**Langkah – Langkah Perbaikan**

**‘Notifikasi dan Konfirmasi Transaksi’**

“Kekurangan keempat adalah **tidak ada notifikasi atau konfirmasi ketika transaksi berhasil diselesaikan**, yang membuat pengguna tidak yakin apakah transaksi mereka telah diproses dengan benar.

Untuk memperbaiki masalah ini, kita dapat menambahkan fitur notifikasi atau tampilan pesan konfirmasi setiap kali transaksi berhasil. Ini bisa berupa popup pesan, alert di halaman, atau bahkan email atau pesan SMS untuk konfirmasi transaksi. “

1. **Tambahkan Pesan Notifikasi di Frontend**

Ketika transaksi selesai, tampilkan notifikasi yang jelas kepada pengguna. Ini bisa berupa modal, popup, atau alert yang langsung muncul di layar.

Contoh menggunakan JavaScript dengan alert() untuk notifikasi sederhana:

*function selesaikanTransaksi(barangTerjual) {*

*var xhr = new XMLHttpRequest();*

*xhr.open('POST', '/selesaikan-transaksi', true);*

*xhr.setRequestHeader('Content-Type', 'application/json');*

*xhr.onreadystatechange = function() {*

*if (xhr.readyState === 4 && xhr.status === 200) {*

*var response = JSON.parse(xhr.responseText);*

*alert('Transaksi berhasil!'); // Tampilkan notifikasi*

*} else if (xhr.readyState === 4) {*

*alert('Transaksi gagal, coba lagi.'); // Tampilkan notifikasi error*

*}*

*};*

*xhr.send(JSON.stringify({*

*barang: barangTerjual*

*}));*

*}*

Alternatifnya, menggunakan modal popup untuk memberikan pengalaman lebih interaktif:

*<!-- Modal -->*

*<div id="transaksiModal" class="modal">*

*<div class="modal-content">*

*<span class="close">&times;</span>*

*<p id="transaksiMessage">Transaksi berhasil!</p>*

*</div>*

*</div>*

*<script>*

*function showModal(message) {*

*var modal = document.getElementById("transaksiModal");*

*var messageElement = document.getElementById("transaksiMessage");*

*messageElement.innerText = message;*

*modal.style.display = "block";*

*}*

*function closeModal() {*

*var modal = document.getElementById("transaksiModal");*

*modal.style.display = "none";*

*}*

*document.querySelector(".close").onclick = closeModal;*

*</script>*

Setelah transaksi berhasil:

*function selesaikanTransaksi(barangTerjual) {*

*var xhr = new XMLHttpRequest();*

*xhr.open('POST', '/selesaikan-transaksi', true);*

*xhr.setRequestHeader('Content-Type', 'application/json');*

*xhr.onreadystatechange = function() {*

*if (xhr.readyState === 4 && xhr.status === 200) {*

*showModal('Transaksi berhasil!'); // Tampilkan modal dengan pesan sukses*

*} else if (xhr.readyState === 4) {*

*showModal('Transaksi gagal, silakan coba lagi.'); // Tampilkan pesan error*

*}*

*};*

*xhr.send(JSON.stringify({*

*barang: barangTerjual*

*}));*

*}*

1. **Tambahkan Notifikasi melalui Email (Opsional)**

Untuk memberikan pengalaman yang lebih profesional, Anda bisa mengirimkan email konfirmasi setiap kali transaksi selesai. Ini akan berguna khususnya pada sistem e-commerce atau aplikasi penjualan skala besar.

Misalnya, menggunakan Laravel untuk mengirimkan email notifikasi:

*use Illuminate\Support\Facades\Mail;*

*public function selesaikanTransaksi(Request $request) {*

*// Logika pengurangan stok dan penyelesaian transaksi*

*// Kirim email notifikasi*

*Mail::to($request->user()->email)->send(new TransaksiBerhasilMail($transaksi));*

*return response()->json(['message' => 'Transaksi berhasil, email konfirmasi dikirim.']);*

*}*

Contoh class TransaksiBerhasilMail di Laravel:

*use Illuminate\Bus\Queueable;*

*use Illuminate\Mail\Mailable;*

*use Illuminate\Queue\SerializesModels;*

*class TransaksiBerhasilMail extends Mailable*

*{*

*use Queueable, SerializesModels;*

*public $transaksi;*

*public function \_\_construct($transaksi)*

*{*

*$this->transaksi = $transaksi;*

*}*

*public function build()*

*{*

*return $this->view('emails.transaksi-berhasil')*

*->with(['transaksi' => $this->transaksi]);*

*}*

*}*

1. **Notifikasi Melalui SMS (Opsional)**

Selain email, menambahkan notifikasi SMS juga bisa menjadi nilai tambah, terutama untuk pelanggan yang lebih suka mendapatkan informasi lewat pesan singkat.

Contoh penggunaan API SMS seperti Twilio di Laravel:

*use Twilio\Rest\Client;*

*public function selesaikanTransaksi(Request $request) {*

*// Logika pengurangan stok dan penyelesaian transaksi*

*$sid = 'TWILIO\_ACCOUNT\_SID';*

*$token = 'TWILIO\_AUTH\_TOKEN';*

*$client = new Client($sid, $token);*

*$client->messages->create(*

*$request->user()->phone\_number, // Nomor telepon penerima*

*[*

*'from' => 'TWILIO\_PHONE\_NUMBER',*

*'body' => 'Transaksi Anda berhasil! Terima kasih sudah berbelanja.'*

*]*

*);*

*return response()->json(['message' => 'Transaksi berhasil, notifikasi SMS dikirim.']);*

*}*

1. **Pastikan Pesan Error Jelas**

Selain notifikasi sukses, penting juga untuk memberikan pesan error yang jelas jika transaksi gagal. Misalnya, jika ada masalah dengan stok barang atau kegagalan pembayaran, tampilkan pesan kesalahan yang sesuai.

Contoh di backend:

*public function selesaikanTransaksi(Request $request) {*

*try {*

*// Proses transaksi*

*} catch (\Exception $e) {*

*return response()->json(['message' => 'Terjadi kesalahan, silakan coba lagi.'], 500);*

*}*

*}*

Di Frontend:

*function selesaikanTransaksi(barangTerjual) {*

*var xhr = new XMLHttpRequest();*

*xhr.open('POST', '/selesaikan-transaksi', true);*

*xhr.setRequestHeader('Content-Type', 'application/json');*

*xhr.onreadystatechange = function() {*

*if (xhr.readyState === 4 && xhr.status === 200) {*

*showModal('Transaksi berhasil!'); // Pesan sukses*

*} else if (xhr.readyState === 4) {*

*showModal('Transaksi gagal, coba lagi.'); // Pesan error*

*}*

*};*

*xhr.send(JSON.stringify({*

*barang: barangTerjual*

*}));*

*}*

* **Testing**
* Uji Transaksi Berhasil: Lakukan transaksi di frontend dan pastikan notifikasi muncul sesuai harapan (baik alert, modal, email, atau SMS).
* Uji Transaksi Gagal: Buat simulasi transaksi gagal (misalnya dengan stok barang yang kurang atau koneksi internet yang putus) dan cek apakah pesan error yang jelas muncul untuk pengguna.
* Uji Pengiriman Email atau SMS: Jika menggunakan email atau SMS, pastikan notifikasi dikirim dengan benar setiap kali transaksi selesai.
* **Hasil Perbaikan**
* Pengguna akan menerima notifikasi langsung ketika transaksi berhasil, baik melalui tampilan di halaman, email, atau SMS.
* Pesan error yang jelas akan muncul ketika transaksi gagal, membantu pengguna memahami apa yang salah.
* Menambah rasa percaya pengguna bahwa transaksi mereka diproses dengan benar dan mengurangi potensi kebingungan.