BAB II
TEORI PENUNJANG

2.1 Halal

Halal adalah membebaskan, melepaskan, memecahkan, membubarkan, dan memperbolehkan. Segala sesuatu yang menyebabkan seseorang tidak dihukum jika menggunakannya [1]. istilah ini biasa dalam kosakata sehari-hari lebih sering digunakan untuk merujuk kepada makanan dan minuman yang diizinkan untuk dikonsumsi menurut dalam Islam. Sedangkan dalam konteks yang lebih luas istilah halal merujuk kepada segala sesuatu yang diizinkan menurut hukum islam (aktivitas, tingkah laku, cara berpakaian, dan lain-lain). Halal dalam cara memperbolehkannya yaitu dengan cara baik dan sah. Produk akan dikatakan halal jika telah melakukan sertifikasi halal pada MUI. Dan harus diperpanjang setiap 2 tahun sekali kurang lebih 3 bulan sebelum masa berlaku habis.

Gambar 2.1 Logo yang sudah sertifikasi halal

2.2 *QR-Code*

QR-Code atau kepanjangan dari *Quick Response Code*, adalah bentuk evolusi kode batang dari suatu dimensi menjadi dua dimensi yang digunakan untuk menyimpan informasi data dan dirancang untuk dapat dibaca oleh smartphone. [4] *QR-Code* dipublikasikan pada tahun 1994 oleh *Denso Wave. Denso Wave* adalah sebuah divisi perusahaan Denso Corporation Jepang. *QR-Code* dapat menyimpan data baik dalam posisi vertikal maupun horizontal [2], [3], [5].

Gambar 2.2 *QR-Code*

QR-Code terdiri dari modul hitam (Titik persegi) diatur dalam kotak persegi pada latar belakang putih yang dapat dibaca oleh perangkat pencitraan, seperti kamera dan diolah menggunakan koreksi kesalahan hingga gambar dapat dengan tepat diinterpretasikan. Data tersebut kemudian diekstraksi dari pola yang ada dari kedua komponen horizontal dan vertikal pada gambar di atas. Sistem kerja pada *QR-Code* yaitu dari teks biasa, *URL* atau data lain diberikan ke *encoder QR-Code* maka akan menghasilkan *QR-Code*, saat kita akan mengakses data *QR-Code* maka akan di dekoderkan melalui *Decoder* Kode QR (*Scanner*) yang mengambil data *QR-Code* [2], [3], [4], [6].

2.2.1 Karakteristik *QR-Code*

Karakteristik dari *QR-Code* sendiri yaitu mampu menampung jumlah data yang besar. Sebanyak 7089 karakter numerik maksimum data dapat tersimpan didalamnya, kerapatan tinggi (100 kali lebih tinggi dari kode simbol linier) dan pembacaan kode dengan cepat. *QR-Code* sendiri memiliki kelebihan lain dalam hal untuk kerja dan fungsi seperti pada tabel dibawah [7].

Tabel 2.1 Karakteristik *QR-Code*

Encodable character set	Numeric (0-9)
	Alpanumeric data (Digits 0-9; upper case letters A-Z; nine other character: space,$ % * + - . / :)
	8-bit byte data
	Kanji character
Color Module	A dark module is a binary 1
	A light module is a binary 0
Versions	Version 1 until 40
	L -7% or less errors can be corrected

Error Level Correction	M 15% or less errors can be corrected Q 25% or less errors can be corrected H 30% or less errors can be corrected
Type of QR Code	Micro with one orientation detecting. iQR with rectangular code, turned-over code, black and white inversion code or dot pattren code (direct part marking). SQRC with limited specific type of scanners. LogoQ with combine designability and readability

2.2.2 Kapasitas Penyimpanan

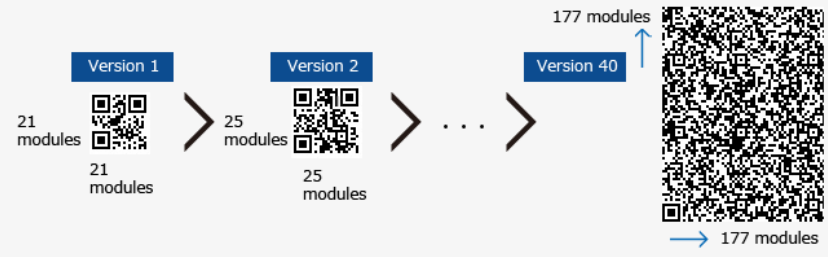
Kapasitas jumlah data yang dapat disimpan pada *QR-Code* tergantung dari tipe data (mode, atau kumpulan karakter masukan) terdapat 3 versi pada *QR-Code* (1,20, dan 40) setiap versi mengidentifikasi keseluruhan dimensi simbol dan tingkat kode koreksi kesalahan. Berikut perbedaan tingkat kapasistas dan versi pada *QR-Code* [7].

Tabel 2.2 Kapasitas Penyimpanan Versi *QR-Code*

Versi	Tingkat Koreksi Kesalahan	Mode Numerik	Mode Alfanumerik	Mode Byte	Mode Kanji
1	L	41	25	17	10
	M	34	20	14	8
	Q	27	16	11	7
	H	17	10	7	4
20	L	2061	1249	858	528
	M	1600	970	666	410
	Q	1159	702	482	297
	H	919	557	382	235
40	L	7089	4296	2953	1817
	M	5596	3391	2331	1435
	Q	3993	2420	1663	1024
	H	3057	1852	1273	784

Setiap versi pada *QR-Code* 1-40 memiliki perbedaan pada modul-modulnya. Konfigurasi modul pada *QR-Code* pada versi 1 (21 x 21 modul), versi 2 (25 x 25 modul) hingga versi 40 (177 x 177 modul) [4]. Setiap modul dari *QR-*

Code adalah muncul sebagai pola acak hitam dan putih yang terkandung pada kota persegi. Pada *QR-Code* menggunakan pola hitam untuk bilangan biner "1", dan menggunakan pola putih untuk bilangan biner "0". Oleh karena itu setiap versi simbol pada *QR-Code* akan memliki kapasitas data maksimal sesuai dengan jumlah data, tipe karakter, dan level koreksi kesalahan, maka seiring dengan bertambahnya jumlah data, akan lebih banyak modul yang diperlukan untuk membentuk *QR-Code* dan akan menghasilkan simbol *QR-Code* yang lebih besar [2], [4].



Gambar 2.3 Versi simbol pada *QR-Code*

2.2.3 Encoder

Encoder QR-Code merupakan proses mengkodekan data pada *QR-Code*. Dimana informasi format mencatat dua hal : tingkat koreksi kesalahan dan *mask pattern* yang digunakan untuk simbol. *Masking* digunakan untuk memecah pola diarea data yang mungkin membingunkan *scanner*, seperti daerah kosong besar atau fitur menyesatkan yang terlihat seperti tanda locator. *Mask pattern* didefinisikan pada *grid* yang diulang gelap sediperlukanya untuk menutupi seluruh simbol. Modul yang sesuai dengan daerah gelap terbalik. Informasi format dilindungi dari kesalahan dengan kode BCH, dan dua salinan lengkap termasuk dalam setiap simbol QR [8].

Sebelum melakukan proses *Encoder* dibutuhkan pemasangan library atau modul. penelitian ini menggunakan library Zlib dan GD agar proses *Encoder QR-Code* dapat dieksekusi secara sekaligus pada tabel, penggunaan Pyson libray pada penelitian ini tidak menggunakan library tersebut karena tidak dapat melakukan proses *Encoder QR-Code* secara serentak.

Terdapat tujuh buah tahapan dalam proses *Encoder QR-Code* [9].

a. Menganalisis dengan memilih model *Encoder*, dapat dilihat pada Tabel 2.1

b. Data *Encoder* memiliki tingkat validasi. Pada tahap ini dilakukan beberapa diantaranya :

1. Pemilihian error correction level dapat dilihat pada Tabel 2.6
2. Menentukan versi pada *QR-Code*, dapat dilihat pada Tabel 2.2
3. Menambahkan mode indikator

Tabel 2.3 Indikator Mode *Encoder*

Mode	Indicator
Numeric	0001
Alpanumeric	0010
Byte	0100
Kanji	1000

4. Menambahkan jumlah karakter pada setiap versi

Tabel 2.4 Panjang Indikator Karakter pada Setiap Versi

Mode/Version	1-9	10-26	27-40
Numeric	10 Bits	12 Bits	14 Bits
Alpanumeric	9 Bits	11 Bits	13 Bits
Byte	8 Bits	16 Bits	16 Bits
Kanji	8 Bits	10 Bits	12 Bits

5. Encode dengan menggunakan mode yang dipilih: untuk mengambil nilai modus alpanumeric maka dapat digunakan pada Tabel 2.

Tabel 2.5 Nilai Modus Alpanumeric

0	0	1	1	2	2	3	3	4	4	5	5	6	6
7	7	8	8	9	9	A	10	B	11	C	12	D	13
E	14	F	15	G	16	H	17	I	18	J	19	K	20
L	21	M	22	N	23	O	24	P	25	Q	26	R	27
S	28	T	29	U	30	V	31	W	32	X	33	Y	34
Z	35		36	$	37	%	38	*	39	+	40	-	41
.	42	/	43	:	44								

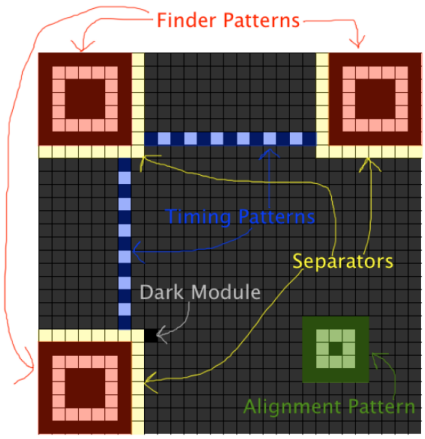
Sebagai contoh untuk memecah kata “HELLO” menjadi HE, LL, O kemudian masing-masing pasangan akan dibuat kedalam bilangan biner. H → 17, E → 14, L → 21, L → 21, O → 24 proses ini menghasilkan data encode 01100001011 (11 bit)

6. Codeword dipecah menjadi 8 bit

c. Koreksi kesalahan coding pada setiap blok codeword data.

d. Struktur akhir pesan dengan mengambil codeword data pada setiap blok sampai semua codeword data memiliki satu blok, maka koreksi kesalahan codeword ditempatkan setelah codeword data.

e. Penempatan modul pada matriks: *QR-Code* harus mencakup pola sebagai fungsi utama.



Gambar 2.4 Pola *QR-Code*

2. Data Masking. Bagian ini akan memodifikasi *QR-Code* sehingga mudah dibaca oleh perangkat pemindai *QR-Code*.
3. Menambahkan format dan versi yang akan digunakan untuk menampung informasi.

2.2.4 Decoder

Decoder adalah proses pembacaan *QR-Code* untuk menghasilkan informasi dari data yang ada pada *QR-Code*. *Decoder* merupakan kebalikan dari proses *Encoder* yang merupakan proses untuk mengubah data ke dalam beentuk *QR-Code* [8].

2.2.5 Struktur *QR-Code*

QR-Code sebenarnya adalah modul hitam dalam pola persegi dengan latar belakang putih. *QR-Code* terdiri dari banyak area yang memiliki pola-pola khusus. [10] Pola-pola ini terdiri dari fungsi untuk membaca mudah dan area data dimana data disimpan. *QR-Code* memiliki delapan bagian signifikan dari arsitektur *QR-Code* [7].

Gambar 2.5 Struktur *QR-Code*

1. *Finder patter,* perangkat lunak decoder mampu mengenali *QR-Code* dan memastikan orientasi yang benar dalam segala arah
2. *Separators*, sebagai pemisah antara pola *finder* dan kode data.
3. *Timing pattern,* untuk memastikan perangkat lunak *decoder* kompensasi gambar.
4. *Aligement patern,* mengaktifkan perangkat lunak *decoder* kompensasi gambar.
5. *Format Information,* untuk menjaga koreksi kesalahan tingkat *QR-Code* dan pola masking yang dipilih.
6. *Data*, data 8 bit codeword.
7. *Error correction,* 8 bit codeword koreksi kesalahan.
8. *Remainder bits,* bit kosong jika data dan bit koreksi kesalahan tidak dapat dibagi menjadi 8 bit codeword tanpa sisa.

2.2.6 Koreksi Kesalahan

Beberapa faktor yang mempengaruhi pada kapasitas *QR-Code*. Seperti tingkat koreksi kesalahan yang dipilih, versi dari kode yang menggunakan ukuran (jumlah modul) dan jenis kapasitas dampak data yang disandikan terlihat empat tingkat

koreksi kesalahan. Semakin tinggi tingkat koreksi kesalahan, kapasitas penyimpanan akan semakin berkurang.

Tabel 2.5 Tingkat Koreksi Kesalahan

Tingkat Kesalahan	Kapasitas Koreksi Kesalahan
L	7%
M	15%
Q	25%
H	30%

Ada empat level koreksi kesalahan, Rendah (L) yang dapat mentolerir kerusakan hingga 7%, Sedang (M) dapat mentolerir hingga 15%, Kuartil (Q) dapat mentolerir kerusakan hingga 25%, dan Tinggi (H) dapat mentolerir kerusakan hingga 30%. Ketika tingkat koreksi yang lebih tinggi digunakan untuk menghasilkan peningkatan persentase codeword, karena itu jumlah data yang dapat disimpan dalam kode menurun [11].

2.3 Android

Android merupakan sebuah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi-aplikasi mereka sendiri untuk digunakan oleh bermacam-macam piranti bergerak [12] [13].

Android akan terus berusaha memperbaharui sistem operasinya agar terus memuaskan kebutuhan pasar global. Kemajuan teknologi saat ini tentunya tidak terlepas dari perkembangan teknologi yang semakin hari semakin terbaharui. Hal tersebut terlihat dari adanya versi yang terus diluncurkan oleh *Android*. Berbagai fitur yang ditawarkan *Android* telah menjadikannya raja *platform* ponsel pintar sampai saat ini [14]. Sistem operasi android atau OS android terdiri dari beberapa versi sebagai berikut :

Tabel 2.6 Versi Android

No	Versi Android	Tahun *Release*
1	Android 1.0 Apple Pie	28 September 2008
2	Android 1.1 Banana Bread	9 Februari 2009
3	Android 1.5. Cupcake	27 April 2009
4	Android 1.6 Donut	15 September 2009
5	Android 2.0 Eclair	26 Oktober 2009
6	Android 2.2 Froyo	20 Mei 2010
7	Android 2.3 Gingerbread	6 Desember 2010
8	Android 3.0 Honeycomb	22 Februari 2011
9	Android 4.0 Ice Cream Sandwich	19 Oktober 2011
10	Android 4.1 Jelly Bean	27 Juni 2012
11	Android 4.4 Kitkat	31 Oktober 2013
12	Android 5.0 Lollipop	12 November 2014
13	Android 6.0 Marshmallow	12 November 2014
14	Android 7.0 Nougat	9 Maret 2016
15	Android 8.0 Oreo	21 Maret 2017
16	Android 9.0 Pie	6 Agustus 2018
17	Android 10.0 Q	3 September 2019

2.4 PHP

PHP atau kepanjangan dari *Hypertext Processor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah skripsi *HTML* [15]. *PHP* merupakan bahasa *scripting server – side,* dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya *server*lah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan [15]. *PHP* dirancang untuk dapat bekerja sama dengan database server dan dibuat sedemikian rupa sehingga pembuatan dokumen *HTML* yang dapat mengakses *database* menjadi begitu mudah [16].

2.5 MySQL

MySQL adalah *server database open source* yang termasuk populer keberadaanya [17]. *MySQL* umumnya digunakan bersamaan dengan *PHP* untuk membuat aplikasi *server* yang dinamis dan powerfull [17]. *MySQL AB* membuat *MySQL* tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License (GPL)*, tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaanya tidak cocok dengan penggunaan *GPL* [17]. *MySQL* adalah *Relational Database Management System (RDBMS)* yang didistribusikan secara gratis dibawah lisensi GPL *(General Public License)* [17].

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL *(Structured Query Language)* [18]. *SQL* adalah sebuah konsep pengoprasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoprasian data dikerjakan dengan mudah secara otomatis [18].

2.6 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menganggambarkan arsitektur dalam pemograman berorientasi objek [19]. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sistem dengan menggunakan diagram dan teks-teks pendukung [19].

UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembangan sistem membuat *blue print* atas visinya dalam bentuk baku [20]

2.6.1 Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [19]. *Class* adalah deskripsi sekumpulan *object* yang memiliki kesamaan atribut, operasi, *relationship* dan

semantics [20]. Dengan kata lain, sebuah *class* merupakan *blueprint* template cetakan dari satu atau lebih *object* [19] [20]. Diagram kelas dibuat agar pembuatan program membuat kelas-kelas sesuai rancangan didalam diagram agar antara dokumentasi perancangan dan perangkat lunak sinkron [19].

2.6.2 *Use case* Diagram

Use case Diagram merupakan pemodelan untuk kelakuan *(behavior)* sistem informasi yang akan dibuat [19]. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat [19] [20]. *Use case* adalah layanan (*services*) atau fungsi-fungsi yang disediakan oleh sistem untuk pengguna-penggunaanya [20]. Setiap *Use case* adalah suatu urut-urutan (*sequence)* transaksi yang saling berhubungan dan dilakukan oleh sebuah aktor dan sistem dalam bentuk sebuah dialog [20]. *Use case Diagram* dibuat untuk memvisualisasikan atau menggambarkan hubungan antara aktor dan *Use case* [20].

2.6.3 *Sequence diagram*

Sequence diagram menggambarkan kelakukan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek [19], [20]. Menggambarkan diagaram sekuen harus diketahui objek-objek yang terlibat dalam *use case* berserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu [19]. *Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian *even* untuk menghasilkan *output* tertentu [20].