, jumlah, harga satuan, dan subtotal, serta memiliki relasi dengan model `Purchase` (sebagai induk transaksi) dan `Product` (sebagai barang yang dibeli).

25. supplier models

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Supplier extends Model
{
    protected $fillable = [
        'user_id',
        'nama_supplier',
        'perusahaan',
        'no_hp',
        'alamat',
        'foto'
    ];

    public function user()
    {
        return $this->belongsTo(User::class);
    }

    public function products()
    {
        return $this->hasMany(Product::class);
    }
}

```

Model `Supplier` digunakan untuk merepresentasikan data pemasok dalam sistem, mencakup informasi seperti identitas pengguna, nama, perusahaan, kontak, alamat, dan foto, serta memiliki relasi `belongsTo` ke model `User` dan `hasMany` ke model `Product` yang menunjukkan bahwa satu supplier dapat menyediakan banyak produk.

26. [users](#) models

```

class User extends Authenticatable
{
    use HasFactory, Notifiable;

    // Tentukan kolom yang bisa diisi pesan
    protected $fillable = [
        'name',
        'email',
        'password',
        'role', // DITAMBAHKAN
        'is_active', // DITAMBAHKAN
    ];

    // Tentukan saat toforget()/TIDAK
    protected $hidden = [
        'password',
        'remember_token',
    ];

    // Casting tipe data
    protected $casts = [
        'email_verified_at' => 'datetime',
        'is_active' => 'boolean', // Tambahkan
    ];

    // Contoh relasi (optional, jika ada di register)
    public function supplier()
    {
        return $this->hasOne(\App\Models\Supplier::class);
    }
    public function mitra()
    {
        return $this->hasOne(\App\Models\Mitra::class);
    }
    public function customer()
    {
        return $this->hasOne(\App\Models\Customer::class);
    }
}

```

Model `User` berfungsi sebagai representasi pengguna sistem secara umum, mencakup autentikasi, data identitas, peran pengguna (`role`), status aktif, dan relasi satu-ke-satu dengan model `Supplier`, `Mitra`, dan `Customer` untuk membedakan hak akses dan fitur berdasarkan jenis pengguna.

Lampiran 5. Sertifikat HKI


REPUBLIK INDONESIA
KEMENTERIAN HUKUM

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC-002025100122, 28 Juli 2025

Pencipta
Nama : Fadillah Surya Irawan, Hepatika Zidny Hnasdina, S.Pd., M.Kom.
dkk
Alamat : Desa Sindang, RT/RW 002/005, Dukuhwaru, Kab. Tegal, Jawa Tengah, 52451
Kewarganegaraan : Indonesia

Pemegang Hak Cipta
Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama
Alamat : Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana, Margadana, Kota Tegal, Jawa Tengah, 52142
Kewarganegaraan : Indonesia
Jenis Ciptaan : Program Komputer
Judul Ciptaan : Sistem Manajemen Untuk Efisiensi Ketersediaan Stok Barang Pada Toko Aby Snack Menggunakan Laravel

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 28 Juli 2025, di Kota Tegal

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor Pencatatan : 000940383

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b
Direktur Hak Cipta dan Desain Industri

Ageng Damarasongko,SH_MH
NIP. 196912261994031001



 **Digital Signatures and Electronic Certificates**

Disclaimer:
1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan data yang tertera, Menteri berwenang untuk mencabut surat pencatatan permohonan.
2. Surat Pencatatan ini telah diunggah secara elektronik menggunakan segel elektronik yang diterbitkan oleh Badan Penyelenggara Sertifikasi Elektronik, Badan Siber dan Sandi Negara.
3. Surat Pencatatan ini dapat dibuktikan keabsahannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.

Lampiran 6. Lembar Bimbingan



D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Fadillah Surya Irawan
NIM : 21090030
No. Ponsel : 089524174289
Judul TA : Rancang Bangun Sistem Untuk Efisiensi
Stock Ketersediaan Barang Pada Studi Kasus
Toko Aby Snack Menggunakan Laravel

Dosen Pembimbing II : Arif Hidayah, S.Tr.Kom., M.Tr.Kom.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
	16/6/2025	- App.	- Ditanya utu lulus ke ini //	
	7/7/2025	- App	- Fitur Profil - Fitur Payment - Breadcrumbs	
	11/7/2025	- App	- Pembagian Midtrans - Transaksi & Pembayaran Jalur 1!	
	19/7/2025	- App	- Payment Midtrans Selesai wallet	

16/2015 7	- App	- Paymen Sematkan waktu Durasinya	
18/2015 7	- App	BETULKAN BUG-NYA	
25/2015 7	- Laporan	- Perbaiki Samai paster k' file	

Tegal,
Dosen Pembimbing II



Arif Hidayat, S.Tr.Kom., M.Tr.Kom.
NIPY. 10.024.548



**D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Fadillah Surya Irawan
NIM : 21090030
No. Ponsel : 089524174289
Judul TA : Rancang Bangun Sistem Untuk Efisiensi Stock
Ketersediaan Barang Pada Studi Kasus Toko Aby
Snack Menggunakan Laravel

Dosen Pembimbing I : Hepatika Zidny Ilmadina, S.Pd., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	12/06/25	x aplikasi	x tole ada q : admin toko ; toko distributor ; supplier pembeli x toko distributor ⇒ membeli barang di toko besar x supplier ⇒ upload barang dagangan ⇒ ↓ buat skema menu tiap user	
2.		x aplikasi	x alamat y pengiriman ada limitasi hanya kota Tegal dan sekitarnya	

3.	18/07/25	*aplikasi	*pengiriman sudah sesuai dg alamat hanya dr kota Tegal *hantutkan ke btkas Hki	~
----	----------	-----------	--	---

Tegal,
Dosen Pembimbing I



Hepatika Zidny Ilmadina, S.Pd., Mkom.
NIPY. 08.017.340

Lampiran 7. Lembar Data Produk

Nama Produk	Harga (Rp)	Estimasi Berat / Ukuran
Pilus Kacang Jumbo	85.000	2 kg
Keripik Pisang	8.000	250 gram
Kue Kering Putri Salju	35.000	350 gram
Cookies Almond Cheese Coklat	40.000	350 gram
Cookies Cheese Kress Regular	35.000	350 gram
Cookies Milky Nastar	40.000	350 gram
Nastar Keju Regular	40.000	350 gram
Kastengel Regular	40.000	350 gram
Kastengel Mocaf Regular	45.000	350 gram
Putri Cheese Nut Regular	40.000	350 gram
Lidah Kucing Keju Kecil	30.000	200 gram
Lidah Kucing Keju Besar	45.000	400 gram
Nastar Besar	65.000	650 gram
Corn Flakes Kecil	35.000	300 gram
Corn Flakes Besar	60.000	600 gram
Soes Kering Keju (Besar)	25.000	300 gram
Soes Kering Keju (Kecil)	15.000	150 gram
Keripik Talas	20.000	300 gram
Stik Balado	35.000	500 gram
Pangsit Kering	40.000	500 gram
Makaroni	25.000	300 gram
Emping	30.000	300 gram
Keripik Tempe	35.000	500 gram
Keripik Tahu	35.000	500 gram
Kerupuk Stik Ikan	50.000	750 gram
Kerupuk Tenggiri	50.000	750 gram
Kuping Gajah	30.000	350 gram
Pastel Abon	35.000	350 gram
Stik Keju	25.000	250 gram
Ring Keju	35.000	350 gram
Untir-Untir	40.000	500 gram

Lampiran 8. Lembar Observasi

