

BAB 2

LANDASAN TEORI

2.1. Sekolah Menengah Pertama

SMP (Sekolah Menengah Pertama) merupakan pendidikan formal pada jenjang pendidikan dasar. Pendidikan dan pembelajaran di tingkat SMP memberikan penekanan peletakan pondasi dalam menyiapkan generasi agar menjadi manusia yang mampu menghadapi era yang semakin berat. Menurut Undang-Undang Sistem Pendidikan Nasional no 20 tahun 2003 pasal 17 tentang pendidikan dasar disebutkan bahwa pendidikan dasar terdiri dari SD (Sekolah Dasar)/sederajat dan SMP (Sekolah Menengah Pertama)/sederajat [3].



Gambar 2.1 Ilustrasi Murid Sekolah Menengah Pertama

2.1.1. Absensi Murid

Absensi murid adalah ketidak hadirnya seorang murid ke tempat belajar yang disebabkan beberapa alasan seperti : alpa, ijin dan sakit. Tinggi rendahnya suatu absensi didalam sekolah dapat digunakan untuk mengukur disiplin tidaknya seseorang murid dalam belajar. Jika semakin besar tingkat absen murid akan menghambat produktivitas seorang murid untuk mendapatkan ilmu dan nilai mata pelajaran.

Absen adalah suatu pendataan atau pencatatan dari kehadiran karyawan ditempat kerja, bagian dari pelaporan aktifitas suatu perusahaan yang berisi sebuah data-data kehadiran karyawan yang disusun dan diatur sedemikian rupa, sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pihak yang berkepentingan didalam perusahaan [4].

2.1.2. Pembayaran Sekolah

Menurut Tirto Waluyo pembayaran adalah suatu tindakan menukarkan sesuatu uang atau barang dengan maksud dan tujuan yang sama yang dilakukan oleh dua orang atau lebih. Sedangkan SPP (Sumbangan Pembinaan Pendidikan) merupakan iuran wajib bagi siswa atau siswi yang dipergunakan oleh pihak sekolah untuk memfasilitasi segala kegiatan pembelajaran yang dilakukan siswa/siswi, dengan waktu pembayaran ditentukan sebelumnya [5].

2.2. Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output* [6]. Seperti yang disebutkan pada kamus besar bahasa Indonesia (2005 : 52), “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu”. Berdasarkan kedua definisi tersebut didapatkan bahwa aplikasi adalah suatu program pada komputer yang berfungsi melakukan suatu kegiatan yang dijalankan

oleh penggunaanya dengan proses *input* dari pengguna yang menghasilkan suatu *output*.

2.3. Basis Data (*Database*)

Database atau basis data dapat diartikan sebagai kumpulan *data store* yang tersimpan dalam *magnetic disc*, *optical disc* atau media penyimpanan sekunder lainnya, yang terdiri dari satu atau lebih tabel yang terintegrasi satu sama lain, di mana setiap pemakai (*user*) diberi wewenang (otorisasi) untuk dapat mengakses (mengubah, menghapus, menganalisis, menambah, memperbaiki) data dalam tabel-tabel tersebut [7]. Dengan kata lain, basis data merupakan sekumpulan data yang saling terkait (berelasi) antara satu dengan yang lainnya yang menggambarkan suatu organisasi (*enterprise*). Sedangkan sistem basis data adalah suatu sistem yang mengelola data dan dapat menyediakan data tersebut ketika diperlukan.

2.3.1. Keuntungan Basis Data

Penggunaan basis data dapat memberikan beberapa keuntungan, diantaranya:

1. Mereduksi redundansi yang akibatnya mengurangi inkonsistensi. Penyimpanan data yang sama berulang-ulang di beberapa file dapat mengakibatkan inkonsistensi. Apabila salah satu dari file yang mengandung data tersebut terlewat diperbaharui maka terjadilah data yang tidak konsisten lagi.
2. Data dapat terintegrasi antar aplikasi. Data dapat dipakai secara bersama-sama oleh beberapa program aplikasi (secara *batch* maupun *on-line*).
3. Standarisasi data dapat dilakukan. Karena adanya pengontrolan yang terpusat, maka *DBA* dapat menerapkan standarisasi data yang disimpan sehingga memudahkan pemakaian, pengiriman maupun pertukaran data. Jika data tersebar dalam beberapa file dalam bentuk format yang tidak sama, maka ini menyulitkan dalam program aplikasi untuk mengambil dan

menyimpan data. Maka haruslah data dalam suatu basis data dibuat format yang standar sehingga mudah dibuat program aplikasinya.

4. Batasan keamanan dapat diterapkan. *DBA* dapat memberikan batasan-batasan pengaksesan data, misalnya dengan memberikan *password* dan pemberian hak akses bagi pengguna.
5. Mengelola integritas (keterjaminan akurasi) data. Jika kerangkapan data dikontrol dan konsistensi data dapat dijaga maka data menjadi akurat.
6. Menyeimbangkan kebutuhan yang saling konflik.
7. Independensi data (objektif DBS). Kekebalan aplikasi terhadap perubahan struktur penyimpanan dan teknik pengaksesan data. Basis data harus dapat berkembang tanpa mempengaruhi aplikasi yang telah ada. Apapun perubahan dalam basis data semua perintah akan mengalami kestabilan tanpa perlu ada yang diubah.

2.3.2. Komponen Utama Sistem Basis Data

Komponen utama sistem basis data adalah sebagai berikut:

- A. Data yang disimpan dalam basis data.
- B. Perangkat keras.

Berupa komputer dan bagian-bagian di dalamnya seperti *processor*, *memory* dan *hardisk*. Komponen inilah yang melakukan pemrosesan dan juga untuk menyimpan basis data.

- C. Perangkat lunak.

Perangkat lunak ini terdiri dari program pengelola basis data. Perangkat lunak inilah yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali. Perangkat lunak ini juga menerapkan mekanisme pengamanan data, pemakaian data secara bersamaan, pemaksaan keakuratan/konsistensi data, dsb.

- D. DBMS, Report-writer, design arts, dll.
- E. Basis data.

Sebuah DBMS dapat memiliki beberapa basis data, setiap basis data dapat berisi sejumlah objek basis data (file, table, indeks, dll). Disamping berisi data, setiap basis data juga menyimpan definisi struktur (baik untuk basis data maupun objek-objeknya secara detail).

F. Pengguna.

1. Pengguna Awam (*Naïve User*)
2. Pengguna Biasa (*Casual User*)
3. *Programmer*.

Orang yang membuat program aplikasi yang menggunakan basis data.

4. *Administrator*.

Orang yang bertanggung jawab terhadap pengelolaan basis data.

2.3.3. Bahasa dalam Sistem Basis Data

Secara umum terdapat dua jenis bahasa dalam sistem basis data, yaitu:

a. *Data Definition Language* (DDL)

Perintah-perintah yang digunakan oleh database administrator untuk mendefinisikan skema ke DBMS. Secara detail hal yang perlu dijabarkan pada DBMS adalah:

1. Nama basis data.
2. Nama seluruh berkas pada basis data.
3. Nama rekaman dan medan.
4. Deskripsi berkas, rekaman dan medan.
5. Nama medan kunci.
6. Nama indeks dan medan yang menjadi indeks.
7. Hal lain seperti ukuran basis data.

DDL juga digunakan untuk menciptakan, mengubah, dan menghapus basis data.

b. *Data Manipulation Language* (DML)

DML adalah perintah-perintah yang digunakan untuk mengubah, memanipulasi dan mengambil data pada basis data. Tindakan seperti menghapus, mengubah, dan mengambil data menjadi bagian dari DML. DML dibagi atas 2 jenis, yaitu:

1. Prosedural

Menuntut pengguna menentukan data apa saja yang diperlukan dan bagaimana cara mendapatkannya.

2. Non-prosedural

Yang menuntut pengguna menentukan data apa yang diperlukan tetapi tidak perlu menyebutkan cara mendapatkannya.

2.3.4. Operasi Basis Data

Operasi dasar basis data diantaranya adalah pembuatan, penghapusan, pembuatan tabel, pengisian tabel yang dijelaskan sebagai berikut:

- a. Pembuatan *database* baru (*create database*) yaitu menciptakan *database* baru.
- b. Penghapusan *database* (*drop database*) yaitu penghapusan suatu *database*.
- c. Pembuatan *file*/tabel baru ke dalam suatu *database* yaitu penambahan *file*/tabel ke dalam *database*.
- d. Penambahan/pengisian data baru ke sebuah *file*/tabel di sebuah *database* (*insert*).
- e. Perubahan data yang ada pada sebuah *file*/tabel di sebuah *database* (*update*).
- f. Penghapusan data yang ada pada sebuah *file*/tabel di sebuah *database* (*delete*).

2.4. Internet

Internet dengan berbagai aplikasinya seperti *web*, *VoIP*, *E-Mail* pada dasarnya merupakan media yang digunakan untuk mengefisienkan proses komunikasi [8]. *Internet* merupakan jaringan komputer yang saling terhubung secara mendunia yang memungkinkan jutaan jaringan terhubung antara satu dengan yang lainnya sehingga *user* lebih mudah terhubung dengan *user* lainnya [9].

Berdasarkan kedua pendapat diatas, dapat ditarik kesimpulan bahwa internet merupakan jaringan komputer yang saling terhubung yang berguna untuk mengefisienkan proses komunikasi.

Asal usul internet dari jaringan komputer yang disusun oleh ARPA (*Advance Research Project Agency*) yang bernaung dibawah departemen pertahanan Amerika Serikat atau DoD (*Departement of Defense*) sehingga terbentuk *ARPAnet*, jaringan komputer bentukan departemen pertahanan Amerika Serikat, tahun 70-an. Pada awalnya, *ARPAnet* hanya menghubungkan empat buah situs saja, yaitu:

- a. SRI (*Stanford Research Institute*)
- b. USCB (*University of California at Santa Barbara*)
- c. UCLA (*University of California at Los Angeles*)
- d. *University of Utah at Utah*

Jaringan tersebut selanjutnya diperbaharui dan dikembangkan, dan saat ini digunakan sebagai tulang punggung jaringan informasi yang disebut dengan internet. Penelitian di Stanford menghasilkan sebuah protokol yang disebut *TCP/IP*, inilah yang berkembang terus hingga sekarang dan menjadi protokol standar dalam *internet*.

2.5. QR Code

Kode Batang atau yang biasa disebut barcode merupakan representasi data yang dapat dibaca oleh mesin optik yang berhubungan dengan objek yang terpasang. Pada awalnya, *barcode* mewakili data dengan memvariasikan lebar dan jarak dari garis paralel secara sistematis, dan dapat disebut sebagai linear atau satu dimensi (*1D*). Kemudian *barcode* berevolusi menjadi persegi panjang, titik, segi enam dan pola geometris lainnya dalam dua dimensi (*2D*). Meskipun sistem *2D* menggunakan berbagai simbol, tapi pada umumnya tetap disebut sebagai *barcode* juga atau dikenal juga dengan sebutan kode matriks. Kelebihan dari kode matriks ini adalah dapat merepresentasikan lebih banyak data untuk setiap area unitnya. Awalnya *barcode* dipindai oleh pemindai (*scanner*) optik khusus yang disebut pembaca *barcode* (*barcode scanner*). Kemudian, *scanner* dan *software interpretative* mulai tersedia

pada perangkat termasuk printer *desktop* dan *smartphones*. *Quick Respons Code* atau biasa disebut *QR code* merupakan salah satu jenis kode matriks yang pertama kali dirancang oleh industri otomotif di Jepang. Sebuah *QR code* menggunakan empat mode standar *encoding* (yaitu numerik, alfanumerik, byte / biner dan kanji) untuk menyimpan data secara efisien. Sistem *QR code* pun telah menjadi populer di luar industri otomotif karena pembacaan yang cepat dan kapasitas penyimpanan yang lebih besar dibandingkan dengan standar *barcode* UPC.



Gambar 2.2 Contoh Gambar *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 Reed-Solomon hingga gambar dapat dengan tepat diinterpretasikan. Data tersebut kemudian diekstraksi dari pola yang ada dari kedua komponen horizontal dan komponen *vertical* pada gambar 2.8.

2.5.1. Standar

Berikut beberapa Standar pengkodean data dalam *QR Code*:

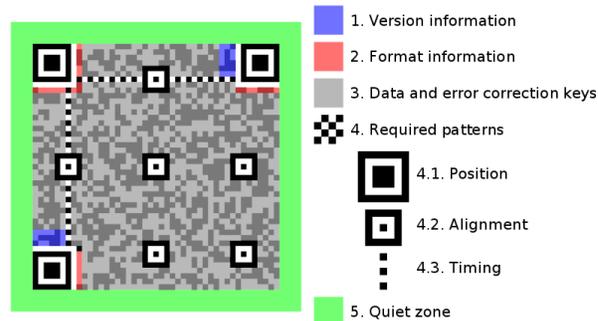
- a. October 1997 – AIM (*Association for Automatic Identification and Mobility*) International
- b. January 1999 – JIS X 0510
- c. June 2000 – ISO/IEC 18004:2000 Teknologi informasi – Identifikasi otomatis dan teknik data capture – Simbologi barcode – *QR code*

- d. 1 September 2006 – ISO/IEC 18004:2006 Teknologi informasi – Indentifikasi otomatis dan teknik data capture – *QR code* 2005 spesifikasi simbologi *barcode*. Mendefinisikan simbol *QR code* 2005, perpanjangan *QR code model 2*. Tidak menentukan cara membaca simbol *QR code model 1*
- e. 1 Februari 2015 - Informasi ISO / IEC 18004: 2015 - Teknik identifikasi dan pengambilan data otomatis - Spesifikasi simbologi kode batang QR Code.
- f. Mengganti nama simbol QR Code 2005 menjadi QR Code dan menambahkan klarifikasi pada beberapa prosedur dan perbaikan kecil.

Pada lapisan aplikasi, terdapat beberapa variasi antara sebagian besar implementasi. NTT Docomo Jepang telah membentuk standar *de facto* untuk pengkodean *URL*, informasi kontak dan beberapa jenis data lainnya. Proyek *open source* “ZXing” menyimpan daftar tipe data *QR code*.

2.5.2. Desain

Tidak seperti *barcode* satu dimensi yang dirancang secara mekanis untuk dipindai oleh sinar sempit cahaya (*narrow beam of light*), *QR code* dideteksi oleh sensor gambar digital dua dimensi dan kemudian dianalisis secara digital oleh prosesor yang terprogram. Prozessor menempatkan togak kotak khas di sudut-sudut gambar *QR code* menggunakan persegi yang lebih kecil (atau beberapa kotak) dekat sudut keempat untuk menormalkan gambar untuk ukuran, orientasi dan sudut pandang. Titik-titik kecil di seluruh *QR code* tersebut kemudian dikonversi ke bilangan biner dan divalidasi dengan kode koreksi kesalahan.



Gambar 2.3 Struktur QR Code Beserta Elemen Fungsi

2.5.3. Storage

Jumlah data yang dapat disimpan pada simbol *QR code* tergantung dari tipe data (mode, atau kumpulan karakter masukan), versi (1, ..., 40, mengindikasikan keseluruhan dimensi simbol), dan tingkat kode koreksi kesalahan. Kapasitas penyimpanan maksimum saat ini adalah untuk simbol 40-L (versi 40 dengan koreksi kesalahan tingkat low).

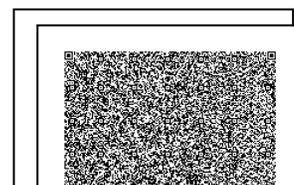
Tabel 2.1 Kapasitas Maksimum Penyimpanan Karakter (40-L)

<i>Input Mode</i>	Maksimal Karakter	Bits/Karakter	Default Encoding
<i>Numeric only</i>	7.089	$3^{1/3}$	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
<i>Alphanumeric</i>	4.296	$5^{1/2}$	0–9, A–Z (upper-case only), space, \$, %, *, +, ,, /, :
<i>Binary/byte</i>	2.953	8	ISO 8859-1
Kanji/Kana	1.817	13	Shift JIS X 0208

Setiap versi akan mempunyai ukuran yang berbeda dengan versi lainnya. Gambar 2.4 sampai Gambar 2.9 adalah contoh simbol *QR code* dengan versi yang



Gambar 2.10 QR Code Versi 25 (117x117)



Gambar 2.9 QR Code Versi 40 (177x177)

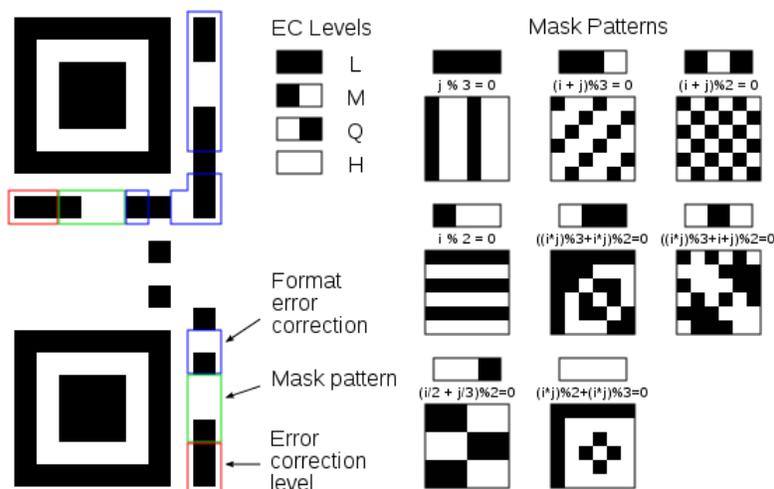
berbeda.

2.5.4. Enkripsi

Proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca dengan bantuan khusus [10]. QR code yang dienkripsi sangatlah tidak umum, hanya sedikit yang mengimplementasikannya.

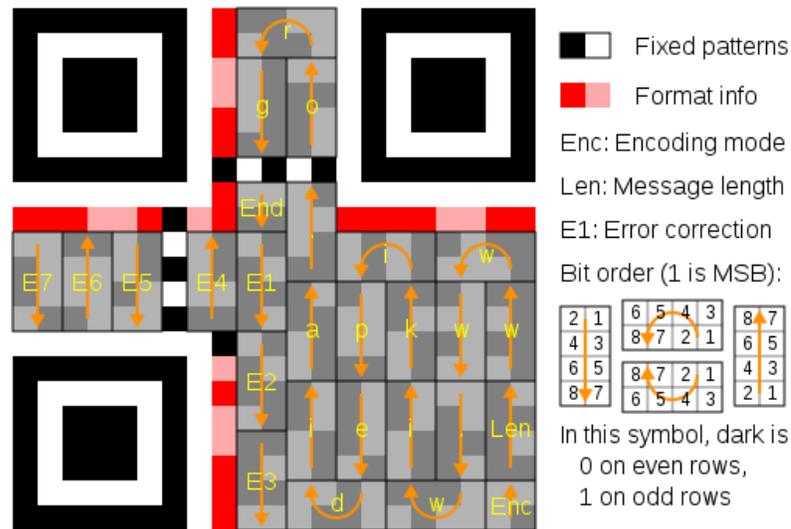
2.5.5. Encoding

Informasi format mencatat dua hal: tingkat koreksi kesalahan dan *mask pattern* yang digunakan untuk simbol. *Masking* digunakan untuk memecah pola di *area data* yang mungkin membingungkan *scanner*, seperti daerah kosong besar atau fitur menyestatkan yang terlihat seperti tanda *locator*. *Mask pattern* didefinisikan pada *grid* yang diulang sediperlukannya untuk menutupi seluruh simbol. Modul yang sesuai dengan daerah gelap *mask* terbalik. Informasi format dilindungi dari kesalahan dengan kode BCH, dan dua salinan lengkap termasuk dalam setiap simbol *QR* [11].



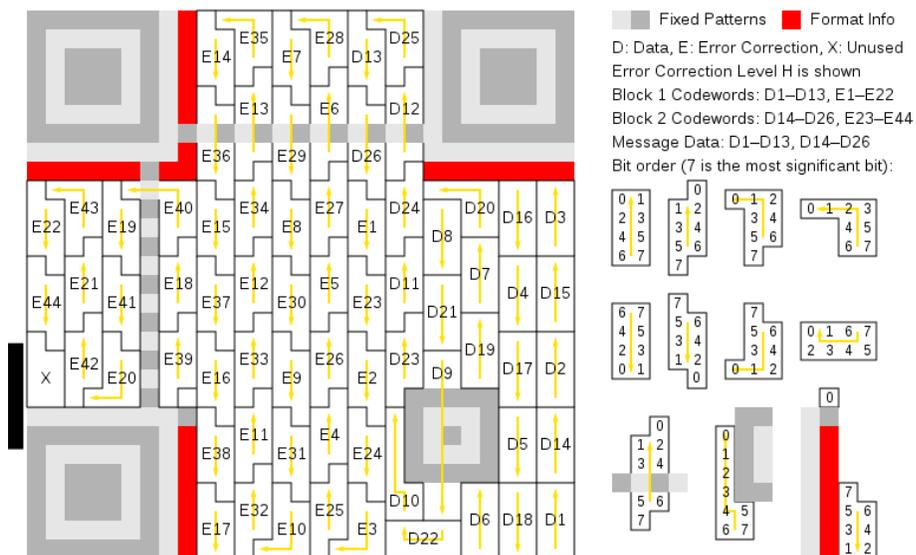
Gambar 2.11 Informasi Format [11]

Data pesan ditempatkan dari kanan ke kiri dalam pola zigzag. Dalam simbol yang lebih besar, cukup rumit karena adanya pola keselarasan dan penggunaan beberapa blok interleaved koreksi kesalahan.



Gambar 2.12 Penempatan Pesan dan Simbol QR [11]

Penempatan pesan dalam simbol *QR*. Pesan dikodekan menggunakan kode merah dengan menggunakan "*padding*" yang dapat memperbaiki kesalahan hingga 3 byte.



Gambar 2.13 Ilustrasi Blok Interleaved dengan Simbol yang Lebih Besar [11]

Indikator 4 bit digunakan untuk memilih *mode encoding* dan menyampaikan informasi lainnya. Mode pengkodean dapat dicampur sesuai kebutuhan dalam simbol *QR*.

Tabel 2.2 Menu Encoding

Indikator	Keterangan
0001	<i>Numeric encoding (10 bits per 3 digits)</i>
0010	<i>Alphanumeric encoding (11 bits per 2 characters)</i>
0100	<i>Byte encoding (8 bits per character)</i>
1000	<i>Kanji encoding (13 bits per character)</i>
0011	<i>Structured append (used to split a message across multiple QR symbols)</i>
0111	<i>Extended Channel Interpretation (select alternate character set or encoding)</i>
0101	<i>FNC1 in first position (see Code 128 for more information)</i>
1001	<i>FNC1 in second position</i>
0000	<i>End of message</i>

Setelah setiap indikator yang memilih *mode encoding*, panjang *field* lah yang memberitahu berapa banyak karakter dikodekan dalam mode tersebut. Jumlah bit pada panjang *field* tergantung pada *encoding* dan versi simbol.

Tabel 2.3 Jumlah Bit Per Panjang Field

Encoding	Ver.1–9	10–26	27–40
<i>Numerik</i>	10	12	14
<i>Alfanumerik</i>	9	11	13
<i>Byte</i>	8	16	16
<i>Kanji</i>	8	10	12

Mode encoding alfanumerik menyimpan pesan lebih ringkas daripada yang *mode byte*, tetapi *mode* alfanumerik tidak bisa menyimpan huruf-huruf non-kapital dan memiliki pilihan tanda baca terbatas yang cukup untuk alamat web yang belum sempurna. Dua karakter dikodekan dalam nilai 11-bit dengan rumus:

$$V = 45 \times C_1 + C_2$$

Tabel 2.4 Kode Karakter Alfanumerik

Kode	Karakter	Kode	Karakter
00	0	23	N
01	1	24	O
02	2	25	P
03	3	26	Q
04	4	27	R
05	5	28	S
06	6	29	T
07	7	30	U
08	8	31	V
09	9	32	W
10	A	33	X
11	B	34	Y
12	C	35	Z
13	D	36	SP
14	E	37	\$
15	F	38	%
16	G	39	*
17	H	40	+
18	I	41	-
19	J	42	.

20	K	43	/
21	L	44	:
22	M		

2.5.6. Koreksi Kesalahan

Codeword dengan panjang 8 bit dan menggunakan algoritma koreksi kesalahan Reed-Solomon dengan empat tingkat koreksi kesalahan. Semakin tinggi tingkat koreksi kesalahan, kapasitas penyimpanan akan semakin kurang. Tabel 2.5 mencantumkan perkiraan kemampuan koreksi kesalahan pada masing-masing empat tingkatan.

Tabel 2.5 Tabel Tingkat Koreksi Kesalahan

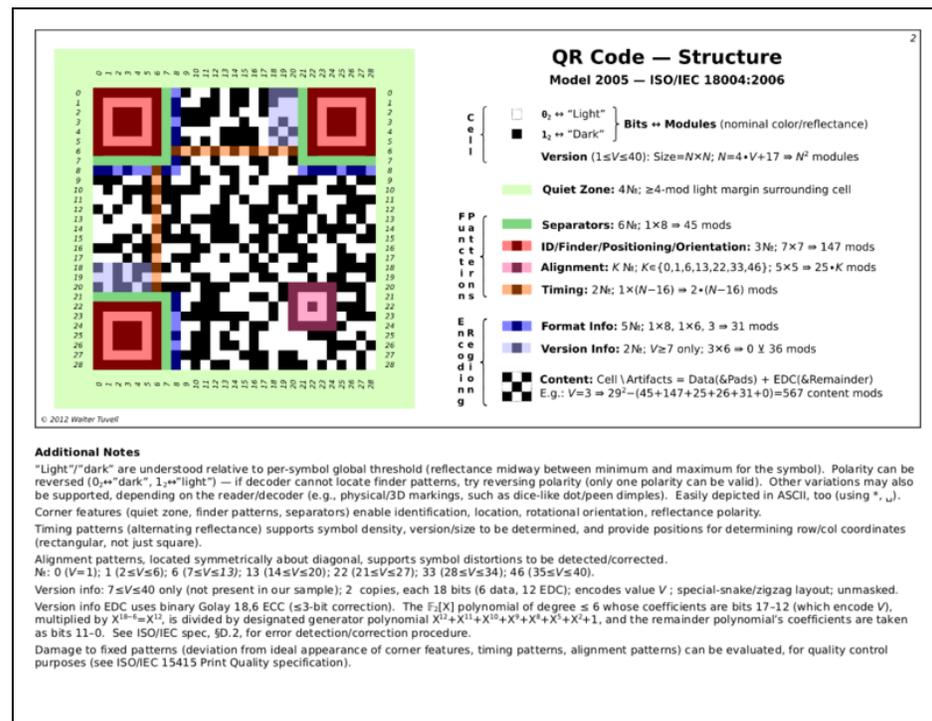
Tingkat EC	Kemampuan Koreksi Kesalahan	Indikator
Level L (<i>Low</i>)	7% dari <i>codewords</i> dapat dikembalikan	01
Level M (<i>Medium</i>)	15% dari <i>codewords</i> dapat dikembalikan	00
Level Q (<i>Quartile</i>)	25% dari <i>codewords</i> dapat dikembalikan	11
Level H (<i>High</i>)	30% dari <i>codewords</i> dapat dikembalikan	10

Dalam simbol *QR code* yang lebih besar, pesan akan dipecah menjadi beberapa blok kode Reed-Solomon. Ukuran dipilih sehingga paling banyak 15 kesalahan dapat diperbaiki di setiap blok, ini membatasi kompleksitas dari algoritma decoding. Blok kode tersebut kemudian disisipkan bersama-sama, sehingga lebih kecil kemungkinannya bahwa kerusakan lokal untuk simbol *QR* akan membanjiri kapasitas setiap blok tunggal.

Tingkat Koreksi kesalahan (*EC Level*) dan versi dari *QR code* ini dapat menentukan jumlah data yang dapat ditampung dalam setiap *QR code*. Rincian dari versi 1 sampai versi 5 dengan tingkat koreksi kesalahan L hingga H dapat dilihat pada Tabel 2.6 untuk seluruh versi *QR code* dapat dilihat pada lampiran.

2.5.7. Decoding

Decoding adalah proses pembacaan *QR code* untuk menghasilkan informasi dari data yang ada pada *QR code*. *Decoding* merupakan kebalikan dari proses *encoding* yang merupakan proses untuk mengubah data ke dalam bentuk *QR code*. Ada beberapa bagian pada *QR code* yang dapat dilihat secara manual seperti pada Gambar 2.15. [12]



Gambar 2.14 Lokasi Area *QR Code* [12]

2.6. Library ZXing (Zebra Crossing)

QR Code merupakan kode 2 dimensi yang digunakan untuk menyimpan informasi. Pembuatan dan pemindaian QR Code pada aplikasi android di dalam sistem ini dilakukan menggunakan *library ZXing (Zebra Crossing)*. ZXing merupakan sebuah *library open-source* pemrosesan gambar barcode multi-format 1D/2D yang diimplementasikan menggunakan bahasa pemrograman java.

2.7. Website

Website merupakan kumpulan dari beberapa halaman *web* yang saling terhubung dan *file-filenya* saling terkait satu dengan lainnya [14]. *Web* terdiri dari beberapa *page* atau halaman, serta kumpulan halaman-halaman tersebut dapat disebut *homepage*. *Homepage* berada pada titik teratas diikuti oleh halaman-halaman yang terkait yang berada dibawahnya. Pada umumnya setiap halaman dibawah *homepage* bisa disebut *child page* yang berisi berbagai *hyperlink* kehalaman lain pada web.

Website awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan surfer atau pengguna internet melakukan penelusuran informasi di internet. Informasi yang disajikan dengan web menggunakan konsep multimedia, informasi dapat disajikan dengan menggunakan banyak media, seperti teks, gambar, animasi, suara, video atau kombinasi dari semuanya.

Selain menyediakan informasi, website juga menyediakan mekanisme bagi pengunjungnya untuk dapat berinteraksi dengan dengan situs tersebut. Untuk dapat berinteraksi dengan website tersebut, perlu ditambahkan kode-kode pemrograman seperti PHP, ASP, JSP, Pearl dan sebagainya. Website yang telah ditambahkan kode program tersebut biasa dikenal dengan nama aplikasi web (*web application*).

2.8. PHP

Menurut Betha Sidik (2017), PHP secara umum dikenal sebagai bahasa pemrograman script-script yang membuat dokumen HTML secara *in the fly* yang di eksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML. Dikenal juga sebagai bahasa pemrograman server side. Dengan menggunakan PHP maka maintenance suatu situs web menjadi lebih mudah. Proses update data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan script PHP [15].

PHP/FI merupakan nama awal dari PHP. PHP – Personal Home Page, FI adalah Form Interface. Dibuat pertama kali oleh Rasmus Lerdoff. PHP, awalnya merupakan program CGI yang dikhususkan untuk menerima input melalui form yang ditampilkan dalam browser web. Software ini disebar dan dilisensikan sebagai perangkat lunak Open Source. CGI (Common Gateway Interface) sendiri menurut Beta Sidik adalah suatu standar yang menghubungkan (interface) aplikasi eksternal dengan server web.

Kini, PHP adalah kependekan dari PHP:HyperText Preprocessor(rekursif mengikut gaya penamaan di *nix), merupakan bahasa utama script server side yang disisipkan pada HTML yang dijalankan di server.

2.9. Android

Android adalah sebuah sistem operasi berbasis *linux kernel* dan dirancang untuk perangkat *mobile* dengan layar sentuh seperti *smartphone* dan komputer tablet. Awal mulanya dikembangkan oleh Android, Inc., yang didukung oleh Google secara finansial dan kemudian dibeli pada tahun 2005. Android ini diresmikan pada tahun 2007 seiring dengan berdirinya *Open Handset Alliance* yang merupakan konsorsium dari *hardware*, *software* dan perusahaan telekomunikasi yang ditujukan untuk memajukan standar terbuka untuk perangkat *mobile*.

Android memiliki banyak sekali fitur. Fitur-fitur pada *android* dapat diklasifikasikan sebagai berikut.

a. General

1) *Messaging*

SMS dan MMS tersedia dalam bentuk pesan, termasuk pesan teks *threaded* dan *Google Cloud Messaging (GCM)*.

2) *Web browser*

Web browser yang tersedia pada *android* adalah berbasis *Blink opensource layout engine*, ditambah dengan *Chrome's V8 JavaScript Engine*.

3) *Voice Based features*

Pencarian *Google* melalui suara telah tersedia sejak awal rilis. Perintah suara untuk perintah lainnya didukung pada *Android 2.2* dan seterusnya.

4) *Multi-touch*

Android memiliki dukungan asli untuk *multi-touch* yang awalnya tersedia pada handset seperti *HTC Hero*.

5) *Multitasking*

Tersedianya *multitasking* bagi aplikasi dengan penanganan alokasi memori yang unik.

6) *Screen capture*

Android mendukung pengambilan *screenshot* dengan cara menekan tombol *power* dan *volume-down* secara bersamaan.

7) *Video calling*

Android tidak mendukung panggilan video asli, tetapi beberapa perangkat memiliki versi kostumisasi dari sistem operasi yang mendukung, baik melalui jaringan UMTS ataupun melalui IP (*VoIP*). Panggilan video melalui aplikasi lainnya telah tersedia saat ini seperti *Skype*, *Hangouts* (*Google Talk*), *Vibe*, dan aplikasi lainnya.

8) *Multi language support*

Android mendukung multi bahasa.

9) *Accessibility*

Built in text to speech disediakan melalui *TalkBack* bagi orang-orang dengan gangguan penglihatan. Penambahan fitur bagi orang-rang dengan gangguan pendengaran tersedia seperti alat bantu lainnya.

b. *Connectivity*

1) *Connectivity*

Android mendukung teknologi konektivitas termasuk *GSM/EDGE*, *WiFi*, *Bluetooth*, *LTE*, *CDMA*, *EV-DO*, *UMTS*, *NFC*, *IDEN* dan *WiMax*.

2) *Bluetooth*

Mendukung panggilan suara dan mengirim kontak antar telepon, mengirim file (OOP), mengakses buku telepon (PBAP), A2DP dan AVRCP. Dukungan keyboard, mouse dan joystick (HID) tersedia pada *Android* versi 3.1 keatas dan pada versi sebelumnya melalui kostumisasi produsen dan aplikasi pihak ketiga.

3) *Tethering*

Android mendukung *tethering*, yang memungkinkan ponsel untuk berbagi konektivitas internet. Dapat dilakukan dengan media kabel ataupun nirkabel sebagai *Wi-Fi hotspot*. Sebelum *Android 2.2*, fitur ini didukung oleh aplikasi pihak ketiga atau kostumisasi pabrikan.

c. Media

1) *Streaming media support*

RTP / RTSP *streaming* (3GPP PSS, ISMA), progresif *download* HTML (HTML5 <video> tag). *Adobe Flash Streaming* (RTMP) dan *Streaming HTTP* Dinamis didukung oleh *plugin flash*. *Apple HTTP Live Streaming* didukung oleh *RealPlayer for Android*, dan oleh sistem operasi *Android 3.0 (Honeycomb)* keatas.

2) *Media support*

Android mendukung format media audio/video sebagai berikut: *WebM*, H.263, H.264, AAC, HE-AAC (*in 3GP or MP4 container*), MPEG-4 SP, AMR, AMR-WB (*in 3GP container*), MP3, MIDI, Ogg Vorbis, FLAC, WAV, JPEG, PNG, GIF, BMP, WebP.

3) *External Storage*

Kebanyakan perangkat *android* menyertakan *slot microSD* dan dapat membaca kartu *microSD* dengan format *Fat32*, *Ext3* atau sistem file *Ext4*. Untuk memungkinkan penggunaan media penyimpanan berkapasitas tinggi seperti *USB flash drive* dan *HDD USB*, banyak tablet *android* yang juga menyertakan *USB 'A' receptable* yang biasa disebut *USB On-The-Go* atau *USB host*..

d. *Hardware Support*

Perangkat *android* dapat menyertakan kamera video, layar sentuh, GPS, akselerometer, giroskop, barometer, magnetometer, kontrol game khusus, sensor kedekatan dan tekanan, thermometer, *2D bit blits* terakselerasi (dengan orientasi perangkat keras, *scaling*, konversi format piksel) dan grafis 3D terakselerasi.

e. Fitur Lainnya

1) *Java support*

Sementara sebagian besar aplikasi *android* ditulis dengan *Java*, tidak ada *Java Virtual Machine* di *platform* dan kode *byte Java* tidak dieksekusi. Kelas *java* dikompilasi kedalam *executable Dalvik* dan berjalan pada *Dalvik*, yaitu mesin virtual khusus untuk *android* dan dioptimalkan untuk perangkat *mobile* bertenaga baterai dengan memori dan CPU yang terbatas. Dukungan J2ME dapat diberikan melalui aplikasi pihak ketiga.

2) *Handset layout*

Platform ini bekerja untuk berbagai ukuran layar dari ukuran *smartphone* hingga ukuran tablet dan dapat terhubung ke layar eksternal, misalnya melalui *HDMI* atau secara nirkabel dengan *Miracast*. Orientasi potret dan lansekap juga didukung dan biasanya beralih antar dengan memutar. *Library grafis 2D*, *library grafis 3D* yang digunakan berbasis *OpenGL ES 2.0*.

3) *Storage*

SQLite, sebuah *database* relasional yang ringan. Digunakan untuk tujuan penyimpanan data. [16]