

LAMPIRAN

Lampiran 1 Surat Kesepakatan Pembimbing

1. Surat Kesepakatan Pembimbing 1

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Helina Putri
NIM : 19090133
Program Studi : DIV Teknik Informatika

Pihak Kedua

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

Pada hari ini Kamis tanggal 18 Juni 2026 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 18 Juni 2026

Pihak Pertama



Helina Putri

Pihak Kedua



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

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika



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

2. Surat Kesepakatan Pembimbing 2

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Helina Putri
NIM : 19090133
Program Studi : DIV Teknik Informatika

Pihak Kedua

Nama : Arif Hidayah, S.Tr.Kom., M.Tr.Kom
Status : Dosen
NUPTK : 2154773674130253
Jabatan Fungsional : -
Pangkat/Golongan : -

Pada hari ini Kamis tanggal 04 Juni 2026 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 04 Juni 2026

Pihak Pertama



Helina Putri

Pihak Kedua



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

Mengetahui
Ketua Program Studi DIV Teknik Informatika



Dyah Apriani, S.T., M.Kom.
NIPY. 09.015.225

Lampiran 2 Surat Pengalihan Hak Cipta

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

- 1 Nama : Helina Putri
Kewarganegaraan : Indonesia
Alamat : Jalan Raya Pakulaut RT 05 / RW 04, Desa Pakulaut, Kecamatan Margasari, Kabupaten Tegal, Provinsi Jawa Tengah, 52463
- 2 Nama : Dyah Apriliani, S.T., M.Kom
Kewarganegaraan : Indonesia
Alamat : Perumahan Mutiara Indah Blok C4, Jalan Nangka Gang 2, RT 02 / RW 02, Kelurahan Procot, Kecamatan Slawi, Kabupaten Tegal, Provinsi Jawa Tengah, 52412
- 3 Nama : Arif Hidayah, S.Tr.Kom., M.Tr.Kom
Kewarganegaraan : Indonesia
Alamat : Jalan Bulu Emas No. 8, Desa Lawatan, Kecamatan Dukuhturi, Kabupaten Tegal, Provinsi Jawa Tengah, 52192

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

Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M)
Universitas Harkat Negeri
Alamat : Jl. Mataram No.9. Pesurungan Lor, Kec. Margadana, Kota Tegal

Adalah Pihak II selaku Pemegang Hak Cipta berupa Webiste dengan judul "**Aplikasi Point Of Sale (POS) Bengkel Berbasis Website**". untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

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

Pemegang Hak Cipta
Ketua P3M
 **Universitas Harkat Negeri**
(Sharfina Febbi Handayani, M.Kom.)

Tegal, 30 Juli 2026

Pencipta

(Helina Putri)

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

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

Lampiran 3 Surat Pernyataan HKI

SURAT PERNYATAAN

Yang bertanda tangan dibawah ini, pemegang hak cipta:

Nama : Helina Putri
Kewarganegaraan : Indonesia
Alamat : Jalan Raya Pakulaut RT 05 / RW 04, Desa Pakulaut, Kecamatan Margasari, Kabupaten Tegal, Provinsi Jawa Tengah, 52463

Nama : Diyah Apriliani, S.T., M.Kom
Kewarganegaraan : Indonesia
Alamat : Perumahan Mutiara Indah Blok C4, Jalan Nangka Gang 2, RT 02 / RW 02, Kelurahan Procot, Kecamatan Slawi, Kabupaten Tegal, Provinsi Jawa Tengah, 52412

Nama : Arif Hidayah, S.Tr.Kom., M.Tr.Kom
Kewarganegaraan : Indonesia
Alamat : Jalan Bulu Emas No. 8, Desa Lawatan, Kecamatan Dukuhturi, Kabupaten Tegal, Provinsi Jawa Tengah, 52192

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:

Berupa : Program Komputer
Berjudul : Aplikasi *Point Of Sale* Bengkel Berbasis Website

Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);

- Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
- Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
- Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
- Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
- Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh 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 kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, 30 Juli 2026



Helina Putri
Pemegang Hak Cipta*

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

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

Pemegang Hak Cipta*

*Semua pemegang hak cipta agar menandatangani di atas materai.



Disusun oleh

Helina Putri

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

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

Daftar Isi

Daftar Isi	3
1. Pengenalan Sistem	4
2. Registrasi dan Login	5
3. Panduan Administrator	10
4. Panduan Pemilik Bengkel	14
5. Panduan Data Master	19
6. Jasa, Pemasok, Pelanggan, dan Kendaraan	21
7. Panduan Pembelian	23
8. Laporan dan Konfirmasi Setor Kas	26
9. Panduan Kasir dan Kendala Umum	27
10. Penutup	30

01

PENGENALAN SISTEM

Gambaran umum fungsi POS Bengkel

1.1 Deskripsi Sistem

Aplikasi Point of Sale (POS) Bengkel merupakan aplikasi berbasis web yang membantu kegiatan operasional bengkel. Sistem ini digunakan untuk mengelola data bengkel, barang dan jasa servis, pembelian dari pemasok, penjualan, pelanggan dan kendaraan, setor kas, serta laporan.

1.2 Kelompok Pengguna

01	02	03
Administrator	Pemilik Bengkel	Kasir
Memantau data bengkel, status langganan, dan riwayat pembayaran.	Mengelola profil bengkel, langganan, kasir, data master, pembelian, laporan, dan verifikasi setor kas.	Melayani penjualan, mengelola pelanggan dan kendaraan, melihat riwayat transaksi, serta menyetor kas.

02

REGISTRASI DAN LOGIN

Panduan membuat akun pemilik bengkel dan mendaftarkan bengkel.

2.1 Registrasi Akun dan Bengkel

Menu Register digunakan oleh calon pemilik bengkel untuk membuat akun dan mendaftarkan bengkel sebelum menggunakan aplikasi POS Bengkel.

1	Buka halaman Register — Pada halaman login, klik Daftar Sekarang .
2	Isi informasi akun — Masukkan nama pemilik, username, email, password, dan konfirmasi password.
3	Isi informasi bengkel — Masukkan nama bengkel, alamat, dan nomor telepon yang aktif.
4	Periksa kembali data — Pastikan seluruh informasi telah ditulis dengan benar.
5	Klik Daftar Sekarang — Sistem memproses pendaftaran dan menampilkan hasil registrasi.

i

Catatan Gunakan email aktif karena tautan verifikasi akun akan dikirim melalui email tersebut.

02

REGISTRASI DAN LOGIN

Tampilan halaman Register pada aplikasi POS Bengkel.

2.2 Halaman Register

Halaman Register terdiri dari bagian Informasi Akun dan Informasi Bengkel. Seluruh kolom wajib diisi menggunakan data yang benar.

Gambar 2.1 Halaman Register Akun POS Bengkel

02

REGISTRASI DAN LOGIN

Langkah masuk ke dalam aplikasi POS Bengkel.

2.3 Login ke Aplikasi

1

Buka aplikasi — Masukkan alamat POS Bengkel pada browser.

2

Isi username atau email — Gunakan akun yang sudah terverifikasi.

3

Masukkan password — Ketik password dengan benar.

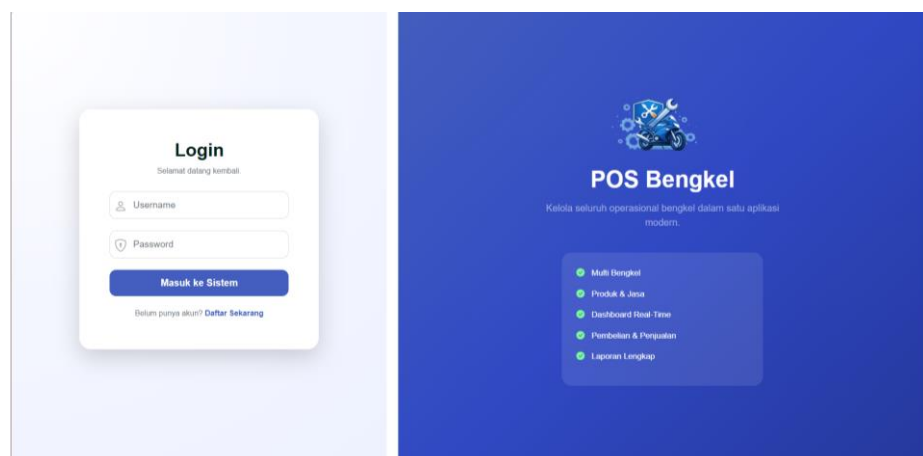
4

Klik tombol Login — Klik Masuk Ke Sistem.

5

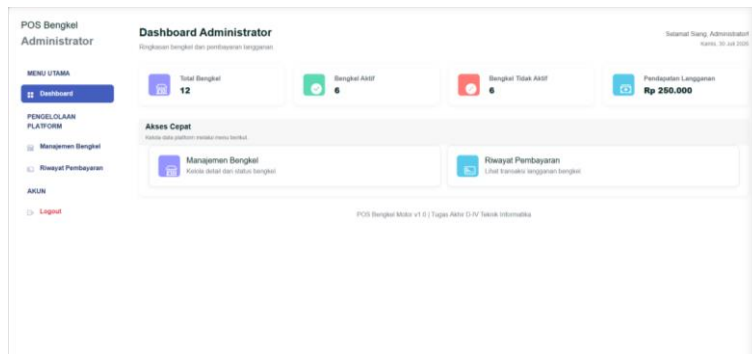
Masuk ke dashboard — Sistem menampilkan dashboard sesuai hak akses pengguna.

2.3.1 Halaman Login



Gambar 2.2 Halaman Login POS Bengkel

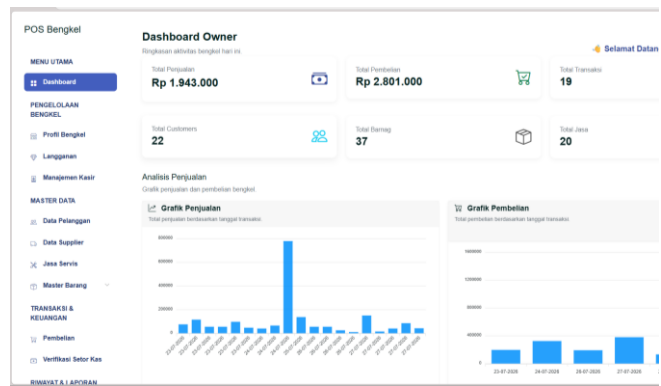
2.3.2 Halaman Dashboard Berdasarkan Hak Akses



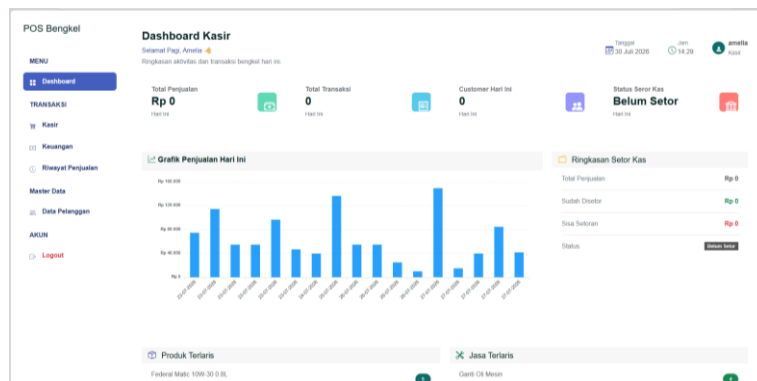
Dashboard Administrator

A. Halaman Dashboard Administrator POS Bengkel

Dashboard Pemilik Bengkel



B. Halaman Dashboard Pemilik Bengkel POS Bengkel

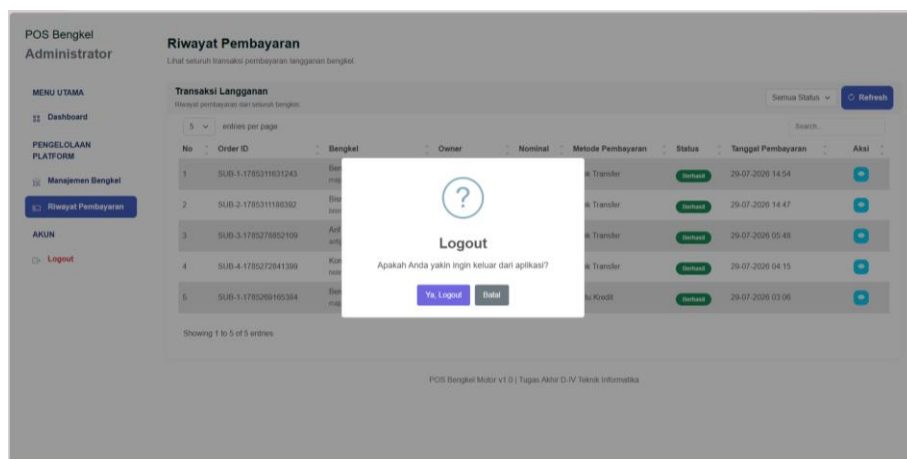


Dashboard Kasir

C. Halaman Dashboard Kasir POS Bengkel

2.4 Keluar dari Aplikasi

- 1 **Pilih menu Logout** — Klik Logout pada sidebar.
- 2 **Konfirmasi** — Klik Ya, Logout pada kotak konfirmasi.
- 3 **Kembali ke halaman login** — Tunggu sampai proses logout selesai.



Gambar 2.3 Tampilan Konfirmasi Proses Logout Aplikasi POS Bengkel

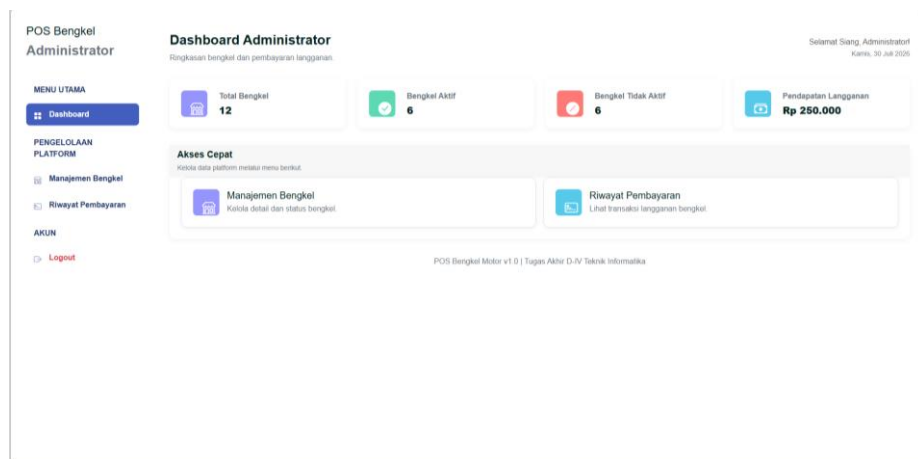
03

PANDUAN ADMINISTRATOR

Panduan memantau bengkel dan pembayaran langganan.

3.1 Dashboard Administrator

Dashboard Administrator tampil setelah administrator berhasil masuk ke sistem. Halaman ini menampilkan jumlah bengkel terdaftar, bengkel aktif dan tidak aktif, serta total pendapatan dari pengguna langganan aplikasi.



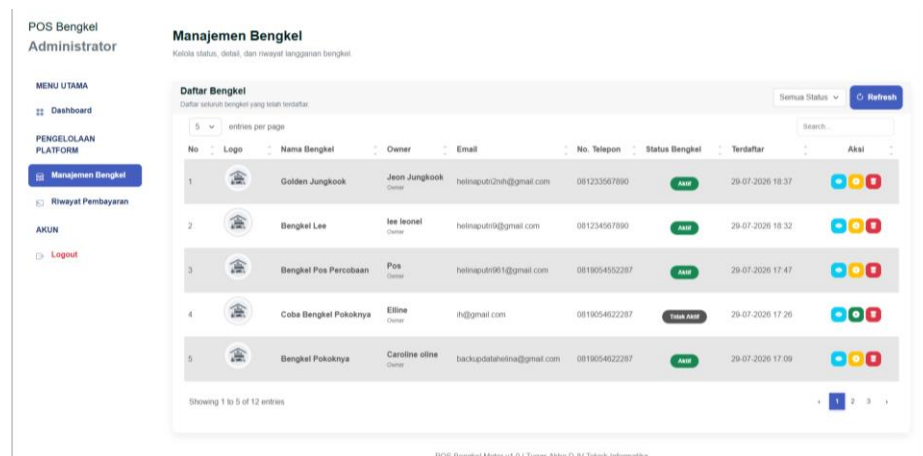
Gambar 3.2 Halaman Dashboard Administrator POS Bengkel

3.2 Membuka Daftar Bengkel

- 1 **Pilih menu** — Klik Manajemen Bengkel pada sidebar.
- 2 **Lihat data bengkel** — Sistem menampilkan seluruh bengkel yang terdaftar.
- 3 **Filter berdasarkan status** — Pilih status bengkel pada menu Semua Status.
- 4 **Kelola data bengkel** — Gunakan tombol aksi untuk melihat detail, mengubah status, atau menghapus data.

5 **Perbarui tampilan** — Klik Refresh untuk memuat ulang data.

6 **Pindah halaman** — Klik nomor halaman atau tanda panah untuk melihat data lainnya.



Gambar 3.3 Halaman Menu Manajemen Bengkel

3.3 Mengubah Status Bengkel atau Langgan

1 **Pilih bengkel** — Tentukan bengkel yang statusnya akan diubah.

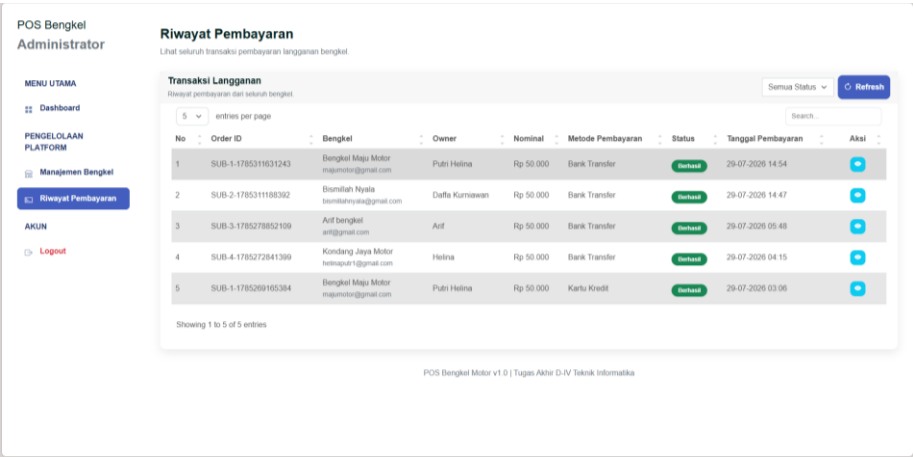
2 **Klik tombol status** — Gunakan tombol aktifkan atau nonaktifkan pada kolom Aksi.

3 **Konfirmasi perubahan** — Setujui perubahan pada kotak konfirmasi.

4 **Periksa hasil** — Pastikan status pada tabel sudah berubah.

3.4 Melihat Riwayat Pembayaran

- 1 **Pilih menu** — Klik Riwayat Pembayaran pada sidebar.
- 2 **Lihat data pembayaran** — Sistem menampilkan riwayat pembayaran langganan bengkel.
- 3 **Filter berdasarkan status** — Pilih status pembayaran pada menu Semua Status.
- 4 **Cari data pembayaran** — Gunakan kolom Search untuk mencari transaksi tertentu.
- 5 **Lihat detail** — Klik ikon mata pada kolom Aksi.
- 6 **Perbarui data** — Klik Refresh untuk memuat ulang daftar pembayaran.



POS Bengkel Administrator

Riwayat Pembayaran
Lihat seluruh transaksi pembayaran langganan bengkel.

Transaksi Langganan

5 entries per page

Semua Status Refresh

No	Order ID	Bengkel	Owner	Nominal	Metode Pembayaran	Status	Tanggal Pembayaran	Aksi
1	SUB-1-1785311031243	Bengkel Maju Motor majumotor@gmail.com	Putri Helina	Rp 50.000	Bank Transfer	Selesai	29-07-2026 14:54	👁️
2	SUB-2-1785311108392	Bismillah Nyata bismillahnyata@gmail.com	Daffa Kurniawan	Rp 50.000	Bank Transfer	Selesai	29-07-2026 14:47	👁️
3	SUB-3-1785278852109	Arit bengkel arit@gmail.com	Arit	Rp 50.000	Bank Transfer	Selesai	29-07-2026 05:48	👁️
4	SUB-4-1785272641399	Kondang Jaya Motor helenapuri1@gmail.com	Helina	Rp 50.000	Bank Transfer	Selesai	29-07-2026 04:15	👁️
5	SUB-1-1785269165384	Bengkel Maju Motor majumotor@gmail.com	Putri Helina	Rp 50.000	Kartu Kredit	Selesai	29-07-2026 03:06	👁️

Showing 1 to 5 of 5 entries

POS Bengkel Motor v1.0 | Tugas Akhir D-IV Teknik Informatika

Gambar 3.4 Tampilan Menu Riwayat Pembayaran Langganan Aplikasi POS Bengkel

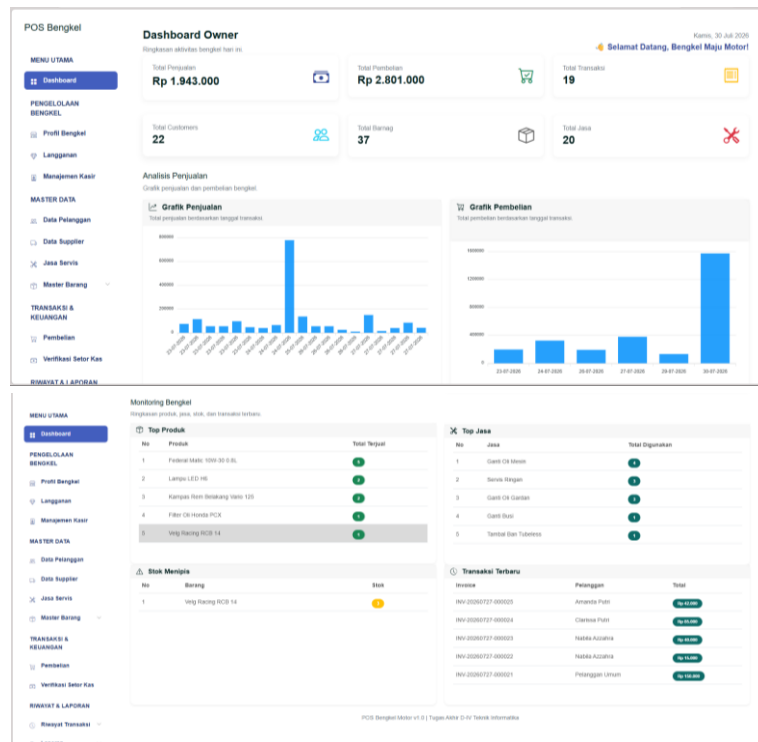
04

PANDUAN PEMILIK BENGKEL

Panduan mengelola kegiatan operasional bengkel.

4.1 Melihat Dashboard Pemilik Bengkel

Dashboard Pemilik Bengkel menampilkan ringkasan penjualan, pembelian, transaksi, pelanggan, barang, dan jasa. Pemilik bengkel juga dapat melihat grafik penjualan dan pembelian serta bagian Monitoring Bengkel yang berisi Top Produk, Top Jasa, Stok Menipis, dan Transaksi Terbaru.



Gambar 4.1 Halaman Dashboard Pemilik Bengkel

4.2 Mengubah Profil Bengkel

Tujuan menu: memperbarui logo dan informasi bengkel.

- 1 **Buka menu** — Klik Profil Bengkel pada sidebar.
- 2 **Ubah logo** — Klik Choose File, lalu pilih gambar logo bengkel.
- 3 **Perbarui data** — Ubah nama bengkel, email, nomor telepon, atau alamat.
- 4 **Periksa kembali data** — Pastikan informasi yang dimasukkan sudah benar.
- 5 **Simpan perubahan** — Klik Simpan Perubahan.

POS Bengkel

PROFIL BENGKEL

Dashboard / Profil Bengkel

Informasi Bengkel
Kekada informasi utama bengkel yang akan digunakan pada sistem POS.

Logo: [Image of a butterfly logo]
Choose File No file chosen

Form Fields:

- Nama Bengkel:
- Email:
- No. Telepon: Status: ADD
- Alamat:

Simpan Perubahan

POS Bengkel Motor v1.0 | Tugas Akhir D-IV Teknik Informatika

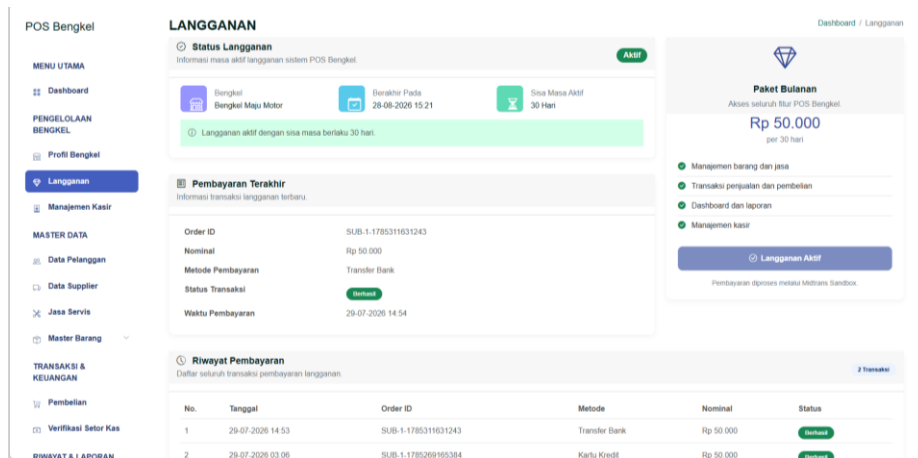
Gambar 4.2 Halaman Profil Bengkel

4.3 Mengelola Langganan

Tujuan menu: melihat status, riwayat pembayaran, dan memperpanjang masa langganan.

1	Buka menu — Klik Langganan pada sidebar.
2	Lihat status langganan — Periksa status, tanggal berakhir, dan sisa masa aktif.
3	Cek pembayaran terakhir — Lihat order ID, nominal, metode, status, dan waktu pembayaran.
4	Lihat riwayat pembayaran — Gulir ke bagian Riwayat Pembayaran.
5	Lakukan pembayaran — Klik tombol pembayaran saat ingin memperpanjang langganan.
6	Periksa status — Pastikan status aktif dan masa langganan sudah bertambah.

!	Penting Pemilik bengkel harus memiliki langganan aktif untuk menggunakan seluruh fitur POS Bengkel.
---	--

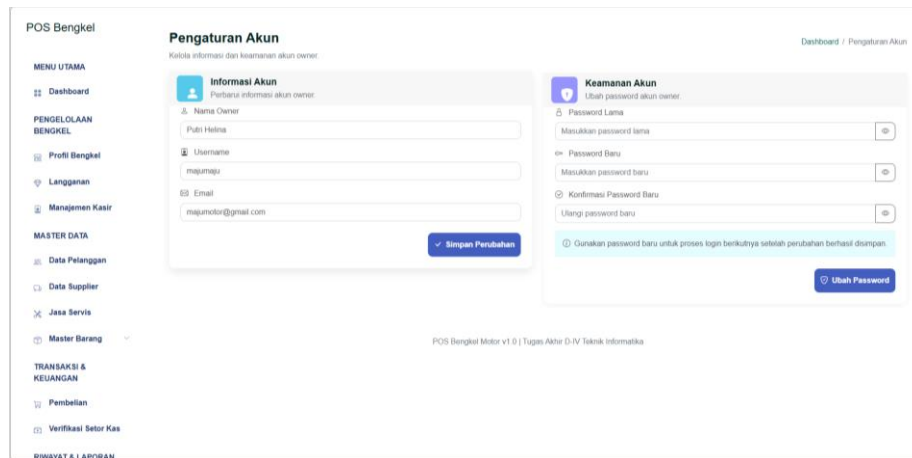


Gambar 4.3 Halaman Berlangganan Aplikasi

4.4 Mengatur Akun Pemilik Bengkel

Tujuan menu: memperbarui informasi akun pemilik bengkel dan mengganti password.

- 1 **Buka menu** — Klik Pengaturan Akun.
- 2 **Ubah informasi akun** — Perbarui nama pemilik, username, atau email.
- 3 **Simpan data akun** — Klik Simpan Perubahan.
- 4 **Isi password lama** — Masukkan password yang sedang digunakan.
- 5 **Buat password baru** — Masukkan dan konfirmasi password baru.
- 6 **Simpan password** — Klik Ubah Password.



Gambar 4.4 Halaman Pengaturan Akun Pemilik Bengkel

4.5 Mengelola Akun Kasir

Tujuan menu: menambah, melihat, mengubah, dan menghapus akun Kasir.

- 1 **Buka menu** — Klik Manajemen Kasir pada sidebar.
- 2 **Lihat daftar kasir** — Sistem menampilkan data Kasir beserta status akunnya.
- 3 **Tambah kasir** — Klik Tambah Kasir, lalu lengkapi datanya.
- 4 **Cari data kasir** — Gunakan kolom Search untuk menemukan akun tertentu.
- 5 **Ubah data kasir** — Klik ikon pensil pada kolom Aksi.
- 6 **Hapus akun kasir** — Klik ikon tempat sampah, lalu konfirmasi penghapusan.

POS Bengkel

MENU UTAMA

Dashboard

PENGELOLAAN BENGKEL

Profil Bengkel

Langganan

Manajemen Kasir

MASTER DATA

Data Pelanggan

Data Supplier

Jasa Servis

Master Barang

TRANSAKSI & KEUANGAN

Pembelian

Verifikasi Setor Kas

Riwayat Laporan

MANAJEMEN KASIR

Dashboard / Manajemen Kasir

Daftar Kasir

Kelola akun kasir yang memiliki akses ke sistem POS

Tambah Kasir

Total Kasir

7

Total akun kasir terdaftar

Kasir Aktif

6

Akun kasir aktif

Kasir Nonaktif

1

Akun kasir nonaktif

10 entries per page

Search...

No	Nama	Username	Email	Status	Aksi
1	Kasir Empat	kasirempat	kasirempat@gmail.com	Aktif	 
2	Kasir Tiga	kasirtiga	kasirtiga@gmail.com	Aktif	 
3	Kasir	Kasirduaa	kasir@gmail.com	Aktif	 
4	Kasir Helina Putri	helinaputri	helina1@gmail.com	Aktif	 
5	Diana Rahma	dianaa	dianahma@gmail.com	Aktif	 
6	Helina Putri	helina	helina@gmail.com	Nonaktif	 
7	Amelia Putri	amelina	amelina@gmail.com	Aktif	 

Gambar 4.5 Halaman Manajemen Kasir POS Bengkel

05

PANDUAN DATA MASTER

Data dasar yang digunakan dalam transaksi dan laporan.

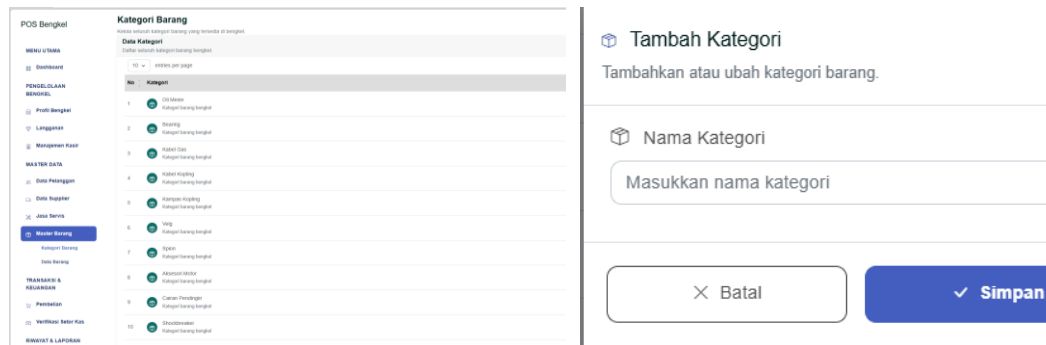
5.1 Mengelola Kategori Barang

Tujuan menu: mengelompokkan barang berdasarkan jenisnya agar lebih rapi dan mudah dicari.

- | | |
|---|--|
| 1 | Buka menu — Klik Master Barang, lalu pilih Kategori Barang. |
| 2 | Lihat daftar kategori — Sistem menampilkan kategori yang sudah tersimpan. |
| 3 | Cari kategori — Gunakan kolom Search untuk menemukan kategori tertentu. |
| 4 | Tambah kategori — Klik Tambah, lalu masukkan nama kategori. |
| 5 | Ubah kategori — Klik ikon pensil pada kolom Aksi. |
| 6 | Hapus kategori — Klik ikon tempat sampah, lalu konfirmasi penghapusan. |



Peringatan Pastikan kategori tidak sedang digunakan oleh data barang sebelum menghapusnya.



Gambar 5.1 Tampilan Halaman Data Kategori Barang

5.2 Menambahkan Data Barang

Tujuan menu: mencatat dan mengelola seluruh barang yang tersedia di bengkel.

- 1 **Buka menu** — Klik Master Barang, lalu pilih Data Barang.
- 2 **Klik Tambah Barang** — Formulir data barang akan ditampilkan.
- 3 **Isi data barang** — Lengkapi kategori, barcode, nama, stok, stok minimum, harga beli, dan harga jual.
- 4 **Periksa kembali data** — Pastikan stok dan harga sudah sesuai.
- 5 **Klik Simpan** — Sistem menyimpan barang dan menampilkan notifikasi berhasil.

POS Bengkel

Menu Utama

Dashboard

Pengelolaan Bengkel

Profil Bengkel

Layanan

Manajemen Kasir

Manajemen Karyawan

Data Pelanggan

Data Supplier

Data Barang

Manajemen Barang

Manajemen Keuangan

Manajemen Laporan

Data Barang

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

06

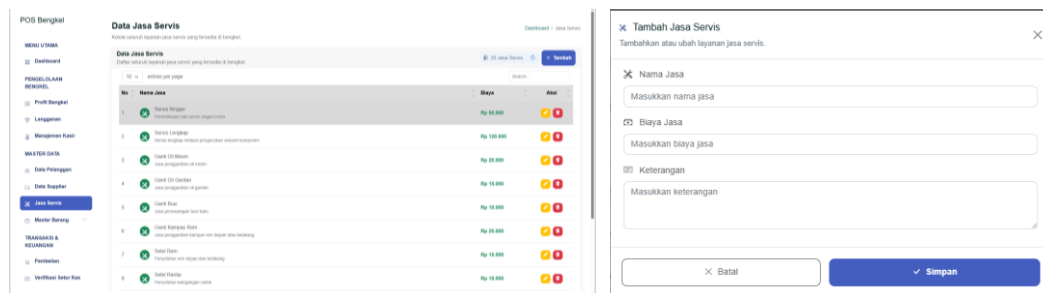
JASA, PEMASOK, PELANGGAN, DAN KENDARAAN

Data pendukung kegiatan operasional bengkel.

6.1 Menambahkan Jasa Servis

Tujuan menu: mencatat dan mengelola layanan servis yang tersedia di bengkel.

- 1 **Buka menu** — Klik Jasa Servis pada sidebar.
- 2 **Klik Tambah Jasa** — Formulir tambah jasa akan ditampilkan.
- 3 **Isi data jasa** — Lengkapi nama jasa, harga, dan keterangan.
- 4 **Periksa kembali data** — Pastikan informasi jasa sudah benar.
- 5 **Klik Simpan** — Sistem menyimpan jasa dan menampilkan notifikasi berhasil.

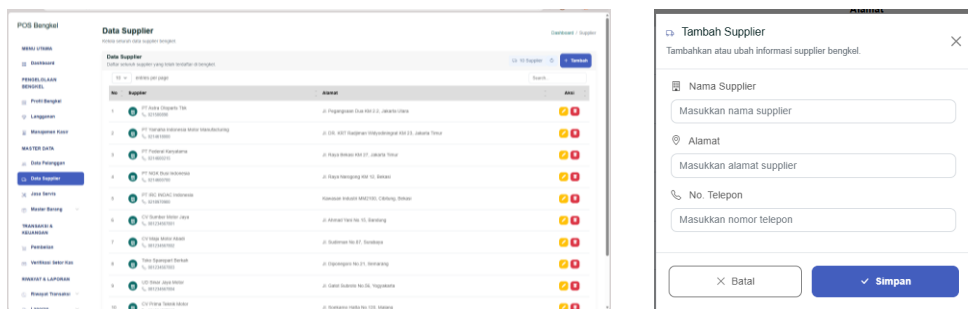


Gambar 6.1 Tampilan Halaman Jasa Servis

6.2 Menambahkan Data Pemasok

Tujuan menu: mencatat pemasok barang yang digunakan dalam proses pembelian.

- 1 **Buka menu** — Klik Data Pemasok pada sidebar.
- 2 **Klik Tambah** — Formulir tambah pemasok akan ditampilkan.
- 3 **Isi data pemasok** — Lengkapi nama, alamat, dan nomor telepon pemasok.
- 4 **Periksa kembali data** — Pastikan seluruh informasi sudah benar.
- 5 **Klik Simpan** — Sistem menyimpan pemasok dan menampilkan notifikasi berhasil.

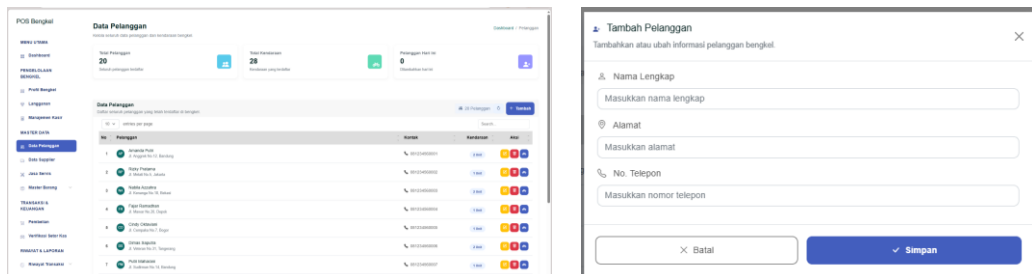


Gambar 6.2 Tampilan Halaman Data Pemasok

6.3 Menambahkan Data Pelanggan

Tujuan menu: mencatat pelanggan untuk kebutuhan transaksi dan riwayat servis.

- 1 **Buka menu** — Klik Data Pelanggan pada sidebar.
- 2 **Klik Tambah** — Formulir tambah pelanggan akan ditampilkan.
- 3 **Isi data pelanggan** — Lengkapi nama, alamat, dan nomor telepon pelanggan.
- 4 **Periksa kembali data** — Pastikan seluruh informasi sudah benar.
- 5 **Klik Simpan** — Sistem menyimpan pelanggan dan menampilkan notifikasi berhasil.



Gambar 6.3 Tampilan Halaman Data Pelanggan

6.4 Menambahkan Data Kendaraan

Tujuan menu: mencatat kendaraan milik pelanggan yang digunakan saat transaksi servis.

- 1 **Buka data pelanggan** — Klik Data Pelanggan, lalu pilih pelanggan.

Klik ikon kendaraan — Sistem menampilkan daftar kendaraan milik pelanggan.

Tambah kendaraan — Isi nomor polisi, merek, tipe, tahun, dan warna kendaraan.

Periksa kembali data — Pastikan seluruh informasi sudah benar.

Klik Simpan — Sistem menyimpan kendaraan dan menampilkan notifikasi berhasil.

Gambar 6.4 Tampilan Halaman Data Kendaraan Pelanggan

07

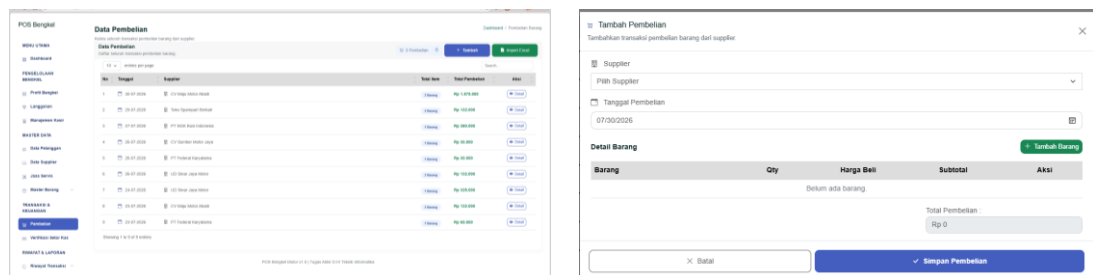
PANDUAN PEMBELIAN

Panduan mencatat pembelian barang dari pemasok.

7.1 Mencatat Pembelian Barang

Tujuan menu: mencatat barang masuk dari pemasok dan memperbarui stok barang.

- Buka menu** — Klik Pembelian pada sidebar.
- Klik Tambah** — Formulir Tambah Pembelian akan ditampilkan.
- Pilih pemasok dan tanggal** — Tentukan pemasok serta tanggal pembelian.
- Tambahkan barang** — Pilih barang, lalu isi jumlah dan harga beli.
- Simpan pembelian** — Periksa total, lalu klik Simpan Pembelian.



Gambar 7.1 Tampilan Halaman Data Pembelian Barang

7.2 Mengimpor Pembelian dari Excel

Tujuan menu: memasukkan banyak data pembelian sekaligus menggunakan file Excel.

1

Buka form impor — Pada halaman Pembelian, klik Import Excel.

2

Unduh template — Klik Download Template Excel.

3

Isi file Excel — Lengkapi kolom Barcode, Jumlah, dan Harga Beli.

4

Pilih data impor — Tentukan pemasok, tanggal pembelian, dan file Excel.

5

Impor file — Klik Import File, lalu perbaiki data jika muncul pesan validasi.

Import Pembelian

Import data pembelian menggunakan template Excel.

Supplier

Pilih Supplier

Tanggal Pembelian

07/30/2026

File Excel

Choose File No file chosen

Format file harus .xlsx

Gunakan template Excel agar proses import berjalan dengan benar.

Download Template Excel

Batal Import File

Gambar 7.2 Formulir Import Excel Data Pembelian

08

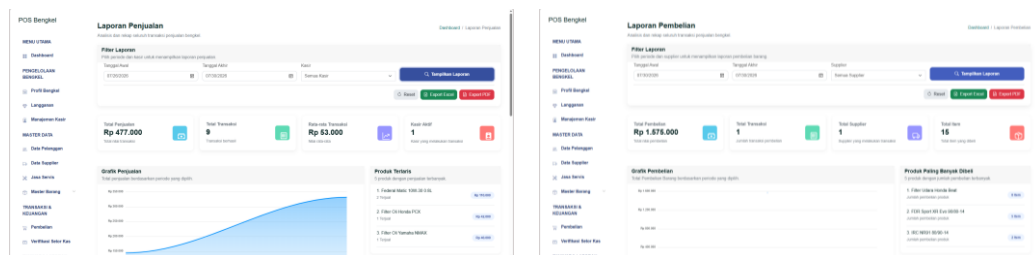
LAPORAN DAN KONFIRMASI SETOR KAS

Panduan melihat laporan dan memeriksa setoran Kasir.

8.1 Melihat Laporan

Tujuan menu: melihat ringkasan transaksi dan mengunduh laporan sesuai periode.

- Buka menu** — Klik Laporan, lalu pilih jenis laporan.
- Tentukan periode** — Pilih tanggal awal dan tanggal akhir.
- Gunakan filter** — Pilih filter tambahan sesuai kebutuhan.
- Tampilkan laporan** — Terapkan filter untuk memuat data.
- Unduh laporan** — Klik Export Excel atau Export PDF.

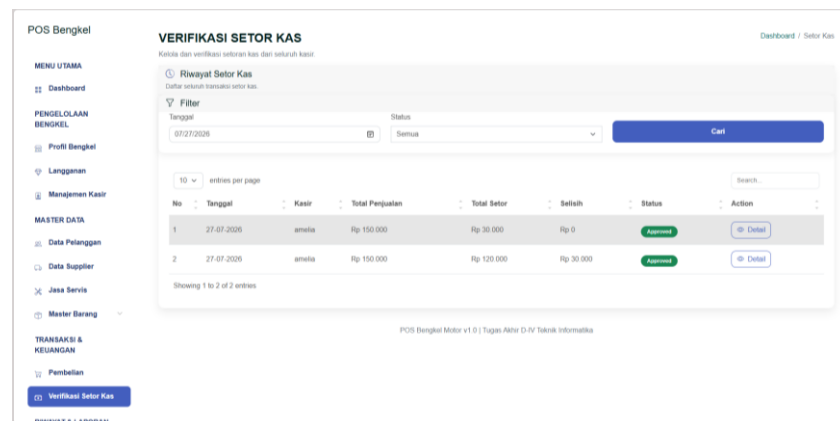


Gambar 8.1 Tampilan Halaman Laporan Penjualan dan Pembelian Barang

8.2 Mengonfirmasi Setor Kas

Tujuan menu: memeriksa dan mengonfirmasi setoran kas yang dikirim oleh Kasir.

- 1 **Buka menu** — Klik Verifikasi Setor Kas pada sidebar.
- 2 **Lihat daftar setoran** — Pilih setoran yang belum diperiksa.
- 3 **Periksa detail** — Bandingkan total penjualan dan jumlah setoran.
- 4 **Beri keputusan** — Konfirmasi atau tolak setoran sesuai hasil pemeriksaan.
- 5 **Periksa status** — Pastikan status setoran sudah berubah.



Gambar 8.2 Tampilan Halaman Verifikasi Setor Kas

09

PANDUAN KASIR

Panduan transaksi penjualan, riwayat, dan setor kas.

9.1 Melihat Dashboard Kasir

Tujuan menu: melihat ringkasan aktivitas penjualan dan status setor kas.

1

Buka dashboard — Dashboard tampil setelah Kasir berhasil login.

2

Lihat ringkasan — Periksa total penjualan, transaksi, dan status setor kas.

3

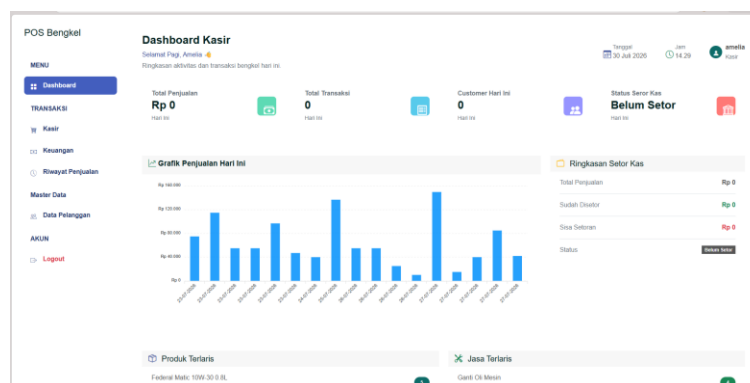
Cek grafik penjualan — Lihat perkembangan penjualan harian.

4

Lihat produk dan jasa — Periksa produk serta jasa yang paling sering terjual.

5

Periksa status setoran — Pastikan setoran hari ini sudah dibuat atau diverifikasi.

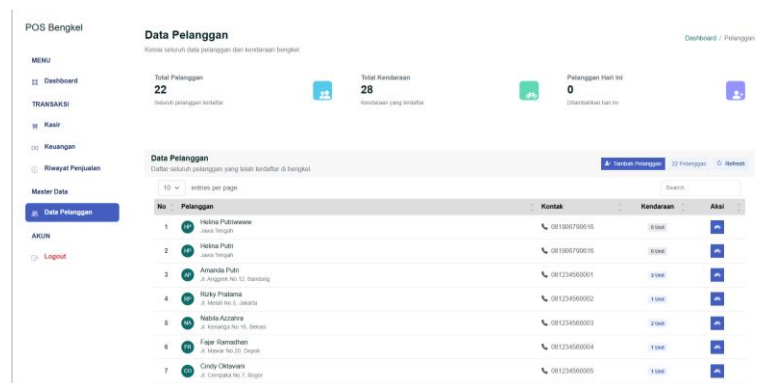


Gambar 9.1 Tampilan Halaman Dashboard Kasir

9.2 Menambahkan Pelanggan dan Kendaraan

Tujuan menu: mencatat pelanggan dan kendaraannya sebelum melakukan transaksi servis.

- 1 **Buka menu** — Klik Data Pelanggan pada sidebar.
- 2 **Tambah pelanggan** — Klik Tambah, lalu isi data pelanggan.
- 3 **Simpan pelanggan** — Periksa data, lalu klik Simpan.
- 4 **Buka data kendaraan** — Klik ikon kendaraan pada pelanggan yang dipilih.
- 5 **Tambah kendaraan** — Isi data kendaraan, lalu klik Simpan.



Gambar 9.2 Tampilan Halaman Data Pelanggan Milik Kasir

9.3 Membuat Transaksi Penjualan

Tujuan menu: mencatat penjualan barang dan jasa kepada pelanggan.

- 1 **Buka menu** — Klik Kasir pada sidebar.
- 2 **Pilih data transaksi** — Pilih pelanggan, kendaraan, barang, atau jasa yang diperlukan.

3

Periksa pembayaran — Pastikan total, uang bayar, dan kembalian sudah benar.

4

Simpan transaksi — Klik Simpan Transaksi, lalu cetak struk jika diperlukan.

The screenshot shows the 'KASIR' (Cashier) interface of the 'POS Bengkel' system. On the left is a sidebar menu with 'KASIR' selected. The main area is divided into sections: 'Caril Barang' (Search Item) with a search bar, 'Keranjang Penjualan' (Sales Cart) with a table of items, and 'Pembayaran' (Payment) section at the bottom with fields for 'Total Belanja' (Total Spent), 'Nominal Bayar' (Amount Paid), and 'Kembalian' (Change).

Gambar 9.3 Tampilan Halaman Transaksi Penjualan

9.4 Melihat Riwayat Penjualan

Tujuan menu: melihat transaksi penjualan yang sudah tersimpan.

1

Buka menu — Klik Riwayat Penjualan pada sidebar.

2

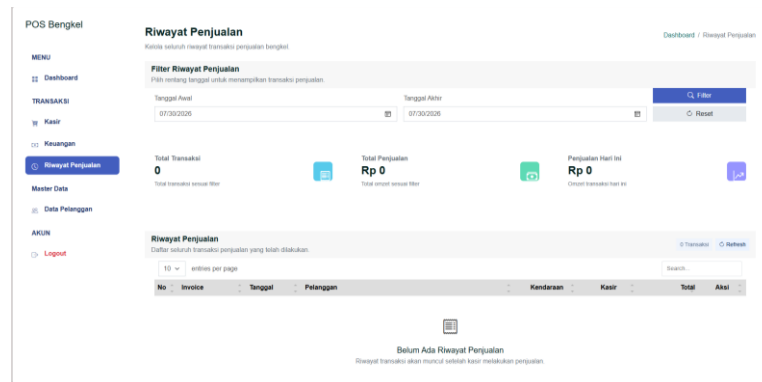
Atur pencarian — Gunakan periode tanggal atau kolom Search.

3

Lihat daftar transaksi — Periksa invoice, pelanggan, total, dan tanggal transaksi.

4

Lihat detail — Klik tombol Detail pada transaksi yang dipilih.



Gambar 9.4 Tampilan Halaman Riwayat Penjualan Kasir

9.5 Melakukan Setor Kas

Tujuan menu: mencatat penyerahan uang hasil penjualan kepada pemilik bengkel.

- 1 **Buka menu** — Klik keuangan pada sidebar.
- 2 **Periksa total penjualan** — Pastikan nominal penjualan hari ini sudah sesuai.
- 3 **Isi data setoran** — Masukkan jumlah setor dan catatan jika diperlukan.
- 4 **Kirim setoran** — Klik Simpan dan tunggu verifikasi dari pemilik bengkel.

POS Bengkel

MENU

Dashboard

TRANSAKSI

Rasio

Kasir

Riwayat Perputaran

Master Data

Data Pelanggan

AKUN

Logout

SETOR KAS

Kasir

Tidak Perputaran Kasir 00

Rp 0

0 kali setor

Tidak Diverifikasi

Rp 0

0 kali setor

Tidak Diverifikasi

Rp 0

0 kali setor

Status: Diverifikasi

Disetujui

Menunggu konfirmasi owner

Form Setor Kas

Tidak Perputaran Kasir 00

Rp 0

Nominal Setor

Masukkan nominal setor

Catatan

Setor Kas

Riwayat Setor Kas

Catatan riwayat transaksi setor kas

10 records per page

No	Tanggal	Nominal Setor	Catatan	Status	Diverifikasi Oleh	Diverifikasi Pada	Action
1	27.07.2020	Rp 50.000	catatan setor kas 50.000	Menunggu		27.07.2020	Detail

Informasi

1. Lakukan setor kas minimal 1 kali setiap hari

Anda dapat melakukan setor kas satu kali per harinya

2. Menunggu konfirmasi owner

Anda akan mendapat notifikasi jika owner telah dikonfirmasi

3. Pastikan nominal sesuai

Harap pastikan nominal sesuai sebelum melakukan setor

Gambar 9.5 Tampilan Halaman Setor Kas Kasir

10

PENUTUP

Bagian akhir manual book penggunaan aplikasi POS Bengkel.

10.1 Penutup

Manual book ini disusun untuk membantu pengguna memahami penggunaan Aplikasi POS Bengkel sesuai dengan hak akses masing-masing. Ikuti langkah-langkah pada setiap menu agar pencatatan data, transaksi, dan laporan dapat dilakukan dengan benar.

Dengan adanya panduan ini, diharapkan Administrator, Pemilik Bengkel, dan Kasir dapat menggunakan aplikasi secara lebih mudah dan dapat memanfaatkan fitur yang tersedia secara optimal. Sehingga proses kegiatan operasional bengkel dapat berjalan lebih efektif.

TERIMA KASIH

POS BENGKEL • Panduan Pengguna

DOKUMEN TEKNIKAL POS BENGKEL

Dokumentasi teknis Aplikasi POS Bengkel
Berbasis Web.

2026

D-IV TEKNIK
INFORMATIKA

UNIVERSITAS
HARKAT NEGERI

Disusun oleh

Helina Putri

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

Daftar Isi

Daftar Isi	2
1. Profil	3
2. Latar Belakang	4
3. Manfaat	5
4. Spesifikasi Teknis	6
5. Source Code	7
6. Penutup	12

1.1 Profil Aplikasi

Aplikasi POS Bengkel merupakan aplikasi berbasis web yang dirancang untuk membantu kegiatan operasional bengkel secara terintegrasi. Aplikasi ini dapat digunakan oleh beberapa bengkel dengan data masing-masing bengkel yang terpisah. Sistem memiliki tiga jenis pengguna, yaitu Administrator, Pemilik Bengkel, dan Kasir dengan hak akses yang berbeda.

Aplikasi menyediakan fitur pengelolaan data barang, jasa servis, pemasok, pelanggan dan kendaraan, transaksi pembelian, transaksi penjualan, pengelolaan kasir, penyetoran kas, pembayaran langganan, serta laporan. Dalam pengembangannya, aplikasi POS Bengkel menggunakan metode Waterfall yang terdiri dari tahap analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan.

Dengan adanya aplikasi ini diharapkan kegiatan pencatatan transaksi dan pengelolaan data bengkel dapat dilakukan secara lebih terstruktur serta memudahkan Pemilik Bengkel dalam memantau kegiatan operasional.

1.2 Kelompok Pengguna

1	Administrator — memantau data bengkel, status aktif atau tidak aktif, serta riwayat pembayaran langganan.
2	Pemilik Bengkel — mengelola profil bengkel, data master, pembelian, kasir, laporan, dan konfirmasi setor kas.
3	Kasir — melakukan transaksi penjualan, mengelola data pelanggan dan kendaraan yang diperlukan, melihat riwayat transaksi, serta melakukan penyetoran kas.

2.1 Latar Belakang

Bengkel merupakan salah satu usaha jasa yang tidak hanya memberikan pelayanan perbaikan dan perawatan kendaraan, tetapi juga melakukan kegiatan penjualan suku cadang, pembelian barang dari pemasok, pengelolaan persediaan, serta pencatatan transaksi pelanggan. Banyaknya aktivitas tersebut menyebabkan bengkel membutuhkan pengelolaan data yang baik agar kegiatan operasional dapat berjalan secara efektif.

Pada proses pengelolaan secara manual, pencatatan transaksi penjualan, pembelian barang, persediaan, pelanggan, serta jasa servis dapat menimbulkan berbagai permasalahan. Data yang dicatat secara terpisah berisiko mengalami kesalahan pencatatan, kehilangan data, kesulitan dalam mengetahui jumlah stok barang, serta membutuhkan waktu lebih lama ketika Pemilik Bengkel ingin mengetahui hasil transaksi atau membuat laporan.

Selain itu, proses transaksi penjualan juga membutuhkan pencatatan yang cepat dan akurat. Kasir perlu mengetahui barang dan jasa yang digunakan dalam transaksi, jumlah pembayaran pelanggan, serta riwayat transaksi yang telah dilakukan. Pemilik Bengkel juga membutuhkan informasi mengenai pembelian barang, penjualan, persediaan, serta hasil setoran kas dari Kasir.

Untuk membantu mengatasi permasalahan tersebut, dibutuhkan sebuah aplikasi Point of Sale (POS) berbasis web yang dapat mengintegrasikan kegiatan operasional bengkel dalam satu sistem. Melalui aplikasi ini, data barang, jasa servis, pemasok, pelanggan, kendaraan, transaksi pembelian, dan transaksi penjualan dapat dikelola secara terpusat.

Aplikasi POS Bengkel juga dilengkapi dengan pembagian hak akses pengguna. Administrator digunakan untuk memantau bengkel yang terdaftar dan pembayaran langganan. Pemilik Bengkel dapat mengelola data dan memantau kegiatan operasional bengkel, sedangkan Kasir dapat melakukan

transaksi penjualan, mengelola data pelanggan dan kendaraan yang diperlukan, melihat riwayat transaksi, serta melakukan penyetoran kas.

Dalam pengembangan aplikasi POS Bengkel digunakan metode Waterfall. Metode ini dilakukan secara bertahap mulai dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, hingga pemeliharaan. Penggunaan metode Waterfall membantu proses pengembangan dilakukan secara terstruktur dan setiap tahapan dapat terdokumentasi dengan baik.

Berdasarkan permasalahan tersebut, dikembangkan Aplikasi POS Bengkel Berbasis Web yang diharapkan dapat membantu pengelolaan data dan transaksi bengkel secara terintegrasi serta mempermudah Pemilik Bengkel dan Kasir dalam menjalankan kegiatan operasional.

3.1 Manfaat Aplikasi

1	Memudahkan Pemilik Bengkel dalam mengelola data barang, jasa servis, pemasok, pelanggan, kendaraan, dan data Kasir.
2	Membantu Kasir dalam melakukan transaksi penjualan secara lebih cepat dan terstruktur.
3	Membantu Pemilik Bengkel dalam mencatat transaksi pembelian dan memantau persediaan barang.
4	Memudahkan proses pencatatan dan konfirmasi penyetoran kas dari Kasir kepada Pemilik Bengkel.
5	Membantu Pemilik Bengkel memperoleh informasi transaksi dan laporan kegiatan operasional bengkel.
6	Membantu Administrator dalam memantau data bengkel dan riwayat pembayaran langganan aplikasi.

4.1 Perangkat Lunak dan Teknologi

Spesifikasi teknis yang digunakan dalam membangun Aplikasi POS Bengkel meliputi:

No.	Teknologi / Perangkat Lunak
1	Python 3.9.13
2	Framework Flask 3.1.3
3	HTML
4	CSS
5	JavaScript
6	Visual Studio Code
7	MySQL/MariaDB
8	XAMPP 10.4.32
9	Flask-SQLAlchemy 3.1.1
10	Flask-JWT-Extended 4.7.4
11	Flask-Migrate 4.1.0 / Alembic
12	Midtrans
13	OpenPyXL
14	ReportLab
15	Web Browser

4.2 Source Code

Pada bagian *source code* akan dijelaskan beberapa kode utama yang digunakan dalam membangun Aplikasi POS Bengkel. *Source code* yang ditampilkan difokuskan pada bagian yang mewakili proses penting dalam sistem, baik pada sisi *backend* menggunakan Python dan Flask maupun pada sisi *frontend* menggunakan HTML dan *JavaScript*.

5.1 Registrasi Akun Pemilik Bengkel

```

18 # SIGN UP ----- Begin
19 def signup(datas):
20     try:
21         # Validation Request Body ----- Start
22         if datas == None:
23             return invalid_params()
24
25         required_data = [
26             "owner_name",
27             "username",
28             "email",
29             "password",
30             "retype_password",
31             "workshop_name",
32             "workshop_address",
33             "workshop_phone"
34         ]
35         for req in required_data:
36             if req not in datas:
37                 return parameter_error(f"Missing {req} in request body.")
38         # Validation Request Body ----- Finish
39
40         # Initialize Data Input ----- Start
41         owner_name = datas["owner_name"].strip()
42         username = datas["username"].strip()
43         email = datas["email"].strip().lower()
44         password = datas["password"]
45         retype_password = datas["retype_password"]
46         workshop_name = datas["workshop_name"]
47         workshop_address = datas["workshop_address"]
48         workshop_phone = datas["workshop_phone"]
49         # Initialize Data Input ----- Finish
50
51         # Data Validation ----- Start
52         checker_result = signup_validator(owner_name, username, email, password, retype_password, workshop_name, workshop_address, workshop_phone)
53
54         if len(checker_result) != 0:
55             return defined_error(checker_result, "Bad Request", status_code=400)
56         # Data Validation ----- Finish
57
58         # Insert Data ----- Start
59         # Initialize
60         password_encrypt = hash_password(password)
61         timestamp = current_timestamp()
62         verification_token = secrets.token_urlsafe(32)
63         verification_token_expired_at = timestamp + (24 * 60 * 60 * 1000)
64
65         user_data = Users(
66             owner_name=owner_name,
67             username=username,
68             email=email,
69             email_verified_at=None,
70             verification_token=verification_token,
71             verification_token_expired_at=verification_token_expired_at,
72             password=password_encrypt,
73             role=1,
74             is_active=1,
75             created_at=timestamp,
76             updated_at=timestamp
77         )
78
79         # Save Data
80         db.session.add(user_data)
81         db.session.commit()
82         # Insert Data ----- Finish
83

```

Gambar 5.1 Code Registrasi Akun Pemilik Bengkel

Code pada Gambar 5.1 menjelaskan proses registrasi akun Pemilik Bengkel. Sistem menerima data berupa nama pemilik, *username*, email, *password*, nama bengkel, alamat, dan nomor telepon. Selanjutnya sistem melakukan validasi terhadap data yang dimasukkan melalui *signup_validator()*. Apabila data telah sesuai, *password* dienkripsi, data akun disimpan ke database, kemudian sistem membuat data bengkel yang terhubung dengan akun Pemilik Bengkel.

5.2 Verifikasi Email

```
256 # VERIFY EMAIL ----- Begin
257 def verify_email(token):
258     try:
259         # Token Validation ----- Start
260         if token is None or token.strip() == "":
261             return bad_request(
262                 "Token verifikasi tidak valid."
263             )
264         # Token Validation ----- Finish
265
266         # Initialize Timestamp ----- Start
267         timestamp = current_timestamp()
268         # Initialize Timestamp ----- Finish
269
270         # Get User ----- Start
271         user_data = Users.query.filter_by(
272             verification_token=token,
273             is_delete=0
274         ).first()
275
276         if not user_data:
277             return not_found(
278                 "Token verifikasi tidak ditemukan atau sudah digunakan."
279             )
280         # Get User ----- Finish
281
282         # Token Expiration Validation ----- Start
283         if (
284             user_data.verification_token_expired_at is None or
285             int(user_data.verification_token_expired_at) <= timestamp
286         ):
287             return bad_request(
288                 "Link verifikasi sudah kedaluwarsa."
289             )
290         # Token Expiration Validation ----- Finish
291
292         # Get Workshop ----- Start
293         workshop_data = Workshops.query.filter_by(
294             owner_id=user_data.id,
295             is_delete=0
296         ).first()
297
298         if not workshop_data:
299             return not_found(
300                 "Data bengkel tidak ditemukan."
301             )
302         # Get Workshop ----- Finish
303
304         # Activate Account ----- Start
305         user_data.email_verified_at = timestamp
306         user_data.verification_token = None
307         user_data.verification_token_expired_at = None
308         user_data.is_active = 1
309         user_data.updated_at = timestamp
310
311         workshop_data.is_active = 1
312         workshop_data.updated_at = timestamp
313
314         db.session.commit()
315         # Activate Account ----- Finish
316
317         # Return Response -----
318         return success(
319             message="Verifikasi email berhasil. Silakan login."
320         )
321
322     except Exception as e:
323         db.session.rollback()
324         return bad_request(str(e))
325 # VERIFY EMAIL ----- End
326
```

Gambar 5.2 Code Verifikasi Email

Code pada Gambar 5.2 menjelaskan proses verifikasi email Pemilik Bengkel setelah melakukan registrasi. Sistem memeriksa token verifikasi yang diterima melalui tautan pada email. Apabila token valid, sistem memperbarui status verifikasi dan mengaktifkan akun Pemilik Bengkel serta bengkel yang terdaftar. Setelah proses verifikasi berhasil, Pemilik Bengkel dapat melakukan login ke aplikasi.

5.3 Login

```
552 # SIGN IN ----- Begin
553 def signin(datas):
554     try:
555         # Checking Request Body ----- Start
556         if datas == None:
557             return invalid_params()
558
559         requiredData = ["username", "password"]
560         for req in requiredData:
561             if req not in datas:
562                 return parameter_error(f"Missing {req} in request body.")
563         # Checking Request Body ----- Finish
564
565         # Initialize Data Input ----- Start
566         username = datas["username"].strip().lower()
567         password = datas["password"].strip()
568         # Initialize Data Input ----- Finish
569
570         # Data Validation ----- Start
571         checker_result, result, stts = signin_validator(username, password)
572
573         if len(checker_result) > 0:
574             return defined_error(checker_result, "Bad Request", status_code=stts)
575         # Data Validation ----- Finish
576
577         # Get Workshop ID ----- Start
578         if str(result.role) == "1":
579             ws_id = result.workshops[0].id if result.workshops else None
580
581         elif str(result.role) == "2":
582             cashier = Cashiers.query.filter_by(
583                 user_id=result.id,
584                 is_delete=0
585             ).first()
586
587             ws_id = cashier.workshop_id if cashier else None
588
589         else:
590             ws_id = None
591         # Get Workshop ID ----- Finish
592
593         # Data Payload ----- Start
594         workshop_data = Workshops.query.filter_by(owner_id=result.id, is_delete=0).first()
595         jwt_payload = {
596             "id": result.id,
597             "email": result.email,
598             "name": result.username,
599             "role": result.role,
600             "ws_id": ws_id
601         }
602         # Data Payload ----- Finish
603
604         # Access Token by Email ----- Start
605         access_token = create_access_token(result.email, additional_claims=jwt_payload)
606
607         # Data Response ----- Start
608         response = success_data({
609             "access_token": access_token,
610             "role": result.role,
611             "name": result.username
612         })
613         set_access_cookies(response, access_token)
614         # Data Response ----- Finish
615
616         # Return Response -----
617         return response
618
619     except Exception as e:
620         return bad_request(str(e))
621 # SIGN IN ----- End
```

Gambar 5.3 Code Login

Code pada Gambar 5.3 menjelaskan proses login pengguna menggunakan *username* atau email dan *password*. Sistem melakukan validasi data login melalui *signin_validator()*. Apabila data sesuai, sistem menentukan bengkel berdasarkan role pengguna, kemudian membuat token autentikasi JWT yang berisi identitas dan hak akses pengguna. Token tersebut selanjutnya digunakan sebagai autentikasi pengguna dalam mengakses aplikasi.

5.4 Pembayaran Langganan

```
subscription.py X
apps > routes > models > subscription.py > SubscriptionModels > create_payment

43 # CREATE SUBSCRIPTION PAYMENT ===== Begin
44 def create_payment(user_role, workshop_id, datas):
45     try:
46         # ACCESS VALIDATION =====
47         access = owner_validator(user_role)
48
49         if not access:
50             return authorization_error()
51
52         # REQUEST VALIDATION =====
53         if datas is None:
54             return invalid_params()
55
56         if "package" not in datas:
57             return parameter_error(
58                 "Missing package in request body."
59             )
60
61         package_code = str(datas["package"]).strip().lower()
62
63         if package_code not in SUBSCRIPTION_PACKAGES:
64             return parameter_error(
65                 "Paket langganan tidak tersedia."
66             )
67
68         # CHECK WORKSHOP =====
69         workshop = Workshops.query.filter_by(
70             id=workshop_id,
71             is_delete=0
72         ).first()
73
74         # INITIALIZE PACKAGE ===== Start
75         package = SUBSCRIPTION_PACKAGES[package_code]
76
77         timestamp = current_timestamp()
78
79         order_id = (
80             f'SUB-{workshop_id}-{timestamp}'
81         )
82
83         amount = package["amount"]
84         # INITIALIZE PACKAGE ===== Finish
85
86         # MIDTRANS TRANSACTION DATA ===== Start
87         transaction_data = {
88             "transaction_details": {
89                 "order_id": order_id,
90                 "gross_amount": amount,
91             },
92             "item_details": [
93                 {
94                     "id": package_code,
95                     "price": amount,
96                     "quantity": 1,
97                     "name": package["name"],
98                 }
99             ],
100         }
101         # MIDTRANS TRANSACTION DATA ===== Finish
102
103         # CREATE SNAP TOKEN ===== Start
104         snap = get_snap()
105
106         snap_response = snap.create_transaction(
107             transaction_data
108         )
109
110         snap_token = snap_response.get("token")
111         redirect_url = snap_response.get("redirect_url")
112
113         if not snap_token:
114             return bad_request(
115                 "Gagal mendapatkan Snap Token dari Midtrans."
116             )
117         # CREATE SNAP TOKEN ===== Finish
118
119         # INSERT PAYMENT ===== Start
120         payment = SubscriptionPayments(
121             workshop_id=workshop_id,
122             order_id=order_id,
123             amount=amount,
124             transaction_status="pending",
125             snap_token=snap_token,
126             created_at=timestamp,
127             updated_at=timestamp,
128             is_delete=0,
129         )
130
131         try:
132             db.session.add(payment)
133             db.session.commit()
134         except:
```

Gambar 5.4 Code Pembayaran Langganan

Code pada Gambar 5.4 menjelaskan proses pembayaran langganan yang dilakukan oleh Pemilik Bengkel. Sistem membuat data transaksi pembayaran dan

menghubungkannya dengan layanan pembayaran Midtrans. Setelah pembayaran berhasil, sistem memperbarui status dan masa aktif langganan bengkel.

5.5 Pengelolaan Data Barang

```
apn > node > model > product.py > ...
10
11 # Initialize Data Input ----- Start
12 category_id = data["category_id"]
13 barcode = str(data.get("barcode", "").strip())
14 product_name = data["product_name"].strip()
15 stock = data["stock"]
16 minimum_stock = data["minimum_stock"]
17 purchase_price = data["purchase"]
18 selling_price = data["price"]
19 # Initialize Data Input ----- Finish
20
21 # Data Validation ----- Start
22 checker_result = product_validator(
23     category_id,
24     barcode,
25     product_name,
26     stock,
27     minimum_stock,
28     purchase_price,
29     selling_price,
30     workshop_id
31 )
32
33 if len(checker_result) != 0:
34     return defined_error(
35         checker_result,
36         "defined error",
37     )
38
39 # Data Validation ----- Finish
40
41 # Check Workshop ----- Start
42 workshop = workshops.query.filter_by(
43     id=workshop_id,
44     is_deleted=False
45 ).first()
46
47 if not workshop:
48     return not_found(
49         "workshop could not be found."
50     )
51 # Check Workshop ----- Finish
52
53 # Insert Data ----- Start
54 timestamp = current_timestamp()
55
56 data = Products(
57     workshop_id=workshop_id,
58     category_id=category_id,
59     barcode=barcode if barcode != "" else None,
60     product_name=product_name,
61     stock=stock,
62     minimum_stock=minimum_stock,
63     purchase_price=purchase_price,
64     selling_price=selling_price,
65     created_at=timestamp,
66     updated_at=timestamp
67 )
68
69 try:
70     db.session.add(data)
71     db.session.commit()
72
73 except Exception as e:
74     db.session.rollback()
75     return parameter_error(str(e))
76 # Insert Data ----- Finish
77
```

Gambar 5.5 Code Pengelolaan Data Barang

Code pada Gambar 5.5 menjelaskan proses pengelolaan data barang oleh Pemilik Bengkel. Data yang dikelola meliputi kategori barang, barcode, nama barang, stok, harga beli, dan harga jual. Sistem juga menghubungkan data barang dengan workshop_id sehingga data yang ditampilkan hanya berasal dari bengkel pengguna.

5.6 Transaksi Pembelian Barang

```
22 def create_purchase(user_role, workshop_id, datas):
23     try:
24         # Access Validation ----- Start
25         access = role_validator(user_role)
26
27         if not access:
28             return authorization_error()
29
30         subscription_access = subscription_validator(user_role, workshop_id)
31
32         if not subscription_access:
33             return subscription_required()
34         # Access Validation ----- Finish
35
36         # Check Request Body ----- Start
37         if datas is None:
38             return invalid_params()
39
40         required_data = [
41             "supplier_id",
42             "purchase_date",
43             "purchase_details"
44         ]
45
46         for req in required_data:
47             if req not in datas:
48                 return parameter_error(
49                     f"Missing {req} in request body."
50                 )
51         # Check Request Body ----- Finish
52
53     # Insert Purchase ----- Start
54     purchase = Purchases(
55         supplier_id=supplier_id,
56         workshop_id=workshop_id,
57         purchase_date=purchase_date,
58         total=0,
59         created_at=timestamp,
60         updated_at=timestamp
61     )
62
63     try:
64         db.session.add(purchase)
65         db.session.flush()
66
67     except Exception as e:
68         db.session.rollback()
69         return parameter_error(str(e))
70     # Insert Purchase ----- Finish
71
72     # Insert Purchase Detail ----- Start
73     for item in purchase_details:
74
75         product = Products.query.filter_by(
76             id=item["product_id"],
77             workshop_id=workshop_id,
78             is_deleted=0
79         ).first()
80
81         if not product:
82             db.session.rollback()
83
84             return not_found(
85                 "Product could not be found."
86             )
87
88         quantity = int(item["quantity"])
89         unit_cost = int(item["unit_cost"])
90         subtotal = quantity * unit_cost
91
92         detail = PurchaseDetails(
93             purchase_id=purchase.id,
94             product_id=product.id,
95             quantity=quantity,
96             unit_cost=unit_cost,
97             subtotal=subtotal
98         )
99
100         db.session.add(detail)
101
102         # Update Stock
103         product.stock -= quantity
104
105         # Update harga beli terakhir
106         product.purchase_price = unit_cost
107
108         # Calculate Total
109         total += subtotal
110     # Insert Purchase Detail ----- Finish
111
112     # Update Purchase Total ----- Start
113     purchase.total = total
114     purchase.updated_at = timestamp
115     # Update Purchase Total ----- Finish
116
117     # Commit Transaction ----- Start
118     try:
119         db.session.commit()
120
121     except Exception as e:
122         db.session.rollback()
123         return parameter_error(str(e))
124     # Commit Transaction ----- Finish
125
126     # Return Response -----
127     return success(
128         status_code=201
129     )
130
131 except Exception as e:
132     db.session.rollback()
133     return bad_request(str(e))
134
135 # CREATE PURCHASE ----- End
```

Gambar 5.6 Code Transaksi Pembelian Barang

Code pada Gambar 5.6 menjelaskan proses transaksi pembelian barang dari pemasok. Pemilik Bengkel memilih pemasok dan barang, kemudian memasukkan jumlah serta harga beli. Sistem menghitung subtotal dan total pembelian, menyimpan data transaksi, serta menambahkan stok barang sesuai jumlah barang yang dibeli.

5.7 Transaksi Penjualan

```
94 # CHECKOUT ***** Begin
95 def checkout(datas):
96     try:
97         claims = get_jwt()
98
99         workshop_id = claims["ws_id"]
100         role = str(claims["role"])
101
102         OWNER = "1"
103         CASHIER = "2"
104
105         # Access Validation ----- Start
106         if role not in [OWNER, CASHIER]:
107             return authorization_error()
108         subscription_access = subscription_validator(
109             role,
110             workshop_id
111         )
112
113         if not subscription_access:
114             return subscription_required()
115         # Access Validation ----- Finish
116         if datas is None:
117             return invalid_params()
118
119         customer_id = datas.get("customer_id")
120         vehicle_id = datas.get("vehicle_id")
121         payment = datas.get("payment")
122         product_details = datas.get("product_details", [])
123         service_details = datas.get("service_details", [])
124
125         validation = sale_validator(
126             customer_id,
127             vehicle_id,
128             payment,
129             product_details,
130             service_details,
131             workshop_id
132         )
133
134         if validation:
135             return bad_request(validation)
136
137         # HITUNG TOTAL BARANG ----- Start
138         total = 0
139         for item in product_details:
140             product = Products.query.filter_by(
141                 id=item["product_id"],
142                 workshop_id=workshop_id,
143                 is_delete=0
144             ).first()
145
146             if not product:
147                 return bad_request("Produk tidak ditemukan.")
148
149             qty = int(item["quantity"])
150
151             if qty > product.stock:
152                 return bad_request(
153                     f"Stok {product.product_name} tidak mencukupi."
154                 )
155     except:
```

```

156     total += product.selling_price * qty
157
158     # HITUNG TOTAL JASA ----- Start
159     for item in service_details:
160         service = Services.query.filter_by(
161             id=item["service_id"],
162             workshop_id=workshop_id,
163             is_delete=0
164         ).first()
165
166         if not service:
167             return bad_request("Jasa tidak ditemukan.")
168
169         qty = int(item["quantity"])
170         total += service.service_fee * qty
171
172     payment = int(payment)
173
174     if payment < total:
175         return bad_request(
176             "Nominal pembayaran kurang."
177         )
178     change = payment - total
179
180     now = int(time.time())
181     # HITUNG TOTAL JASA ----- Finsih
182
183     # HEADER TRANSAKSI ----- Start
184     trx = Payments(
185         cashier_id=claims["id"],
186         workshop_id=workshop_id,
187         customer_id=customer_id or None,
188         vehicle_id=vehicle_id or None,
189         payment_date=now,
190         total=total,
191         paid=payment,
192         change=change,
193         created_at=now,
194         updated_at=now
195     )
196
197     db.session.add(trx)
198     # simpan dulu untuk mendapatkan id
199     db.session.flush()
200
201     # Generate nomor invoice
202     trx.invoice = (
203         f"INV-{datetime.now().strftime('%Y%m%d')}-{trx.id:06d}"
204     )
205     # HEADER TRANSAKSI ----- Finish

```

Gambar 5.7 Code Transaksi Penjualan

Code pada Gambar 5.7 menjelaskan proses transaksi penjualan yang dilakukan oleh Kasir. Kasir dapat mencari barang menggunakan nama atau barcode, memilih jasa servis, menentukan pelanggan dan kendaraan, serta memasukkan jumlah pembayaran. Sistem menghitung total transaksi dan uang kembalian, menyimpan transaksi, serta mengurangi stok barang yang terjual.

5.8 Penyetoran Kas

```
18 def create_cash_deposit(user_role, workshop_id, user_id, datas):
19     try:
20         # Access Validation ----- Start
21         access = role_validator(user_role)
22         if not access:
23             return authorization_error()
24
25         subscription_access = subscription_validator(user_role, workshop_id)
26
27         if not subscription_access:
28             return subscription_required()
29         # Access Validation ----- Finish
30
31         # Check Request Body ----- Start
32         if datas is None:
33             return invalid_params()
34
35         required_data = {
36             "total_deposit",
37             "notes"
38         }
39
40         for req in required_data:
41             if req not in datas:
42                 return parameter_error(
43                     f"Missing {req} in request body."
44                 )
45         # Check Request Body ----- Finish
46
47         # Initialize Data Input ----- Start
48         total_deposit = int(datas["total_deposit"])
49         notes = datas["notes"].strip()
50         # Initialize Data Input ----- Finish
51
52         # Check Workshop ----- Start
53         workshop = Workshops.query.filter_by(
54             id=workshop_id,
55             is_delete=0
56         ).first()
57
58         if not workshop:
59             return not_found(
60                 "Workshop could not be found."
61             )
62         # Check Workshop ----- Finish
63
64         # Get Total Sales ----- Start
65         today = datetime.now().date()
66
67         start_timestamp = int(
68             datetime.combine(today, datetime.min.time()).timestamp()
69         )
70
71         end_timestamp = int(
72             datetime.combine(today, datetime.max.time()).timestamp()
73         )
74
75         # Calculate Difference ----- Start
76         difference = remaining - total_deposit
77         # Calculate Difference ----- Finish
78
79         # Insert Data ----- Start
80         data = CashDeposits(
81             workshop_id=workshop_id,
82             user_id=user_id,
83             deposit_date=timestamp,
84             total_sales=total_sales,
85             total_deposit=total_deposit,
86             difference=difference,
87             notes=notes,
88             status=0,
89             created_at=timestamp,
90             updated_at=timestamp
91         )
92
93         try:
94             db.session.add(data)
95             db.session.commit()
96
97         except Exception as e:
98             db.session.rollback()
99             return parameter_error(str(e))
100         # Insert Data ----- Finish
```

Gambar 5.8 Code Penyetoran Kas

Code pada Gambar 5.8 menjelaskan proses penyetoran kas oleh Kasir. Sistem mengambil total penjualan Kasir, kemudian Kasir memasukkan jumlah uang yang disetorkan. Sistem menghitung selisih antara total penjualan dan total setoran serta menyimpan data dengan status menunggu konfirmasi.

5.9 Konfirmasi Setoran Kas

```

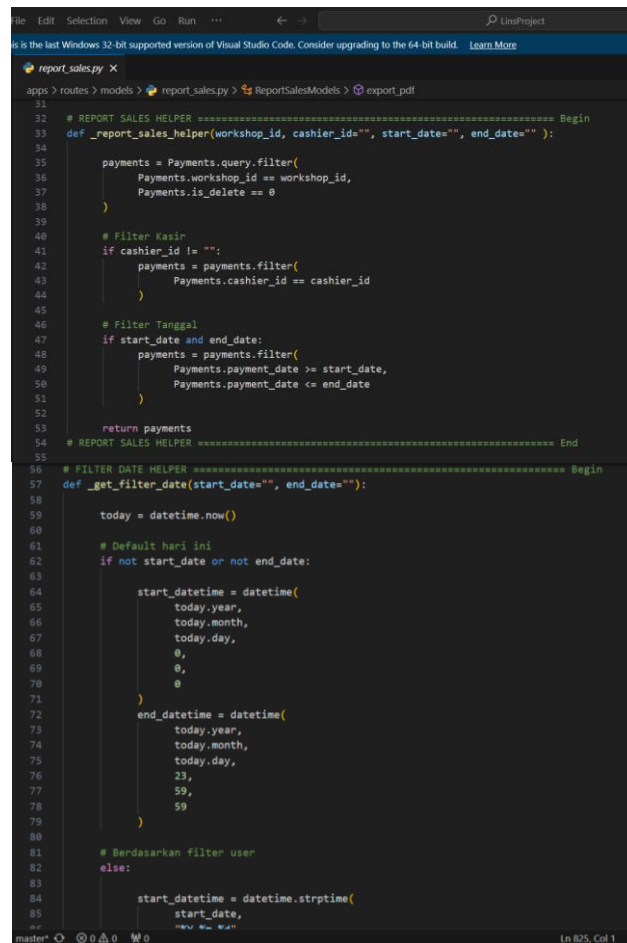
386 # VERIFY CASH DEPOSIT ----- Begin
387 def verify_cash_deposit(user_role, workshop_id, user_id, datas):
388     try:
389         # Access Validation ----- Start
390         access = role_validator(user_role)
391
392         if not access:
393             return authorization_error()
394
395         subscription_access = subscription_validator(user_role, workshop_id)
396
397         if not subscription_access:
398             return subscription_required()
399         # Access Validation ----- Finish
400
401         # Check Request Body ----- Start
402         if datas is None:
403             return invalid_params()
404
405         required_data = [
406             "deposit_id",
407             "status"
408         ]
409
410         for req in required_data:
411             if req not in datas:
412                 return parameter_error(
413                     f"Missing {req} in request body."
414                 )
415         # Check Request Body ----- Finish
416
417         # Initialize Data ----- Start
418         deposit_id = int(datas["deposit_id"])
419         status = int(datas["status"])
420         # Initialize Data ----- Finish
421
422         # Validate Status ----- Start
423         if status not in [1, 2]:
424             return parameter_error(
425                 "Invalid verification status."
426             )
427         # Validate Status ----- Finish
428
429         # Check Workshop ----- Start
430         workshop = Workshops.query.filter_by(
431             id=workshop_id,
432             is_delete=0
433         ).first()
434
435         if not workshop:
436             return not_found(
437                 "Workshop could not be found."
438             )
439         # Check Workshop ----- Finish
440
441         # Check Cash Deposit ----- Start
442         cash_deposit = CashDeposits.query.filter_by(
443             id=deposit_id,
444             workshop_id=workshop_id,
445             is_deleted=0
446         ).first()
447
448         if not cash_deposit:
449             return not_found(
450                 "Cash deposit could not be found."
451             )
452
453         if cash_deposit.status != 0:
454             return defined_error(
455                 "Cash deposit has already been verified.",
456                 "Defined Error",
457                 499
458             )
459         # Check Cash Deposit ----- Finish
460
461         # Update Data ----- Start
462         timestamp = current_timestamp()
463
464         cash_deposit.status = status
465         cash_deposit.verified_by = user_id
466         cash_deposit.verified_at = timestamp
467         cash_deposit.updated_at = timestamp
468
469         try:
470             db.session.commit()
471
472         except Exception as e:
473             db.session.rollback()
474             return parameter_error(str(e))
475         # Update Data ----- Finish
476
477         return success(

```

Gambar 5.9 Code Konfirmasi Setoran Kas

Code pada Gambar 5.9 menjelaskan proses konfirmasi setoran kas oleh Pemilik Bengkel. Pemilik Bengkel dapat melihat data setoran dari Kasir kemudian memilih untuk menerima atau menolak setoran. Sistem selanjutnya memperbarui status setoran berdasarkan tindakan yang dilakukan.

5.10 Pembuatan Laporan



```
31
32 # REPORT SALES HELPER ===== Begin
33 def _report_sales_helper(workshop_id, cashier_id="", start_date="", end_date=""):
34     payments = Payments.query.filter(
35         Payments.workshop_id == workshop_id,
36         Payments.is_delete == 0
37     )
38
39     # Filter Kasir
40     if cashier_id != "":
41         payments = payments.filter(
42             Payments.cashier_id == cashier_id
43         )
44
45     # Filter Tanggal
46     if start_date and end_date:
47         payments = payments.filter(
48             Payments.payment_date >= start_date,
49             Payments.payment_date <= end_date
50         )
51
52     return payments
53 # REPORT SALES HELPER ===== End
54
55 # FILTER DATE HELPER ===== Begin
56 def _get_filter_date(start_date="", end_date=""):
57     today = datetime.now()
58
59     # Default hari ini
60     if not start_date or not end_date:
61         start_datetime = datetime(
62             today.year,
63             today.month,
64             today.day,
65             0,
66             0,
67             0
68         )
69         end_datetime = datetime(
70             today.year,
71             today.month,
72             today.day,
73             23,
74             59,
75             59
76         )
77
78     # Berdasarkan filter user
79     else:
80         start_datetime = datetime.strptime(
81             start_date,
82             "%Y-%m-%d %H:%M:%S"
83         )
84         end_datetime = datetime.strptime(
85             end_date,
86             "%Y-%m-%d %H:%M:%S"
87         )
88
89     return start_datetime, end_datetime
90 # FILTER DATE HELPER ===== End
```

```

698 # Create Workbook ----- Start
699 workbook = Workbook()
700 worksheet = workbook.active
701 worksheet.title = "Report Sales"
702 worksheet.merge_cells("A1:I1")
703
704 worksheet["A1"] = "LAPORAN PENJUALAN"
705
706 worksheet["A1"].font = Font(
707     bold=True,
708     size=16
709 )
710
711 worksheet["A1"].alignment = Alignment(
712     horizontal="center"
713 )
714
715 worksheet.append([])
716
717 worksheet.append([
718     "Nama Bengkel",
719     workshop.workshop_name
720 ])
721
722 worksheet.append([
723     "Periode",
724     f"{format_date(start_date)} s.d. {format_date(end_date)}"
725 ])
726 worksheet.append([])
727
728 # Fill Data ----- Finish
729 worksheet.append([
730     "Jumlah Transaksi",
731     total_transaction
732 ])
733
734 worksheet.append([
735     "Total Penjualan",
736     f"Rp {total_sales:,}".replace(",", ".")
737 ])
738
739 # Response File ----- Start
740 buffer = BytesIO()
741 workbook.save(buffer)
742
743 buffer.seek(0)
744
745 return send_file(
746     buffer,
747     as_attachment=True,
748     download_name="sales_report.xlsx",
749     mimetype="application/vnd.openxmlformats-officedocument.spreadsheetml.sheet"
750 )
751 # Response File ----- Finish
752
753 except Exception as e:
754     return bad_request(str(e))
755
756 # EXPORT EXCEL ----- End
757
758 # Summary ----- Finish
759 # Generate PDF ----- Start
760
761 table_data = [
762     {
763         "No",
764         "Invoice",
765         "Tanggal",
766         "Customer",
767         "Plat Nomor",
768         "Kasir",
769         "Total"
770     }
771 ]
772
773 for index, payments in enumerate(payments, start=1):
774
775     table_data.append([
776         index,
777         payments.invoice,
778         format_date(payments.payment_date),
779         payments.customers.customer_name if payments.customers else "Pelanggan Umum",
780         payments.vehicles.plate_number if payments.vehicles else "-",
781         payments.cashier.username if payments.cashier else "-",
782         f"Rp {payments.total:,}".replace(",", ".")
783     ])
784
785 # Initialize PDF ----- Start
786 buffer = BytesIO()

```

```

862 # Initialize PDF ----- Start
863 buffer = BytesIO()
864
865 document = SimpleDocTemplate(
866     buffer,
867     pagesize=A4,
868     leftMargin=1.5 * cm,
869     rightMargin=1.5 * cm,
870     topMargin=2 * cm,
871     bottomMargin=2 * cm
872 )
873
874 styles = getSampleStyleSheet()
875
876 elements = []
877 # judul
878 elements.append(
879     Paragraph(
880         "<b>LAPORAN PENJUALAN</b>",
881         styles["Title"]
882     )
883 )
884 elements.append(
885     Paragraph(
886         f"<b>Nama Bengkel :</b> {workshop.workshop_name}",
887         styles["Normal"]
888     )
889 )
890
891 elements.append(
892     ("VALIGN", (0, 0), (-1, -1), "MIDDLE"),
893     ("BOTTOMPADDING", (0, 0), (-1, 0), 8),
894     ("TOPPADDING", (0, 1), (-1, -1), 8),
895     ("BOTTOMPADDING", (0, 1), (-1, -1), 8),
896     ])
897
898 elements.append(table)
899
900 elements.append(Spacer(1,0.5*cm))
901
902 elements.append(
903     Paragraph(
904         f"<b>Jumlah Transaksi :</b> {total_transaction}",
905         styles["Normal"]
906     )
907 )
908
909 elements.append(
910     Paragraph(
911         f"<b>Total Penjualan :</b> Rp {total_sales:.} ".replace(".", "."),
912         styles["Normal"]
913     )
914 )
915
916 document.build(elements)
917
918 buffer.seek(0)
919 # Initialize PDF ----- Finish

```

Gambar 5.10 Code Pembuatan Laporan

Code pada Gambar 5.10 menjelaskan proses pembuatan laporan pada Aplikasi POS Bengkel. Sistem mengambil data transaksi berdasarkan workshop_id dan periode yang dipilih, kemudian mengolah data tersebut untuk ditampilkan dalam bentuk laporan. Laporan penjualan dan pembelian juga dapat diekspor ke dalam format Excel dan PDF.

6.1 Penutup

Dokumen teknis Aplikasi POS Bengkel ini memberikan gambaran mengenai profil aplikasi, latar belakang pengembangan, manfaat, spesifikasi teknis, serta beberapa source code utama yang digunakan dalam sistem. Dokumen ini diharapkan dapat memberikan informasi teknis mengenai proses pengembangan dan fungsi utama Aplikasi POS Bengkel Berbasis Web.

TERIMA KASIH

POS BENGKEL • Dokumen Teknis

Lampiran 6 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	: EC002026131427, 3 Agustus 2026
Pencipta	
Nama	: Helina Putri, Dyah Apriliani, S.T., M. Kom. dkk
Alamat	: Jalan Raya Pakulaut Rt 05 / Rw 04, Desa Pakulaut, Margasari, Kab. Tegal, Jawa Tengah, 52463
Kewarganegaraan	: Indonesia
Pemegang Hak Cipta	
Nama	: Unit Penelitian Abdimas Universitas Harkat Negeri
Alamat	: Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana 52142, Margadana, Kota Tegal, Jawa Tengah, 52142
Kewarganegaraan	: Indonesia
Jenis Ciptaan	: Program Komputer
Judul Ciptaan	: Aplikasi Point Of Sale Bengkel Berbasis Website
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	: 3 Agustus 2026, di Kota Tegal
Jangka waktu perlindungan	: Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor Pencatatan	: 001391074
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 PENCIPTA

No	Nama	Alamat
1	Helina Putri	Jalan Raya Pakulaut Rt 05 / Rw 04, Desa Pakulaut Margasari, Kab. Tegal
2	Dyah Apriliani, S.T., M. Kom.	Perumahan Mutiara Indah Blok C4, Jalan Nangka Gang 2, RT 02 Rw 02, Kelurahan Procot Slawi, Kab. Tegal
3	Arif Hidayah, S.Tr.Kom., M.Tr.Kom.	Jalan Bulu Emas No. 8, Desa Lawatan Dukuhturi, Kab. Tegal







Balai Besar
Sertifikasi
Elektronik

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 7 Lembar Bimbingan



DIV TEKNIK INFORMATIKA
UNIVERSITAS HARKAT NEGERI

LEMBAR BIMBINGAN TUGAS AKHIR

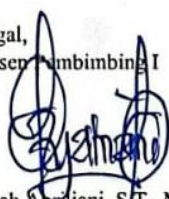
Nama : Helina Putri
NIM : 19090133
No. Ponsel : 081906790616
Judul TA : Aplikasi *Point Of Sale* Bengkel Berbasis
Website Menggunakan Flask
Dosen Pembimbing I : Dyah Apriliani, S. T., M. Kom

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	18/6 2026	Appl	- Jasa → Indc - Tampilan H. - Admin fang - Multi:	
2.	20/6 2026	Appl	- per - cant → H - & modul	
3.	2/7 2026	Appl	- Lanjutan.	

4.	28/2016 7	Apresiasi.	Perini - Pif for name book data.	
5.	30/2016 7	Apresiasi	Perini kan (H/4)	
6.	31/2016 7	Apresiasi - HKI	- OK - OK - Target for	
7.	4/2016 8	Apresiasi - Capas	OK Perini.	
8.	11/2016 8	Laporan	Perini.	

g.	18/2026	lapu	lapu	dy.
6,	21/2026	lapu	lapu	dy.

Tegal, 2026
Dosen Pembimbing I



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



DIV TEKNIK INFORMATIKA
UNIVERSITAS HARKAT NEGERI

LEMBAR BIMBINGAN TUGAS AKHIR

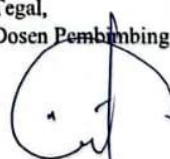
Nama : Helina Putri
NIM : 19090133
No. Ponsel : 081906790616
Judul TA : Aplikasi Point Of Sale Bengkel Berbasis
Website Menggunakan Flask
Dosen Pembimbing II : Arif Hidayah, S.Tr.Kom., M.Tr.Kom

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	9/6/26	APP	- Perbaiki DB - Lanjutkan !	
2.	18/6/26	APP	- Perbaiki Sebagian catatan	
3	23/6/26	APP	- Perbaiki Sebagian catatan	
4.	28/6/26	APP	- Perbaiki alur bisnis - validasi form	

5	30/6/26	APP	SB	
	7/8/26	APP	terima Lupa Password	
	11/8/26	APP Laporan	hosting Perbaiki Setor Cekoran	
	13/8/26	Laporan	- Perbaiki Setor Cakupan - tambah ke Pengujian	
	26/8/26	Laporan	- Lengkapi SB	

--	--	--	--	--

Tegal, 2026
Dosen Pembimbing II



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

NIPY. 10.024.548