

## BAB 2

### LANDASAN TEORI

#### 2.1 Teori Umum Mengenai Sistem dan Digitalisasi

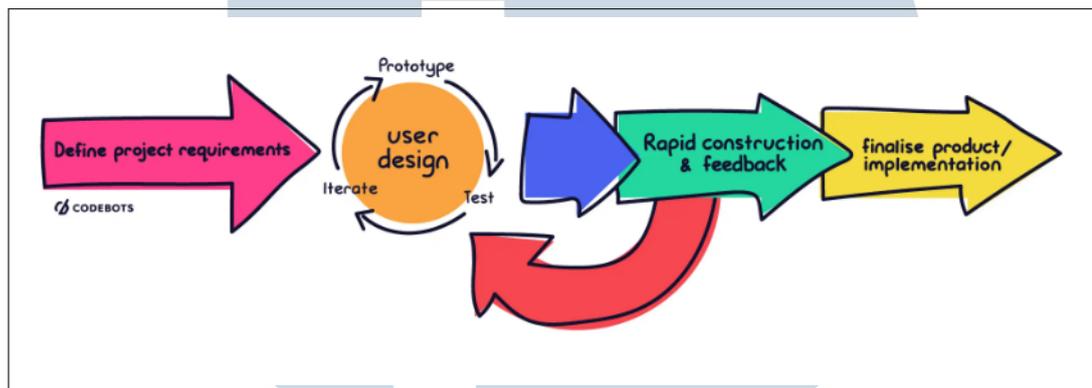
Sistem informasi merupakan kombinasi dari teknologi informasi, proses, dan sumber daya manusia yang digunakan untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi guna mendukung proses pengambilan keputusan dan pengendalian dalam suatu organisasi. Dalam era digital, transformasi sistem manual menjadi sistem informasi digital telah menjadi strategi utama perusahaan dalam meningkatkan efisiensi operasional [7]. Digitalisasi memungkinkan akses data secara *real-time*, otomatisasi proses kerja, serta integrasi antar departemen dalam suatu sistem yang terstruktur. Dalam dunia bisnis, penerapan sistem informasi digital juga terbukti dapat mengurangi biaya operasional, meningkatkan akurasi data, dan mempercepat proses transaksi. Dalam konteks distribusi dan penjualan, digitalisasi memainkan peran penting dalam mendukung kelancaran alur kerja *sales*, seperti pemrosesan pesanan, pengecekan stok barang, dan pelaporan performa penjualan [8]. Oleh karena itu, penerapan sistem informasi digital bukan hanya sebuah kebutuhan teknis, tetapi menjadi bagian dari strategi bisnis *modern* yang menjamin daya saing perusahaan di tengah perkembangan teknologi.

#### 2.2 Framework dan Teknologi yang Digunakan

Dalam penelitian ini, framework yang digunakan adalah Laravel 12 sebagai *backend* dan *React Native* untuk *frontend*. *Laravel* merupakan salah satu *framework* PHP *modern* yang menggunakan arsitektur *Model View Controller* (MVC), menawarkan berbagai fitur seperti routing, ORM (*Eloquent*), *middleware*, serta sistem autentikasi yang terintegrasi, menjadikannya sangat cocok untuk pengembangan aplikasi skala menengah hingga besar [9]. Sementara itu, *React Native* dipilih untuk pengembangan aplikasi mobile karena kemampuannya dalam membuat aplikasi *Android* dan *IOS* secara lintas *platform* menggunakan satu basis kode *JavaScript*.

## 2.3 Metode *Rapid Application Development* (RAD)

Penelitian ini menggunakan metode *Rapid Application Development* (RAD) sebagai kerangka utama dalam proses pengembangan sistem. RAD merupakan pendekatan pengembangan perangkat lunak yang menekankan pada kecepatan pengembangan melalui proses iteratif yang melibatkan pengguna secara aktif [10].



Gambar 2.1. Tahapan *Rapid Application Development* (RAD)

### 2.3.1 Define Project Requirements (Pendefinisian Kebutuhan)

Tahapan awal ini bertujuan untuk mengidentifikasi kebutuhan sistem melalui diskusi dan wawancara langsung dengan *stakeholder*, khususnya pemilik bisnis. Informasi yang diperoleh menjadi fondasi awal dalam proses perancangan sistem dan akan digunakan untuk memastikan bahwa sistem yang dikembangkan sesuai dengan tujuan utama organisasi [11].

### 2.3.2 User Design (Desain Iteratif oleh Pengguna)

Pada tahap ini, dilakukan proses *prototyping* secara iteratif, di mana perancang dan pengguna bekerja sama dalam merancang antarmuka sistem, menguji rancangan, serta memberikan umpan balik. Iterasi ini memungkinkan adanya penyempurnaan desain sebelum masuk ke tahap pembangunan sistem utama. Tahap ini mencakup: pembuatan *prototype* awal, Uji coba antarmuka dan fungsionalitas, Revisi desain berdasarkan *feedback* [12].

### **2.3.3 Rapid Construction and Feedback (Pengembangan Cepat dan Umpan Balik)**

Setelah desain dikonfirmasi, pengembangan sistem dilakukan secara modular dan bertahap. Proses ini bersifat cepat namun tetap terbuka terhadap *feedback* dari pengguna. Setiap modul yang selesai diuji dan disempurnakan berdasarkan masukan dari pengguna sebelum diintegrasikan secara penuh ke dalam sistem utama [13].

### **2.3.4 Final Implementation (Implementasi Akhir)**

Tahap akhir adalah implementasi sistem ke lingkungan operasional. Setelah sistem selesai dikembangkan dan diuji secara menyeluruh, sistem akan diimplementasikan untuk digunakan secara penuh oleh pengguna akhir. *Final testing* dilakukan untuk memastikan sistem telah memenuhi seluruh kebutuhan bisnis yang telah didefinisikan pada tahap awal.

## **2.4 Sistem Informasi Penjualan dan Pencatatan Stok**

Sistem informasi penjualan merupakan perangkat lunak yang dirancang untuk membantu proses penjualan, mulai dari pencatatan transaksi hingga pengecekan stok barang dan laporan penjualan. Sistem ini bertujuan untuk menggantikan proses manual yang memerlukan banyak waktu dan rawan kesalahan, seperti pencatatan melalui buku atau *file spreadsheet* [14]. Salah satu fungsi penting dari sistem ini adalah integrasi antara proses penjualan dan data inventaris, sehingga informasi ketersediaan stok dapat diperoleh secara *real-time* oleh tim *sales* maupun manajemen [15]. Dalam sistem digital, proses pencarian produk dapat diotomatisasi berdasarkan nama item atau kategori produk, yang membantu meminimalisir waktu yang dibutuhkan saat transaksi. Sistem juga memungkinkan pelacakan historis penjualan serta analisis tren yang bisa digunakan sebagai dasar pengambilan keputusan. Dalam konteks penelitian ini, fitur-fitur tersebut diimplementasikan melalui integrasi antara antarmuka pengguna berbasis *React* dan *backend Laravel* yang menyediakan API untuk pengelolaan data stok dan pesanan.

## 2.5 Konsep *Usability Testing*

*Usability Testing* adalah proses evaluasi sistem dari perspektif *user* untuk mengetahui seberapa efektif, efisien, dan memuaskan suatu aplikasi digunakan oleh target *user*. Evaluasi dilakukan dengan cara memberikan tugas spesifik kepada pengguna untuk diselesaikan menggunakan aplikasi, kemudian diamati apakah mereka mengalami kendala dalam interaksi tersebut. Salah satu metode umum yang digunakan adalah pengujian langsung dengan beberapa orang pengguna disertai observasi, wawancara, atau kuesioner [16]. Dalam penelitian ini, *usability testing* digunakan untuk menilai kualitas pengalaman pengguna (*user experience*) terhadap aplikasi yang dikembangkan untuk tim *sales* PT Pacific Metalindo Perkasa. Tujuan dari *usability testing* adalah mengidentifikasi apakah fitur-fitur seperti pembuatan *sales order*, pencarian produk, dan pengecekan stok mudah digunakan serta dipahami oleh *user*. Hasil dari pengujian ini nantinya digunakan untuk melakukan perbaikan pada antarmuka dan alur kerja aplikasi agar semakin selaras dengan kebutuhan pengguna nyata di lapangan. Metode ini penting untuk memastikan aplikasi tidak hanya fungsional secara teknis, tetapi juga optimal dalam hal kenyamanan dan efektivitas penggunaannya.

## 2.6 Skala *Likert*

Skala *Likert* adalah alat ukur yang digunakan untuk mengetahui tingkat sikap, opini, atau persepsi seseorang terhadap suatu pernyataan. Skala ini pertama kali diperkenalkan oleh Rensis Likert, seorang psikolog sosial asal Amerika Serikat, pada tahun 1932 dalam penelitiannya yang berjudul "*A Technique for the Measurement of Attitudes.*" [17] Tujuan utama dari penggunaan skala ini adalah untuk menyederhanakan proses pengukuran variabel yang bersifat subjektif, seperti kepuasan, kepercayaan, atau tingkat persetujuan terhadap layanan, produk, atau kebijakan tertentu. Dengan menyajikan pilihan tanggapan dari "Sangat Tidak Setuju" hingga "Sangat Setuju". Selain itu, skala *Likert* juga memfasilitasi pengambilan keputusan berbasis data dengan mengidentifikasi pola kecenderungan responden terhadap suatu isu, sehingga hasil penelitian menjadi informatif.

Tabel 2.1. Bentuk Skala *Likert* [1]

Skor	Kategori Respon
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Netral
4	Setuju
5	Sangat Setuju

### 2.6.1 Perhitungan Skala *Likert*

Berikut beberapa rumus yang digunakan dalam perhitungan Skala *Likert* [2]:

- (a) Total skor yang diperoleh didapatkan dari hasil perhitungan sebagai berikut:

$$\text{Total Skor} = \text{Jumlah Responden pada Skala} \times \text{Nilai Skala} \quad (2.1)$$

- (b) Skor Maksimal diperoleh dari hasil perhitungan sebagai berikut:

$$\text{Jumlah Responden} \times \text{Skor Tertinggi} \times \text{Jumlah Pernyataan pada Komponen} \quad (2.2)$$

- (c) Skor Persentase untuk menghitung hasil dari perhitungan pada setiap komponen dalam bentuk persentase, dengan rumus sebagai berikut:

$$\text{Skor Persentase} = \left( \frac{\text{Total Skor yang Diperoleh}}{\text{Skor Maksimal}} \right) \times 100\% \quad (2.3)$$

Setelah memperoleh nilai persentase dari hasil perhitungan skor total berdasarkan skala *Likert*, langkah selanjutnya adalah melakukan interpretasi terhadap nilai tersebut guna menentukan tingkat kelayakan atau efektivitas dari sistem yang diuji. Interpretasi ini dilakukan dengan mengacu pada rentang persentase tertentu yang telah ditetapkan, sehingga hasil evaluasi

dapat diklasifikasikan ke dalam kategori kualitatif yang lebih mudah dipahami, seperti "Sangat Baik", "Baik", "Cukup", "Kurang Baik", atau "Buruk". Adapun tabel berikut menyajikan kategori interpretasi hasil berdasarkan persentase yang telah dihitung:

Tabel 2.2. Kriteria Interpretasi Skor Berdasarkan Interval Persentase [2]

<b>Rentang Persentase</b>	<b>Interpretasi</b>
0% – 19,99%	Sangat Tidak Setuju / Buruk / Kurang Sekali
20% – 39,99%	Tidak Setuju / Kurang Baik
40% – 59,99%	Cukup / Netral
60% – 79,99%	Setuju / Baik / Suka
80% – 100%	Sangat Setuju / Sangat Baik / Sangat Suka

UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA