

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Teori

Berdasarkan pada bab ini pengkaji menerangkan terkait penerangan para ahli yang dipergunakan pada pengkajian ini.

2.1.1 Warehouse

Warehouse merupakan tempat penyimpanan barang dan tempat berlangsungnya proses produksi seperti kegiatan penerimaan, kegiatan penyimpanan, kegiatan pemeliharaan, dan kegiatan distribusi. Dalam proses kelola di *warehouse* yang harus diperhatikan jumlah serta waktu dalam agar terjaminnya kualitas dan kuantitas barang yang terdapat di gudang serta dalam pendistribusiannya (Hamidin et al. 2018).

Gudang disebut sebagai lokasi logistik perusahaan yang berfungsi sebagai fasilitas penyimpanan sementara sebelum barang didistribusikan. Selain memberikan laporan penyimpanan berupa dokumen yang menjelaskan tentang pengeluaran dan penerimaan barang yang disimpan di gudang atau *warehouse* (Ardika and Ratama 2022)

2.1.2 Aplikasi

Aplikasi ialah subkategori *software computer* dimana secara langsung menggunakan keahlian *computer* dalam melaksanakan pekerjaan pengguna. Umumnya dari pada *software* sistem yang menggabungkan banyak fungsi komputer namun tidak dapat menggunakan fungsi tersebut untuk kepentingan pengguna. Contoh penerapannya ialah penganalisisan kalimat, *spreadsheet*, serta

menjalankan sumber. Banyak program dimana dibundel terkadang disebut sebagai "*bundel*" yaitu *Microsoft Office* dan *Open Office* mengintegrasikan program analisa kalimat, *spreadsheet* dan banyak program lain. Program dengan integrasi menggunakan fungsi *online* dengan sesama pengguna, membuat setiap aplikasi lebih mudah dipelajari dan digunakan. Dalam kebanyakan kasus, *user* untuk berinteraksi program tersebut dapat mendukung satu sama lain untuk kepentingan pengguna. Misalnya, *spreadsheet* dapat dimasukkan ke dalam dokumen pengolah kata meskipun dibuat dalam aplikasi *spreadsheet* terpisah (Wikipedia, 2022).

Aplikasi *Android* dibangun pakai memeruntukkan bahasa penyusunan *Java* dan *SDK*. *Editor* yang digunakan menjelang membentuk program *Android* ialah *Android Studio*, *Netbeans IDE*, *Eclipse*. Pada penyusunan praktik *mobile parking* ini, pengembara memeruntukkan *Android Studio Editor* bagian dalam proyeknya. Praktik *Android* telah mempunyai pasarnya sendiri, sehingga *developer* tak mesti cemas pakai perdagangan praktik (Prayoga and Simanjuntak 2020).

2.1.3 Android

Android merupakan sistem operasi terbuka (*open source*) serta *Google* mempublikasi merknya atas nama *Apache*. Situasi ini memperlihatkan *Google* mengizinkan siapa saja untuk memodifikasi, membagikan, mengembangkan, dan memodifikasi program *Android*. watak *Android* yang terbuka serta dapat diskalakan mendorong penggunaan metode *Android* saat menggunakan alat elektro lainnya yaitu laptop, buku pintar, TV pintar, dan kamera. lalu, sistem *Android* juga disematkan pada kaca mata data, arloji, *CD/DVD* mobil, pemutar *multimedia portabel*, telepon *VoIP*, dan lain-lain (Kie and Simanjuntak 2022).

Pada tahun 2000, android diciptakan oleh perusahaan kecil yang bernama *Android Inc*, yang selanjutnya di beli oleh perusahaan *Google Inc*. *Android* ialah sistem operasi berbasis kernel *linux*. *Android* seperti tumpukan *software* yang berisi program pendukung yang memiliki fungsi spesifik dari sistem operasi. pembangunan *Android* dilaksanakan di *Open Handset Alliance (OHA)* dan banyak perusahaan lainnya. Menggunakan *Open Handset Alliance (OHA)* untuk mengembangkan teknologi ponsel. *Open Handset Alliance (OHA)* mencakup 34 perusahaan yang didukung oleh Google Android. Google Android adalah peluasan bermula pada instruksi dasar susunan penggunaan *Android*. Sistem penggunaan yang digunakan pada *Android* berfundamen *Linux* yang melingkupi sistem penggunaan aplikasi, *middleware*, serupa penyedia prosedur terbuka kepada pengeditan aplikasi (Saputra and Putri 2021).

Sistem pelaksanaan *Android* perkiraan berfungsi jauh sejak didirikan di *Open Handset Alliance* pada tahun 2007. *Android* adalah sistem pelaksanaan terbuka yang dipromosikan secara radikal oleh *Google*. Banyak instrumen nirkabel di berbagai dunia memperuntukkan kaidah pelaksanaan *Android*. Perangkat lain serupa *tablet*, *e-book*, *set-top box* bahkan otomiobil juga memakai sistem pelaksanaan *Android* (Simanjuntak and Alfisyahri 2020).

2.1.4 Unified Modeling Language (UML)

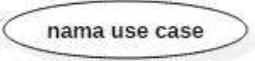
Unified Modeling Language bisa disebut dengan bahasa yang mempergunakan foto dan objek, membuat dan mendeskripsikan perangkat lunak berorientasi objek. Tujuan *UML* adalah memberikan arsitektur sistem, desain perangkat lunak, dan pengembangan perangkat lunak sebagai alat analisis untuk

implementasi sistem berbasis perangkat lunak. Pemodelan berorientasi objek dan visual mempergunakan banyak tipe diagram *UML* dasar. Diagram dasar datang dalam bentuk diagram *use case*, diagram kelas, diagram urutan, diagram diagram keadaan, diagram aktivitas (Huda, Ir. Fariani Hermin Indiyah, and Ratna Widyati, S.Si. 2021)

1. Use Case Diagram

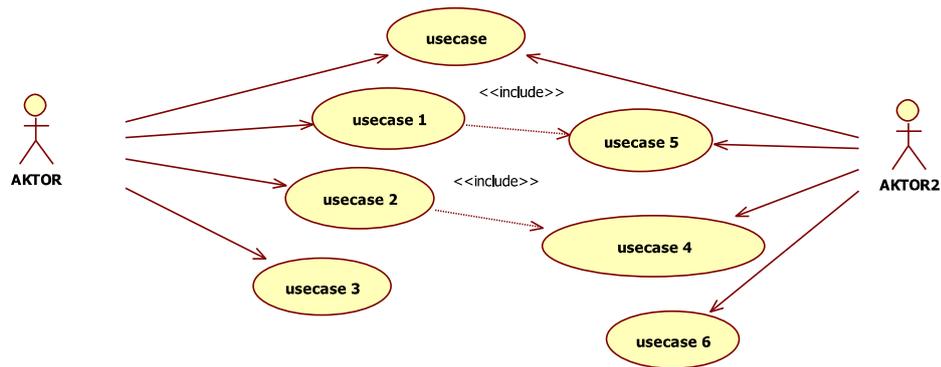
Sebuah *use case* diagram berkaitan dengan perilaku (*behaviour*) dari sistem. Model *use case* adalah setiap sistem operasi yang berkomunikasi diantara aktor dan sistem pada bagian obrolan (Huda et al. 2021)

Tabel 2. 1 Simbol Use Case Diagram

Bentuk	Keterangan
<p><i>Use case</i></p> 	<p>Utilitas <i>use case</i> ini sebagai penyedia sistem menjadikan entitas yang bertukar kalimat diantar entitas atau tokoh.</p>
<p>Aktor/<i>actor</i></p> 	<p>Orang, proses, atau sistem lain yang bertindak menjadi penggerak sistem tujuan. harapan dari sistem ini adalah untuk mengidentifikasi operator item pekerjaan yang terkait pada tujuan target dari sistem itu sendiri. Aktor serta <i>use case</i> berinteraksi, tetapi mereka tidak dapat mengontrol <i>use case</i>.</p>

<p>asosiasi/<i>association</i></p> 	<p>Tautan antara aktor & <i>use case</i> menjadi tanda siapa atau apa yang secara pribadi meminta interaksi</p>
<p>Ekstensi/<i>extend</i></p> <p><<<i>extend</i>>></p> 	<p>Hubungan <i>use case</i> tambahan dengan sebuah <i>use case</i>, dimana <i>use case</i> yang ditambahkan juga dapat berdiri sendiri tanpa <i>use case</i> lainnya. Panah menunjuk ke kasus penggunaan yang ditambahkan.</p>
<p>generalisasi/<i>generalization</i></p> 	<p>Hubungan antara generalisasi dan spesialisasi (umum – khusus) antara dua <i>use case</i> dimana satu fitur lebih umum dari yang lain. Arah panah menunjuk ke <i>use case</i> yang merupakan generalisasi (umum).</p>
<p>Menggunakan/<i>include/uses</i></p> <p><<<i>include</i>>></p>  <p><<<i>uses</i>>></p> 	<p>Hubungan <i>use case</i> tambahan ke <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> tersebut untuk melakukan tugasnya atau merupakan kondisi untuk kinerja <i>use case</i> tersebut. Panah menunjuk ke kasus penggunaan yang ditambahkan</p>

Sumber : (Hendini 2016)



Gambar 2. 1 Contoh *Usecase Diagram*

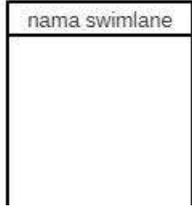
Sumber : (Data Peneliti 2022)

2. *Activity Diagram*

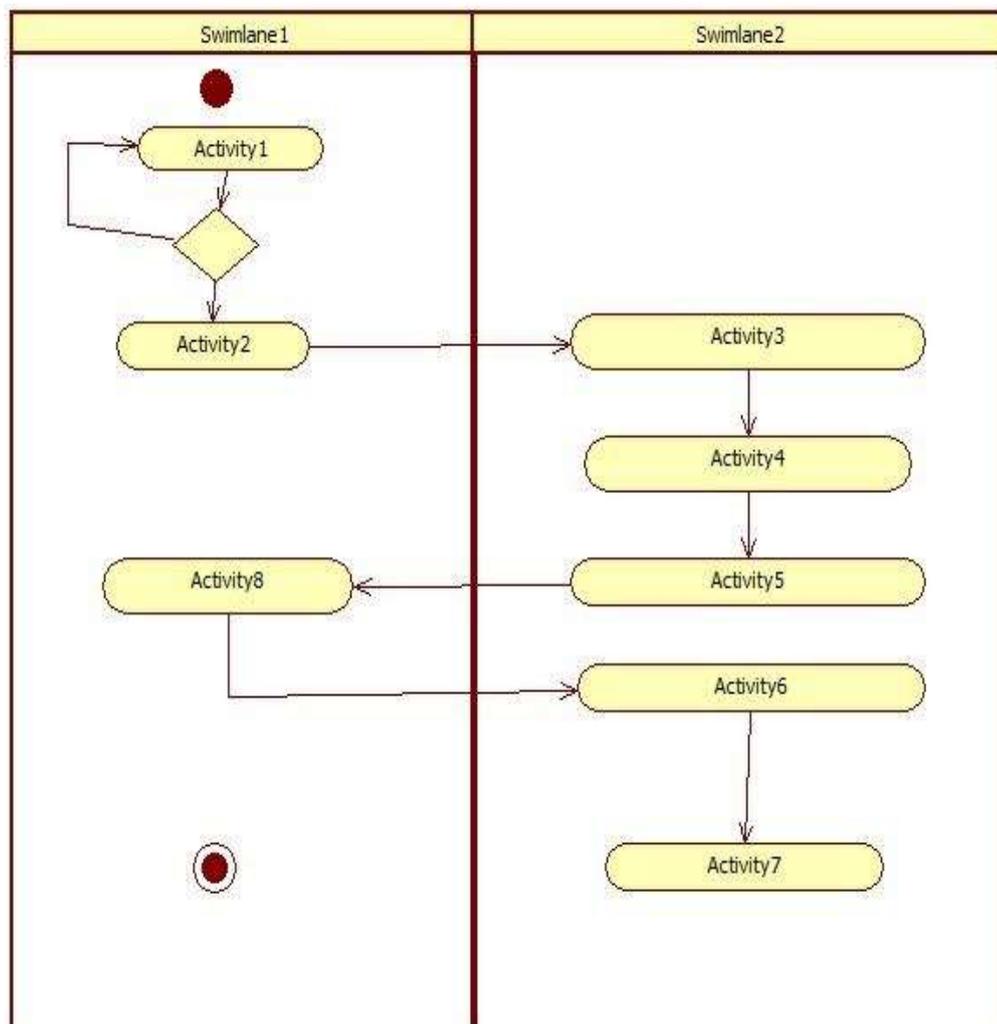
Diagram fungsi adalah diagram antara dua atau lebih objek kelas yang menjelaskan kontrol saat memproses fungsi. titik awal yang terkait dengan fungsi mulai sebagai awal dari diagram tindakan. Tugasnya adalah persegi panjang bulat dengan nama di dalamnya. Menghubungkan satu aktivitas dengan aktivitas lainnya menggunakan garis transisi (*decision point*). Juga, penyelesaian proses pemodelan terkait dengan titik akhir (Huda et al. 2021)

Tabel 2. 2 Simbol *Activity Diagram*

Simbol	Keterangan
Status awal 	Keadaan awal operasi sistem, diagram operasi memiliki keadaan awal
Aktifitas 	Tindakan yang dilakukan oleh sistem Tindakan biasanya dimulai dengan kata kerja
Percabangan/ <i>decision</i>	Asosiasi industri dengan lebih dari satu peluang operasi

	
<p>Penggabungan/<i>join</i></p> 	<p>Asosiasi peracikan, di mana lebih dari satu kegiatan digabungkan menjadi satu</p>
<p>Status akhir</p> 	<p>Keadaan akhir dari sistem, diagram fungsional memiliki keadaan akhir</p>
<p><i>Swimlane</i></p>  <p style="text-align: center;">atau</p> 	<p>Sebuah organisasi bisnis yang bertanggung jawab untuk kegiatan tertentu</p>

Sumber : (Hendini 2016)



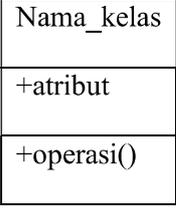
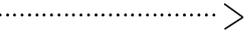
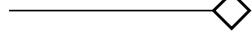
Gambar 2. 2 Activity Diagram

Sumber : (Data Peneliti, 2022)

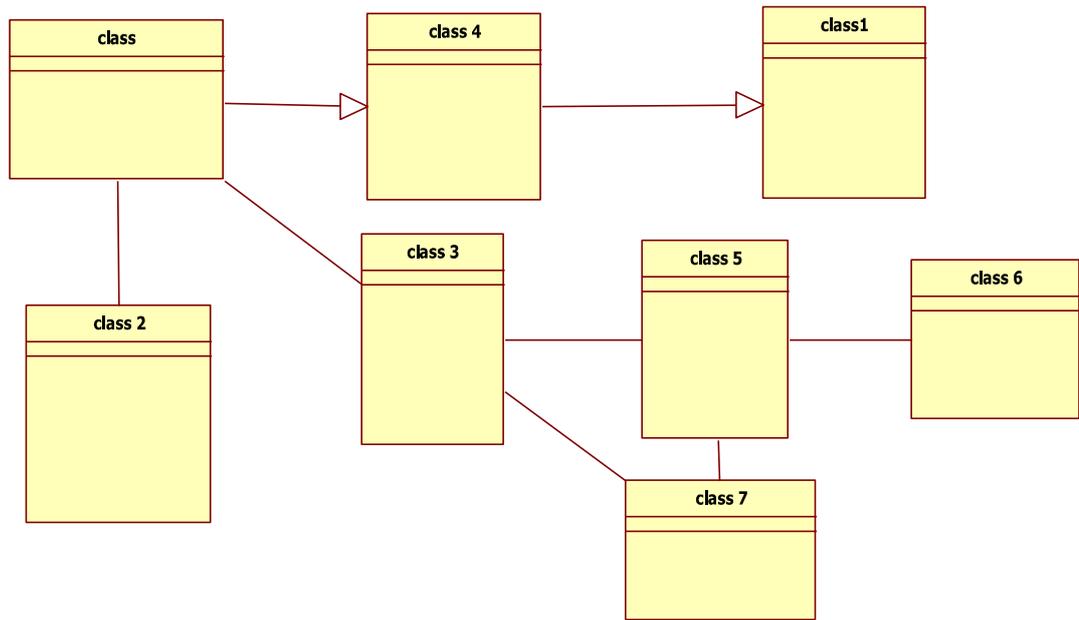
3. *Class Diagram*

Class diagram merupakan penjelasan dari masing-masing *class* yang saling terkait satu sama lain dalam model desain sistem, mendefinisikan perilaku sistem, menunjukkan aturan dan tanggung jawab entitas. Diagram kelas bertindak sebagai atribut dan fungsi kelas, dan kendala terkait dengan objek terkait (Hendini 2016)

Tabel 2. 3 Simbol Class Diagram

Simbol	Deskripsi
<p>Kelas</p> 	<p>Kelas pada struktur <i>system</i></p>
<p>Antarmuka / <i>interface</i></p> 	<p>Hal yang sama berlaku untuk konsep antarmuka pengguna pemrograman berorientasi objek</p>
<p>Asosiasi / <i>association</i></p> 	<p>Asosiasi antar kelas yang memiliki arti umum biasanya melibatkan himpunan</p>
<p>Asosiasi berarah / <i>directed association</i></p> 	<p>Hubungan antar class dengan makna suatu class digunakan oleh class lain, asosiasi biasanya melibatkan berbagai macam</p>
<p>Generalisasi</p> 	<p>Hubungan antar kelas dengan makna generalisasi-spesialisasi (umum)</p>
<p>Kebergantungan / <i>dependency</i></p> 	<p>Hubungan antar kelas menurut ketergantungan antar kelas</p>
<p>Agregasi / <i>aggregation</i></p> 	<p>Semua-bagian (<i>whole-part</i>)</p>

Sumber : (Hendini 2016)



Gambar 2.3 Class Diagram

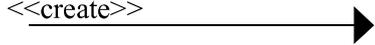
Sumber : (Data Peneliti, 2022)

4. Sequence Diagram

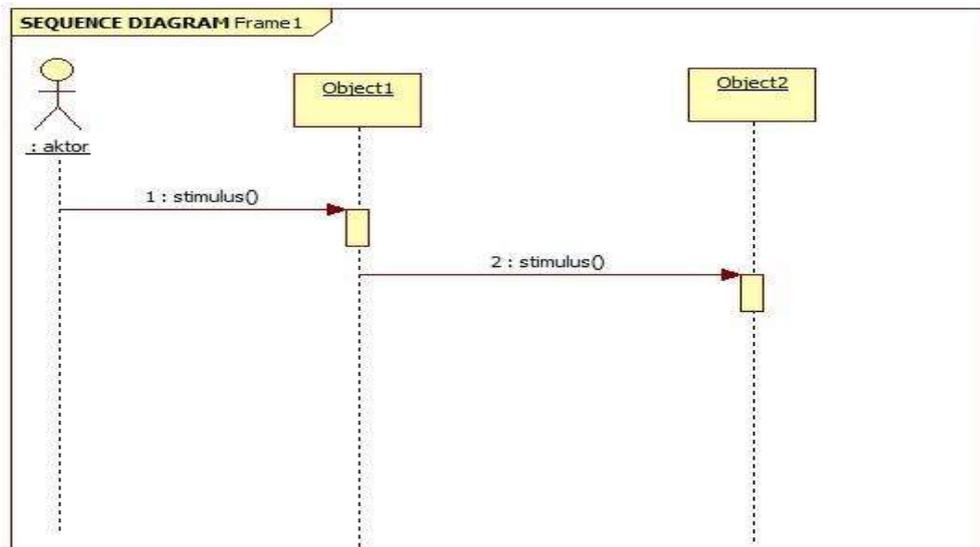
Sequence diagram adalah representasi dari perilaku objek dalam *use case* yang memperlihatkan masa pakai barang serta pesan yang dikirim dan diterima antar objek (Hendini 2016).

Tabel 2.4 Simbol *Sequence* Diagram

Simbol	Deskripsi
<p>Aktor/<i>actor</i></p>  <p>nama aktor</p>	<p>Seseorang, proses dan sistem lain yang bekerja satu sama lain dengan sistem informasi yang dirancang di luar sistem informasi yang dikembangkan. Aktor belum tentu orang, tetapi biasanya diwakili oleh kata benda di awal <i>bibliografi</i> aktor.</p>

<p>Garis hidup/<i>lifeline</i></p> 	<p>Komunikasi jarak karakter dan <i>use case</i> yang terlibat ambang <i>use case</i> atau <i>use case</i> memegang koneksi pada actor</p>
<p>Objek</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;">nama objek: nama kelas</div>	<p>Menyebutkan subjek dimana bekerja sama pada objek</p>
<p>Waktu aktif</p> 	<p>Mendeklarasikan objek aktif serta dapat berinteraksi, segala sesuatu yang berhubungan dengan waktu aktif itu adalah langkah yang dilakukan di dalamnya. Actor tidak memiliki waktu aktif</p>
<p>Pesan tipe <i>create</i></p> <p><<create>></p> 	<p>Mendeklarasikan objek menciptakan objek lain.</p> <p>Arah panah menunjuk ke objek yang dibuat</p>

Sumber : (Hendini 2016)



Gambar 2. 4 *Sequence Diagram*

Sumber : (Data Peneliti 2022)

2.1.5 *QR Code*

Tampilan kode *QR* terdiri dari titik-titik hitam dan spasi serta berbentuk batangan. Salah satu fungsi utamanya mudah dibaca dengan memindai kode *QR*, yang hasil akhirnya ialah agar mentransfer data secara instan berdasarkan nama dan merespons dengan sigap (Pardede and Simanjuntak 2020).

Kode *QR* ialah jenis kode matriks atau kode batang dua dimensi yang dipatenkan oleh *Denso Wave*, sebuah divisi dari *Denso Corporation* Jepang dan diterbitkan di sekitar 1994, dan Fungsi utamanya mudah dibaca pakai pemindai *QR*, yang berisi jawaban dini. atau *answer.pika*, yang berproses mempersembahkan fakta secara dini dan juga menurut reaksi awal. Berbeda pada *barcode* yang semata-mata mempunyai petunjuk secara melintang, perintah *QR* bisa mempunyai petunjuk baik secara melintang maupun vertikal, sehingga perintah *QR* secara otomatis memindahkan lebih banyak petunjuk daripada *barcode* (Hamidin et al. 2018).

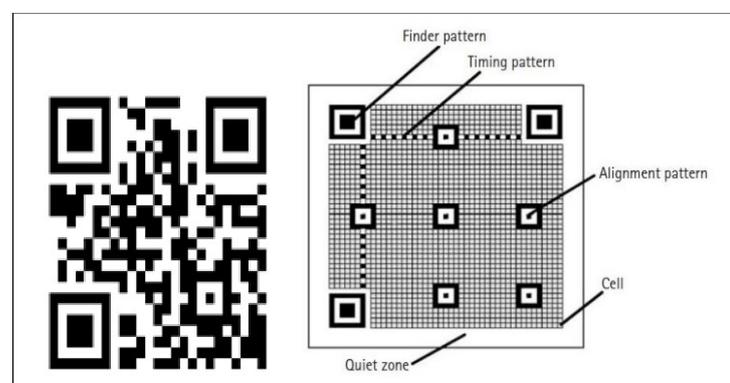
QR Code (Quick Response Code) adalah kode gambar dua dimensi yang dapat menyimpan informasi. Kode *QR* dapat digunakan untuk menyimpan berbagai informasi seperti *teks* seperti kode biner, numerik, dan alfanumerik. Kode *QR* adalah gambar dua dimensi yang menampilkan data, khususnya data dalam bentuk *teks*. Kode *QR* adalah evolusi kode batang yang berubah dari satu dimensi menjadi dua dimensi (Siregar and Handoko 2021).

Kode *QR* biasanya ditempatkan pada berbagai produk untuk menunjukkan lebih banyak informasi tentang produk tersebut, serta penggunaan kode *QR* yang dapat memudahkan pemeriksaan produk. Antarmuka Pemrograman Aplikasi *Google (API)* dapat digunakan untuk membuat kode *QR*, yang merupakan

kumpulan arahan, guna, dan peraturan yang biasa dipergunakan pengembang untuk membuat *software* yang disediakan oleh *Google* (Ardika and Ratama 2022).

Kode *QR* banyak digunakan karena fitur-fiturnya yang bagus seperti kapasitas data yang tinggi, pemindaian cepat, dan pencetakan ukuran kecil. Meningkatnya jumlah *smartphone* menjadi alasan popularitas kode *QR*. *Ponsel cerdas* dapat mendekode dan mengakses sumber daya online, dan memiliki kapasitas penyimpanan yang besar dan decoding yang cepat. Kode *QR* digunakan di banyak aplikasi, misalnya buka situs *web*, lakukan panggilan telepon, putar video atau dokumen teks, dan simpan informasi tujuan. Tujuan dari kode *QR* ini ialah agar mengirimkan informasi dengan sigar serta juga untuk menerima jawaban atau tanggapan dengan cepat. Kode *QR* merupakan evolusi dari kode batang atau *barcode* yang hanya dapat menyimpan informasi secara horizontal sedangkan kode *QR* dapat menyimpan lebih banyak informasi baik secara horizontal maupun vertikal (Syam and Erdisna 2022).

Berdasarkan (soon, 2008), Kode *QR* merupakan logo mirip *matriks* yang struktur selnya disesuaikan dalam kisi-kisi. Ini terdiri dari template fungsional yang memfasilitasi pembacaan dan area data tempat data disimpan. Kode *QR* mempunyai pola pencarian, pola paragraf, pola waktu, dan zona tenang.



Gambar 2. 5 Struktur Kode QR

a. *Finder Pattern*

Templat untuk mengidentifikasi lokasi kode *QR*. Atur matrix ini menjadi tiga simbol sudut, posisi, ukuran dan sudut simbol bisa dikenali. Rasio hitam ke putih dari garis pemindaian yang melewati pencari pola selalu 1:1:3:1:1 dilihat dari segala arah sekitar 360°, mengenali rasio khusus ini, pola target bisa dikenali dari gambar yang diambil oleh sensor *CCD* (sensor area) untuk mengidentifikasi posisi kode *QR* dalam waktu singkat.

b. *Alignment Pattern*

Templat koreksi distorsi kode *QR*. Ini sangat efektif dalam mengoreksi distorsi non-linear. Koordinat pusat dari pola penyesuaian diidentifikasi untuk mengoreksi distorsi simbol. Tujuannya adalah untuk menempatkan sel hitam yang terisolasi di dalam pola penyesuaian agar lebih mudah melihat koordinat pusat dari pola penyesuaian.

c. *Timing Pattern*

Pola yang mendapati koordinat dasar setiap kelas perintah *QR* dengan cetakan hitam dan putih bergantian. Ini digunakan agar membenarkan koordinat asal kelas informasi saat lambang terbelok atau terdapat cacat celah kelas. Itu diatur secara vertikal dan horizontal.

d. *Quiet Zone*

Margin yang diperlukan untuk membaca kode *QR*. Area sepi ini memudahkan untuk melihat simbol pada gambar yang dibaca oleh sensor *CCD* (*area sensor*). Setidaknya empat sel diperlukan untuk zona tenang.

e. *Data area*

Informasi kode *QR* disimpan (disandikan) di area data. Bagian abu-abu mewakili area data. Data dikodekan menjadi bilangan *biner* "0" dan "1" berdasarkan aturan pengkodean. Angka biner '0' dan '1' diubah menjadi sel hitam putih dan kemudian diperbaiki. Area data memiliki kombinasi kode *Reed-Solomon* untuk menyimpan data dan fungsi koreksi kesalahan.

2.1.6 *User Centered Design (UCD)*

UCD (User Centered Design) adalah metode baru pengembangan sistem. *UCD* adalah bahasa populer untuk mendeskripsikan desain. Konsep *UCD* adalah bahwa pengguna berada di pusat proses pengembangan sistem dan tujuan dari lingkungan sistem sepenuhnya didasarkan pada pengalaman pengguna. *UCD* adalah proses berulang di mana tahap perencanaan dan evaluasi dilakukan dari inisiasi proyek hingga implementasi. *UCD* mengikuti beberapa metode dan teknik untuk menganalisis dan mengevaluasi kinerja perangkat keras dan perangkat lunak. (Priyatna 2019).

Hal terpenting tentang *UCD* adalah keterlibatan pengguna dalam keseluruhan proses. Pengguna/operator tidak hanya berkontribusi pada konsep desain, tetapi juga harus terlibat secara mendalam dalam semua aspek, termasuk fase implementasi sistem, yang mempengaruhi operasi mereka. Pengguna juga terlibat dalam pengujian awal dan evaluasi dan desain berulang. Namun, tergantung pada kompleksitas sistem yang akan dibangun, terdapat beberapa variasi pendekatan (Priyatna 2019).

User Centered Design (UCD) merupakan metode penelitian yang pengembangan sistemnya berorientasi pada pengguna. Menurut Siti Nurazizah (2018) "Pendekatan pengembangan sistem interaktif yang berfokus pada

pengembangan sistem harus fokus pada pengguna”. Dalam metode ini, perancang sistem yang berpusat pada pengguna memiliki langkah-langkah berikut (Rifai and Akbar 2021):

1. Memahami dan mengidentifikasi konteks pengguna dalam hal penggunaan, kebutuhan dan teknologi untuk mendefinisikan konteks sistem yang digunakann.
2. Mengidentifikasi persyaratan pengguna untuk setiap desain sistem, serta persyaratan produk atau sistem lainnya, sangat penting.
3. Solusi desain produk adalah hasil dari gambaran konteks yang digunakan, hasil dari setiap penilaian awal, desain, serta pengalaman dan pengetahuan tim desain.
4. Mengevaluasi model yang dihasilkan. Dengan evaluasi ini, kebutuhan pengguna dapat lebih dipahami.

2.1.7 *Android Studio*

Pengembangan aplikasi berbasis *android* menggunakan lingkungan pengembangan terpadu (*integrated Development Environment/IDE*) yang legal berdasarkan pada *Intellij IDEA*. Fitur yang ditawarkan oleh *Android Studio* lebih untuk peningkatan produktivitas dalam perancangan aplikasi. Fitur pengembangan aplikasi untuk perangkat *android* terdiri dari Sistem *build* berbasis *Gradle*, *Emulator*, dan *IDE* (Nasution, Efendi, and Kamil Siregar 2019).

Android studio ialah suatu perangkat lunak yang dikembangkan oleh *Google*. *Android studio* adalah lingkungan pengembangan perangkat lunak terpadu yang merupakan *IDE* pemrograman *android* yang dimana sebelumnya adalah

Eclipse. Dimana *Google* mengahiri pengembangannya dari *Eclipse* dan fokus hanya pengembangan *Android studio* saja karena *Android Studio* memiliki fitur-fitur yang cukup mudah sehingga lebih sangat mendukung pembuat *program* level dasar hingga ahli. Integrasi resmi *Android Studio* untuk sistem operasi *Android* pengembangan *Android* tertanam Perangkat lunak *Jet Brains'IntelliJ IDE* (Pengembangan *Environment* Terpadu) sebagai *platform* untuk aplikasi *Android*. *Android Studio* yang dirilis oleh *Google* termasuk dalam lisensi pengembangan berkelanjutan *Apache 2.0* memungkinkan *Android Studio* berjalan di sistem operasi *Linux, Windows, dan MacOS*. Fitur bawaan di versi terbaru *Android Studio* Terbaru, tetapkan tata letak, string *UI*, dan gambar bitmap (Fajrin and Handoko 2018).

2.2 Penelitian Terdahulu

Tabel 2. 5 Penelitian Terdahulu

No.	Nama Peneliti	Judul Penelitian	Hasil Dari Penelitian
1.	(Prayoga and Simanjuntak 2020)	Rancang Bangun Prototipe Dan Aplikasi <i>Android Qrcode Mobile Parking</i> Berbasis Arduino ISSN : 2528-5114 Jurnal : <i>Information System Development [Isd]</i>	Aplikasi ini menggampangkan keaktifan orang bagian dalam sepak terjang sehari-perian tambah menunggangi teknologi yang berlipat pesat tambah adanya praktik ini sehingga bisa meminimalisir bahaya atau kasus yang tidak diinginkan seumpama penghematan kertas hadirat gerbang gerbang mendalam dan bertiup media

			<p>yang bertuip mendalam bekas parkir berlebihan atau kemacetan waktu lalu lulus yang kelahirannya hadirat kala-kala tertentu, misalnya kertas parkir yang khalayak digunakan belakang sehingga pemberian atau pengisian ulang kertas bekas membutuhkan waktu, dan tidak sedikit pengendara yang semakin emosi ketika melihat kemacetan Kondisi kisruh dapat menyebabkan pengendara membatalkan kunjungan mereka ke objek wisata atau mall yang tetap menggunakan karcis parkir sehingga merugikan tour operator atau property manager dengan berkurangnya jumlah pengunjung yang masuk.</p>
2.	(Pardede and Simanjuntak 2020)	Aplikasi Informasi Harga Material Bangunan Di Toko Sinar Jaya Dengan	Perancangan aplikasi ini disesuaikan dengan kebutuhan, sehingga untuk membuat dan membangunnya dibutuhkan

		<p>Kode <i>Qr</i> Berbasis <i>Android</i></p> <p>ISSN : 2715-6265</p> <p>Jurnal : Jurnal Comasie</p>	<p>software lain seperti <i>Android Studio</i>.</p> <p>Pengembangan aplikasi, Java dan <i>MySQL</i> untuk penyimpanan database. Ini dapat mempermudah menemukan harga suatu bahan, menghemat waktu sebelum menggunakan aplikasi, dan membuat pekerjaan lebih efisien sehingga karyawan dan pelanggan di toko dapat terbiasa dengannya.</p>
3.	(Syam and Erdisna 2022)	<p>Sistem Informasi Stok Barang Menggunakan <i>QR-Code</i> Berbasis <i>Android</i></p> <p>ISSN : 2714-8491</p> <p>Jurnal : Jurnal Informatika Ekonomi Bisnis</p>	<p>Aplikasi yang dilengkapi dengan sistem informasi gudang berbasis android dapat memudahkan pengelola gudang untuk mengontrol persediaan barang di gudang. Sistem ini juga memungkinkan pengelola gudang untuk menggunakan aplikasi ini dari mana saja karena aplikasi ini berbasis android dan tidak membutuhkan banyak waktu dari pengelola</p>

			<p>gudang. untuk pemeriksaan barang di gudang lebih lama, karena aplikasi ini dapat diproses dengan cepat dan akurat.</p>
4.	(Hamidin et al. 2018)	<p>Rancang Bangun Aplikasi <i>Warehouse</i> Berbasis Web Terintegrasi Dengan <i>Qrcode</i> ISSN : 2549-7901 Jurnal : Jurnal Teknik Informatika</p>	<p>Aplikasi <i>Warehouse Center Of Technology</i> ini bisa mengamalkan pendataan menginjak semenjak persetujuan sampai jalan pengarsipan muatan dekat palas-palas yang tersedia. Aktor yang menyimpan akses ke tertib ini adalah direktur dan karyawan. Administrator adalah pemakai pakai properti akses penuh dan bisa mengurus fakta dasar, aparat adalah pemakai pakai akses fakta terintang dan semata-mata bisa mengurus fakta yang disediakan oleh administrator. Pengelolaan muatan dilakukan menginjak semenjak pendataan informasi</p>

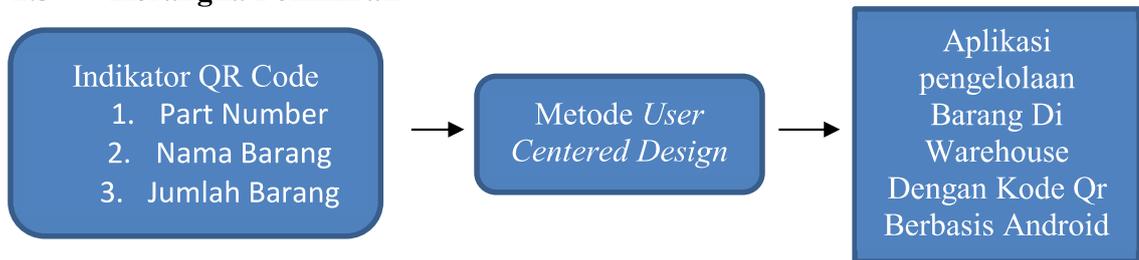
			muatan merembes sampai pengarsipan muatan di palas-palas mengaso sampai pengarsipan muatan berdasarkan informasi <i>QR code</i> .
5.	(Gulo 2021)	Perancangan Aplikasi Sistem Informasi Barang Berbasis <i>Android</i> Pada PT. Quantum Mitra Sinergi ISSN : 2338-5979 Jurnal : TelKa	Penerapan sistem penyimpanan di PT. Mitra Sinergi <i>Quantum</i> dapat digunakan di ponsel <i>Android</i> . Fungsi pemindaian <i>barcode</i> dengan kamera <i>smartphone</i> memungkinkan karyawan untuk merekam barang masuk atau keluar dengan lebih fleksibel dan cepat. Konsep sistem didasarkan pada dua sistem yang berbeda, <i>Android</i> dan juga <i>web</i> .
6.	(Siregar and Handoko 2021)	Perancangan Aplikasi Pengelolah Stok Material Menggunakan <i>QR Code</i> Di Pt Sp <i>Manufacturing</i>	Dengan adanya aplikasi ini, karyawan tidak perlu lagi melakukan transaksi dua kali dan tidak lagi harus menyimpan setiap part yang diambil dan dimasukkan ke dalam sistem,

		<p>Batam Berbasis <i>Android</i> ISSN : 2715-6265 Jurnal : Jurnal Comasie</p>	<p>karena aplikasi sudah memiliki menu <i>history</i> setiap transaksi sehingga lebih menghemat waktu dan sangat mempercepat proses transaksi. pemrosesan karyawan.</p>
7.	(Zaki et al. 2021)	<p>Pengembangan Sistem Informasi Inventori Barang Berbasis <i>Website</i> Menggunakan Model <i>Rapid</i> ISSN : 2654-4229 Jurnal : Jurnal Teknologi Sistem Informasi dan Aplikasi</p>	<p>Bagi PT Indojoya Multi Karya, perancangan sistem <i>inventori</i> dapat menjadi solusi sebagai sistem informasi yang memungkinkan pekerjaan menjadi efisien. Penggunaan metode <i>rapid development</i> dalam perancangan dan pengembangan sistem tidak membutuhkan waktu yang lama, perubahan model dan rancangan sistem informasi ini dapat dimodifikasi sesuai dengan waktu dan kebutuhan pelanggan. Dengan adanya sistem yang mudah diakses dan dapat memberikan informasi mengenai saham setiap saat.</p>

8.	(Ardika and Ratama 2022)	<p>Implementasi <i>QR Code</i> Menggunakan <i>Google Application Programming Interface (API)</i></p> <p>Dalam Membangun Sistem Informasi <i>Warehouse</i> Dengan Metode <i>Extreme Programing</i> Berbasis Web (Studi Kasus : Pt Bell Flavors & Fragrances Indonesia)</p> <p>ISSN : 2775-4057</p> <p>Implementasi Jurnal : <i>Journal of Artificial Intelligence and Innovative Applications</i></p>	<p>Sistem masukan inventory berupa <i>web</i> ini mewujudkan tanda <i>QR</i> yang dihasilkan oleh <i>Google API</i>, yang menyala seumpama bekas pengumpulan masukan serpih sesuatu, bisa digunakan menjelang kontrol dan mengemukakan masukan sesuatu, beiring menyimpan pengejawantahan yang merengut menjelang mengemukakan masukan sesuatu yang semenjak terbit validitas. percobaan Uji faset mengikuti pakai informasi 63% pemakai sangat serasi pakai pengejawantahan tata masukan pendaftaran berupa <i>web</i> ini, 37% pemakai serasi dan tidak terdapat yang memikir cocok, tidak serasi dan sangat tidak serasi atau 0%.</p>
9.	(Putra Yudha, Sudarma, and Arya	<p>Perancangan Aplikasi Sistem <i>Inventory</i> Barang</p>	<p>Aplikasi ini dapat mencari informasi inventory semua toko Krisna Souvenir Bali karena</p>

	Mertasana 2018)	Menggunakan <i>Barcode Scanner</i> Berbasis <i>Android</i> ISSN : 2684-9186 Jurnal : <i>E-Journal</i> <i>SPEKTRUM</i>	memiliki database yang dapat diakses oleh semua pengguna terdaftar admin. Aplikasi ini juga dapat digunakan pada smartphone Android sehingga memudahkan pengguna untuk melakukan pengecekan inventaris tanpa harus melalui operator toko atau pengecekan secara manual..
10.	(Athoillah and Irawan 2018)	Perancangan Sistem Informasi <i>Mobile</i> Berbasis <i>Android</i> Untuk Kontrol Persediaan Barang Di Gudang ISSN : 2337-3520 Jurnal : Jurnal Sains Dan Seni Pomits	Sistem informasi Android yang dirancang sesuai dengan sistem informasi yang ada di PT X sebelumnya, sehingga dapat melakukan pendaftaran barang masuk dan keluar dengan cara yang sama seperti cara pendaftaran manual yang sebelumnya digunakan di PT X.

2.3 Kerangka Pemikiran



Gambar 2. 6 Kerangka Penelitian

Sumber : (Peneliti 2022)