

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Landasan Teori**

Teori-teori yang mendasar sebagai penunjang proses penelitian yaitu sebagai berikut:

##### **2.1.1 Sistem Informasi**

Sistem informasi adalah suatu sistem yang tersusun dari elemen-elemen atau komponen-komponen yang saling terhubung dan berinteraksi satu sama lain untuk mengumpulkan, memproses, menyimpan, mengambil kembali dan berbagi informasi atau data untuk pengambilan keputusan, pengendalian dan operasi dalam satu organisasi atau lingkungan bisnis. Sistem informasi juga dapat berperan sebagai alat untuk menghasilkan laporan atau hasil yang dibutuhkan oleh manajemen atau pengguna lainnya[2].

Tujuan utama pengembangan sistem informasi adalah untuk meningkatkan kinerja suatu bisnis atau organisasi melalui penggunaan teknologi informasi secara efektif dan efisien. Sistem informasi dapat meningkatkan produktivitas, serta dapat membantu meningkatkan kualitas informasi yang tersedia. Adapun komponen dalam system informasi yaitu sebagai berikut:

1. Perangkat keras (*hardware*), merupakan komponen fisik seperti *computer*, *server*, *printer*, dan lainnya.
2. Perangkat lunak (*software*), berupa program atau aplikasi yang dijalankan pada perangkat keras untuk memproses data atau informasi.
3. Data dapat berupa data *user*, produk, dan transaksi.
4. Prosedur, berupa aturan yang digunakan dalam mengumpulkan, memproses, dan menyimpan data dalam sistem informasi.
5. Personil yaitu terdiri atas orang-orang yang bertugas untuk mengoperasikan sistem informasi.
6. Jaringan untuk menghubungkan berbagai perangkat keras atau *software* dalam sistem informasi[2].

##### **2.1.2 Pemesanan**

Pemesanan adalah tindakan meminta atau memesan barang atau jasa dari penyedia tertentu. Proses ini termasuk mengirimkan permintaan, menentukan persyaratan, memilih produk atau layanan yang diinginkan, dan menyepakati waktu dan harga. Pesanan sering ditempatkan dalam perjalanan bisnis atau perdagangan oleh konsumen atau pelanggan yang membeli barang atau jasa secara langsung, selain itu oleh perusahaan atau organisasi yang membutuhkan barang atau jasa

untuk tujuan operasional. Dunia bisnis, pesanan sering ditempatkan melalui system informasi pesanan terintegrasi untuk memfasilitasi proses pemesanan dan manajemen persediaan[3].

### 2.1.3 QR Code

Menurut (soon,2008), kode QR adalah suatu jenis kode matriks atau kode batang dua dimensi yang dikembangkan oleh Denso Wave, sebuah divisi Denso Corporation yang merupakan sebuah perusahaan Jepang dan dipublikasikan pada tahun 1994 dengan fungsionalitas utama yaitu dapat dengan mudah dibaca oleh pemindai. QR merupakan singkatan dari *quick response* atau respon cepat, yang sesuai dengan tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respon yang cepat pula. Berbeda dengan kode batang yang hanya menyimpan informasi secara *horizontal*, kode QR mampu menyimpan informasi secara *horizontal* dan *vertical*, oleh karena itu secara otomatis kode QR dapat menampung informasi yang lebih banyak daripada kode batang[4].



**Gambar 2. 1** Kode QR

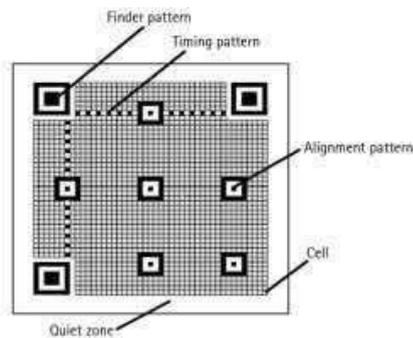
Awalnya kode QR digunakan untuk pelacakan kendaraan bagian di manufaktur, namun kini kode QR digunakan dalam konteks yang lebih luas, termasuk aplikasi komersial dan kemudahan pelacakan aplikasi berorientasi yang ditujukan untuk pengguna telepon seluler. Di Jepang, penggunaan kode QR sangat populer, hamper semua jenis ponsel di Jepang bisa membaca kode QR sebab Sebagian besar pengusaha disana telah memilih kode QR sebagai alat tambahan dalam program promosi produknya, baik yang bergerak dalam perdagangan maupun dalam bidang jasa. Pada umumnya kode QR digunakan untuk menanamkan informasi alamat situs suatu perusahaan[4].

**Tabel 2. 1** Kelebihan dan Kekurangan QR Code

Kelebihan	Kekurangan
Mudah digunakan	Tergantung pada teknologi
Kapasitas penyimpanan besar	Terbatas cakupan penggunaan

Dapat dihasilkan dengan mudah	Memerlukan koneksi internet
Fleksibel	Membutuhkan ruang penyimpanan yang cukup

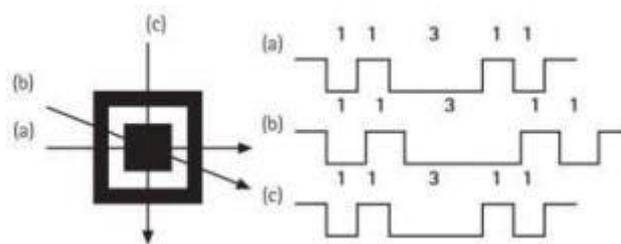
Menurut (soon, 2008), *QR Code* adalah simbol jenis matriks dengan struktur sel yang disusun dalam kotak. Ini terdiri dari pola fungsi untuk membuat pembacaan mudah dan area data tempat data disimpan. *QR code* memiliki pola pencari, pola pelurusan, pola waktu, dan zona tenang.



**Gambar 2. 2** Struktur Kode QR

#### a. Finder Pattern

Pola untuk mendeteksi posisi *QR Code*. Dengan mengatur pola ini pada tiga sudut simbol, posisi, ukuran, dan sudut simbol dapat dideteksi. Pola finder ini terdiri dari struktur yang dapat dideteksi di semua arah ( $360^\circ$ ).



**Gambar 2. 3** Finder Pattern (Soon,2008)

Rasio antara hitam dan putih di antara garis pemindaian yang berjalan melalui pola *finder* selalu 1: 1: 3: 1: 1 bila dilihat dari arah mana saja di sekitar  $360^\circ$  yang mengelilinginya, dengan mendeteksi rasio khusus ini, pola penemu dapat dideteksi dari antara gambar yang ditangkap oleh sensor CCD (area sensor) untuk mengidentifikasi posisi Kode QR dalam waktu singkat.

#### b. Timing Pattern

Suatu pola untuk mengidentifikasi koordinat pusat setiap sel dalam QR Code dengan pola-pola hitam dan putih yang disusun bergantian. Ini digunakan untuk memperbaiki koordinat pusat

sel data ketika simbol terdistorsi atau ketika ada kesalahan untuk pitch sel, diatur dalam arah vertical dan horizontal.

c. *Quiet Zone*

Ruang margin yang diperlukan untuk membaca QR Code. Zona tenang ini membuatnya lebih mudah untuk memiliki simbol yang terdeteksi dari antara gambar yang dibaca oleh sensor CCD (area sensor). Empat atau lebih sel diperlukan untuk zona tenang.

d. *Data Area*

Data QR Code akan disimpan (encoded) ke dalam area data. Data akan dikodekan ke dalam angka biner „0“ dan „1“ berdasarkan pada aturan encoding. Angka biner „0“ dan „1“ akan diubah menjadi sel hitam dan putih kemudian akan diatur. Area data akan memiliki kode Reed-Solomon yang digabungkan untuk data yang disimpan dan fungsi koreksi kesalahan[5].

### 2.1.4 Metode Waterfall

Metode air terjun, yang juga dikenal sebagai metode waterfall, sering disebut sebagai siklus hidup klasik (classic life cycle). Nama sebenarnya dari model ini adalah "Linear Sequential Model," yang menggambarkan pendekatan sistematis dan berurutan dalam pengembangan perangkat lunak. Prosesnya dimulai dengan spesifikasi kebutuhan pengguna dan berlanjut melalui tahapan-tahapan perencanaan, permodelan, konstruksi, serta penyerahan sistem kepada pengguna, dan diakhiri dengan dukungan terhadap perangkat lunak yang telah dihasilkan.

1) *Analisa Kebutuhan*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan Batasan perangkat lunak tersebut. Informasi dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2) *Desain*

Tahap ini pengembang membuat desain sistem yang dapat membantu menentukan perangkat keras (*hardware*) dan sistem persyaratan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3) *Implementation*

Tahap ini sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit *testing*.

4) *Verification*

Sistem dilakukan verifikasi dan pengujian apakah sistem sepenuhnya atau Sebagian memenuhi

persyaratan sistem. Pengujian dapat dikategorikan ke dalam unit *testing* (dilakukan pada modul tertentu kode), sistem pengujian (untuk melihat bagaimana sistem bereaksi ketika semua modul yang terintegrasi) dan penerimaan pengujian (dilakukan dengan atau nama pelanggan untuk melihat apakah semua kebutuhan pelanggan puas).

#### 5) *Maintenance*

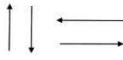
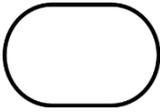
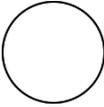
Tahap terakhir dari metode *waterfall* yaitu perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya[6].

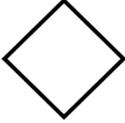
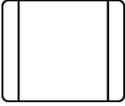
### 2.1.5 Rekayasa Perangkat Lunak

#### A. Flowchart

*Flowchart* merupakan suatu diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Tabel 2.2 merupakan simbol-simbol *flowchart*:

**Tabel 2. 2** Simbol *Flowchart*

	<p><b><i>Flow</i></b></p> <p>Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain.</p>
	<p><b><i>Terninator</i></b></p> <p>Simbol yang menyatakan awal atau akhir suatu program</p>
	<p><b><i>On-Page Reference</i></b></p> <p>Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama</p>
	<p><b><i>Off-Page Reference</i></b></p> <p>Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang berbeda</p>

	<p><b>Process</b></p> <p>Simbol yang menyatakan suatu proses yang dilakukan komputer</p>
	<p><b>Decision</b></p> <p>Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak</p>
	<p><b>Input/output</b></p> <p>Simbol yang menyatakan proses input atau output tanpa tergantung peralatan</p>
	<p><b>Manual Operation</b></p> <p>Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer</p>
	<p><b>Document</b></p> <p>Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik atau output yang perlu dicetak</p>
	<p><b>Predefine Proses</b></p> <p>Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur</p>
	<p><b>Display</b></p> <p>Simbol yang menyatakan peralatan output yang digunakan</p>
	<p><b>Preparation</b></p> <p>Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal</p>

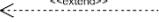
## B. Unified Modeling Language (UML)

Terdapat beberapa diagram UML yang digunakan dalam pengembangan sebuah sistem yaitu:

### a. Use Case Diagram

*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan system informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat[7]. Simbol *use case* dapat dilihat pada Tabel 2.3

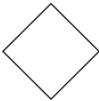
**Tabel 2. 3** Simbol *Use Case*

No	Simbol	Nama Simbol dan Keterangan
1		<i>Actor</i> Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		<i>Dependency</i> Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri ( <i>independent</i> )
3		<i>Generalization</i> Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atas objek induk ( <i>ancestor</i> ).
4		<i>Include</i> Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
5		<i>Extend</i> Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

**b. Entity Relationship Diagram (ERD)**

Suatu model atau desain yang digunakan untuk membangun database, dengan tujuan mempermudah dalam menggambarkan data yang memiliki hubungan atau relasi dalam bentuk desain. Adanya *entity relationship diagram*, sistem database yang dihasilkan dapat digambarkan secara lebih terstruktur dan rapi[8]. Simbol ERD dapat dilihat pada table 2.4.

**Tabel 2. 4** Simbol ERD

No	Simbol	Nama Simbol dan Keterangan
1		Entitas Suatu simbol yang digunakan untuk mewakili objek dengan karakteristik serupa, dilengkapi dengan atribut.
2		Relasi Simbol yang digunakan untuk mengaitkan beberapa entitas berdasarkan fakta dalam suatu konteks.
3		Atribut Simbol yang menggambarkan karakteristik suatu entitas serta relasinya.
4		<i>Connection</i> Simbol yang digunakan untuk menghubungkan keterkaitan antara notasi-notasi seperti entitas, relasi, dan atribut.
5	*	<i>Primary key</i> Kunci utama yang digunakan untuk mengidentifikasi data secara unik dalam tabel.
6	**	<i>Foreign key</i> Penghubung antar tabel yaitu untuk menjalankan fungsi kedua dari key dalam database.

### 2.1.6 Pemrograman Berorientasi Objek (PBO)

Pemrograman Berorientasi Objek (OOP) merupakan suatu pendekatan baru dalam berpikir dan bernalar untuk menyelesaikan masalah-masalah dengan bantuan komputer, di mana setiap objek merupakan entitas tunggal yang menggabungkan struktur data dan fungsi tertentu. Objek-objek ini bisa berupa orang, tempat, benda, kejadian, atau konsep-konsep yang ada di dunia nyata yang relevan dengan suatu aplikasi. Misalnya, objek dapat berupa benda seperti mesin, gedung, komputer, mobil, dan lain-lain. Objek juga bisa berupa kejadian seperti pembayaran uang pendidikan, registrasi biodata siswa, membaca buku, dan sebagainya. Dengan kata lain, objek adalah sesuatu yang dapat dilihat, disentuh, dirasakan, atau diraba untuk mendapatkan manfaat. Pemrograman yang menggunakan “objek” -entitas yang menyatukan data dan metode yang bekerja pada data tersebut[9]. OOP menawarkan cara baru untuk berpikir dan menyelesaikan masalah dengan memodelkan dunia nyata dalam bentuk program komputer.

Konsep utama dalam OOP:

1. Kelas (*Class*): Kelas adalah cetak biru atau template untuk membuat objek. Menentukan atribut dan metode yang dimiliki oleh objek.
2. Objek (*Object*): Objek adalah instansi dari kelas, yang memiliki state (data) dan perilaku (metode dan fungsi).
3. Enkapsulasi (*Encapsulation*): Proses melindungi detail implementasi dari pengguna dan hanya menampilkan fitur penting, dilakukan melalui modifier akses seperti *private*, *protected*, dan *public*.
4. Pewarisan (*Inheritance*): Mekanisme dimana sebuah kelas dapat mewarisi atribut dan metode dari kelas lain, mendukung penggunaan ulang kode dan mengurangi redundansi.
5. Polimorfisme (*Polymorphism*): Kemampuan untuk menggunakan metode atau fungsi yang sama dengan berbagai cara, memungkinkan satu antarmuka untuk berfungsi dengan berbagai tipe objek.
6. Abstraksi (*Abstraction*): Proses penyederhanaan kompleksitas dengan menyembunyikan detail yang tidak diperlukan dan hanya menampilkan informasi yang relevan.

### 2.1.7 Basis Data

Basis data, atau sering disebut database, merupakan sekumpulan informasi yang tersimpan di dalam komputer secara terstruktur, memungkinkan pengguna untuk mengaksesnya melalui program komputer guna mendapatkan informasi yang diperlukan. Perangkat lunak yang bertugas mengelola dan menjalankan kueri pada basis data ini dikenal sebagai Sistem Manajemen Basis Data (*Database Management System*, DBMS). Studi tentang sistem basis data merupakan bagian dari bidang ilmu informasi[10].

Dasar dari konsep basis data adalah koleksi catatan atau segmen pengetahuan. Sebuah basis data memiliki struktur terorganisir yang menjelaskan jenis fakta yang tersimpan di dalamnya, yang disebut sebagai skema. Skema menggambarkan objek yang terdapat dalam basis data serta hubungan antara objek tersebut. Ada berbagai pendekatan untuk menyusun skema atau memodelkan struktur basis data, yang dikenal sebagai model basis data atau model data. Salah satu model yang umum digunakan saat ini adalah model relasional, yang secara sederhana mempresentasikan informasi dalam bentuk tabel-tabel yang saling berhubungan, di mana setiap tabel terdiri dari baris dan kolom (dalam istilah matematika). Dalam model ini, hubungan antara tabel dijelaskan dengan menggunakan nilai yang sama di antara tabel-tabel tersebut. Model lain, seperti model hierarkis dan jaringan, menggunakan pendekatan yang lebih eksplisit untuk mewakili hubungan antar tabel. Struktur dasar sebuah basis data terdiri dari beberapa tingkatan sebagai berikut:

- a. Karakter, merupakan elemen data terkecil yang bisa berupa angka, huruf, atau karakter spesifik lainnya yang membentuk suatu item data atau kolom.
- b. Field/item, adalah representasi dari suatu atribut dan sekumpulan record (rekaman/tupel) yang serupa, menunjukkan suatu item data tertentu.
- c. Record/rekaman/tupel: Sekelompok field membentuk satu rekaman atau tupel. Rekaman menggambarkan satu unit data individu yang spesifik.
- d. File, adalah kumpulan rekaman yang serupa yang menggambarkan satu kesatuan data yang sama. Masing-masing file mewakili satu set data tertentu.
- e. Database, merupakan koleksi file atau tabel yang membentuk satu kesatuan basis data[10].

### 2.1.8 Hypertext Preprocessor (PHP)

PHP adalah bahasa pemrograman web yang diciptakan untuk pengembangan aplikasi berbasis web. Awalnya dikembangkan oleh Rasmus Lerdorf pada akhir tahun 1994 sebagai alat untuk memantau pengunjung situs pribadinya. PHP dirancang untuk membangun situs web yang dinamis dan interaktif, di mana konten dapat berubah sesuai dengan kondisi tertentu dan pengguna dapat memberikan umpan balik. Sebagai bahasa pemrograman *server-side*, PHP diproses oleh *server* dan hasilnya dikirim kembali ke *browser*. Oleh karena itu, untuk menggunakan PHP, diperlukan akses ke *server*. PHP digunakan bersama dengan HTML untuk membuat halaman web dinamis, termasuk menerima, memproses, dan menampilkan data dari *database server*. PHP juga sering digunakan untuk mengelola informasi di internet dan merupakan perangkat lunak sumber terbuka atau gratis.

Salah satu keunggulan PHP adalah kemudahannya dalam berintegrasi dengan database. PHP dapat secara langsung mendukung beberapa database tanpa perlu menginstal konektor tambahan seperti yang diperlukan dalam bahasa pemrograman Java. Hal ini membuat PHP sangat fleksibel dalam kaitannya dengan berbagai jenis *database*. Di antara beberapa database yang sering dipadukan dengan PHP, MySQL adalah yang paling umum. Untuk menghubungkan PHP dengan *database*, Anda hanya perlu mengetahui nama dan lokasi *database*, serta *username* dan *password* untuk mengaksesnya. Berikut adalah beberapa keuntungan menggunakan PHP:

- a. Respons cepat: Karena PHP ditulis di tengah-tengah kode HTML, waktu respon programnya lebih cepat.
- b. Biaya rendah: PHP seringkali gratis dan tidak memerlukan biaya lisensi.
- c. Mudah digunakan: Fitur dan fungsinya lengkap, sangat cocok untuk membuat halaman web dinamis.
- d. Kompatibilitas sistem operasi: PHP dapat dijalankan pada berbagai sistem operasi, termasuk *Windows*, *Linux*, *Mac OS*, dan berbagai varian *Unix*.
- e. Dukungan teknis yang luas: Ada banyak forum dan situs yang didedikasikan untuk membantu menyelesaikan masalah terkait PHP.
- f. Keamanan: Kode PHP tidak terlihat oleh pengunjung, sehingga meningkatkan keamanan.
- g. Mendukung berbagai *database*.
- h. Kemampuan untuk disesuaikan: Karena PHP adalah perangkat lunak sumber terbuka, dapat menyesuaikannya sesuai kebutuhan.



