

LAMPIRAN

Lampiran 1. Surat Kesepakatan Bimbingan Skripsi

SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Dika Dzikri Pratama
NIM : 21090121
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Dyah Apriliani, S.T., M.Kom.
Status : Dosen
NIDN : 0614049002
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata III-D

Pada hari ini Jum'at tanggal 07 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Tugas Akhir Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Tugas Akhir 2 minggu sekali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

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

Tegal, 07 Maret 2025

Pihak Pertama



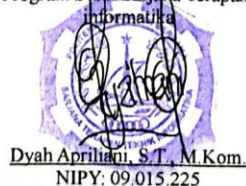
Dika Dzikri Pratama

Pihak Kedua



Dyah Apriliani, S.T., M.Kom.

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik
Informatika



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

SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Dika Dzikri Pratama
NIM : 21090121
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : M. Nishom, S.Kom., M.Kom.
Status : Dosen
NIDN : 0619048701
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata / III/C

Pada hari ini Kamis tanggal 13 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Tugas Akhir 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, 13 Maret 2025

Pihak Pertama



Dika Dzikri Pratama

Pihak Kedua



M. Nishom, S.Kom., M.Kom.

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik
Informatika



Dyah Apriyanti, S.T., M.Kom.
NIPN 09.015.225

Lampiran 2. Surat Pernyataan Pengajuan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : Dika Dzikri Pratama
Kewarganegaraan : Indonesia
Alamat : Dukuh Kedawon RT 02 RW 06, Kecamatan Larangan,
Kabupaten Brebes, 52262
2. Nama : Dyah Apriliani, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Perumahan Mutiara Indah Slawi, Jalan Nangka No.2, Procot,
Procot Blok C4 SLAWI SLAWI JAWA TENGAH.
3. Nama : M. Nishom, S.Kom., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jl. Jepara, Perumahan Griya Putri Land Blok A6, RT 03 RW
04, Margadana, Tegal, 52143

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : Sistem Informasi Penjualan dan Persediaan Barang Elektronik Untuk
Pengelolaan Stok Berbasis Website (Studi Kasus Toko Wayang Elektronik
Dukuh Kedawon)
 - 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;

CS Dipindai dengan CamScanner

- 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 perkara 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 23 Juni 2025



Dika Dzikri Pratama
Pemegang Hak Cipta*



Dyah Apriliani, S.T., M.Kom.
Pemegang Hak Cipta*



M. Nishom, S.Kom., M.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. Nama : Dika Dzikri Pratama
Kewarganegaraan : Indonesia
Alamat : Dukuh Kedawon RT 02 RW 06, Kecamatan Larangan, Kabupaten Brebes,
52262
2. Nama : Dyah Apriliani, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Perumahan Mutiara Indah Slawi, Jalan Nangka No.2, Procot, Procot Blok C4
SLAWI SLAWI JAWA TENGAH.
3. Nama : M. Nishom, S.Kom., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jl. Jepara, Perumahan Griya Putri Land Blok A6, RT 03 RW 04, Margadana,
Tegal, 52143

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

Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M)
Politeknik Harapan Bersama
Alamat : Jl. Mataram No.9, Pesurungan Lor, Kec. Margadana,
Kota Tegal, Provinsi Jawa Tengah, 52147

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul
“Sistem Informasi Penjualan dan Persediaan Barang Elektronik Untuk Pengelolaan
Stok Berbasis Website (Studi Kasus Toko Wayang Elektronik Dukuh Kedawon)” untuk
didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan
Intellectual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

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

Tegal, 24 Juni 2025

Pemegang Hak Cipta

Kepala P3M

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

Pencipta



(Dika Dzikri Pratama)

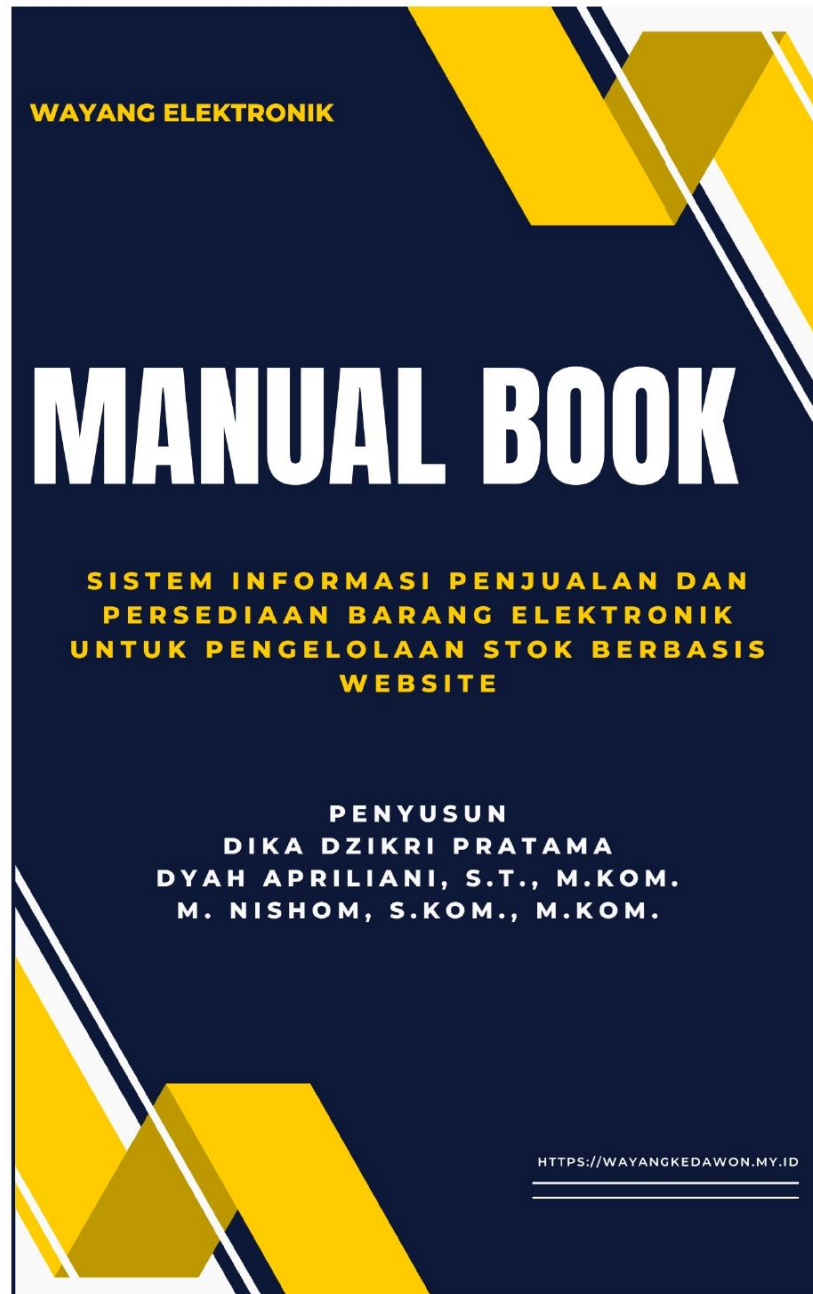


(Dyah Apriani, S.T., M.Kom.)



(M. Nishom, S.Kom., M.Kom.)

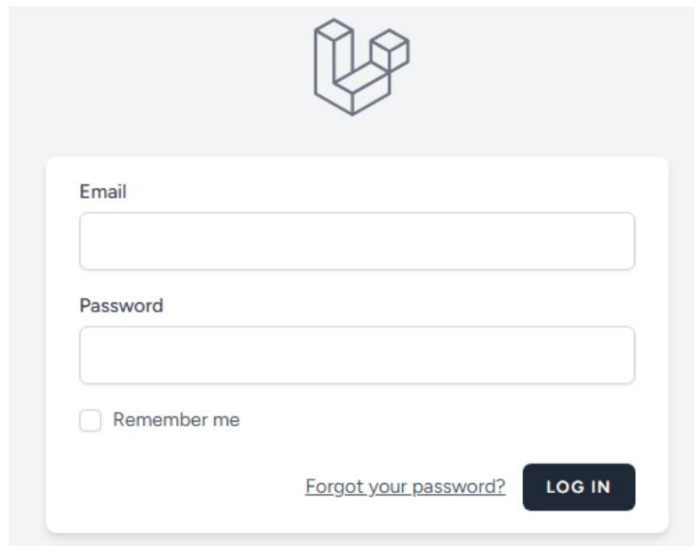
Lampiran 4. Manual Book dan Dokumen Teknikal



MANUAL BOOK PENGGUNAAN WEBSITE

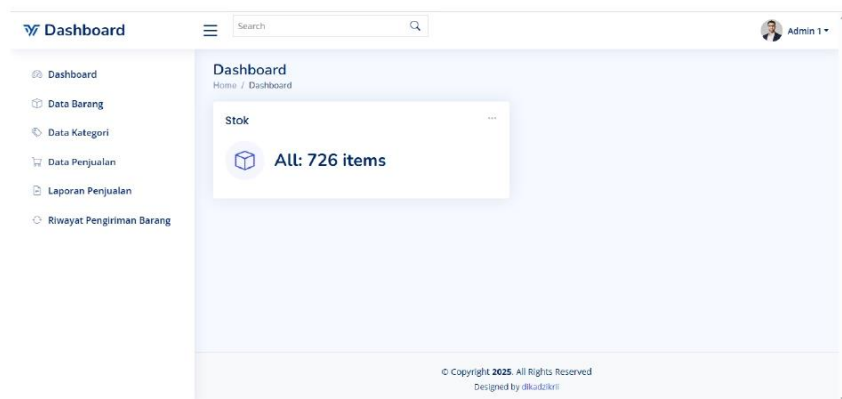
Login

- Buka Browser
- Ketikkan alamat <https://wayangedawon.my.id>
- Isikan username dan password lalu tekan login



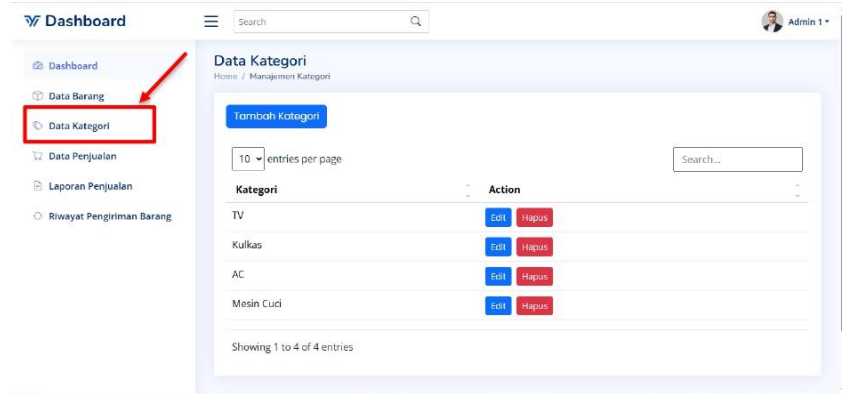
The login form is centered on a light gray background. At the top, there is a logo consisting of three stacked cubes. Below the logo, the form contains two input fields: 'Email' and 'Password'. Under the 'Password' field, there is a checkbox labeled 'Remember me'. At the bottom right of the form, there is a link that says 'Forgot your password?' and a dark blue button labeled 'LOG IN'.

Jika berhasil login, maka akan diarahkan ke halaman dashboard



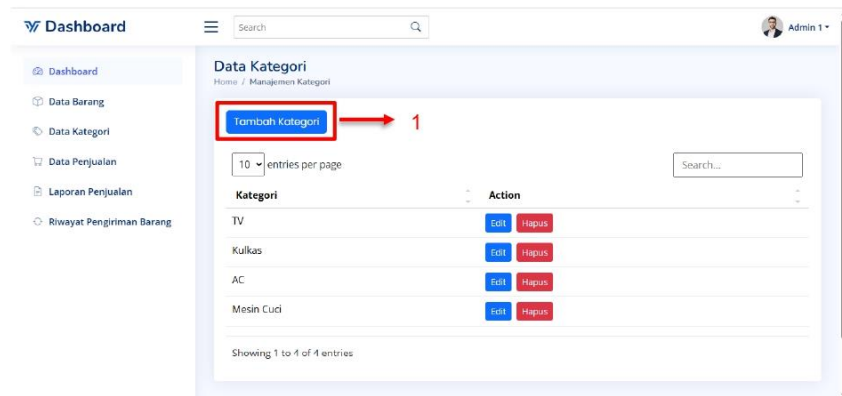
Modul Kategori

Untuk masuk ke Data Kategori, silakan klik Kategori seperti terlihat pada gambar berikut ini.



Untuk menambahkan kategori bisa ikuti langkah dibawah:

- Tambah kategori
- Input nama kategori barang
- Kemudian klik submit



Dashboard

Home / Manajemen Stok / Tambah Kategori

Tambah Kategori Baru

Kategori Baru

Masukkan Kategori Baru

Submit Kembali

© Copyright 2025. All Rights Reserved
Designed by dkkadzikri

Untuk mengubah kategori:

- Klik edit data yang akan diubah
- Ubah data agar sesuai
- Klik submit

Dashboard

Home / Manajemen Kategori

Data Kategori

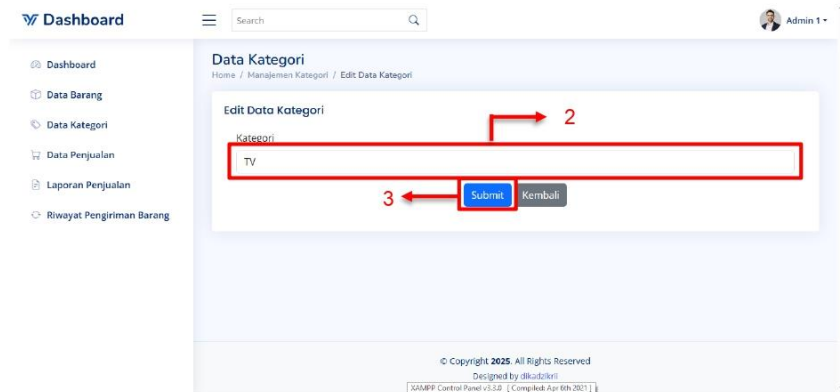
Tambah kategori

10 entries per page

Search...

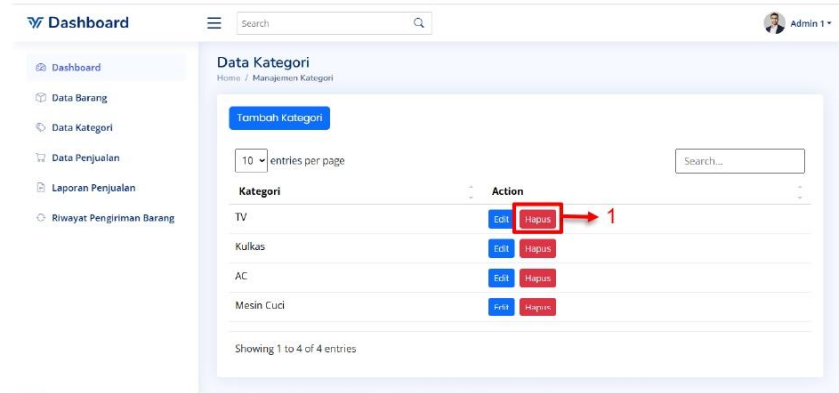
Kategori	Action
TV	Edit Hapus
Kulkas	Edit Hapus
AC	Edit Hapus
Mesin Cuci	Edit Hapus

Showing 1 to 4 of 4 entries



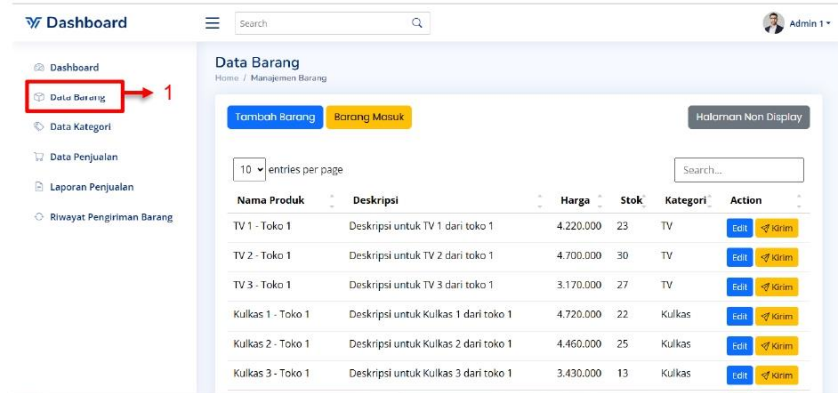
Untuk menghapus data kategori:

- Klik hapus, pada data yang akan dihapus



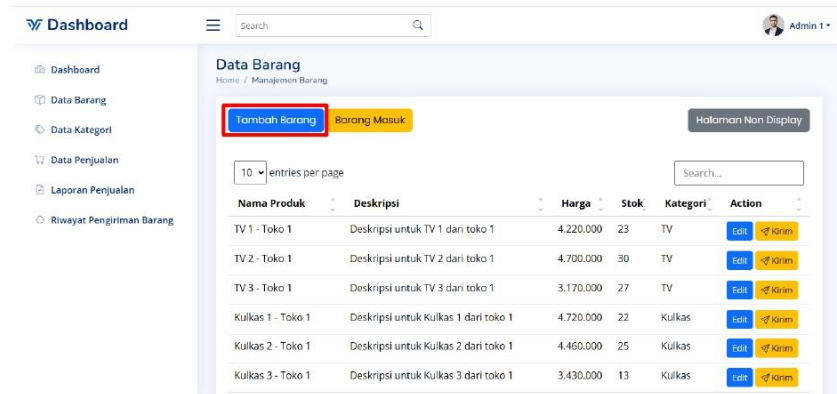
Modul Barang

Untuk membuka halaman bisa klik data barang



Untuk menambahkan data barang:

- Klik tambah barang, kemudian
- Isi form
- Klik submit



Dashboard Search Admin 1

- Dashboard
- Data Barang
- Data Kategori
- Data Penjualan
- Laporan Penjualan
- Riwayat Pengiriman Barang

Nama Barang
Sharp SJ-X196MG

Deskripsi
Dekripsi A

Harga Beli
2300000

Harga
2800000

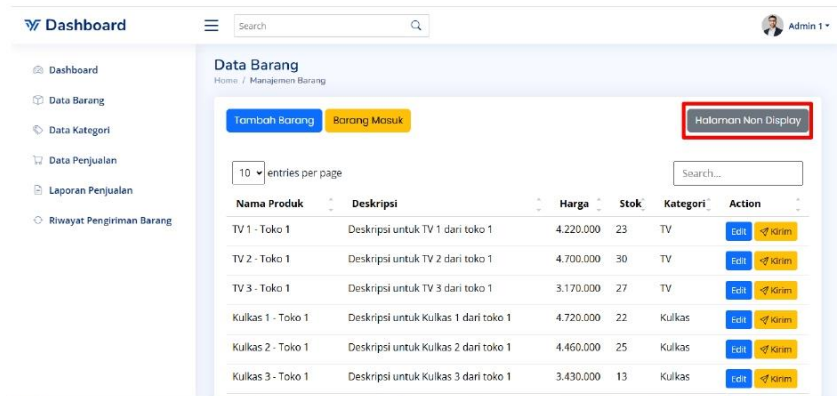
Stok
12

Kategori
Mesin Cuci

3 Simpan Kembali

RAMPP Corp. | Email: info@rampp.com | Copyright © 2021

Untuk tombol Halaman Non-Display berfungsi untuk data barang memiliki 0 stok atau tidak ada stok tercatat
Kemudian terdapat tombol Barang Masuk yang berfungsi untuk mengecek barang yang dikirim dari toko cabang



Dashboard | Search | Admin 1

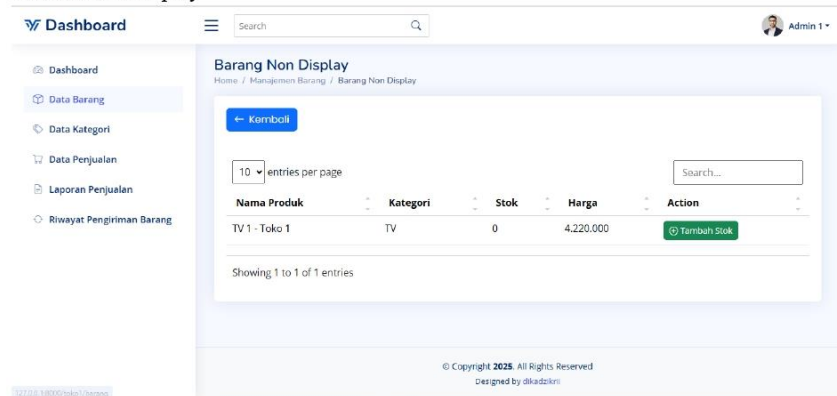
Data Barang
Home / Manajemen Barang

Tambah Barang | Barang Masuk | **Halaman Non Display**

10 entries per page | Search...

Nama Produk	Deskripsi	Harga	Stok	Kategori	Action
TV 1 - Toko 1	Deskripsi untuk TV 1 dari toko 1	4.220.000	23	TV	Edit Kirim
TV 2 - Toko 1	Deskripsi untuk TV 2 dari toko 1	4.700.000	30	TV	Edit Kirim
TV 3 - Toko 1	Deskripsi untuk TV 3 dari toko 1	3.170.000	27	TV	Edit Kirim
Kulkas 1 - Toko 1	Deskripsi untuk Kulkas 1 dari toko 1	4.720.000	22	Kulkas	Edit Kirim
Kulkas 2 - Toko 1	Deskripsi untuk Kulkas 2 dari toko 1	4.460.000	25	Kulkas	Edit Kirim
Kulkas 3 - Toko 1	Deskripsi untuk Kulkas 3 dari toko 1	3.430.000	13	Kulkas	Edit Kirim

Halaman non display



Dashboard | Search | Admin 1

Barang Non Display
Home / Manajemen Barang / Barang Non Display

Kembali

10 entries per page | Search...

Nama Produk	Kategori	Stok	Harga	Action
TV 1 - Toko 1	TV	0	4.220.000	Tambah Stok

Showing 1 to 1 of 1 entries

© Copyright 2025. All Rights Reserved
Designed by dkkadzikri

Untuk menambahkan kembali stok barang yang non display

- Klik tambah stok
- Perbarui jumlah stok
- Klik submit

Dashboard

Search

Admin 1

Dashboard

Data Barang

Data Kategori

Data Penjualan

Laporan Penjualan

Riwayat Pengiriman Barang

Barang Non Display

Home / Manajemen Barang / Barang Non Display

Kembali

10 entries per page

Search...

Nama Produk	Kategori	Stok	Harga	Action
TV 1 - Toko 1	TV	0	4,220.000	Tambah Stok

Showing 1 to 1 of 1 entries

© Copyright 2025. All Rights Reserved
Designed by dkkadikri

Dashboard

Search

Admin 1

Dashboard

Data Barang

Data Kategori

Data Penjualan

Laporan Penjualan

Riwayat Pengiriman Barang

Nama Barang

TV 1 - Toko 1

Deskripsi

Deskripsi untuk TV 1 dari toko 1

Harga Beli

2660000,00

Harga

4220000,00

Stok

0

Kategori

Pilih Kategori

Submit

Kembali

Untuk edit barang

- Klik edit pada data barang
- Perbarui data
- Klik submit

Dashboard

Search

Admin 1

Data Barang

Home / Manajemen Barang

Tambah Barang Barang Masuk Halaman Non Display

10 entries per page Search...

Nama Produk	Deskripsi	Harga	Stok	Kategori	Action
TV 1 - Toko 1	Deskripsi untuk TV 1 dari toko 1	4.220.000	23	TV	1 Edit Kirim
TV 2 - Toko 1	Deskripsi untuk TV 2 dari toko 1	4.700.000	30	TV	Edit Kirim
TV 3 - Toko 1	Deskripsi untuk TV 3 dari toko 1	3.170.000	27	TV	Edit Kirim
Kulkas 1 - Toko 1	Deskripsi untuk Kulkas 1 dari toko 1	4.720.000	22	Kulkas	Edit Kirim
Kulkas 2 - Toko 1	Deskripsi untuk Kulkas 2 dari toko 1	4.460.000	25	Kulkas	Edit Kirim
Kulkas 3 - Toko 1	Deskripsi untuk Kulkas 3 dari toko 1	3.430.000	13	Kulkas	Edit Kirim

Dashboard

Search

Admin 1

2

Nama Barang

TV 1 - Toko 1

Deskripsi

Deskripsi untuk TV 1 dari toko 1

Harga Beli

2660000,00

Harga

4220000,00

Stok

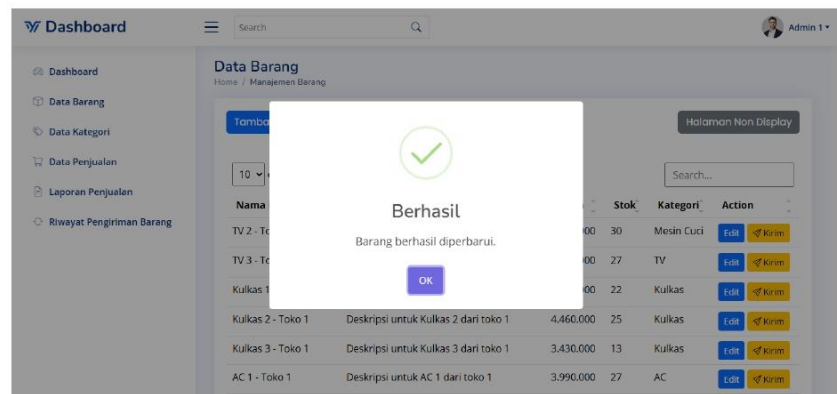
0

Kategori

Pilih Kategori

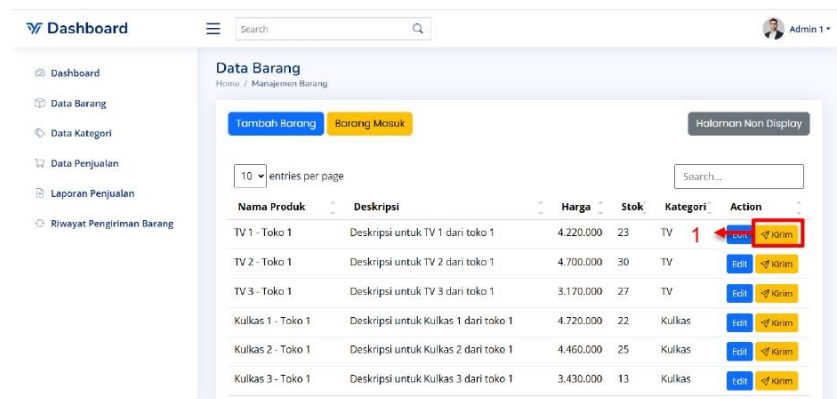
3

Submit Kembali



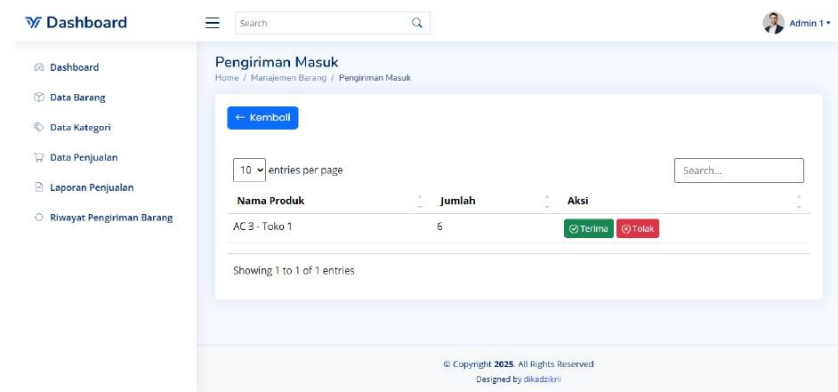
Kirim barang

- Klik data barang
- Isi form
- Klik kirim



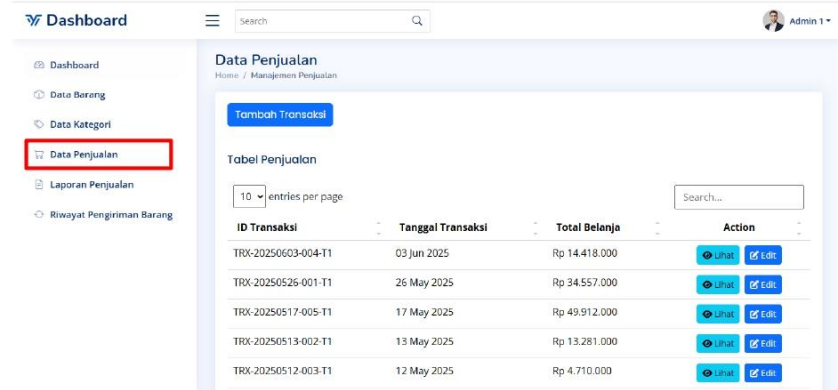
Halaman barang masuk

Untuk menerima barang bisa klik terima, maka data barang dari toko asal akan berkurang dan data barang di toko tujuan akan bertambah(jika data barang sudah ada). Jika klik tolak maka data barang di toko asal akan kembali seperti semula



Modul Penjualan

Untuk membuka halaman klik data penjualan

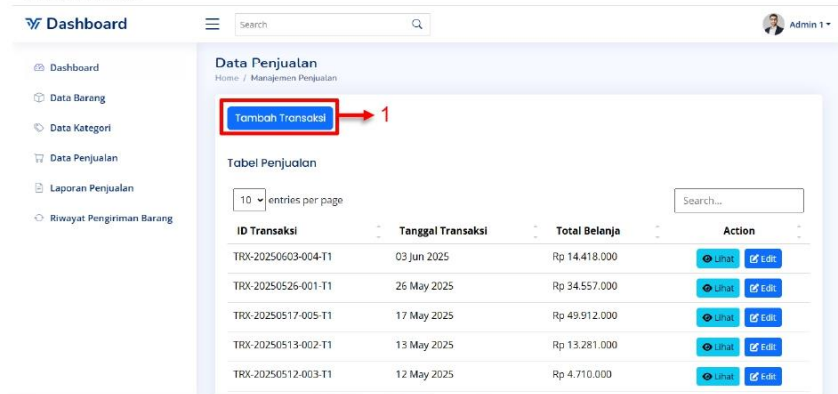


The screenshot shows a web application interface for 'Data Penjualan'. On the left is a sidebar menu with options: Dashboard, Data Bereng, Data Kategori, Data Penjualan (highlighted with a red box), Laporan Penjualan, and Riwayat Pengiriman Barang. The main content area has a 'Tambah Transaksi' button at the top. Below it is a table titled 'Tabel Penjualan' with columns: ID Transaksi, Tanggal Transaksi, Total Belanja, and Action. The table contains five rows of transaction data. A search bar and a dropdown for 'entries per page' (set to 10) are located above the table.

ID Transaksi	Tanggal Transaksi	Total Belanja	Action
TRX-20250603-004-T1	03 Jun 2025	Rp 14.418.000	Ubat Edit
TRX-20250525-001-T1	26 May 2025	Rp 34.557.000	Ubat Edit
TRX-20250517-005-T1	17 May 2025	Rp 49.912.000	Ubat Edit
TRX-20250513-002-T1	13 May 2025	Rp 13.281.000	Ubat Edit
TRX-20250512-003-T1	12 May 2025	Rp 4.710.000	Ubat Edit

Untuk menambah data penjualan:

- Klik tambah transaksi
- Pilih nama produk
- Pilih jumlah
- Jika lebih dari 1 klik Tambah Produk
- Jika barang perlu dikirim, isi biaya pengiriman dan alamat
- Klik Simpan



This screenshot is identical to the one above, but with a red box around the 'Tambah Transaksi' button and a red arrow pointing to it, labeled with the number '1'.

Dashboard

Home / Manajemen Penjualan / Tambah Data Penjualan

Tambah Produk

1

Mesin Cuci 2 - Toko 1 (Stok: 28)

3360000,00

1

Rp 3.360.000

2

TV 1 - Toko 1 (Stok: 23)

4990000,00

2

Rp 9.980.000

3

Pilih Produk

Rp 0

Total Produk

Rp 13.340.000

Biaya Pengiriman

0

Alamat Pengiriman

Tanggal Transaksi *

12/06/2025

Total Keseluruhan

Rp 13.340.000

Simpan

Kembali

Edit data penjualan:

- Klik edit
- Perbarui data yang diperlukan
- Isi alasan kenapa diedit
- Klik simpan

Dashboard

Home / Manajemen Penjualan

Tambah Transaksi

10 entries per page

Search...

ID Transaksi	Tanggal Transaksi	Total Belanja	Action
TRX-20250603-004-T1	03 Jun 2025	Rp 14.418.000	
TRX-20250526-001-T1	26 May 2025	Rp 34.557.000	
TRX-20250517-005-T1	17 May 2025	Rp 49.912.000	
TRX-20250513-002-T1	13 May 2025	Rp 13.281.000	
TRX-20250512-003-T1	12 May 2025	Rp 4.710.000	

D-16

Dashboard

Dashboard

Data Barang

Data Kategori

Data Penjualan

Laporan Penjualan

Riwayat Pengiriman Barang

Search

Admin 1

Edit Data Penjualan

Home / Manajemen Penjualan / Edit Data Penjualan

Detail Produk

+ Tambah Produk

No	Produk	Harga	Jumlah	Subtotal	Aksi
1	Mesin Cuci 3 - Toko 1 (Stok: 23) Stok tersedia: 23	4360000,00	3	Rp 13.080.000	
2	Kulkas 1 - Toko 1 (Stok: 28) Stok tersedia: 28	4330000,00	5	Rp 21.650.000	
Total Produk				Rp 34.730.000	

Biaya Pengiriman

45290,00

Alamat Pengiriman

Psr. Peta No. 626, Malang 46807, Rizu

Biaya Pengiriman

35818,00

Alamat Pengiriman

Jln. Babadak No. 87, Bengkulu 45453, Pabar

Tanggal Transaksi *

hh/bb/tttt

Total Keseluruhan

Rp 25.235.818

Alasan Edit

3

Simpan Perubahan

Kembali

Lihat detail penjualan
- Klik icon mata

Dashboard

Dashboard

Data Barang

Data Kategori

Data Penjualan

Laporan Penjualan

Riwayat Pengiriman Barang

Search

Admin 1

Data Penjualan

Home / Manajemen Penjualan

Tambah Transaksi

10 entries per page

Search...

ID Transaksi	Tanggal Transaksi	Total Belanja	Action
TRX-20250603-004-T1	03 Jun 2025	Rp 14.418.000	1
TRX-20250526-001-T1	26 May 2025	Rp 34.557.000	
TRX-20250517-005-T1	17 May 2025	Rp 49.912.000	
TRX-20250513-002-T1	13 May 2025	Rp 13.281.000	
TRX-20250512-003-T1	12 May 2025	Rp 4.710.000	

D-17

Dashboard

Data Barang

Data Kategori

Data Penjualan

Laporan Penjualan

Riwayat Pengiriman Barang

search

Admin 1

Detail Transaksi TRX-20250526-001-T1

Export PDFPreview PDF

ID Transaksi: TRX-20250526-001-T1

Toko: toko1

Item Produk:

No	Produk	Kategori	Jumlah	Harga Satuan	Subtotal
1	Mesin Cuci 3 - Toko 1 <i>In husto ulkan amnis est qua sit.</i>	Mesin Cuci	3	Rp 4.184.000	Rp 12.552.000
2	Kulkas 1 - Toko 1 <i>Cum ante et qua velit efficit sed debitis.</i>	Kulkas	5	Rp 4.401.000	Rp 22.005.000
Total Harga Produk:					Rp 34.557.000

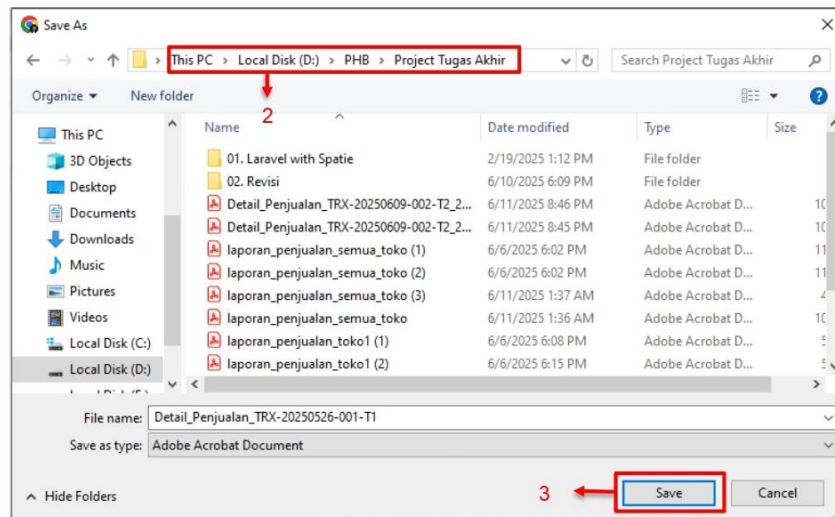
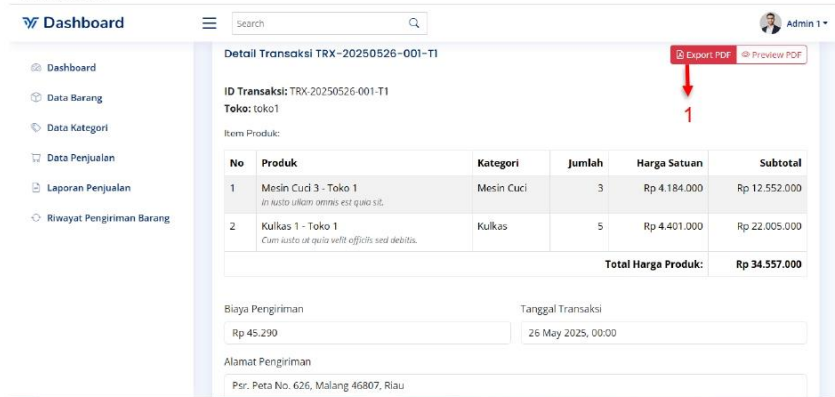
Biaya Pengiriman
Rp 45.290

Tanggal Transaksi
26 May 2025, 00:00

Alamat Pengiriman
Psr: Peta No. 626, Malang 46807, Riau

Export detail penjualan pdf:

- Klik export PDF
- Pilih path
- Klik save



Preview pdf:
- Klik preview pdf

Dashboard

Data Barang

Data Kategori

Data Penjualan

Laporan Penjualan

Riwayat Pengiriman Barang

Search

Admin 1

Detail Transaksi TRX-20250526-001-T1

Export PDF

Preview PDF

ID Transaksi: TRX-20250526-001-T1

Toko: toko1

Item Produk:

No	Produk	Kategori	Jumlah	Harga Satuan	Subtotal
1	Mesin Cuci 3 - Toko 1 <i>in aucto ullam omnis est quia sit.</i>	Mesin Cuci	3	Rp 4.184.000	Rp 12.552.000
2	Kulkas 1 - Toko 1 <i>Cam iusto ut quia velit efficit sed debitis.</i>	Kulkas	5	Rp 4.401.000	Rp 22.005.000
Total Harga Produk:					Rp 34.557.000

Biaya Pengiriman

Rp 45.290

Tanggal Transaksi

26 May 2025, 00:00

Alamat Pengiriman

Per: Peta No. 626, Malang 46807, Riau

Detail Penjualan - TRX-20250526-001-T1

1 / 1

67%

DETAIL TRANSAKSI PENJUALAN

ID Transaksi: TRX-20250526-001-T1

Toko: toko1

Tanggal Transaksi: 26 May 2025, 00:00

Item Produk:

No	Produk	Kategori	Jumlah	Harga Satuan	Subtotal
1	Mesin Cuci 3 - Toko 1 <i>in aucto ullam omnis est quia sit.</i>	Mesin Cuci	3	Rp 4.184.000	Rp 12.552.000
2	Kulkas 1 - Toko 1 <i>Cam iusto ut quia velit efficit sed debitis.</i>	Kulkas	5	Rp 4.401.000	Rp 22.005.000
Total Harga Produk:					Rp 34.557.000

Biaya Pengiriman:

Rp 45.290

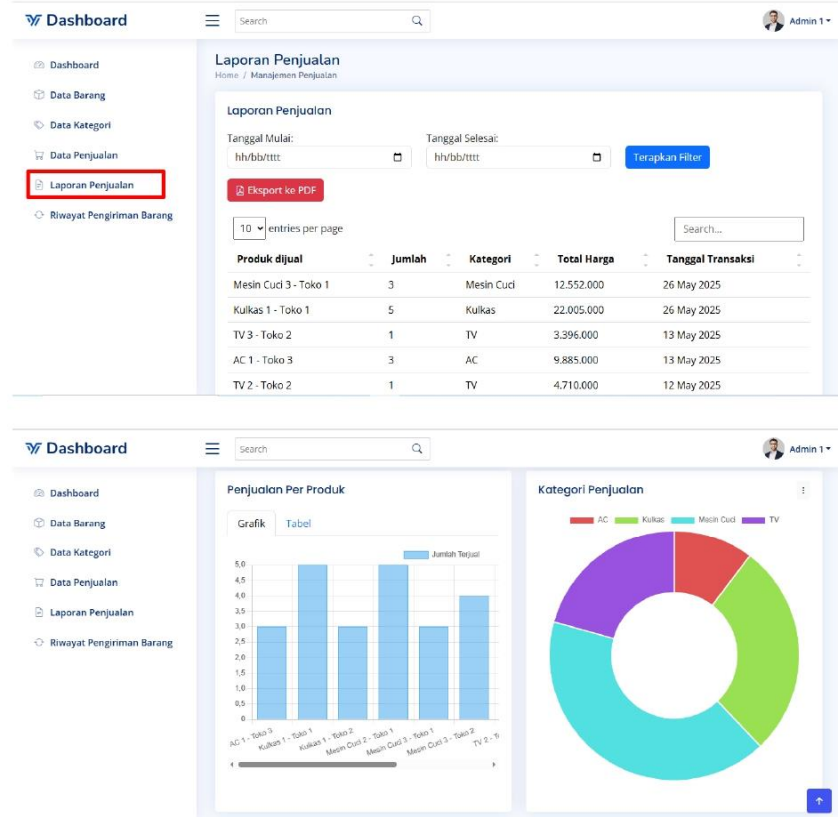
Alamat Pengiriman:

Per: Peta No. 626, Malang 46807, Riau

GRAND TOTAL: Rp 34.557.000

Modul Laporan Penjualan

Untuk membuka halaman laporan penjualan klik laporan penjualan



Export data penjualan:

- Pilih rentang tanggal
- Klik export pdf

Jika klik export pdf tanpa melakukan input rentang tanggal maka akan mengexport semua data penjualan

Dashboard

Laporan Penjualan

Tanggal Mulai: 01/05/2025 Tanggal Selesai: 13/06/2025

Terapkan Filter

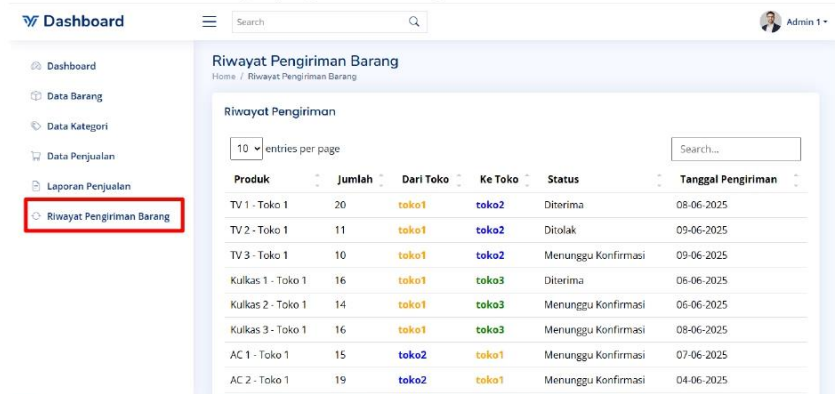
Ekspor ke PDF

10 entries per page

Produk dijual	Jumlah	Kategori	Total Harga	Tanggal Transaksi
Mesin Cuci 3 - Toko 1	3	Mesin Cuci	12.552.000	26 May 2025
Kulkas 1 - Toko 1	5	Kulkas	22.005.000	26 May 2025
TV 3 - Toko 2	1	TV	3.396.000	13 May 2025
AC 1 - Toko 3	3	AC	9.885.000	13 May 2025
TV 2 - Toko 2	1	TV	4.710.000	12 May 2025

Modul Riwayat Pengiriman

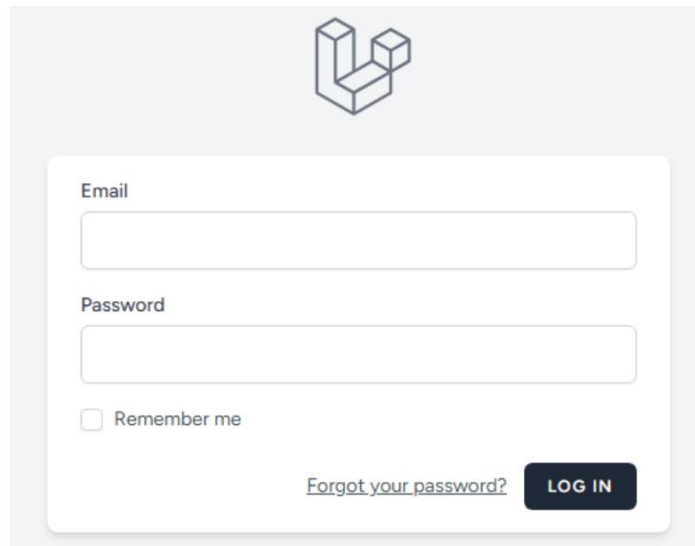
Untuk membuka halaman Riwayat pengiriman klik Riwayat Pengiriman Barang, halaman ini untuk melihat daftar Riwayat pengiriman barang antar toko



Produk	Jumlah	Dari Toko	Ke Toko	Status	Tanggal Pengiriman
TV 1 - Toko 1	20	toko1	toko2	Diterima	08-06-2025
TV 2 - Toko 1	11	toko1	toko2	Ditolak	09-06-2025
TV 3 - Toko 1	10	toko1	toko2	Menunggu Konfirmasi	09-06-2025
Kulkas 1 - Toko 1	16	toko1	toko3	Diterima	06-06-2025
Kulkas 2 - Toko 1	14	toko1	toko3	Menunggu Konfirmasi	06-06-2025
Kulkas 3 - Toko 1	16	toko1	toko3	Menunggu Konfirmasi	08-06-2025
AC 1 - Toko 1	15	toko2	toko1	Menunggu Konfirmasi	07-06-2025
AC 2 - Toko 1	19	toko2	toko1	Menunggu Konfirmasi	04-06-2025

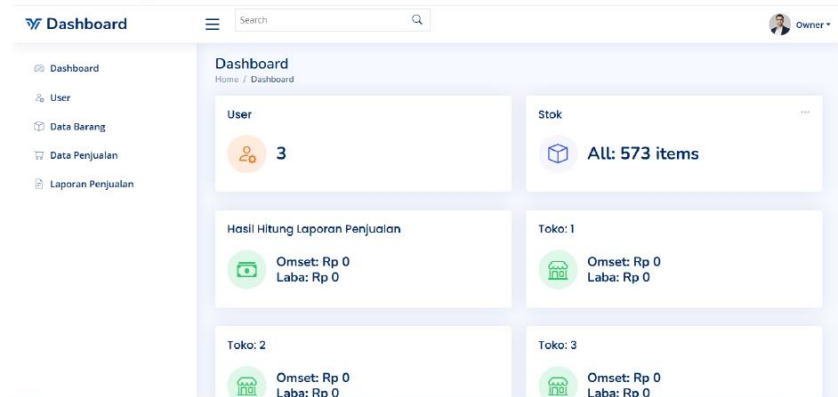
Login

- Login
- Ketikkan alamat <https://wayangedawon.my.id>
- Isikan username dan password lalu tekan login



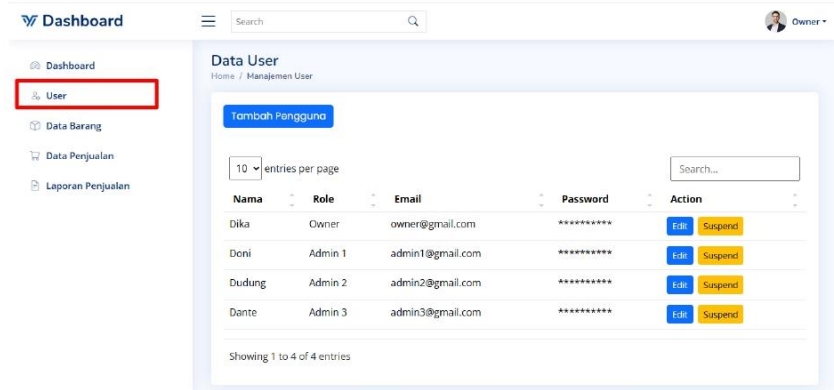
The login form is centered on a light gray background. At the top is a logo consisting of three stacked cubes. Below the logo is a white rectangular box containing the login fields. The 'Email' field is a simple text input. The 'Password' field is a text input with a small eye icon to toggle visibility. Below these fields is a checkbox labeled 'Remember me'. At the bottom right of the box is a dark blue 'LOG IN' button. To the left of the button is a link that says 'Forgot your password?'.

Jika berhasil login maka akan diarahkan ke halaman dashboard



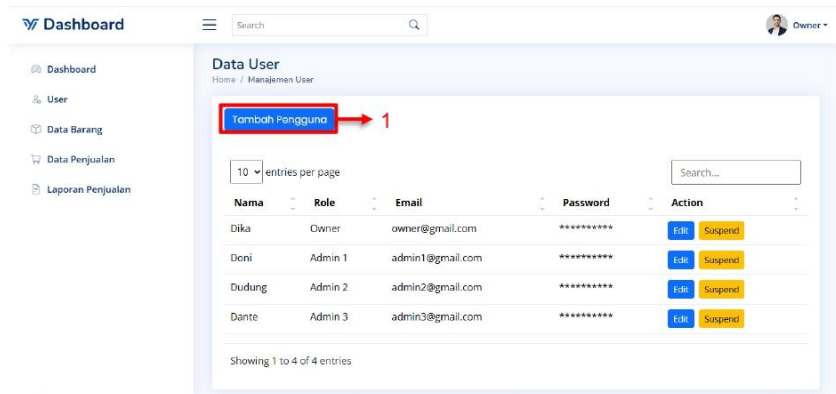
Modul User

Untuk membuka halaman user klik menu user



Tambah user

- Klik tambah pengguna
- Masukkan nama
- Pilih role
- Tambahkan email
- Masukkan password
- Klik submit



Dashboard Search Owner

Data Pengguna
Home / Tambah Data Pengguna

Tambah Data Pengguna

Nama
Fajar → 2

Role
Pilih Role → 3

Pilih Role

Admin 1

Admin 2

Admin 3

Masukkan Password

Submit **Kembali**

Dashboard Search Owner

Data Pengguna
Home / Tambah Data Pengguna

Tambah Data Pengguna

Nama
Fajar

Role
Admin 2

Email
fajar@gmail.com → 4

Password
***** → 5

6 ← **Submit** **Kembali**

Edit user

- Pilih data user yang akan diedit
- Ubah data yang perlu diedit
- Klik simpan

The top screenshot shows the 'Data User' page. It has a sidebar with links to Dashboard, User, Data Barang, Data Penjualan, and Laporan Penjualan. The main content area shows a table of users with columns: Nama, Role, Email, Password, and Action. The table contains four rows of data. The 'Action' column has 'Edit' and 'Suspend' buttons for each user. A red box highlights the 'Edit' button for the user 'Dudung'.

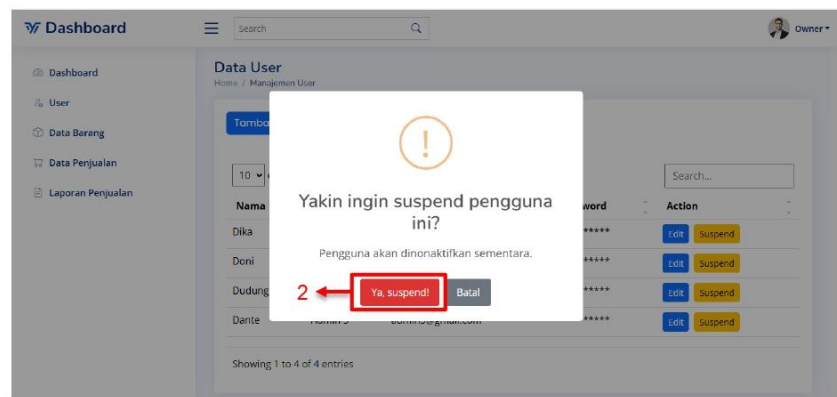
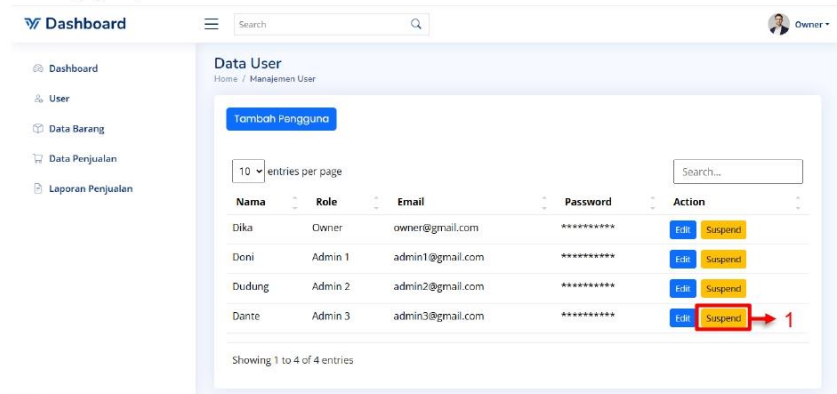
Nama	Role	Email	Password	Action
Dika	Owner	owner@gmail.com	*****	Edit Suspend
Doni	Admin 1	admin1@gmail.com	*****	Edit Suspend
Dudung	Admin 2	admin2@gmail.com	*****	Edit Suspend
Dante	Admin 3	admin3@gmail.com	*****	Edit Suspend

The bottom screenshot shows the 'Edit User' page. It has the same sidebar. The main content area shows a form to edit a user. The form has fields for Nama, Role, Email, and Password. The 'Nama' field is filled with 'Dudung', 'Role' with 'Admin 2', and 'Email' with 'admin2@gmail.com'. The 'Password' field is empty. A red box highlights the 'Update' button.

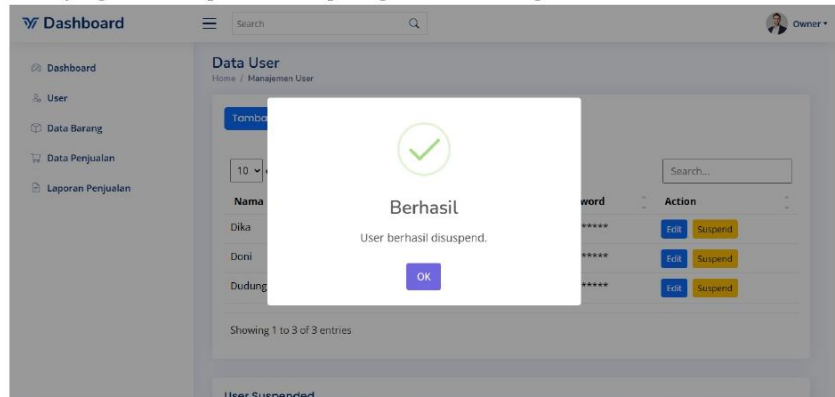
1 → [Edit](#) → 2 → [Update](#) → 3 → [Kembali](#)

Suspend user

- Klik suspend pada akun yang akan disuspend
- Klik iya, suspend



Akun yang telah disuspend tidak dapat digunakan untuk login



Mengaktifkan akun yang disuspend

- Klik aktifkan pada akun yang disuspend
- Klik “Ya, aktifkan”

The first screenshot shows the 'User Suspended' table with one entry: Dante (Admin 3, admin3@gmail.com, Suspended). The 'Aktifkan' button is highlighted with a red box and a red arrow labeled '1'.

The second screenshot shows a confirmation dialog box: 'Aktifkan kembali pengguna? Pengguna akan kembali bisa login dan mengakses sistem.' The 'Ya, aktifkan!' button is highlighted with a red box and a red arrow labeled '2'.

The third screenshot shows the 'Data User' table with four entries. A success message dialog box is displayed: 'Berhasil User berhasil diaktifkan kembali.' The 'Aktifkan' button for Dante is highlighted with a red box.

Nama	Role	Email	Status	Action
Dante	Admin 3	admin3@gmail.com	Suspended	Aktifkan

Showing 1 to 1 of 1 entries

© Copyright 2025. All Rights Reserved
Designed by dikaditkrii

Dashboard

Showing 1 to 3 of 3 entries

User Su...

10 entries per page

Search...

Nama

Dante

Status

Suspended

Action

Aktifkan

Showing

2

Ya, aktifkan!

Batal

© Copyright 2025. All Rights Reserved
Designed by dikaditkrii

Dashboard

Home / Manajemen User

Tambah

10 entries per page

Search...

Nama

Dika

Doni

Dudung

Dante

Admin 3

admin3@gmail.com

word

Action

edit Suspend

edit Suspend

edit Suspend

edit Suspend

Showing 1 to 4 of 4 entries

Berhasil

User berhasil diaktifkan kembali.

OK

Modul Data Barang

Membuka halaman data barang

- Klik data barang

Dashboard | Search | Owner

Data Barang
Home / Manajemen Barang

Data Barang

5 entries per page | Search...

Nama Produk	Deskripsi	Harga	Stok	Kategori	Toko
TV 1 - Toko 1	Deskripsi untuk TV 1 dari toko 1	4.990.000	23	TV	1
TV 2 - Toko 1	Deskripsi untuk TV 2 dari toko 1	3.290.000	18	TV	1
TV 3 - Toko 1	Deskripsi untuk TV 3 dari toko 1	3.550.000	13	TV	1
Kulkas 1 - Toko 1	Deskripsi untuk Kulkas 1 dari toko 1	4.330.000	28	Kulkas	1
Kulkas 2 - Toko 1	Deskripsi untuk Kulkas 2 dari toko 1	3.930.000	23	Kulkas	1

Showing 1 to 5 of 36 entries | 1 2 3 4 5 6 7 8 >

Dashboard | Search | Owner

Data Barang
Home / Manajemen Barang

Data Barang

5 entries per page | Search...

Nama Produk	Deskripsi	Harga	Stok	Kategori	Toko
AC 3 - Toko 2	Deskripsi untuk AC 3 dari toko 2	3.140.000	13	AC	2
Mesin Cuci 1 - Toko 2	Deskripsi untuk Mesin Cuci 1 dari toko 2	4.530.000	14	Mesin Cuci	2
Mesin Cuci 2 - Toko 2	Deskripsi untuk Mesin Cuci 2 dari toko 2	3.760.000	16	Mesin Cuci	2
Mesin Cuci 3 - Toko 2	Deskripsi untuk Mesin Cuci 3 dari toko 2	3.970.000	12	Mesin Cuci	2
TV 1 - Toko 3	Deskripsi untuk TV 1 dari toko 3	4.100.000	22	TV	3

Showing 21 to 25 of 36 entries | < 1 2 3 4 5 6 7 8 >

Modul Data Penjualan

Untuk membuka data penjualan bisa klik Data Penjualan, untuk melihat detail data penjualan
- Klik lihat

The image shows two screenshots of a web application interface for managing sales data.

Top Screenshot: Data Penjualan Dashboard

The dashboard has a sidebar with the following menu items: Dashboard, User, Data Barang, **Data Penjualan** (highlighted with a red box), and Laporan Penjualan. The main content area is titled "Data Penjualan" and contains a table labeled "Tabel Penjualan". The table has columns: ID Transaksi, Tanggal Transaksi, Total Belanja, Toko, and Action. There is a search bar and a dropdown for "entries per page" set to 10. A red box highlights the "lihat" (view) button in the Action column for the transaction with ID TRX-20250603-004-T1, which is also indicated by a red arrow and the number "1".

ID Transaksi	Tanggal Transaksi	Total Belanja	Toko	Action
TRX-20250609-002-T2	09 Jun 2025	Rp 36,298.000	2	lihat
TRX-20250608-004-T3	08 Jun 2025	Rp 40,226.000	3	lihat
TRX-20250608-002-T3	08 Jun 2025	Rp 36,643.000	3	lihat
TRX-20250607-001-T2	07 Jun 2025	Rp 53,900.000	2	lihat
TRX-20250603-004-T1	03 Jun 2025	Rp 14,418.000	1	lihat
TRX-20250530-003-T2	30 May 2025	Rp 39,929.000	2	lihat
TRX-20250526-001-T1	26 May 2025	Rp 34,557.000	1	lihat

Bottom Screenshot: Detail Penjualan

The detailed view shows the transaction "Detail Transaksi TRX-20250608-004-T3". It includes buttons for "Export PDF" and "Preview PDF". The transaction details are: ID Transaksi: TRX-20250608-004-T3, Toko: toko3. Below this is a table for "Item Produk":

No	Produk	Kategori	Jumlah	Harga Satuan	Subtotal
1	Mesin Cuci 1 - Toko 2 <i>Qui itaque voluptatem dolorem optio quo et blanditiis.</i>	Mesin Cuci	5	Rp 4,744.000	Rp 23,720.000
2	AC 3 - Toko 3 <i>Et inventore aut modi est.</i>	AC	2	Rp 3,334.000	Rp 6,668.000
3	TV 1 - Toko 2 <i>Aut provident distinctio quia facere rerum non sequi excepturi.</i>	TV	2	Rp 4,919.000	Rp 9,838.000
Total Harga Produk:					Rp 40,226.000

Dashboard
User
Data Barang
Data Penjualan
Laporan Penjualan

Search

Owner

3

TV 1 - Toko 2

TV

2

Rp 4.919.000

Rp 9.838.000

Total Harga Produk: Rp 40.226.000

Biaya Pengiriman

Rp 43.834

Tanggal Transaksi

08 Jun 2025, 00:00

Alamat Pengiriman

Dk. Kiaracandong No. 556, Palopo 26664, Sultra

Grand Total

Rp 40.226.000

Kembali

Untuk mencetak ke pdf klik export pdf

Dashboard
User
Data Barang
Data Penjualan
Laporan Penjualan

Search

Owner

Detail Penjualan

Home / Manajemen Penjualan / Detail Penjualan

Detail Transaksi TRX-20250608-004-T3

ID Transaksi: TRX-20250608-004-T3

Toko: toko3

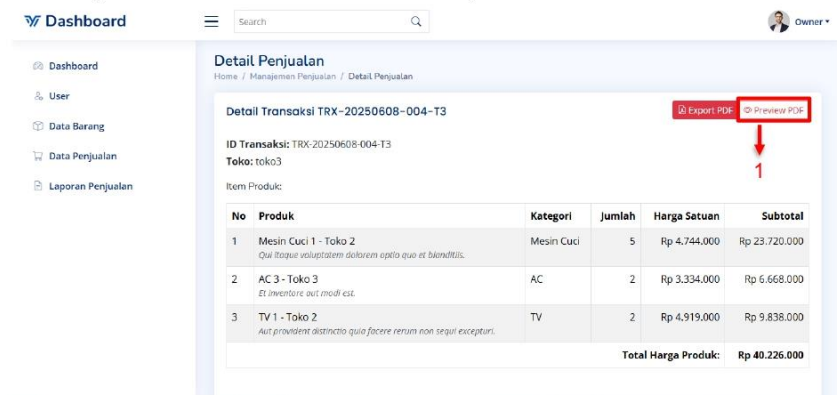
Item Produk:

No	Produk	Kategori	Jumlah	Harga Satuan	Subtotal
1	Mesin Cuci 1 - Toko 2	Mesin Cuci	5	Rp 4.744.000	Rp 23.720.000
2	AC 3 - Toko 3	AC	2	Rp 3.334.000	Rp 6.668.000
3	TV 1 - Toko 2	TV	2	Rp 4.919.000	Rp 9.838.000
Total Harga Produk:					Rp 40.226.000

Export PDF

Preview PDF

Jika mau preview sebelum dicetak kedalam bentuk pdf bisa klik Preview PDF



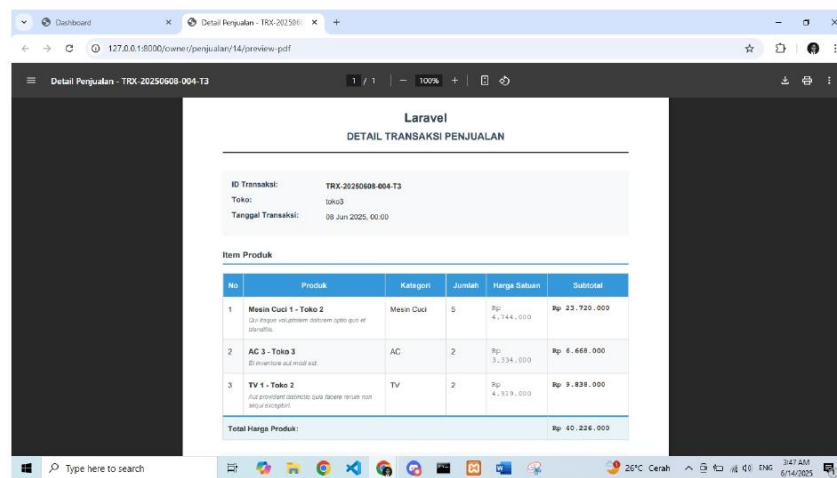
Detail Penjualan
Home / Manajemen Penjualan / Detail Penjualan

Detail Transaksi TRX-20250608-004-T3

ID Transaksi: TRX-20250608-004-T3
Toko: toko3

Item Produk:

No	Produk	Kategori	Jumlah	Harga Satuan	Subtotal
1	Mesin Cuci 1 - Toko 2 <i>Qui itaque voluptatem dolorem optio quo et blanditiis.</i>	Mesin Cuci	5	Rp 4.744.000	Rp 23.720.000
2	AC 3 - Toko 3 <i>Et inventore aut modi est.</i>	AC	2	Rp 3.334.000	Rp 6.668.000
3	TV 1 - Toko 2 <i>Aut provident distinctio quia facere rerum non sequi excepturi.</i>	TV	2	Rp 4.919.000	Rp 9.838.000
Total Harga Produk:					Rp 40.226.000



Dashboard x Detail Penjualan - TRX-20250608-004-T3 x +

127.0.0.1:8000/owner/penjualan/14/preview-pdf

Laravel
DETAIL TRANSAKSI PENJUALAN

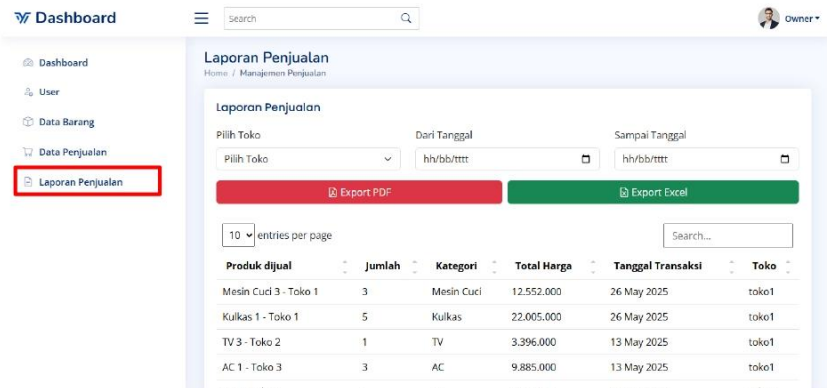
ID Transaksi: TRX-20250608-004-T3
Toko: toko3
Tanggal Transaksi: 08 Jun 2025, 00:00

Item Produk

No	Produk	Kategori	Jumlah	Harga Satuan	Subtotal
1	Mesin Cuci 1 - Toko 2 <i>Qui itaque voluptatem dolorem optio quo et blanditiis.</i>	Mesin Cuci	5	Rp 4.744.000	Rp 23.720.000
2	AC 3 - Toko 3 <i>Et inventore aut modi est.</i>	AC	2	Rp 3.334.000	Rp 6.668.000
3	TV 1 - Toko 2 <i>Aut provident distinctio quia facere rerum non sequi excepturi.</i>	TV	2	Rp 4.919.000	Rp 9.838.000
Total Harga Produk:					Rp 40.226.000

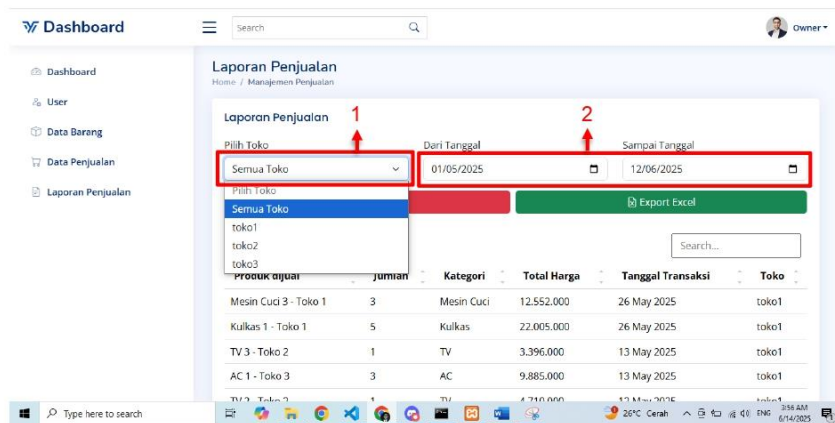
Modul Laporan Penjualan

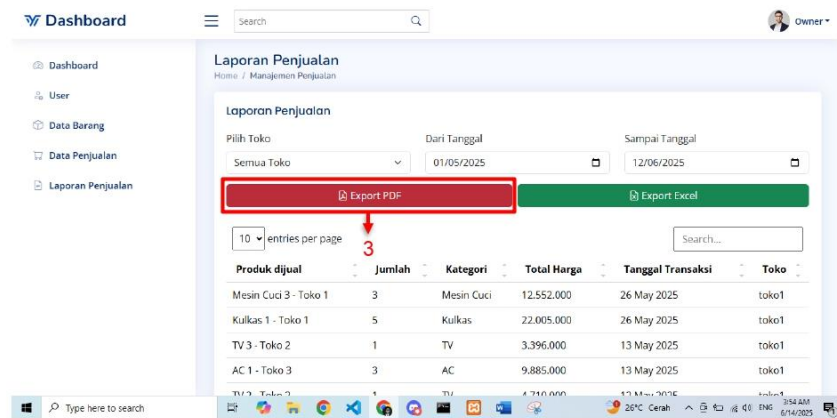
Untuk membuka halaman laporan penjualan klik menu laporan penjualan



Mengexport dalam bentuk pdf

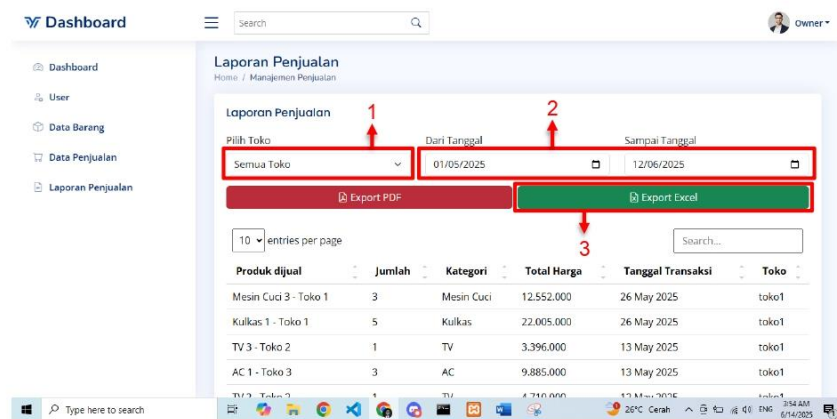
- Pilih toko yang mau diexport
- Pilih rentang tanggal
- Klik export pdf





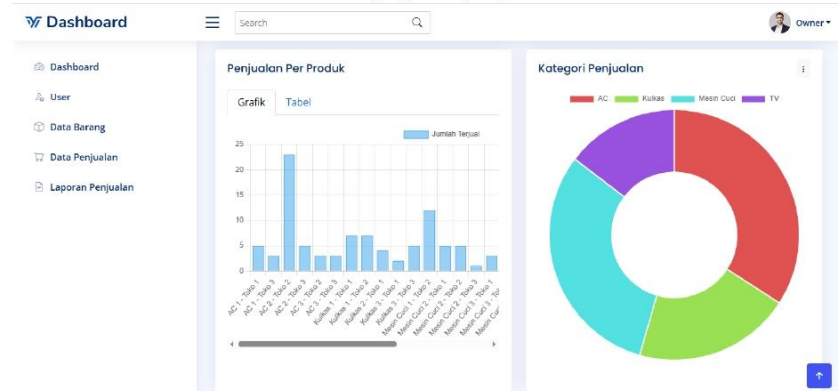
Mengexport dalam bentuk excel

- Pilih toko yang mau diexport
- Pilih rentang tanggal
- Klik export excel

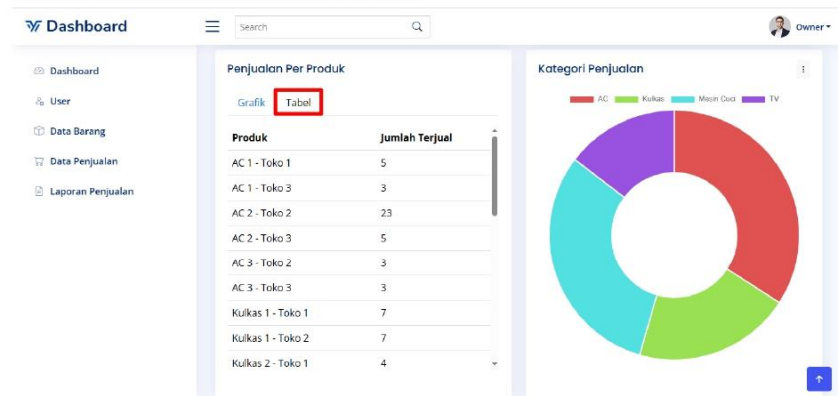


Cek grafik

Geser scroll bar ke kanan untuk melihat penjualan per produk



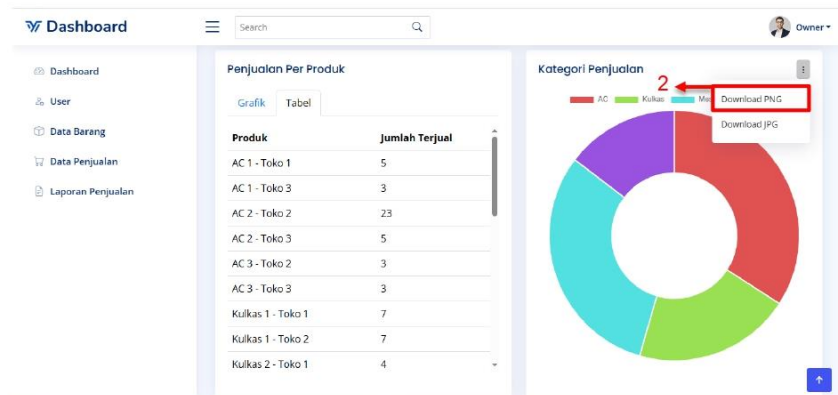
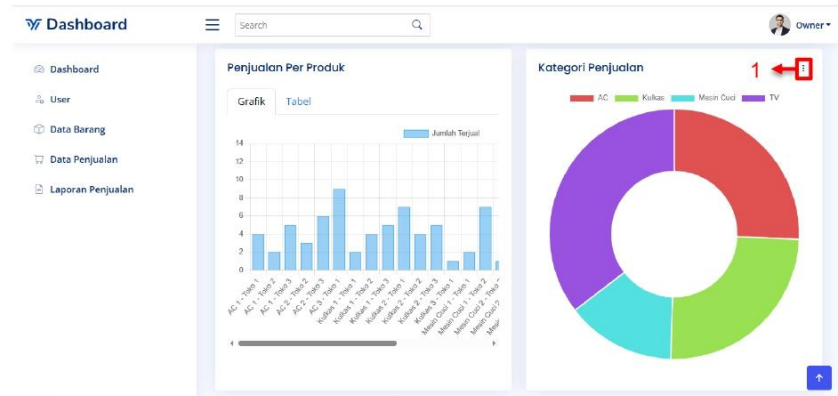
Jika mau dilihat dalam bentuk tabel klik tabel



Export donut chart

- Klik titik tiga

- Pilih download JPG



Hitung omset dan laba

- Pilih toko
- Pilih rentang tanggal
- Klik hitung

Dashboard

Search

Owner

Dashboard

User

Data Barang

Data Penjualan

Laporan Penjualan

Laporan Omset, Laba, dan Rugi (Livewire)

Pilih Toko

Semua Toko

Dari Tanggal

01/06/2025

Sampai Tanggal

30/06/2025

Hitung

1

2

3

© Copyright 2025. All Rights Reserved
Designed by dikadzikri

Dashboard

Search

Owner

Dashboard

User

Data Barang

Data Penjualan

Laporan Penjualan

Laporan Omset, Laba, dan Rugi (Livewire)

Pilih Toko

Semua Toko

Dari Tanggal

20/05/2025

Sampai Tanggal

30/06/2025

Hitung

toko1

Periode	Omset (Rp)	Total Biaya (Rp)	Laba (Rp)	Rugi (Rp)
26 May 2025	Rp 34.557.000,00	Rp 13.100.000,00	Rp 21.457.000,00	Rp 0,00
03 Jun 2025	Rp 14.418.000,00	Rp 6.420.000,00	Rp 7.998.000,00	Rp 0,00
Total	Rp 48.975.000,00		Rp 29.455.000,00	

toko2

Periode	Omset (Rp)	Total Biaya (Rp)	Laba (Rp)	Rugi (Rp)
---------	------------	------------------	-----------	-----------

Dashboard

search

Owner

Dashboard

User

Data Barang

Data Penjualan

Laporan Penjualan

toko2

Periode	Omset (Rp)	Total Biaya (Rp)	Laba (Rp)	Rugi (Rp)
30 May 2025	Rp 39.929.000,00	Rp 11.520.000,00	Rp 28.409.000,00	Rp 0,00
07 Jun 2025	Rp 53.900.000,00	Rp 17.640.000,00	Rp 36.260.000,00	Rp 0,00
09 Jun 2025	Rp 36.298.000,00	Rp 16.690.000,00	Rp 19.608.000,00	Rp 0,00
Total	Rp 130.127.000,00		Rp 84.277.000,00	

toko3

Periode	Omset (Rp)	Total Biaya (Rp)	Laba (Rp)	Rugi (Rp)
21 May 2025	Rp 55.817.000,00	Rp 29.030.000,00	Rp 26.787.000,00	Rp 0,00
08 Jun 2025	Rp 76.869.000,00	Rp 37.990.000,00	Rp 38.879.000,00	Rp 0,00
Total	Rp 132.686.000,00		Rp 65.666.000,00	

Laporan berhasil dihitung dan dashboard telah diperbarui!

© Copyright 2025. All Rights Reserved
Designed by dikadikini

Hasil perhitungan akan ditampilkan dimenu dashboard

Dashboard

Search

Owner

Dashboard

User

Data Barang

Data Penjualan

Laporan Penjualan

User

3

Stok

All: 573 items

Hasil Hitung Laporan Penjualan

Omset: Rp 311.788.000

Laba: Rp 179.398.000

Toko: 1

Omset: Rp 48.975.000

Laba: Rp 29.455.000

Toko: 2

Omset: Rp 130.127.000

Laba: Rp 84.277.000

Toko: 3

Omset: Rp 132.686.000

Laba: Rp 65.666.000

WAYANG ELEKTRONIK

DOKUMEN TEKNIKAL

**SISTEM INFORMASI PENJUALAN DAN
PERSEDIAAN BARANG ELEKTRONIK
UNTUK PENGELOLAAN STOK BERBASIS
WEBSITE**

**PENYUSUN
DIKA DZIKRI PRATAMA
DYAH APRILIANI, S.T., M.KOM.
M. NISHOM, S.KOM., M.KOM.**

Authenticated Session Controller

Fungsi create untuk menampilkan login form

```
public function create(): View
{
    return view('auth.login');
}
```

Fungsi store untuk memproses autentikasi pengguna

```
public function store(LoginRequest $request): RedirectResponse
{
    $request->authenticate();
    $request->session()->regenerate();
    $user = Auth::user();

    if ($user->status === 'suspend') {
        Auth::logout();
        return redirect()->route('login')->withErrors([
            'email' => 'Akun Anda telah disuspend. Silakan hubungi owner.',
        ]);
    }

    if ($user->hasRole('Owner')) {
        return redirect()->to('/owner/dashboard');
    }
    if ($user->hasRole('Admin 1')) {
        return redirect()->to('/toko1/dashboard');
    }
    if ($user->hasRole('Admin 2')) {
        return redirect()->to('/toko2/dashboard');
    }
    if ($user->hasRole('Admin 3')) {
        return redirect()->to('/toko3/dashboard');
    }

    // Fallback
    return redirect()->intended(RouteServiceProvider::HOME);
}
```

User Controller

Fungsi index digunakan untuk mengambil semua data pengguna dari database, lalu memisahkan pengguna berdasarkan status: aktif dan suspend. Data pengguna tersebut kemudian dikirim ke tampilan data-user untuk ditampilkan secara terpisah berdasarkan statusnya.

```
public function index()
{
    $users = User::all();

    $usersAktif = User::where('status', 'aktif')->get();
    $usersSuspend = User::where('status', 'suspend')->get();

    return view('pages.user.data-user', compact('usersAktif', 'usersSuspend'));
}
```

Fungsi create digunakan untuk menampilkan halaman form tambah user dengan terlebih dahulu mengambil semua data role yang tersedia. Role ini akan digunakan untuk menentukan peran pengguna saat pendaftaran dilakukan.

```
public function create()
{
    $roles = Role::all();
    return view('pages.user.tambah-user', compact('roles'));
}
```

Fungsi store digunakan untuk menangani proses penyimpanan user baru ke database. Setelah melakukan validasi input, data user disimpan ke tabel users dengan password yang sudah di-hash, lalu role yang dipilih langsung diberikan ke user melalui fungsi assignRole(). Setelah berhasil, user diarahkan ke halaman daftar user dengan pesan sukses.

```
public function store(Request $request)
{
    $request->validate([
        'name' => 'required|string|max:255',
        'role' => 'required|string',
        'email' => 'required|email|unique:users',
        'password' => 'required|min:6',
    ]);

    $user = User::create([
        'name' => $request->name,
        'role' => $request->role,
        'email' => $request->email,
        'password' => Hash::make($request->password),
    ]);

    $user->assignRole($request->role);

    return redirect()->route('users.index')->with('success', 'User berhasil ditambahkan.');
```

Fungsi edit digunakan untuk menampilkan form edit user dengan mengambil data user yang akan diedit dan seluruh role yang tersedia. Data ini kemudian dikirim ke view edit-user untuk ditampilkan dalam form edit.

```
public function edit(User $user)
{
    $roles = Role::all();
    return view('pages.user.edit-user', compact('user', 'roles'));
}
```

Fungsi update digunakan untuk memproses pembaruan data user. Setelah validasi, data pengguna diperbarui di database, dan jika password baru diisi, maka akan di-hash ulang. Role pengguna juga disinkronisasi ulang menggunakan syncRoles() agar role lama tidak bertumpuk. Setelah berhasil, akan diarahkan kembali ke halaman index user.

```
public function update(Request $request, User $user)
{
    $request->validate([
        'name' => 'required|string|max:255',
        'role' => 'required|string',
        'email' => 'required|email|unique:users,email,' . $user->id,
        'password' => 'nullable|min:6',
    ]);

    $user->update([
        'name' => $request->name,
        'role' => $request->role,
        'email' => $request->email,
        'password' => $request->password ? Hash::make($request->password) : $user->password,
    ]);

    $user->syncRoles([$request->role]);

    return redirect()->route('users.index')->with('success', 'User berhasil diperbarui.');
```

Fungsi suspend digunakan untuk mengubah status user menjadi suspend berdasarkan ID yang diberikan. Setelah status diperbarui, user akan diarahkan kembali ke halaman index dengan pesan berhasil disuspend.

```
public function suspend($id)
{
    $user = User::findOrFail($id);

    $user->status = 'suspend';
    $user->save();

    return redirect()->route('users.index')->with('success', 'User berhasil disuspend.');
```

Fungsi activate digunakan untuk mengaktifkan kembali user yang sebelumnya disuspend dengan mengubah status menjadi aktif.

```
public function activate($id)
{
    $user = User::findOrFail($id);
    $user->status = 'aktif';
    $user->save();

    return redirect()->route('users.index')->with('success', 'User berhasil diaktifkan kembali.');
```

Dashboard Controller

Fungsi index digunakan untuk mengisi data dashboard berupa data barang per toko, total stok barang keseluruhan serta menampilkan jumlah admin

```
class DashboardController extends Controller
{
    public function index(Request $request)
    {
        $kategori = Barang::select('kategori')->distinct()->get();
        $jumlahAdmin = User::whereIn('role', ['Admin 1', 'Admin 2', 'Admin 3'])->count();

        if (!$request->has('filter') || $request->filter == 'all') {
            $totalStok = Barang::sum('stok');

            $stok = Barang::join('toko', 'barang.toko_id', '=', 'toko.id')
                ->select('barang.kategori', 'toko.nama_toko', DB::raw('SUM(barang.stok) as total_stok'))
                ->groupBy('barang.kategori', 'toko.nama_toko')
                ->get();
        } else {
            $totalStok = null;
            $stok = Barang::join('toko', 'barang.toko_id', '=', 'toko.id')
                ->where('barang.kategori', $request->filter)
                ->select('barang.kategori', 'toko.nama_toko', DB::raw('SUM(barang.stok) as total_stok'))
                ->groupBy('barang.kategori', 'toko.nama_toko')
                ->get();
        }

        return view('dashboard', compact('stok', 'kategori', 'totalStok', 'jumlahAdmin'));
    }
}
```

Fungsi filter ajax digunakan untuk merespon permintaan filter kategori secara asinkron

```
public function filterAjax(Request $request)
{
    if (!$request->has('filter') || $request->filter == 'all') {
        $totalStok = Barang::sum('stok');
        return "<h6>All: {$totalStok} items</h6>";
    } else {
        $stok = Barang::where('kategori', $request->filter)
            ->join('toko', 'barang.toko_id', '=', 'toko.id')
            ->select('kategori', 'toko.nama_toko', DB::raw('SUM(stok) as total_stok'))
            ->groupBy('kategori', 'toko.nama_toko')
            ->get();

        $output = '';
        foreach ($stok->groupBy('kategori') as $kategori => $items) {
            $output .= "<h6>{$kategori}</h6><ul>";
            foreach ($items as $item) {
                $output .= "<li>{$item->nama_toko}: {$item->total_stok} items</li>";
            }
            $output .= "</ul>";
        }
        return $output;
    }
}
```

Stok Controller

Fungsi index digunakan untuk menampilkan data kategori barang berdasarkan role yang diambil dari segmen url.

```
class StokController extends Controller
{
    public function index()
    {
        $toko = request()->segment(1);

        if ($toko == 'owner') {
            $stok = Stok::with('toko')->get();
        } else {
            $stok = Stok::whereHas('toko', function ($query) use ($toko) {
                $query->whereRaw('LOWER(nama_toko) = ?', [strtolower($toko)]);
            })->with('toko')->get();
        }

        return view("pages.$toko.manajemenbarang.stok.data-stok", compact('stok', 'toko'));
    }
}
```

Fungsi create untuk menampilkan form tambah kategori dengan menyesuaikan toko yang sedang diakses

```
public function create()
{
    $toko = request()->segment(1);
    return view("pages.$toko.manajemenbarang.stok.tambah-stok", compact('toko'));
}
```

Fungsi store digunakan untuk memvalidasi dan menyimpan table stok untuk toko yang diambil dari segmen

```
public function store(Request $request)
{
    $toko = request()->segment(1);
    $stokModel = Toko::where('nama_toko', $toko)->firstOrFail();

    $request->validate([
        'kategori_baru' => [
            'required',
            'string',
            Rule::unique('stok', 'kategori')->where(function ($query) use ($stokModel) {
                return $query->where('toko_id', $stokModel->id);
            })
        ],
    ]);

    DB::transaction(function () use ($request, $stokModel) {
        Stok::create([
            'kategori' => $request->kategori_baru,
            'toko_id' => $stokModel->id,
        ]);
    });

    return redirect()->route("$toko.stok.index")->with('success', 'Kategori baru berhasil ditambahkan.');
```

Fungsi edit untuk mengambil data berdasarkan id dan toko dari segmen url, lalu menampilkannya ke halaman form edit.

```
public function edit($id)
{
    $toko = request()->segment(1);
    $stok = Stok::where('id', $id)->whereHas('toko', function ($query) use ($toko) {
        $query->where('nama_toko', $toko);
    }->firstOrFail());

    return view("pages.$toko.manajemenbarang.stok.edit-stok", compact('stok', 'toko'));
}
```

Fungsi update digunakan untuk memproses pembaruan berdasarkan id dan toko yang digunakan dari segmen url.

```
public function update(Request $request, $id)
{
    $toko = request()->segment(1);
    $stok = Stok::where('id', $id)->whereHas('toko', function ($query) use ($toko) {
        $query->where('nama_toko', $toko);
    }->firstOrFail());

    // Validasi Input
    $request->validate([
        'kategori' => [
            'required',
            'string',
        ],
    ]);

    DB::transaction(function () use ($request, $stok) {
        $stok->update([
            'kategori' => $request->kategori,
        ]);
    });

    return redirect()->route("$toko.stok.index")->with('success', 'Kategori berhasil diperbarui.');
```

Fungsi destroy digunakan untuk menghapus berdasarkan id dan toko yang digunakan dari segmen url.

```
public function destroy($id)
{
    $toko = request()->segment(1);

    $stok = Stok::where('id', $id)->whereHas('toko', function ($query) use ($toko) {
        $query->where('nama_toko', $toko);
    }->firstOrFail());

    DB::transaction(function () use ($stok) {
        $stok->delete();
    });

    return redirect()->route("$toko.stok.index")->with('success', 'Kategori berhasil dihapus.');
```

Barang Controller

Fungsi index digunakan untuk menampilkan daftar barang yang stoknya lebih dari 0 berdasarkan role yang diambil dari segmen url.

```
class BarangController extends Controller
{
    public function index()
    {
        $toko = request()->segment(1);

        if ($toko == 'owner') {
            $barang = Barang::where('stok', '>', 0)->get();
        } else {
            $barang = Barang::whereHas('toko', function ($query) use ($toko) {
                $query->whereRaw('LOWER(nama_toko) = ?', [strtolower($toko)]);
            })
                ->where('stok', '>', 0)
                ->get();
        }

        return view("pages.$toko.manajemenbarang.produk.data-produk", compact('barang', 'toko'));
    }
}
```

Fungsi create untuk menampilkan form tambah barang sekaligus mengambil data kategori.

```
public function create()
{
    $toko = request()->segment(1);
    $tokoModel = Toko::where('nama_toko', $toko)->firstOrFail();

    $kategori = Stok::where('toko_id', $tokoModel->id)
        ->select('kategori')
        ->distinct()
        ->pluck('kategori');

    return view("pages.$toko.manajemenbarang.produk.tambah-produk", compact('kategori', 'toko'));
}
```

Fungsi store untuk menyimpan barang ke tabel barang berdasarkan input user.

```
public function store(Request $request)
{
    $toko = request()->segment(1);
    $tokoModel = Toko::where('nama_toko', $toko)->firstOrFail();

    $request->validate([
        'nama_produk' => 'required|string|max:255',
        'deskripsi' => 'nullable|string',
        'harga_beli' => 'required|numeric|min:1',
        'harga' => 'required|numeric|min:1',
        'stok' => 'required|integer|min:1',
        'kategori' => 'required|string|max:255',
    ]);

    DB::beginTransaction();
    try {
        $barang::create([
            'nama_produk' => $request->nama_produk,
            'deskripsi' => $request->deskripsi,
            'harga_beli' => $request->harga_beli,
            'harga' => $request->harga,
            'stok' => $request->stok,
            'kategori' => $request->kategori,
            'toko_id' => $tokoModel->id,
        ]);

        DB::commit();
        return redirect()->route("$toko.barang.index")->with('success', 'Barang berhasil ditambahkan.');
```

```
    } catch (\Exception $e) {
        DB::rollBack();
        return back()->with('error', 'Terjadi kesalahan: ' . $e->getMessage());
    }
}
```

Fungsi edit menampilkan form edit berdasarkan id.

```
public function edit($id)
{
    $toko = request()->segment(1);
    $tokoModel = Toko::where('nama_toko', $toko)->firstOrFail();

    $barang = Barang::where('id', $id)
        ->whereHas('toko', function ($query) use ($toko) {
            $query->where('nama_toko', $toko);
        })->firstOrFail();

    $kategori = Stok::where('toko_id', $tokoModel->id)
        ->select('kategori')
        ->distinct()
        ->pluck('kategori');
```

```
    return view("pages.$toko.manajemenbarang.produk.edit-produk", compact('barang', 'kategori', 'toko'));
```

Fungsi update digunakan untuk memproses pembaruan berdasarkan id dan toko yang digunakan dari segmen url.

```
public function update(Request $request, $id)
{
    $toko = request()->segment(1);

    $request->validate([
        'nama_produk' => 'required|string|max:255',
        'deskripsi' => 'nullable|string',
        'harga_beli' => 'required|numeric|min:1',
        'harga' => 'required|numeric|min:1',
        'stok' => 'required|integer|min:1',
        'kategori' => 'required|string|max:255',
    ]);

    $barang = Barang::where('id', $id)
        ->whereHas('toko', function ($query) use ($toko) {
            $query->where('nama_toko', $toko);
        })->firstOrFail();

    $barang->update($request->all());

    return redirect()->route("$toko.barang.index")->with('success', 'Barang berhasil diperbarui.');
```

Fungsi destroy digunakan untuk menghapus berdasarkan id dan toko yang digunakan dari segmen url.

```
public function destroy($id)
{
    $toko = request()->segment(1);
    $barang = Barang::where('id', $id)
        ->whereHas('toko', function ($query) use ($toko) {
            $query->where('nama_toko', $toko);
        })->firstOrFail();
    $barang->delete();

    return redirect()->route("$toko.barang.index")->with('success', 'Barang berhasil dihapus.');
```

Fungsi kirim digunakan untuk menampilkan form pengiriman barang antar toko.

```
public function kirim($id)
{
    $toko = request()->segment(1);
    $barang = Barang::findOrFail($id);
    return view("pages.$toko.manajemenbarang.produk.kirim-produk", compact('barang'));
}
```

Fungsi proses kirim digunakan untuk memproses pengiriman barang ke toko lain mengurangi stok toko asal, menyimpan data ke tabel pengiriman, dan menetapkan status awal "Menunggu Konfirmasi".

```
public function prosesKirim(Request $request, $id)
{
    $tokoAsal = request()->segment(1);
    $tokoTujuan = $request->toke_tujuan;

    if ($tokoAsal === $tokoTujuan) {
        return redirect()->back()->withErrors(['toko_tujuan' => 'Tidak bisa mengirim barang ke toko sendiri!'])->withInput();
    }

    $barangAsal = Barang::where('id', $id)
        ->whereHas('toko', function ($query) use ($tokoAsal) {
            $query->where('nama_toko', $tokoAsal);
        })->firstOrFail();

    DB::beginTransaction();
    try {
        $barangAsal->decrement('stok', $request->jumlah);

        $tokoAsal = request()->segment(1);
        $tanggal_pengiriman = now();

        Pengiriman::create([
            'barang_id' => $barangAsal->id,
            'toko_asal' => $tokoAsal,
            'toko_tujuan' => $tokoTujuan,
            'jumlah' => $request->jumlah,
            'status' => 'Dikirim',
            'status_penerimaan' => 'Menunggu Konfirmasi',
            'tanggal_pengiriman' => $tanggal_pengiriman,
        ]);

        DB::commit();
        return redirect()->route("$tokoAsal.barang.index")->with('success', 'Barang berhasil dikirim! Menunggu konfirmasi dari toko tujuan.');
```

```
    } catch (\Exception $e) {
        DB::rollback();
        return back()->withErrors(['error' => 'Terjadi kesalahan: ' . $e->getMessage()])->withInput();
    }
}
```

Fungsi daftar pengiriman masuk digunakan untuk menampilkan daftar pengiriman masuk yang ditujukan ke toko saat ini, yang belum dikonfirmasi penerimaannya.

```
public function daftarPengirimanMasuk()
{
    $toko = request()->segment(1);
    $pengiriman = Pengiriman::where('toko_tujuan', $toko)
        ->where('status_penerimaan', 'Menunggu Konfirmasi')
        ->get();

    return view("pages.$toko.manajemenbarang.produk.cek-pengiriman", compact('pengiriman', 'toko'));
}
```

Fungsi konfirmasi pengiriman digunakan untuk Memproses konfirmasi pengiriman, jika diterima, stok ditambahkan di toko tujuan. Jika ditolak, stok dikembalikan ke toko asal. Status pengiriman juga diperbarui.

```
public function konfirmasiPengiriman(Request $request, $id)
{
    $pengiriman = Pengiriman::findOrFail($id);

    DB::beginTransaction();
    try {
        if ($request->konfirmasi == 'terima') {
            // Ambil toko tujuan
            $tokoTujuan = Toko::where('nama_toko', $pengiriman->toko_tujuan)->firstOrFail();

            // Cek apakah barang sudah ada di toko tujuan
            $barangTujuan = Barang::where('nama_produk', $pengiriman->barang->nama_produk)
                ->where('toko_id', $tokoTujuan->id)
                ->first();

            if ($barangTujuan) {
                // Jika barang sudah ada, tambahkan stoknya
                $barangTujuan->increment('stok', $pengiriman->jumlah);
            } else {
                // Jika barang belum ada, buat data baru di toko tujuan
                Barang::create([
                    'nama_produk' => $pengiriman->barang->nama_produk,
                    'deskripsi' => $pengiriman->barang->deskripsi,
                    'harga_beli' => $pengiriman->barang->harga_beli,
                    'harga' => $pengiriman->barang->harga,
                    'stok' => $pengiriman->jumlah,
                    'kategori' => $pengiriman->barang->kategori,
                    'toko_id' => $tokoTujuan->id,
                ]);
            }

            $pengiriman->update(['status_penerimaan' => 'Diterima']);
        } else {
            $barangAsal = Barang::findOrFail($pengiriman->barang_id);
            $barangAsal->increment('stok', $pengiriman->jumlah);

            $pengiriman->update(['status_penerimaan' => 'Ditolak']);
        }

        DB::commit();
        $toko = strtolower($pengiriman->toko_tujuan);

        return redirect()->route("$toko.pengiriman.masuk")->with(
            'success',
            $request->konfirmasi == 'terima' ? 'Pengiriman berhasil diterima!' : 'Pengiriman berhasil ditolak!'
        );
    } catch (\Exception $e) {
        DB::rollBack();
        return back()->with('error', 'Terjadi kesalahan: ' . $e->getMessage());
    }
}
```

Fungsi non display Menampilkan daftar barang yang stoknya habis atau nol ($\text{stok} \leq 0$) dari toko saat ini, sebagai barang yang tidak tampil.

```
public function nonDisplay()
{
    $toko = request()->segment(1);

    $tokoModel = Toko::where('nama_toko', $toko)->firstOrFail();

    $barang = Barang::where('toko_id', $tokoModel->id)
        ->where('stok', '<=', 0)
        ->get();

    return view("pages.$toko.manajemenbarang.produk.non-display", compact('barang', 'toko'));
}
```

Pengiriman Controller

Fungsi index digunakan untuk menampilkan halaman riwayat pengiriman barang yang melibatkan toko saat ini, baik sebagai pengirim (toko_asal) maupun penerima (toko_tujuan). Data ditampilkan berdasarkan urutan terbaru.

```
class PengirimanController extends Controller
{
    public function index()
    {
        $toko = request()->segment(1);

        $pengiriman = Pengiriman::with('barang')
            ->where('toko_asal', $toko)
            ->orWhere('toko_tujuan', $toko)
            ->latest()
            ->get();

        return view("pages.$toko.manajemenbarang.produk.riwayat-pengiriman", compact('pengiriman'));
    }
}
```

Penjualan Controller

Fungsi `index` digunakan menampilkan semua data penjualan yang dilakukan. Jika pengguna adalah owner, maka semua transaksi dari seluruh toko akan ditampilkan. Namun jika pengguna adalah admin toko, maka hanya transaksi yang berkaitan dengan toko tersebut yang akan ditampilkan berdasarkan nama toko pada URL segment pertama.

```
class PenjualanController extends Controller
{
    public function index()
    {
        $toko = request()->segment(1);

        if ($toko == 'owner') {
            $transaksi = Penjualan::with(['toko'])
                ->orderBy('tanggal_transaksi', 'desc')
                ->orderBy('id', 'desc')
                ->get();
        } else {
            $transaksi = Penjualan::whereHas('toko', function ($query) use ($toko) {
                $query->whereRaw('LOWER(nama_toko) = ?', [strtolower($toko)]);
            })
                ->with(['toko'])
                ->orderBy('tanggal_transaksi', 'desc')
                ->orderBy('id', 'desc')
                ->get();
        }

        return view("pages.$toko.manajemenpenjualan.penjualan.data-penjualan", compact('transaksi', 'toko'));
    }
}
```

Fungsi `create` digunakan untuk menampilkan form tambah penjualan. Di dalamnya akan dimuat semua produk dari toko yang sedang aktif (dengan stok lebih dari 0), yang nantinya bisa dipilih oleh pengguna saat ingin menambahkan transaksi penjualan.

```
public function create()
{
    $toko = request()->segment(1);
    $tokoModel = Toko::whereRaw('LOWER(nama_toko) = ?', [strtolower($toko)])->firstOrFail();

    $produk = Barang::where('toko_id', $tokoModel->id)
        ->where('stok', '>', 0)
        ->select('id', 'nama_produk', 'stok', 'harga', 'kategori', 'deskripsi')
        ->orderBy('nama_produk', 'asc')
        ->get();

    return view("pages.$toko.manajemenpenjualan.penjualan.tambah-penjualan", compact('produk', 'toko'));
}
```

Fungsi store digunakan untuk menyimpan data transaksi penjualan baru. Input produk akan divalidasi, stok produk akan dikurangi, lalu dibuat entri baru pada tabel penjualan dan detail_penjualan untuk setiap item yang dijual. Jika ada kesalahan validasi atau transaksi, proses dibatalkan dan stok tidak berubah.

```
public function store(Request $request)
{
    $toko = request()->segment(1);
    $tokoModel = Toko::whereRaw('LOWER(nama_toko) = ?', [strtolower($toko)])->firstOrFail();

    Log::info('Request data:', $request->all());

    $request->validate([
        'produk' => 'required|array|min:1',
        'produk.*.id' => [
            'required',
            Rule::exists('barang', 'id')->where(function ($query) use ($tokoModel) {
                return $query->where('toko_id', $tokoModel->id);
            }),
        ],
        'produk.*.jumlah' => 'required|integer|min:1',
        'tanggal_transaksi' => 'required|date',
        'biaya_pengiriman' => 'nullable|numeric|min:0',
        'alamat_pengiriman' => 'nullable|string|max:1000',
    ]);

    try {
        DB::transaction(function () use ($request, $tokoModel) {
            $totalHargaProduk = 0;
            $detailProdukUntukSimpan = [];

            foreach ($request->produk as $item) {
                $produkId = $item['id'] ?? $item['produk_id'] ?? null;
                if (!$produkId) {
                    throw new \Exception("ID produk tidak ditemukan dalam data yang dikirim.");
                }
                $barang = Barang::find($produkId);
                if (!$barang || $barang->toko_id != $tokoModel->id) {
                    throw new \Exception("Produk dengan ID {$produkId} tidak valid atau bukan milik toko ini.");
                }
                if ($barang->stok < $item['jumlah']) {
                    throw new \Exception("Stok produk '{$barang->nama_produk}' (ID: {$barang->id}) tidak mencukupi. Sisa stok: {$barang->stok}.");
                }
                $subtotal = $barang->harga * $item['jumlah'];
                $totalHargaProduk += $subtotal;

                $detailProdukUntukSimpan[] = [
                    'barang_id' => $barang->id,
                    'nama_produk' => $barang->nama_produk,
                    'kategori' => $barang->kategori,
                    'deskripsi' => $barang->deskripsi,
                    'jumlah' => $item['jumlah'],
                    'harga_satuan' => $barang->harga,
                    'subtotal' => $subtotal,
                    'alasan_edit' => '',
                ];
            }
        });
    }
}
```

```

$barang->stok -= $item['jumlah'];
$barang->save();
}
$idTransaksi = Penjualan::generateTransactionId($ tokoModel->id);
$penjualanHeader = Penjualan::create([
    'id_transaksi' => $idTransaksi,
    'toko_id' => $ tokoModel->id,
    'tanggal_transaksi' => $request->tanggal_transaksi,
    'biaya_pengiriman' => $request->biaya_pengiriman ?? 0,
    'alamat_pengiriman' => $request->alamat_pengiriman,
    'harga' => $totalHargaProduk + ($request->biaya_pengiriman ?? 0),
]);

foreach ($detailProdukUntukSimpan as $detailData) {
    $penjualanHeader->detail()->create($detailData);
}
});

return redirect()->route("$ toko.penjualan.index")->with('success', 'Transaksi penjualan berhasil ditambahkan.');
```

```

} catch (\Illuminate\Validation\ValidationException $e) {
    Log::error('Kesalahan validasi saat menyimpan transaksi: ' . $e->getMessage() . " Errors: " . json_encode($e->errors()));
    return redirect()->back()->withInput()->withErrors($e->errors())->with('error', 'Terjadi kesalahan validasi. Periksa kembali input Anda.');
```

```

} catch (\Exception $e) {
    Log::error('Error umum saat menyimpan transaksi: ' . $e->getMessage() . " Trace: " . $e->getTraceAsString());
    return redirect()->back()->withInput()->with('error', 'Terjadi kesalahan saat menyimpan transaksi: ' . $e->getMessage());
}
}

```

Fungsi show digunakan untuk menampilkan detail dari suatu transaksi penjualan tertentu, termasuk informasi toko dan semua item produk yang terjual dalam transaksi tersebut. Fungsi ini juga memastikan bahwa data hanya dapat diakses oleh toko yang sesuai.

```

public function show($id)
{
    $toko = request()->segment(1);

    $penjualan = Penjualan::with(['toko', 'detail', 'detail.barang'])
        ->where('id', $id)
        ->when($toko != 'owner', function ($query) use ($toko) {
            $query->whereHas('toko', function ($subQuery) use ($toko) {
                $subQuery->whereRaw("LOWER(nama_toko) = ?", [strtolower($toko)]);
            });
        })
        ->firstOrFail();

    return view("pages.$ toko.manajemenpenjualan.penjualan.detail-penjualan", compact('penjualan', 'toko'));
}

```

Fungsi edit digunakan untuk menampilkan halaman edit untuk transaksi penjualan tertentu. Data transaksi, detail produk yang dijual, serta semua produk dari toko terkait ditampilkan untuk memudahkan admin mengedit transaksi. Fungsi ini juga mencatat log untuk keperluan debugging.

```
public function edit($id)
{
    $toko = request()->segment(1);

    Log::info("Editing penjualan ID: {$id} for toko: {$toko}");

    $penjualan = Penjualan::with(['toko', 'detail', 'detail.barang'])
        ->where('id', $id)
        ->when($toko != 'owner', function ($query) use ($toko) {
            $query->whereHas('toko', function ($subQuery) use ($toko) {
                $subQuery->whereRaw('LOWER(nama_toko) = ?', [strtolower($toko)]);
            });
        })
        ->firstOrFail();

    $tokoModel = Toko::whereRaw('LOWER(nama_toko) = ?', [strtolower($toko)])->firstOrFail();

    $produk = Barang::where('toko_id', $tokoModel->id)
        ->select('id', 'nama_produk', 'stok', 'kategori', 'deskripsi', 'harga')
        ->orderBy('nama_produk', 'asc')
        ->get();

    Log::info("Penjualan details count: " . $penjualan->detail->count());

    foreach ($penjualan->detail as $detail) {
        Log::info("Detail check:", [
            'detail_id' => $detail->id,
            'barang_id' => $detail->barang_id,
            'nama_produk' => $detail->nama_produk,
            'jumlah' => $detail->jumlah,
            'has_barang' => $detail->barang ? 'YES' : 'NO',
            'barang_nama' => $detail->barang ? $detail->barang->nama_produk : 'N/A',
            'barang_stok' => $detail->barang ? $detail->barang->stok : 'N/A'
        ]);
    }

    $missingProducts = [];
    foreach ($penjualan->detail as $detail) {
        if (!$detail->barang) {
            $missingProducts[] = [
                'detail_id' => $detail->id,
                'barang_id' => $detail->barang_id,
                'nama_produk' => $detail->nama_produk
            ];
        }
    }

    if (!empty($missingProducts)) {
        Log::warning("Missing products found:", $missingProducts);
    }

    return view("pages.$toko.manajemenpenjualan.penjualan.edit-penjualan", compact('penjualan', 'produk', 'toko'));
}
```

Fungsi update digunakan untuk memperbarui data transaksi penjualan. Stok produk yang lama dikembalikan terlebih dahulu, lalu detail transaksi lama dihapus. Setelah validasi, transaksi baru disimpan dan stok baru dikurangi berdasarkan perubahan. Semua aktivitas penting dicatat dalam log.

```
public function update(Request $request, $id)
{
    $stoko = request()->segment(1);
    $stokoModel = Toko::whereRaw('LOWER(nama_toko) = ?', [strtolower($stoko)])->firstOrFail();

    Log::info('Update request data:', [
        'id' => $id,
        'toko' => $stoko,
        'request_data' => $request->all()
    ]);

    $request->validate([
        'produk' => 'required|array|min:1',
        'produk.*.id' => [
            'required',
            Rule::exists('barang', 'id')->where(function ($query) use ($stokoModel) {
                return $query->where('toko_id', $stokoModel->id);
            }),
        ],
        'produk.*.jumlah' => 'required|integer|min:1',
        'tanggal_transaksi' => 'required|date',
        'biaya_pengiriman' => 'nullable|numeric|min:0',
        'alamat_pengiriman' => 'nullable|string|max:1000',
        'alasan_edit' => 'required|string|max:1000|regex:/^\s*\S+(?:\s+\S+)+\s*$/ ',
    ]);

    try {
        DB::transaction(function () use ($request, $id, $stokoModel) {
            $penjualan = Penjualan::with('detail.barang')
                ->where('id', $id)
                ->where('toko_id', $stokoModel->id)
                ->firstOrFail();

            Log::info("Updating penjualan:", $penjualan->toArray());

            foreach ($penjualan->detail as $detaillama) {
                $baranglama = $detaillama->barang;
                if ($baranglama) {
                    $baranglama->stok += $detaillama->jumlah;
                    $baranglama->save();
                    Log::info("Returned stock for barang ID {$baranglama->id}: +{$detaillama->jumlah}");
                } else {
                    Log::warning("Barang relation not found for detail ID: {$detaillama->id}");
                }
            }

            $penjualan->detail()->delete();

            $totalHargaProduk = 0;
            $detailProdukBaru = [];

            foreach ($request->produk as $index => $item) {
                $produkId = $item['id'] ?? null;

                Log::info("Processing product {$index}:", $item);
            }
        });
    } catch (\Exception $e) {
        Log::error($e->getMessage());
    }
}
```

```

if (!$produkId) {
    throw new \Exception("ID produk tidak ditemukan dalam data yang dikirim pada index {$index}.");
}

$barang = Barang::find($produkId);

if (!$barang || $barang->stok_id !== $ tokoModel->id) {
    throw new \Exception("Produk dengan ID {$produkId} tidak valid atau bukan milik toko ini.");
}

if ($barang->stok < $item['jumlah']) {
    throw new \Exception("Stok produk '{$barang->nama_produk}' (ID: {$barang->id}) tidak mencukupi. Sisa stok: {$barang->stok}.");
}

$subtotal = $barang->harga * $item['jumlah'];
$totalHargaProduk += $subtotal;

$detaiProdukBaru[] = [
    'barang_id' => $barang->id,
    'nama_produk' => $barang->nama_produk,
    'kategori' => $barang->kategori,
    'deskripsi' => $barang->deskripsi,
    'jumlah' => $item['jumlah'],
    'harga_satuan' => $barang->harga,
    'subtotal' => $subtotal,
    'alasan_edit' => $request->alasan_edit,
    'edited_by' => Auth::id(),
];

$barang->stok -= $item['jumlah'];
$barang->save();

Log::info("Updated stock for barang ID {$barang->id}: -{$item['jumlah']}");
}

$penjualan->update([
    'tanggal_transaksi' => $request->tanggal_transaksi,
    'biaya_pengiriman' => $request->biaya_pengiriman ?? 0,
    'alamat_pengiriman' => $request->alamat_pengiriman,
    'harga' => $totalHargaProduk + ($request->biaya_pengiriman ?? 0),
]);
foreach ($detaiProdukBaru as $detailData) {
    $penjualan->detail()->create($detailData);
}

Log::info("Successfully updated penjualan ID: {$id}");
});

return redirect()->route($ toko.penjualan.index)->with('success', 'Data penjualan berhasil diperbarui.');
```

```

} catch (\Illuminate\Validation\ValidationException $e) {
    Log::error("Kesalahan validasi saat mengupdate transaksi: " . $e->getMessage() . " Errors: " . json_encode($e->errors()));
    return redirect()->back()->withInput()->withErrors($e->errors())->with('error', 'Terjadi kesalahan validasi. Periksa kembali input Anda.');
```

```

} catch (\Exception $e) {
    Log::error("Error umum saat mengupdate transaksi: " . $e->getMessage() . " Trace: " . $e->getTraceAsString());
    return redirect()->back()->withInput()->with('error', 'Terjadi kesalahan saat mengupdate transaksi: ' . $e->getMessage());
}
}

```

Fungsi get produk digunakan untuk mengembalikan data JSON dari suatu produk berdasarkan ID dan nama toko. Fungsinya biasa digunakan untuk AJAX saat memilih produk dalam form transaksi agar data produk dapat ditampilkan secara otomatis (seperti harga, stok, dll).

```
public function getProduk($toko, $id)
{
    $tokoModel = Toko::whereRaw('LOWER(nama_toko) = ?', [strtolower($toko)])->first();

    if (!$tokoModel) {
        return response()->json(['message' => 'Toko tidak ditemukan.'], 404);
    }

    $produk = Barang::where('id', $id)
        ->where('toko_id', $tokoModel->id)
        ->select('id', 'nama_produk', 'deskripsi', 'kategori', 'harga', 'stok')
        ->first();

    if (!$produk) {
        return response()->json(['message' => 'Produk tidak ditemukan atau tidak tersedia.'], 404);
    }

    return response()->json($produk);
}
```

Fungsi export detail pdf digunakan untuk menghasilkan dan mengunduh file PDF berisi detail transaksi penjualan. Tampilan PDF akan disesuaikan berdasarkan toko, dan pengguna bisa memilih apakah ingin menampilkan transaksi dari toko tertentu atau semua toko.

```
public function exportDetailPdf($id, Request $request)
{
    try {
        $penjualan = Penjualan::with(['detail', 'toko'])->findOrFail($id);

        $tokoId = $request->input('toko_id');
        $tokoFromUrl = request()->segment(1);

        if ($tokoId === 'all') {
            $toko = 'semua_toko';
        } else {
            if ($tokoId) {
                $tokoModel = Toko::find($tokoId);
                $toko = $tokoModel->nama_toko ?? $tokoFromUrl;
            } else {
                $toko = $penjualan->toko->nama_toko ?? $tokoFromUrl;
            }
        }

        $view = $toko == 'semua_toko'
            ? 'pages.Owner.manajemenpenjualan.penjualan.detail-penjualan-pdf'
            : "pages.$toko.manajemenpenjualan.penjualan.detail-penjualan-pdf";
        $pdf = Pdf::loadView($view, compact('penjualan'));

        $pdf->setPaper('A4', 'portrait');
    }
}
```

```

        $pdf->setOptions([
            'isRemoteEnabled' => true,
            'isHtml5ParserEnabled' => true,
            'isPhpEnabled' => true,
            'defaultFont' => 'Arial'
        ]);

        $filename = 'Detail_Penjualan_' . $penjualan->id_transaksi . '.pdf';

        return $pdf->download($filename);
    } catch (\Exception $e) {
        return back()->with('error', 'Gagal mengexport PDF: ' . $e->getMessage());
    }
}

```

Fungsi preview pdf digunakan untuk menampilkan preview detail transaksi penjualan dalam bentuk file PDF langsung di browser tanpa mengunduhnya

```

public function previewPdf($id)
{
    try {
        $penjualan = Penjualan::with(['detail', 'toko'])->findOrFail($id);

        $pdf = Pdf::loadView('pages.Owner.manajemenpenjualan.penjualan.detail-penjualan-pdf', compact('penjualan'));
        $pdf->setPaper('A4', 'portrait');

        return $pdf->stream('Detail_Penjualan_' . $penjualan->id_transaksi . '.pdf');
    } catch (\Exception $e) {
        return back()->with('error', 'Gagal menampilkan PDF: ' . $e->getMessage());
    }
}

```

Laporan Controller

Fungsi index digunakan untuk menampilkan halaman laporan penjualan baik untuk admin toko maupun owner. Jika yang mengakses adalah owner, maka laporan yang ditampilkan mencakup semua toko, lengkap dengan transaksi, grafik bar jumlah produk terjual, dan grafik donut kategori. Jika yang mengakses adalah toko tertentu, data yang ditampilkan hanya terbatas pada toko itu. Fungsi ini juga mendukung filter tanggal dan perhitungan total laporan berdasarkan rentang waktu yang dipilih.

```
public function index()
{
    $toko = request()->segment(1);
    $startDate = request('start_date');
    $endDate = request('end_date');
    $laporan = [];

    if ($toko === 'owner') {
        $daftarToko = Toko::all();

        $transaksi = DB::table('detail_penjualan')
            ->join('penjualan', 'detail_penjualan.penjualan_id', '=', 'penjualan.id')
            ->join('barang', 'detail_penjualan.barang_id', '=', 'barang.id')
            ->join('toko', 'penjualan.toko_id', '=', 'toko.id')
            ->select(
                'barang.nama_produk',
                'detail_penjualan.jumlah',
                'barang.kategori',
                'detail_penjualan.subtotal',
                'penjualan.tanggal_transaksi',
                'toko.nama_toko'
            )
            ->when($startDate && $endDate, function ($query) use ($startDate, $endDate) {
                $query->whereBetween('penjualan.tanggal_transaksi', [$startDate, $endDate]);
            })
            ->get();
    }
}
```

```

$penjualanPerProduk = DB::table('detail_penjualan')
->join('barang', 'detail_penjualan.barang_id', '=', 'barang.id')
->select('barang.nama_produk', DB::raw('SUM(detail_penjualan.jumlah) as total_jumlah'))
->groupBy('barang.nama_produk')
->get();

$kategoriPenjualan = DB::table('barang')
->join('detail_penjualan', 'barang.id', '=', 'detail_penjualan.barang_id')
->select('barang.kategori', DB::raw('SUM(detail_penjualan.jumlah) as total_kategori'))
->groupBy('barang.kategori')
->get();

return view("pages.owner.manajemenpenjualan.laporan", compact(
    'transaksi',
    'penjualanPerProduk',
    'kategoriPenjualan',
    'toko',
    'startDate',
    'endDate',
    'laporan',
    'daftarToko'
));

$stokoModel = Toko::where('nama_toko', $toko)->first();
if (!$stokoModel) {
    return abort(404, 'Toko tidak ditemukan di database.');
```

```

$transaksi = DB::table('penjualan')
->join('detail_penjualan', 'penjualan.id', '=', 'detail_penjualan.penjualan_id')
->select(
    'detail_penjualan.nama_produk',
    'detail_penjualan.kategori',
    'detail_penjualan.jumlah',
    'detail_penjualan.harga_satuan',
    'detail_penjualan.subtotal',
    'penjualan.tanggal_transaksi'
)
->where('penjualan.toko_id', $stokoModel->id)
->when($startDate && $endDate, function ($query) use ($startDate, $endDate) {
    return $query->whereBetween('penjualan.tanggal_transaksi', [$startDate, $endDate]);
})
->get();

$penjualanPerProduk = DB::table('penjualan')
->join('detail_penjualan', 'penjualan.id', '=', 'detail_penjualan.penjualan_id')
->select(
    'detail_penjualan.nama_produk',
    DB::raw('SUM(detail_penjualan.jumlah) as total_jumlah')
)
->where('penjualan.toko_id', $stokoModel->id)
->groupBy('detail_penjualan.nama_produk')
->get();
```

```

$kategoriPenjualan = DB::table('barang')
->join('detail_penjualan', 'barang.id', '=', 'detail_penjualan.barang_id')
->join('penjualan', 'detail_penjualan.penjualan_id', '=', 'penjualan.id')
->select('barang.kategori', DB::raw('SUM(detail_penjualan.jumlah) as total_kategori'))
->where('penjualan.toko_id', $tokoModel->id)
->groupBy('barang.kategori')
->get();

if ($startDate && $endDate) {
    $laporan = $this->hitungLaporan($tokoModel->id, $startDate, $endDate);
}

return view("pages.$toko.manajemenpenjualan.laporan", compact(
    'transaksi',
    'penjualanPerProduk',
    'kategoriPenjualan',
    'toko',
    'startDate',
    'endDate',
    'laporan'
));
}

```

Fungsi export pdf digunakan untuk menghasilkan file PDF berisi laporan penjualan berdasarkan toko, tanggal, kategori, dan produk yang difilter. Laporan mencakup data transaksi, total penjualan per produk dan kategori, omset harian/mingguan/bulanan, serta laba/rugi. View PDF disesuaikan dengan apakah laporan mencakup semua toko atau satu toko tertentu, lalu di-render menggunakan DomPDF dan disediakan untuk diunduh.

```
public function exportPdf($toko, Request $request)
{
    $tokoId = $request->input('toko_id');
    $tokoFromUrl = request()->segment(1);

    if ($tokoId === 'all') {
        $tokoModel = null;
        $toko = 'semua_toko';
    } else {
        if ($tokoId) {
            $tokoModel = Toko::find($tokoId);
            $toko = $tokoModel->nama_toko ?? $tokoFromUrl;
        } else {
            $tokoModel = Toko::where('nama_toko', $tokoFromUrl)->first();
            $toko = $tokoFromUrl;
        }

        if (!$tokoModel) {
            return abort(404, 'Toko tidak ditemukan di database.');
        }
    }

    $startDate = request('start_date');
    $endDate = request('end_date');
    $filterKategori = request('kategori');
    $filterProduk = request('produk');
```

```

$transaksi = DB::table('penjualan')
->join('detail_penjualan', 'detail_penjualan.penjualan_id', '=', 'penjualan.id')
->join('barang', 'detail_penjualan.barang_id', '=', 'barang.id')
->join('toko', 'penjualan.toko_id', '=', 'toko.id')
->select(
    'penjualan.toko_id',
    'barang.nama_produk',
    'detail_penjualan.jumlah',
    'barang.kategori',
    'detail_penjualan.harga_satuan',
    'penjualan.tanggal_transaksi'
)
->when($tokoModel, function ($query) use ($tokoModel) {
    return $query->where('penjualan.toko_id', $tokoModel->id);
})
->when($startDate && $endDate, function ($query) use ($startDate, $endDate) {
    return $query->whereBetween('penjualan.tanggal_transaksi', [$startDate, $endDate]);
})
->when($filterKategori, function ($query) use ($filterKategori) {
    return $query->where('barang.kategori', $filterKategori);
})
->when($filterProduk, function ($query) use ($filterProduk) {
    return $query->where('barang.nama_produk', 'like', '%' . $filterProduk . '%');
})
->get();

$baseOmset = DB::table('penjualan')
->join('detail_penjualan', 'detail_penjualan.penjualan_id', '=', 'penjualan.id');

```

```

if ($tokoModel) {
    $baseOmset->where('penjualan.toko_id', $tokoModel->id);
}

$omset_harian = (clone $baseOmset)
->whereDate('penjualan.tanggal_transaksi', Carbon::today())
->sum(DB::raw('detail_penjualan.harga_satuan * detail_penjualan.jumlah'));

$omset_mingguan = (clone $baseOmset)
->whereBetween('penjualan.tanggal_transaksi', [Carbon::now()->startOfWeek(), Carbon::now()->endOfWeek()])
->sum(DB::raw('detail_penjualan.harga_satuan * detail_penjualan.jumlah'));

$omset_bulanan = (clone $baseOmset)
->whereMonth('penjualan.tanggal_transaksi', Carbon::now()->month)
->sum(DB::raw('detail_penjualan.harga_satuan * detail_penjualan.jumlah'));

$laba_rugi = DB::table('penjualan')
->join('detail_penjualan', 'detail_penjualan.penjualan_id', '=', 'penjualan.id')
->join('barang', 'detail_penjualan.barang_id', '=', 'barang.id')
->when($tokoModel, function ($query) use ($tokoModel) {
    return $query->where('penjualan.toko_id', $tokoModel->id);
})
->sum(DB::raw('(detail_penjualan.harga_satuan - barang.harga beli) * detail_penjualan.jumlah'));

$status = $laba_rugi > 0 ? 'Laba' : ($laba_rugi < 0 ? 'Rugi' : 'Break Even');

```

```

$penjualanPerProduk = DB::table('penjualan')
->join('detail_penjualan', 'detail_penjualan.penjualan_id', '=', 'penjualan.id')
->join('barang', 'detail_penjualan.barang_id', '=', 'barang.id')
->select('barang.nama_produk', DB::raw('SUM(detail_penjualan.jumlah) as total_jumlah'))
->when($stokoModel, function ($query) use ($stokoModel) {
    return $query->where('penjualan.toko_id', $stokoModel->id);
})
->when($startDate && $endDate, function ($query) use ($startDate, $endDate) {
    return $query->whereBetween('penjualan.tanggal_transaksi', [$startDate, $endDate]);
})
->groupBy('barang.nama_produk')
->get();

$kategoriPenjualan = DB::table('penjualan')
->join('detail_penjualan', 'detail_penjualan.penjualan_id', '=', 'penjualan.id')
->join('barang', 'detail_penjualan.barang_id', '=', 'barang.id')
->select('barang.kategori', DB::raw('SUM(detail_penjualan.jumlah) as total_jumlah'))
->when($stokoModel, function ($query) use ($stokoModel) {
    return $query->where('penjualan.toko_id', $stokoModel->id);
})
->groupBy('barang.kategori')
->get();

$view = $stoko == 'semua_toko'
    ? 'pages.Owner.manajemenpenjualan.penjualan-pdf'
    : "pages.$stoko.manajemenpenjualan.penjualan-pdf";
$pdf = Pdf::loadView($view, compact(
    'transaksi',
    'penjualanPerProduk',
    'kategoriPenjualan',
    'toko',
    'startDate',
    'endDate',
    'omset_harian',
    'omset_mingguan',
    'omset_bulanan',
    'laba_rugi',
    'status'
));

return $pdf->download('laporan_penjualan_' . $stoko . '.pdf');
}

```

Fungsi export excel digunakan untuk mengeksport data laporan penjualan dalam format Excel menggunakan library Laravel Excel. Laporan disesuaikan berdasarkan ID toko serta rentang tanggal yang dipilih.

```
public function exportExcel(Request $request)
{
    $tokoId = $request->input('toko_id');

    $startDate = $request->input('start_date');
    $endDate = $request->input('end_date');

    return Excel::download(new PenjualanExport($tokoId, $startDate, $endDate), 'laporan_penjualan.xlsx');
}
```

Fungsi get laporan omset digunakan untuk mengambil data laporan omset, laba, dan rugi dari penjualan suatu toko dalam rentang tanggal tertentu dalam format JSON. Data di-group by per hari (periode) dan menampilkan total omset, biaya modal, laba, serta kerugian berdasarkan perhitungan selisih antara penjualan dan harga beli.

```
public function getLaporanOmset(Request $request, $toko)
{
    $startDate = $request->query('start_date');
    $endDate = $request->query('end_date');

    if (!$startDate || !$endDate) {
        return response()->json(['error' => 'Tanggal tidak boleh kosong!'], 400);
    }

    $toko = Toko::find($toko);

    if (!$toko) {
        return response()->json(['error' => 'Toko tidak ditemukan!'], 404);
    }

    $laporan = DB::table('penjualan')
        ->join('detail_penjualan', 'penjualan.id', '=', 'detail_penjualan.penjualan_id')
        ->join('barang', 'detail_penjualan.barang_id', '=', 'barang.id')
        ->where('penjualan.toko_id', $toko->id)
        ->whereBetween('penjualan.tanggal_transaksi', [$startDate, $endDate])
        ->selectRaw('
            DATE_FORMAT(penjualan.tanggal_transaksi, "%Y-%m-%d") as periode,
            SUM(detail_penjualan.subtotal) as omset,
            SUM(barang.harga_beli * detail_penjualan.jumlah) as biaya,
            SUM(detail_penjualan.subtotal - (barang.harga_beli * detail_penjualan.jumlah)) as laba,
            SUM(CASE
                WHEN (detail_penjualan.subtotal - (barang.harga_beli * detail_penjualan.jumlah)) < 0
                THEN ((barang.harga_beli * detail_penjualan.jumlah) - detail_penjualan.subtotal)
                ELSE 0
            END) as rugi
        ')
        ->groupBy('periode')
        ->get();

    return response()->json($laporan);
}
```

Fungsi get laporan semua toko digunakan untuk menghasilkan data laporan omset, biaya, laba, dan rugi untuk setiap toko yang ada di sistem, berdasarkan rentang tanggal yang dipilih. Data ini digunakan untuk membandingkan performa penjualan antar toko, dan hasilnya dikembalikan dalam bentuk JSON berisi nama toko dan detail laporan per hari.

```
public function getLaporanSemuaToko(Request $request)
{
    $startDate = $request->query('start_date');
    $endDate = $request->query('end_date');

    if (!$startDate || !$endDate) {
        return response()->json(['error' => 'Tanggal tidak boleh kosong!'], 400);
    }

    $tokolist = Toko::all();
    $hasil = [];

    foreach ($tokolist as $t) {
        $laporan = DB::table('penjualan')
            ->join('detail_penjualan', 'penjualan.id', '=', 'detail_penjualan.penjualan_id')
            ->join('barang', 'detail_penjualan.barang_id', '=', 'barang.id')
            ->where('penjualan.toko_id', $t->id)
            ->whereBetween('penjualan.tanggal_transaksi', [$startDate, $endDate])
            ->selectRaw('
                DATE_FORMAT(penjualan.tanggal_transaksi, "%Y-%m-%d") as periode,
                SUM(detail_penjualan.subtotal) as omset,
                SUM(barang.harga_beli * detail_penjualan.jumlah) as biaya,
                SUM(detail_penjualan.subtotal - (barang.harga_beli * detail_penjualan.jumlah)) as laba,
                SUM(CASE
                    WHEN (detail_penjualan.subtotal - (barang.harga_beli * detail_penjualan.jumlah)) < 0
                    THEN ((barang.harga_beli * detail_penjualan.jumlah) - detail_penjualan.subtotal)
                    ELSE 0
                END) as rugi
            ')
            ->groupBy('periode')
            ->get();

        // Tambahkan data laporan toko
        $hasil[] = [
            'toko' => $t->nama_toko,
            'laporan' => $laporan
        ];
    }

    return response()->json($hasil);
}
```

Model User

Model user digunakan untuk mewakili data pengguna dan menangani proses login, registrasi, serta manajemen role dan permission.

```
class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable, HasRoles;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'role',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
        'password' => 'hashed',
    ];
}
```

Model Toko

Model ini digunakan untuk mewakili data toko dalam database dan mengatur relasi toko dengan entitas lain seperti barang, stok, pengiriman, penjualan, dan ringkasan keuangan. Model ini mengatur nama tabel yang digunakan (toko) dan menentukan kolom yang bisa diisi (nama_toko). Model ini juga menggunakan `getRouteKeyName()` agar routing bisa menggunakan nama_toko sebagai parameter URL.

```
class Toko extends Model
{
    use HasFactory;

    protected $table = 'toko';

    protected $fillable = [
        'nama_toko',
    ];

    public function getRouteKeyName()
    {
        return 'nama_toko';
    }

    public function barang()
    {
        return $this->hasMany(Barang::class, 'toko_id');
    }

    public function stok()
    {
        return $this->hasMany(Stok::class, 'toko_id');
    }

    public function pengirimansAsTokoAsal()
    {
        return $this->hasMany(Pengiriman::class, 'toko_asal');
    }

    public function pengirimansAsTokoTujuan()
    {
        return $this->hasMany(Pengiriman::class, 'toko_tujuan');
    }

    public function summary()
    {
        return $this->hasOne(SummaryKeuangan::class, 'toko_id', 'id');
    }

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

Model Stok

Model stok digunakan untuk merepresentasikan data kategori barang yang dimiliki oleh setiap toko dalam database.

```
class Stok extends Model
{
    use HasFactory;

    protected $table = 'stok';

    protected $fillable = [
        'kategori',
        'toko_id',
    ];

    public function toko()
    {
        return $this->belongsTo(Toko::class, 'toko_id');
    }
}
```

Model Barang

Model barang digunakan untuk mewakili data barang yang dijual oleh sebuah toko dalam database. Dengan relasi belongsTo ke model Toko, setiap barang dikaitkan dengan satu toko pemiliknya.

```
class Barang extends Model
{
    use HasFactory;

    protected $table = 'barang';
    protected $fillable = [
        'nama_produk',
        'deskripsi',
        'harga_beli',
        'harga',
        'stok',
        'kategori',
        'toko_id',
    ];

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

Model Pengiriman

Model pengiriman digunakan untuk merepresentasikan data proses pengiriman barang antar toko dalam database. Model ini penting untuk mencatat dan melacak distribusi stok antar cabang atau unit toko.

```
class Pengiriman extends Model
{
    use HasFactory;

    protected $table = 'pengiriman';
    protected $fillable = [
        'barang_id',
        'toko_asal',
        'toko_tujuan',
        'jumlah',
        'tanggal_pengiriman',
        'status_penerimaan'
    ];

    public function barang()
    {
        return $this->belongsTo(Barang::class);
    }

    public function toko_asal()
    {
        return $this->belongsTo(Toko::class, 'toko_asal');
    }

    public function toko_tujuan()
    {
        return $this->belongsTo(Toko::class, 'toko_tujuan');
    }
}
```

Model Penjualan

Model penjualan digunakan untuk merepresentasikan data transaksi penjualan barang oleh toko dalam database. Fitur `generateTransactionId()` digunakan untuk membuat kode transaksi otomatis dengan format unik berdasarkan tanggal dan ID toko, yang akan diproses saat data penjualan dibuat.

```
class Penjualan extends Model
{
    use HasFactory;

    protected $table = 'penjualan';

    protected $fillable = [
        'id_transaksi',
        'toko_id',
        'harga',
        'biaya_pengiriman',
        'alamat_pengiriman',
        'tanggal_transaksi',
    ];

    protected $casts = [
        'tanggal_transaksi' => 'date',
        'harga' => 'decimal:2',
        'biaya_pengiriman' => 'decimal:2'
    ];

    public static function generateTransactionId($tokoid)
    {
        $today = now();
        $datePrefix = $today->format('Ynd');

        $searchPattern = "TRX-{$datePrefix}-X-1({$tokoid})";

        $lastTransaction = self::where('id_transaksi', 'like', $searchPattern)
            ->orderBy('id', 'desc')
            ->first();

        if (!$lastTransaction) {
            $sequence = 1;
        } else {
            $parts = explode('-', $lastTransaction->id_transaksi);
            $sequencePart = $parts[2];
            $lastSequence = (int) $sequencePart;
            $sequence = $lastSequence + 1;
        }

        return 'TRX-' . $datePrefix . '-' . str_pad($sequence, 3, '0', STR_PAD_LEFT) . '-' . $tokoid;
    }

    protected static function boot()
    {
        parent::boot();

        static::creating(function ($penjualan) {
            if (empty($penjualan->id_transaksi)) {
                $penjualan->id_transaksi = self::generateTransactionId($penjualan->toko_id);
            }
        });
    }

    public function toko()
    {
        return $this->belongsTo(Toko::class, 'toko_id');
    }

    public function barang()
    {
        return $this->belongsTo(Barang::class, 'barang_id');
    }

    public function detail()
    {
        return $this->hasMany(DetailPenjualan::class, 'penjualan_id');
    }
}
```

Model Detail Penjualan

Model detail penjualan digunakan untuk menyimpan rincian setiap item yang dijual dalam satu transaksi penjualan. Model ini merupakan bagian penting dari sistem penjualan karena mencatat detail setiap produk dalam transaksi penjualan secara lengkap dan fleksibel.

```
class DetailPenjualan extends Model
{
    use HasFactory;

    protected $table = 'detail_penjualan';

    protected $fillable = [
        'penjualan_id',
        'barang_id',
        'nama_produk',
        'kategori',
        'deskripsi',
        'jumlah',
        'harga_satuan',
        'subtotal',
        'alasan_edit',
        'edited_by',
    ];

    public function penjualan()
    {
        return $this->belongsTo(Penjualan::class, 'penjualan_id');
    }

    public function barang()
    {
        return $this->belongsTo(Barang::class, 'barang_id');
    }

    public function editedBy()
    {
        return $this->belongsTo(User::class, 'edited_by');
    }
}
```

Model Summary Keuangan

Model ini digunakan untuk menyimpan hasil perhitungan omset sebuah toko dalam periode waktu tertentu. Model ini juga digunakan untuk menampilkan hasil perhitungan ke halaman dashboard.

```
class SummaryKeuangan extends Model
{
    use HasFactory;
    protected $table = 'summary_keuangan';

    protected $fillable = [
        'toko_id',
        'omset',
        'laba',
        'start_date',
        'end_date',
    ];

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

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	: EC002025076918, 26 Juni 2025
Pencipta	
Nama	: Dika Dziki Pratama, Dyah Aprilliani, S.T., M.Kom. dkk
Alamat	: Dukuh Kedawon RT02 RW06, Kecamatan Larangan, Kabupaten Brebes 52262, Larangan, Kab. Brebes, Jawa Tengah, 52262
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 Informasi Penjualan dan Persediaan Barang Elektronik Untuk Pengelolaan Stok Berbasis Website (Studi Kasus Toko Wayang Elektronik Dukuh Kedawon)
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	: 26 Juni 2025, di Kota Tegal
Jangka waktu perlindungan	: Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor Pencatatan	: 000917179
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  Agung Damarsasongko.SH..MH. NIP. 196912261994031001
	Disclaimer: 1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan. 2. Surat Pencatatan ini telah disegel secara elektronik menggunakan segel elektronik yang diterbitkan oleh Balai Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara. 3. Surat Pencatatan ini dapat dibuktikan keasliannya 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 : Dika Dzikri Pratama
 NIM : 21090121
 No. Ponsel : 0895377386177
 Judul TA : RANCANG BANGUN SISTEM INFORMASI
 PENJUALAN DAN PERSEDIAAN BARANG
 ELEKTRONIK UNTUK PENGELOLAAN
 STOK BERBASIS WEBSITE (Studi kasus
 Toko Wayang Elektronik Dukuh Kedawon)

Dosen Pembimbing I : Dyah Apriliani, S.T., M.Kom

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	16/4/2023	Aplikasi	Konsep diagram ↓ Toko ↓ toko, 120121.2023	
2.	23/4/2023	Aplikasi	Alur proses bar aplikasi → data	
3.	21/5/2023	Aplikasi	Revisi - Tambahkan Bagian lain	

4.	21/6 2025	Aplikasi	Optimalisasi halaman dashboard	
5.	23/2025 /6	De Hki	Peru	
6.	8/2025 /8	Laporan	<u>Peru</u>	
7.	9/2025 /7	Laporan	Peru	
8.	9/2025 /7	Laporan	Peru	

--	--	--	--	--

Tanggal Juli 2025
Dosen Pembimbing I



Dyah Apriliani, S.T., M.Kom.
NIPY. 09.015.255



**D IV. TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Dika Dzikri Pratama
NIM : 21090121
No. Ponsel : 0895377386177
Judul TA : RANCANG BANGUN SISTEM INFORMASI
PENJUALAN DAN PERSEDIAAN BARANG
ELEKTRONIK UNTUK PENGELOLAAN
STOK BERBASIS WEBSITE (Studi kasus
Toko Wayang Elektronik Dukuh Kedawon)

Dosen Pembimbing II : M. Nishom, S.Kom., M.Kom

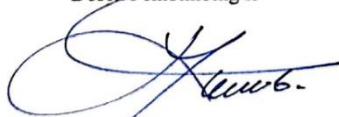
No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	2/5 - 2025	- Rancangan fungsionalitas - Aplikasi / Sistem	- Susun gambaran sistem (use case). - Bangun sesuai rancangan fungsionalitas.	
2.	8/5 - 2025	- Sistem	- Lanjutkan pembuatan sistem ✓ pada laporan keuangan tambahkan fitur untuk "Bemua folo" ✓ pada Role "folo" tambahkan Menu Riwayat perpindahan produk. ✓ cek semua logic dan Validasi form.	  

3.	3/6-2025	Sistem	- form data penjualan belum kompatibel dengan kebutuhan pendataan transaksi penjualan. → Perbaiki sesuai kebutuhan (logic wajib didasarkan pada proses bisnis).	↗
4.	10/6-25	- sistem	- Di dashboard owner sediakan fitur untuk cek omset & laba/rugi. (custom).	↗
5.	16/6-2025	- Sistem	- Akuntabilitas. (tingkatkan)	↗
		- HCI	- uji sistem - Persiapkan dokumen persyaratan usulan HCI	↗
		- Laporan	- susun Laporan (BAB I-III).	↗
6.	26/6-2025	- Laporan	- Perbaiki, sesuai ketentuan pada Laporan	↗

			Bas U : Susun laporan berdasarkan kan hasil / ltkj	✓
		- serkon -	- Belum Ada Persiapan!	
		- Taggl. Jl.	- Persiapan -	✓
7	7/7-25	- Pendaftaran Fidalg	- lengkap kelengkapan berkas pa. daftar	✓

--	--	--	--	--

Tegal, 7 Juli 2025
Dosen Pembimbing II



M. Nishom, S.Kom., M.Kom.
NIPY. 09.017.337