

# Laporan Progres Mingguan - **SmartBizAdmin**

---

## Kelompok: 5

- **Muhammad Fikri Haikal Ariadma / 10231063**
- **Irfan Zaki Riyanto / 10231045**
- **Micka Mayulia Utama / 10231053**
- **Ika Agustin Wulandari / 10231041**

 Mitra: Kost Al-Fitri D'Carjoe

 Pekan ke-: 13

 Tanggal: 09/05/2025

---

## Progress Summary

Pada pekan ke-13, tim SmartBizAdmin telah menyelesaikan implementasi fitur inti keempat yang mencakup sistem kasir untuk coffee shop dan pengelolaan informasi kost, termasuk data kamar, harga, dan ketersediaannya. Tim juga berhasil mengembangkan panel super admin dengan lima menu utama, yaitu fitur keuangan untuk memantau arus kas, fitur kost untuk mengelola data penyewaan kamar, fitur coffee shop untuk mengatur menu dan transaksi, fitur hak akses untuk pengelolaan pengguna dan peran, serta fitur inventaris untuk memantau stok barang. Selain itu, dashboard admin khusus juga telah dibuat untuk operasional coffee shop, sehingga pengelola dapat fokus pada pemesanan, menu, dan transaksi harian. Visualisasi data sederhana ditambahkan dalam bentuk grafik batang yang menampilkan statistik penjualan menu pada coffee shop serta tingkat hunian pada kost.

## Accomplished Tasks

-  Implementasi fitur kost dan fitur coffee shop
  -  Pembuatan dashboard super admin dan admin
  -  Visualisasi grafik data keuangan
  -  Demo progress ke mitra
- 

## Challenges & Solutions

-  **Challenge 1:** Grafik lingkaran yang menampilkan pemasukan dan pengeluaran bulanan tidak mencerminkan data yang ada di tabel keuangan. Ketidaksesuaian ini menyebabkan pengguna mengalami kebingungan dalam membaca data keuangan dari visualisasi yang seharusnya

membantu.

**Solution:** Menambahkan pengecekan format dan urutan bulan (labels: months) agar sesuai dengan data dari backend.

---

## Next Week Plan

-  Penyempurnaan seluruh fitur
  -  Usability testing
  -  Bugfixing
  -  Persiapan deployment (jika diperlukan)
  -  Demo progress ke mitra
- 

## Contributions

-  **Muhammad Fikri Haikal Ariadma / 10231063**  
→ Memperbaiki UI Web, Memperbaiki Grafik Keuangan, dan Membuat halaman dashboard admin dan superadmin
  -  **Irfan Zaki Riyanto / 10230145**  
→ Aktivasi sistem manajemen kos dan manajemen coffeeshop
  -  **Micka Mayulia Utama / 10231053**  
→ Membuat laporan
  -  **Ika Agustin Wulandari / 10231041**  
→ Membuat laporan
- 

## Screenshots / Demo

### Kode Backend:

- Kode backend kosController.js:

```

const pool = require('../config/db');

// @desc Get all kos rooms
// @route GET /api/kos
// @access Private (Superadmin)
const getAllKosRooms = async (req, res) => {
    try {
        const result = await pool.query(
'SELECT * FROM rooms ORDER BY id ASC');
        res.status(200).json(result.rows);
    } catch (error) {
        console.error('Error in getAllKosRooms: '
, error);
        res.status(500).json({ message: 'Server error'
, error: error.message });
    }
};

// @desc Get a single kos room by ID
// @route GET /api/kos/:id
// @access Private (Superadmin)
const getKosRoomById = async (req, res) => {
    try {
        const roomId = parseInt(req.params.id, 10);
        if (isNaN(roomId)) {
            return res.status(400).json({ message:
'Invalid room ID' });
        }
        const result = await pool.query(
'SELECT * FROM rooms WHERE id = $1', [roomId]);
        if (result.rows.length === 0) {
            return res.status(404).json({ message:
'Kos room not found' });
        }
        res.status(200).json(result.rows[0]);
    } catch (error) {

```

Potongan kode di atas adalah controller Node.js yang menggunakan Express dan PostgreSQL untuk melakukan operasi CRUD pada data kamar kos di tabel rooms. Terdiri dari lima fungsi: mengambil semua kamar, mengambil kamar berdasarkan ID, menambah, memperbarui, dan menghapus kamar. Setiap fungsi menggunakan async/await, validasi input, dan penanganan error dengan try-catch. Semua response diberikan dalam format JSON dan hanya dapat diakses oleh Superadmin.

- **Kode kos.js:**

```
const kosController = require(
  '../controllers/kosController');
const express = require('express');
const router = express.Router();
// const { protect } = require('../middleware/authMiddleware'); // Temporarily commented out

// @route   GET /api/kos
// @desc    Get all kos rooms
// @access  Private (Superadmin)
router.get('/', kosController.getAllKosRooms);
// Removed protect

// @route   GET /api/kos/:id
// @desc    Get a single kos room by ID
// @access  Private (Superadmin)
router.get('/:id', kosController.getKosRoomById);
// Removed protect

// @route   POST /api/kos
// @desc    Create a new kos room
// @access  Private (Superadmin)
router.post('/', kosController.createKosRoom);
// Removed protect

// @route   PUT /api/kos/:id
// @desc    Update a kos room by ID
// @access  Private (Superadmin)
router.put('/:id', kosController.updateKosRoom);
// Removed protect

// @route   DELETE /api/kos/:id
// @desc    Delete a kos room by ID
// @access  Private (Superadmin)
router.delete('/:id', kosController.deleteKosRoom);
// Removed protect
```

Kode di atas merupakan konfigurasi routing menggunakan Express.js untuk mengatur endpoint API /api/kos yang terhubung ke fungsi-fungsi pada kosController. Setiap route menangani operasi CRUD: GET untuk mengambil semua atau satu kamar kos, POST untuk menambah kamar, PUT untuk memperbarui, dan DELETE untuk menghapus kamar berdasarkan ID. Middleware protect yang biasanya digunakan untuk autentikasi Superadmin sementara dinonaktifkan (dikomentari). Akhirnya, router diekspor agar dapat digunakan di file utama aplikasi.

- **Kode coffeeShopController.js:**

```
// backend/controllers/coffeeShopController.js
const pool = require('../config/db');
// Assuming your DB config is here

// Placeholder functions - we will implement these later

// Menu Management
exports.getAllMenus = async (req, res) => {
    try {
        const result = await pool.query(
            'SELECT id, name, price, category, description, availability FROM public.menu_items ORDER BY name ASC'
        );
        res.status(200).json(result.rows);
    } catch (err) {
        console.error('Error fetching menus:', err);
        res.status(500).json({ message: 'Internal Server Error', error: err.message });
    }
};

exports.getMenuById = async (req, res) => {
    const { id } = req.params;
    if (isNaN(parseInt(id))) {
        return res.status(400).json({ message: 'Invalid menu ID provided.' });
    }
    try {
        const result = await pool.query(
            'SELECT id, name, price, category, description, availability FROM public.menu_items WHERE id = $1'
        );
        res.status(200).json(result.rows[0]);
    } catch (err) {
        console.error('Error fetching menu by ID:', err);
        res.status(500).json({ message: 'Internal Server Error', error: err.message });
    }
};
```

```

        , [id]);
        if (result.rows.length === 0) {
            return res.status(404).json({ message:
            'Menu item not found' });
        }
        res.status(200).json(result.rows[0]);
    } catch (err) {
        console.error(`Error fetching menu with ID
${id}:`, err);
        res.status(500).json({ message:
        'Internal Server Error', error: err.message });
    }
}

```

File coffeeShopController.js berisi controller untuk manajemen menu dan pemesanan pada aplikasi coffee shop menggunakan Node.js dan PostgreSQL. Fungsi-fungsinya mencakup: mengambil semua menu (getAllMenus), mengambil menu berdasarkan ID (getMenuById), menambahkan menu baru dengan validasi input (createMenu), memperbarui menu secara dinamis berdasarkan input yang dikirim (updateMenu), serta menghapus menu dengan penanganan error seperti foreign key. Selain itu, terdapat fungsi untuk membuat pesanan (createOrder) yang memvalidasi input pengguna, mengecek ketersediaan menu, menghitung total harga, dan menyimpan data pesanan ke database dengan transaksi untuk menjaga konsistensi data.

- **Kode coffeeShop.js:**



```

// backend/routes/coffeeShop.js
const express = require('express');
const router = express.Router();
const coffeeShopController = require(
    '../controllers/coffeeShopController');
const authenticateToken = require(
    '../middleware/authMiddleware');
// const checkRole = require('../middleware/roleMidd
leware'); // Placeholder for role checking middlewar
e

// Middleware to apply authentication to all coffee
shop routes
router.use(authenticateToken);

```

```

// Middleware for role checking (Admin and Superadmin) - Placeholder
const checkAdminOrSuperAdmin = (req, res, next) => {

  // In a real implementation, you'd check req.user.role

  // For now, we'll just proceed. Replace with actual
  // role check later.

  // if (req.user.role !== 'admin' && req.user.role !== 'superadmin') {
    //   return res.status(403).json({ message: 'Forbidden: Insufficient privileges' });
  // }
  console.log('Role check bypassed for now. User:', req.user); // Log user info if available
  next();
};

// Apply role check middleware to all subsequent routes in this file
router.use(checkAdminOrSuperAdmin);

// --- Menu Routes ---
// GET all menus
router.get('/menus'
, coffeeShopController.getAllMenus);

// GET a single menu by ID
router.get('/menus/:id'
, coffeeShopController.getMenuById);

```

Kode ini mengatur routing fitur coffee shop dengan Express.js, mencakup autentikasi token dan (placeholder) pengecekan peran admin/superadmin. Disediakan endpoint untuk CRUD menu, pemesanan, dan pencetakan struk, dengan logika ditangani oleh coffeeShopController.

## Kode Frontend:

- **Front End ManageCoffeeShopMenu.jsx:**

```
import React, { useState, useEffect, useRef } from
'react';
import axios from 'axios';
import
{ Modal, Button, Form, Table, Alert, Spinner } from
'react-bootstrap';
import { FontAwesomeIcon } from
'@fortawesome/react-fontawesome';
import
{ faShoppingCart, faPlus, faArrowLeft, faInfoCircle,
faEdit }
from '@fortawesome/free-solid-svg-icons';
// Added faEdit and faInfoCircle
import './ManageCoffeeShopMenu.css';
import { useNavigate } from 'react-router-dom';
import { toast, ToastContainer } from
'react-toastify';
import 'react-toastify/dist/ReactToastify.css';

const API_URL = 'http://localhost:3001/coffee-shop'
; // Backend URL

function ManageCoffeeShopMenu() {
    const navigate = useNavigate();
    const [menus, setMenus] = useState([]);
    const [cart, setCart] = useState([]);
    const
[showCartModal, setShowCartModal] = useState(false)
;
    const [showModal, setShowModal] = useState(false)
;
    const [currentItem, setCurrentItem] = useState(
null);
    const
[showDescriptionModal, setShowDescriptionModal] = u
seState(
false); // State for description modal
    const
[currentDescriptionItem, setCurrentDescriptionItem]
= useState(
```

```

    null); // State for item in description modal
    const cartIconRef = useRef(null);
    // Ref for the cart icon

    const increaseQuantity = (item) => {
        const updatedCart = cart.map((cartItem) =>
            cartItem.id === item.id ? { ...cartItem,
        quantity: cartItem.quantity +
        1 } : cartItem
        );
        setCart(updatedCart);
        localStorage.setItem('cart'
        , JSON.stringify(updatedCart));
        // Update local storage
    };

```

Kode ini memungkinkan pengguna untuk mengelola menu di coffee shop dengan fitur tambah, edit, dan hapus menu menggunakan modal. Pengguna dapat menambah item ke keranjang belanja, mengubah jumlah, dan melihat deskripsi menu. Terdapat animasi ketika item ditambahkan ke keranjang, dan modal keranjang menampilkan daftar item yang dipilih beserta total harga. Setelah itu, pengguna dapat melanjutkan ke halaman pembayaran. Kode ini juga menggunakan notifikasi toast untuk feedback dan memanfaatkan Bootstrap untuk tampilan modal serta elemen UI lainnya.

- **Front end ManageKos.jsx:**



```

import React, { useState, useEffect, useCallback } from "react";
import axios from "axios";
import { Button, Modal, Form, Table, Dropdown, DropdownButton } from "react-bootstrap";
import { FaTrashAlt, FaEdit, FaPlus, FaArrowLeft, FaPrint } from "react-icons/fa";
import { useNavigate } from "react-router-dom";
import Footer from './Footer';
import './ManageKos.css';

```

```
// Ensure this CSS file is adapted or created

const ManageKos = () => {
  const navigate = useNavigate();
  const [kosRooms, setKosRooms] = useState([]);
  const [error, setError] = useState(null);
  const [showModal, setShowModal] = useState(false);
  const [isEditing, setIsEditing] = useState(false);
  const [currentRoom, setCurrentRoom] = useState({
    id: null,
    room_name: '',
    price: '',
    facilities: '',
    availability: true,
  });

  const API_URL = 'http://localhost:3001/api/kos';

  const fetchKosRooms = useCallback(async () => {
    try {
      const token = localStorage.getItem('token');
      if (!token) {
        throw new Error(
          "Token tidak ditemukan. Harap login terlebih dahulu."
        );
      }
    
```

Kode ini mengelola fitur untuk menambah, mengedit, dan menghapus kamar kos. Pengguna dapat melihat daftar kamar yang ada, lengkap dengan nama, harga, fasilitas, dan ketersediaan. Ada tombol untuk menambah kamar baru, serta modal untuk mengisi detail kamar yang baru ditambahkan atau diedit. Fungsi edit mengisi form dengan data kamar yang dipilih, sementara fungsi hapus meminta konfirmasi sebelum menghapus data kamar. Data kamar yang ditampilkan diambil dari API dan disimpan dengan token autentikasi untuk keamanan. Desain menggunakan Bootstrap untuk tampilan tabel dan modal.

- **Front end PaymentForm.jsx:**



```
import React, { useState } from 'react';
```

```

import
  { Container, Row, Col, Card, Form, Button, Table, Alert }
from 'react-bootstrap'; // Added Alert
import { useNavigate } from 'react-router-dom';
import { FontAwesomeIcon } from
'@fortawesome/react-fontawesome';
import
  { faUserCircle, faCreditCard, faShoppingCart, faArrowLeft, faCheckCircle, faSpinner }
from '@fortawesome/free-solid-svg-icons';
// Added faSpinner
import axios from 'axios'; // Import axios
import './PaymentForm.css';

function PaymentForm() {
  const navigate = useNavigate();
  const
  [isProcessing, setIsProcessing] = useState(false);
// State for loading indicator
  const
  [paymentError, setPaymentError] = useState('');
// State for payment error messages
  const
  [customerName, setCustomerName] = useState('');
  const [paymentType, setPaymentType] = useState(
'Tunai'); // Default to Tunai
  const cart = JSON.parse(localStorage.getItem(
'cart') || '[]');

  const handleBackToMenu = () => {
    navigate('/manage-coffee-shop-menu');
};

  const calculateTotal = () => {
    return cart.reduce((total, item) =>
total + (item.price * item.quantity), 0);
};

  const handlePayNow = async () => {
    setPaymentError('');
// Clear previous errors
    if (!customerName.trim()) {
      setPaymentError(
'Nama pelanggan harus diisi ');

```

```
        Nama pelanggan harus diisi. ",
```

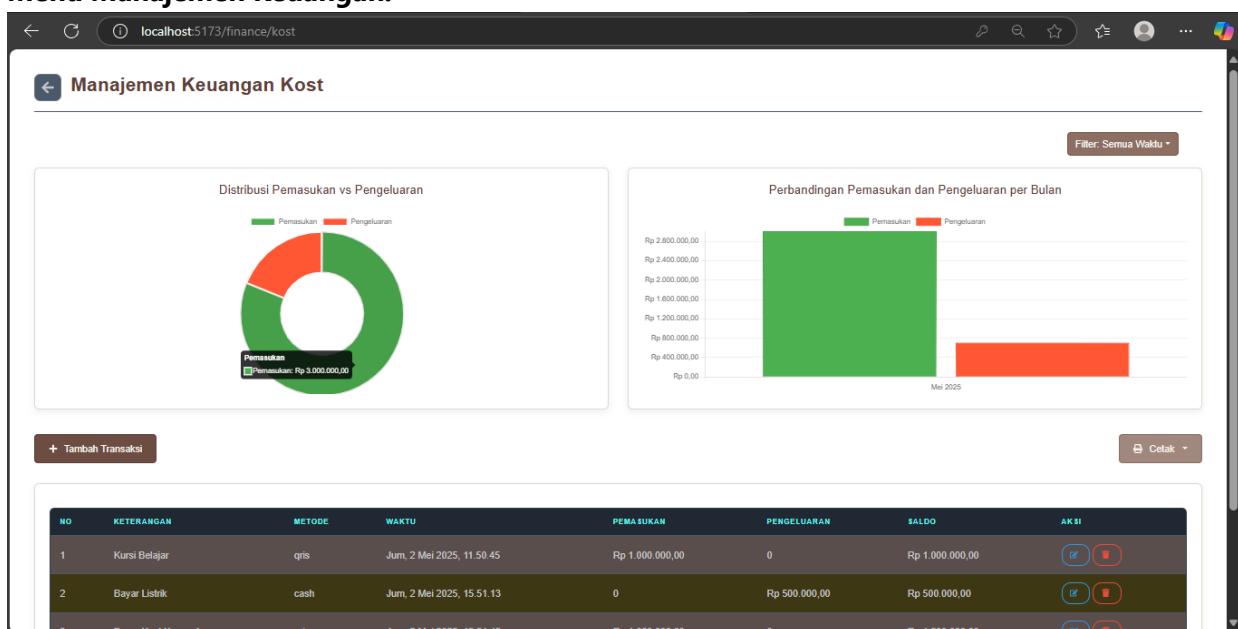
```
    return;
```

```
}
```

Kode ini berfungsi untuk menampilkan form pembayaran, mengelola transaksi, dan memproses pembayaran di aplikasi. Pengguna dapat memilih nama pelanggan dan metode pembayaran (Tunai atau Qris), serta melihat ringkasan pesanan dalam bentuk tabel yang menunjukkan daftar menu, jumlah, dan harga. Ketika pelanggan mengonfirmasi pembayaran, data transaksi dikirim ke backend menggunakan axios. Sistem memeriksa apakah keranjang belanja kosong dan jika nama pelanggan sudah diisi. Jika terjadi kesalahan, pesan kesalahan ditampilkan. Proses pembayaran dikendalikan dengan indikator loading (isProcessing) dan tombol dinonaktifkan hingga kondisi valid dipenuhi. Setelah pembayaran berhasil, pengguna diarahkan ke halaman struk dengan detail transaksi.

## ⌚ Tampilan Sistem Smartbiz Admin:

- **Menu Manajemen Keuangan:**



Gambar diatas adalah tampilan menu manajemen keuangan pada user super admin yang difungsikan untuk melihat data keuangan coffee shop dan kost.

- **Tampilan Dashboard Super Admin:**

The screenshot shows the Smartbiz dashboard for a super admin. The left sidebar lists five menu items: Dashboard, Manajemen Keuangan, Kamar Kos, Coffee Shop Menu, Manajemen User, and Manajemen Inventaris. The main content area displays a welcome message "Selamat Datang, SAD!" and "Role: superadmin". Below this is a section titled "Ringkasan Statistik" with four cards: "TOTAL PENGGUNA" (5), "PENDAPATAN BULAN INI" (Rp 3.112.334), "PENGELUARAN BULAN INI" (Rp 701.000), "KAMAR TERSEDIA" (0), and "ITEM MENU KOPI" (2). A footer bar at the bottom contains the text "© 2025 SmartBiz Admin. All rights reserved."

Tampilan dashboard yang bisa di akses oleh super admin ada 5 fitur. Yang pertama ada menu keuangan, menu kost, menu coffee shop, menu hak akses, menu inventaris

- **Tampilan Dashboard Admin:**

The screenshot shows the Smartbiz dashboard for an admin. The left sidebar lists three menu items: Dashboard, Coffee Shop Menu. The main content area displays a welcome message "Selamat Datang, Alip!" and "Role: admin". Below this is a section titled "Ringkasan Statistik" with three cards: "TOTAL PENGGUNA" (5), "PENDAPATAN BULAN INI" (Rp 3.112.334), and "PENGELUARAN BULAN INI" (Rp 701.000). A footer bar at the bottom contains the text "© 2025 SmartBiz Admin. All rights reserved."

Tampilan dashboard yang bisa di akses oleh admin ada 1 fitur, yaitu menu coffee shop

- **Menu Manajemen Kost:**

#	NAMA KAMAR	HARGA	FA SILITAS	KETERSEDIAAN	AKSI
1	Asik Room	Rp 500.000	ADA	Tidak Tersedia	<button>Edit</button> <button>Hapus</button>

© 2025 SmartBiz Admin. All rights reserved.

Tampilan ini berisikan pengelolaan crud kost, dari pengisian kamar kost fasilitas yang ada.

- **Menu Manajemen Coffee Shop:**

Kopi Aren Rp 20.000	Pop Mie Rp 10.000
------------------------	----------------------

Tambah Hapus

Tampilan ini berisikan crud untuk menu pada coffee shop

- **Keranjang Coffee Shop:**

The screenshot shows a web-based application for managing a coffee shop menu. At the top, there's a header "Manage Coffee Shop Menu". Below it, there are two cards for menu items: "Kopi Aren" (Rp 20.000) and "Pop Mie" (Rp 10.000). Each card has "Tambah" and "Hapus" buttons. To the right, a "Keranjang" (Cart) section displays a table with one item: "Pop Mie" (1 unit, Rp 10.000). The cart table has columns for #, Nama Menu, Jumlah, and Total. Buttons "Kosongkan Keranjang" and "Lanjut" are at the bottom.

Tampilan ini merupakan menu keranjang pesanan sebelum diproses menjadi transaksi, sehingga bisa dibatalkan jika pemesan ingin mengubah menu pesanan.

- **Transaksi Coffee Shop:**

The screenshot shows a "Checkout" page from a browser. The URL is "localhost:5173/payment-form". It has sections for "Customer & Payment Information" (Customer Name: "Enter customer name", Payment Type: "Tunai (Cash)"), and "Order Summary" (a table showing "Pop Mie" (1 unit, Rp 10.000)). A green text "Total: Rp 10.000" is displayed below the summary. At the bottom are "Cancel" and "Pay Now" buttons.

Tampilan ini merupakan fiksasi pesanan customer dari pesanan yang dipilih di menu keranjang sebelumnya. Isi dari ini ada nama customer, total tagihan, metode pembayaran.

- **Struk Transaksi Coffee Shop:**

The screenshot displays a digital receipt from "AREA 9 COFFEE". At the top, it shows the shop's logo and address: "Jl. Sei Walin KM 15 Balikpapan Utara, D'Carjoe Cluster". Below this, transaction details are listed: ID Transaksi: TXN225519XX5, Nama Pelanggan: irfan, Tanggal: 5/9/2025, 5:00:25 PM, Metode Pembayaran: Qris. The receipt is titled "Detail Pesanan:" and contains a table of the order items:

#	Nama Menu	Jumlah	Total Harga
1	Pop Mie	1	Rp 10.000

Below the table, a message reads: "Terima kasih telah berbelanja di AREA 9 COFFEE! ~ Have a nice day ~". At the bottom, there are two buttons: "Cetak Resi" (Print Receipt) and "Kembali ke Dashboard" (Return to Dashboard).

Tampilan ini struk dari pemesanan yang telah di order dan juga telah dibayarkan. Berisikan timestamp transaksi dibuat, total tagihan, pembayaran dan juga kop dari nama coffee shop.