BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Quick Response Code atau yang biasa disebut dengan *QR Code* merupakan sebuah *barcode* dua dimensi yang diperkenalkan oleh Perusahaan Jepang Denso Wave pada tahun 1994. Jenis barcode ini awalnya digunakan untuk pendataan inventaris produksi suku cadang kendaraan dan sekarang sudah digunakan dalam berbagai bidang layanan bisnis dan jasa untuk aktivitas marketing dan promosi. Pada dasarnya bahwa *QR Code* dikembangkan sebagai suatu kode yang memungkinkan isinya untuk dapat diterjemahkan dengan kecepatan tinggi(*Rouillard*, 2008). Keunggulan dari *QR Code* adalah mampu menyimpan informasi secara *horizontal* dan *vertikal*. Oleh karena itu, *QR Code* dapat menampung informasi yang lebih banyak dibandingkan dengan *barcode* satu dimensi (David, 2007). Saat ini, untuk penggunaan *QR Code* telah banyak diimplementasikan dalam bentuk aplikasi *QR Code Reader* dan *QR Code Generator*, sehingga seseorang akan sangat mudah untuk membuat informasi dalam bentuk *QR Code* dan mendapatkan informasi yang ingin diketahuinya, hanya dengan melakukan proses scanning dan pemindaian data melalui media dari kamera *handphone* (Anastasia, Istiadi, dan Hidayat, 2010).

Dalam bidang pelayanan bisnis, *QR Code* telah banyak digunakan oleh perusahaan ataupun penyedia jasa layanan tertentu untuk dapat mengarahkan pelanggannya langsung ke alamat *URL* yang dituju (*Anonymous*, 2011), yaitu dengan memasang gambar *QR Code* pada majalah, poster, atau media cetak

lainnya, dimana *QR code* itu akan memaparkan segala sesuatu yang ingin disampaikan oleh perusahaan ataupun penyedia jasa layanan tersebut melalui situs mereka, seperti pada penelitian yang telah dilakukan terhadap Layanan Taman Nasional di *Fort Smith* (Cramer danTheresa, 2010), tiga Dealer Acura di kota *New York* (Sawyers dan Arlena, 2010), industri toko perhiasan (*Anonymous*, 2010), perusahaan *Cannondale* dan *Fox Racing Shox* (Norman dan Jason, 2010), perusahaan *J Vineyards & Winery* (*Anonymous*, 2011) dan aplikasi pemesanan tiket nonton bioskop berbasis android (Habibi, Purwantoro dan Akbar, 2012).

Selain itu, salah satu bentuk layanan bisnis yang paling menguntungkan bagi perusahaan ataupun penyedia jasa layanan tertentu pada saat ini adalah dengan menyediakan tiket dalam bentuk elektronik. Berikut ini merupakan beberapa contoh penelitian yang terkait dengan penggunaan aplikasi berbasis *QR Code* pada tiket elektronik.

Pada tahun 2011 perusahaan vendini melakukan penelitian terhadap respon dari pelanggannya mengenai model penjualan tiket elektronik yang ditawarkan oleh perusahaan ini dalam bentuk aplikasi *QR Code* untuk berbagai acara konser, olahraga, dan pertunjukan langsung. Hasil penelitian menunjukan respon pelanggan sangat positif terhadap penggunaan aplikasi *QR Code* dalam bentuk tiket, dikarenakan mereka merasa lebih dipermudah dengan hanya menunjukan bukti tiket *QR Code* tersebut dari dalam *smartphone* yang dibawa.

Crocker, Paul, Nicolau, dan Vasco (2011) juga melakukan penelitian dengan mengusulkan sebuah arsitektur inovatif baru untuk keamanan pengiriman tiket elektronik dalam bentuk *QR Code* yang dapat digunakan pada berbagai jasa

layanan di Negara Portugis, sedangkan untuk sistem pembayarannya menggunakan teknologi NFC, dimana untuk keamanan tiket elektronik tersebut akan diintegrasikan dengan e-KTP Nasional Portugis yang didalamnya telah terdapat PIN kriptografi dan otentikasi sistem biometrik.

Masih pada tahun yang sama 2011, Finzgar dan Trebar melakukan penelitian dengan mengembangkan suatu implementasi sistem tiket online yang memungkinkan pengguna *smartphone* untuk memperoleh dan membayar tiket elektronik pada bagian transportasi umum dalam bentuk aplikasi *QR Code* dan teknologi *NFC*.

L. Hu, Y. Wang, D. Li dan J. Li (2010), juga melakukan penelitian dengan mengembangkan sebuah sistem tiket terpadu menggunakan *QR Code* untuk tiket elektronik pada *handphone* yang didalamnya telah di enkripsi dengan algoritma MD5. Dengan adanya pengembangan sistem ini, maka akan memberikan kemudahan dan kenyamanan bagi pengguna dalam melakukan transaksi tiket, sekaligus menerapkan sistem keamanan data pada tiket.

Pada tahun 2010, Li, Wang, Hu, Li, Guo, Lin dan Liu juga melakukan penelitian dengan mengembangkan sebuah arsitektur klien atau server berdasarkan sistem tiket elektronik pada bagian jalur penumpang dengan menggunakan *QR Code* di *handphone*. Dengan adanya pengembangan sistem ini, maka akan memberikan kemudahan dan kenyamanan bagi penumpang dalam melakukan transaksi tiket.

Di tahun 2012, Zhang, Yao dan Zhou melakukan penelitian dengan mengembangkan aplikasi tiket elektronik berbasis QR Code pada *handphone*

yang telah di enkripsi dengan menggunakan algoritma RSA untuk industri pariwisata di taman Shenzhen Happy Valley China. Dengan adanya pengembangan aplikasi ini, maka akan memberikan kemudahan dan keamanan bagi wisatawan dalam melakukan transaksi tiket.

Conde-Lagoa, Costa-Montenegro, Gonzalez-Castano, dan Gil-Castineira (2010) juga melakukan penelitian dengan membangun sebuah aplikasi untuk tiket elektronik menggunakan *QR Code* dengan data yang dapat di *password* oleh pengguna. Sistem *password* yang terenkripsi menggunakan algoritma Rijndael 128 bit. Dengan adanya pengembangan aplikasi ini, maka akan memberikan kenyamanan dan keamanan bagi pengguna.

Di tahun 2010 Canadi, Hopken, dan Fuchs juga melakukan penelitian dengan mengembangkan sebuah aplikasi tiket *online* dalam bentuk *QR code* untuk objek wisata budaya di museum Mercedez Benz Jerman, dimana dengan adanya ketersediaan tiket *online* tersebut, maka para pengunjung museum dapat dengan mudah untuk memesan dan mendapatkan tiket melalui *smartphone* mereka.

Pada tahun 2012, Suparta juga melakukan penelitian dengan mengembangkan sebuah sistem tiket elektronik dalam bentuk *QR Code* pada bandara internasional di Yogyakarta, dimana tiket elektronik tersebut akan dikombinasikan dengan teknologi *NFC* sebagai media untuk pembayaran dan validasi tiket.

Tabel 2.1 berikut ini merupakan gambaran dari beberapa perbandingan penelitian yang dibahas.

Tabel 2.1. Perbandingan Penelitian.

No	Penelitian	Tujuan	Media	Hasil
1.	Wireless News, 2011, *Vendini Releases New QR Code Capability for Mobile Ticketing, Journal of Communications, ProQuest document ID* 900988114	Memberikan penawaran untuk model penjualan tiket dalam bentuk elektronik kepada pelanggannya.	*QR Code*	Respon pelanggan sangat positif terhadap penggunaan *QR Code* dalam bentuk tiket, dikarenakan mereka merasa lebih dipermudah dengan hanya menunjukan bukti tiket *QR Code* tersebut dari dalam *smartphone* yang dibawa.
2.	Crocker, Paul, Nicolau, and Vasco, 2011, *A Secure Architecture for Electronic Ticketing Based on the Portuguese e-ID Card, Journal of Computer Security, ProQuest document ID* 1010346768, *Pages.* 38-VII	Mengusulkan sebuah arsitektur inovatif baru untuk keamanan pengiriman tiket elektronik dan sistem pembayarannya yang akan diintegrasikan dengan Portugis e-ID.	*QR Code* dan NFC	Sistem ini akan memberikan kemudahan dalam melakukan proses pembayaran dan transaksi tiket pada berbagai jasa layanan di Negara Portugis, serta memberikan jaminan keamanan bagi penggunanya, karena sudah dilengkapi dengan penggunaan PIN kriptografi dan otentikasi sistem biometrik pada Portugis e-ID.
3.	Finzgar, L., and Trebar, M., 2011, *Use of NFC and QR code identification in an electronic ticket system for public transport, Conference International, Slovenia, Pages* 1-6.	Mengembangkan suatu implementasi sistem tiket online pada bagian transportasi umum.	*QR Code* dan NFC	Sistem ini akan memberikan kemudahan dan kecepatan bagi pengguna *smartphone* untuk memperoleh dan membayar tiket elektronik pada bagian transportasi umum.

4.	Hu, L., Wang, Y., Li, D., and Li, J., 2010, *A hybrid client/server and browser/server mode-based universal mobile ticketing system, IEEE, International Conference on Information Management and Engineering, Pages.* 691-695.	Mengembangkan sebuah sistem tiket terpadu dalam bentuk elektronik pada *handphone*.	*QR Code,* dan Algoritma MD5	Sistem ini akan memberikan kemudahan dan kenyamanan bagi pengguna dalam melakukan transaksi tiket, sekaligus menerapkan sistem keamanan data pada tiket.
5.	Li, D., Wang, Y., Hu, L., Li, J., Guo, X., Lin, J., and Liu, J., 2010, *Client/Server Framework-Based Passenger Line Ticket System Using 2-D Barcode on Mobile Phone, IEEE, International Conference on E-Business and E-Government*, *Pages.* 97-100.	Mengembangkan sebuah arsitektur klien atau server berdasarkan sistem tiket elektronik pada bagian jalur penumpang.	*QR Code*	Sistem ini akan memberikan kemudahan dan kenyamanan bagi penumpang dalam melakukan transaksi tiket.
6.	Zhang, M., Yao, D., and Zhou, Q., 2012, *The Application and Design of QR Code in Scenic Spot's eTicketing System -A Case Study of Shenzhen Happy Valley, International Journal of Science and Technology*, Vol. 2, No. 12.	Mengembangkan aplikasi tiket elektronik pada *handphone* untuk industri pariwisata di taman Shenzhen Happy Valley China.	*QR Code* dan Algoritma RSA	Dapat memberikan kemudahan dan keamanan bagi wisatawan dalam melakukan transaksi tiket.

7.	Conde-Lagoa, D., Costa-Montenegro, E, Gonzalez-Castano, F.J., and Gil-Castineira, F., 2010, *Secure eTickets Based on QR-Codes with User-Encrypted Content, IEEE*, *International Conference on Consumer Electronics*, *Pages.* 257-258.	Membangun sebuah aplikasi untuk tiket elektronik dengan data yang dapat di *password* oleh pengguna.	*QR Code* dan Algoritma Rijndael 128 bit	Dengan adanya pengembangan aplikasi ini, maka akan memberikan kenyamanan dan keamanan bagi pengguna.
8.	Canadi, M., Hopken, W., and Fuchs, M., 2010, *Application of QR Codes in Online Travel Distribution, Information and Communication Technologies in Tourism* 2010*, pp* 137-148.	Mengembangkan sebuah aplikasi tiket *online* dalam bentuk *QR code* untuk objek wisata budaya di museum Mercedez Benz Jerman	*QR Code*	Dengan adanya ketersediaan tiket *online* pada objek wisata budaya di museum Mercedez Benz Jerman, maka para pengunjung museum akan dapat dengan mudah untuk memesan dan mendapatkan tiket melalui *smartphone* mereka.
9.	Suparta, W., 2012, *Application of Near Field Communication Technology for Mobile Airline Ticketing, Journal of Computer Science, ISSN* 1549-3636*,* © 2012 *Science Publications.*	Mengembangkan sebuah aplikasi sistem tiket elektronik pada bandara Internasional di Yogyakarta	NFC dan *QR Code*	Calon penumpang pesawat akan lebih mudah dalam mendapatkan, membayar, dan memvalidasi tiket di bandara Internasional Yogyakarta.

Berdasarkan perbandingan pada tabel 2.1, dapat disimpulkan bahwa permasalahan yang akan diteliti penulis terkait dengan kasus antrian dalam pembelian tiket secara konvensional yang terjadi pada Stadion Utama Gelora Bung Karno, bisa diatasi dengan menggunakan penerapan tiket berbasis *mobile* dalam bentuk aplikasi *QR Code*. Disamping itu, untuk mencegah terjadinya pemalsuan tiket yang pernah dilakukan oleh oknum-oknum tertentu, maka pada pengembangan sistem ini akan digunakan algoritma *Data Encryption Standard* (DES) untuk mengamankan data pada tiket, sehingga keaslian data tiket dapat tetap terjaga dan disisi lain juga memberikan kemudahan bagi pihak PSSI dalam mendistribusikan tiket secara efektif.

2.2. Landasan Teori

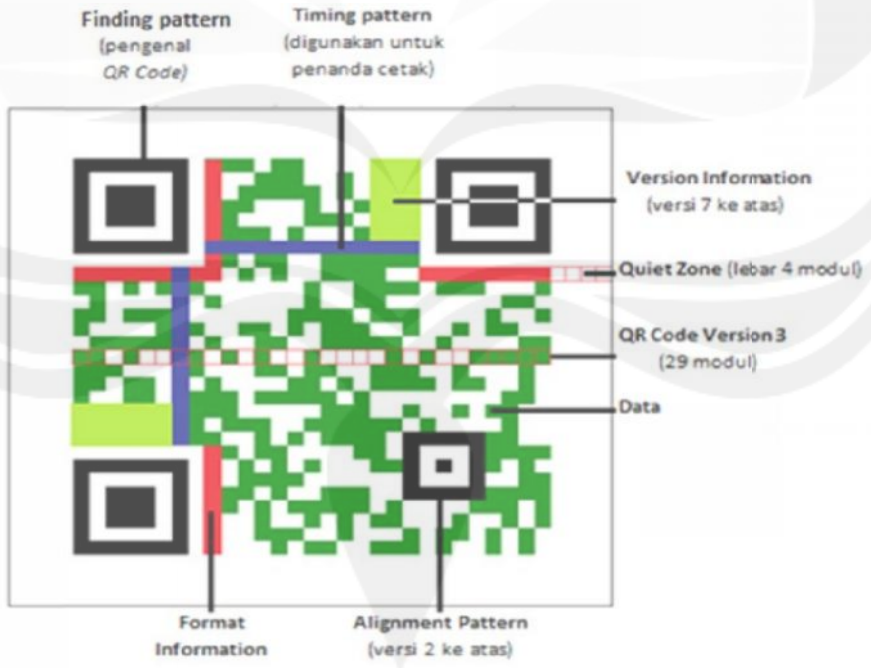
2.2.1. *Quick Response Code* (*QR Code*).

Quick Response Code atau yang sering disingkat dengan *QR Code* merupakan sebuah barcode dua dimensi yang diperkenalkan oleh Perusahaan Jepang Denso Wave pada tahun 1994. Jenis barcode ini awalnya digunakan untuk melacak persediaan di bagian manufaktur kendaraan dan sekarang sudah digunakan dalam berbagai industri perdagangan dan jasa. Pada dasarnya bahwa *QR Code* dikembangkan sebagai suatu kode yang memungkinkan isinya untuk dapat diterjemahkan dengan kecepatan tinggi(*Rouillard*, 2008). *QR Code* terdiri dari sebuah untaian kotak persegi yang disusun dalam suatu pola persegi yang lebih besar, yang disebut sebagai modul. Gambar 2.1 berikut ini, menunjukan gambaran dari sebuah *QR Code*.

Gambar 2.1. *QR Code* (Ariadi, 2011).

2.2.2. Struktur *QR Code*

QR Code memiliki bagian-bagian struktur yang akan penulis jelaskan pada gambar 2.2 dibawah ini (Ariadi, 2011).



Gambar 2.2. Struktur *QR Code* (Ariadi, 2011).

Berikut ini merupakan penjelasan dari istilah-istilah yang berkenaan dengan gambar *QR Code* di atas :

1) *Finding Pattern* merupakan pola untuk mendeteksi posisi dari *QR Code*.
2) *Timing pattern* merupakan pola yang digunakan untuk identifikasi koordinat pusat dari *QR Code*, dibuat dalam bentuk modul hitam putih bergantian.
3) *Version Information* merupakan Versi dari sebuah *QR Code,* versi terkecil adalah 1 (21 x 21) modul dan versi terbesar adalah 40 (177 x 177) modul.
4) *Quiet Zone* merupakan daerah kosong dibagian terluar *QR Code* yang mempermudah mengenali pengenal *QR* oleh sensor *CCD*.
5) *QR Code version* merupakan versi *QR Code.* Pada contoh gambar, versi yang digunakan adalah versi 3 (29 x 29 modul).
6) Data merupakan daerah tempat data tersimpan atau data dikodekan.
7) *Alignment Pattern* merupakan pola yang digunakan untuk memperbaiki penyimpangan *QR Code* terutama distorsi non linier.
8) *Format information* merupakan informasi tentang *error correction level* dan *mask pattern*.

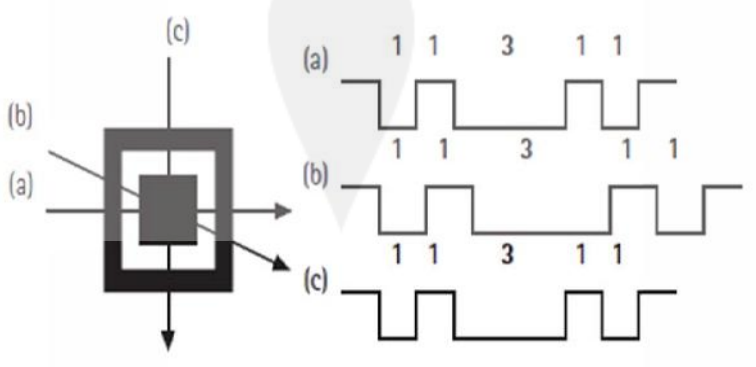
2.2.3. Karakteristik dari *QR Code*

Karakteristik dari *QR Code* yaitu dapat menampung jumlah data yang besar. Secara teori sebanyak 7089 karakter numerik maksimum data dapat tersimpan di dalamnya, kerapatan tinggi (100 kali lebih tinggi dari kode simbol *linier*) dan pembacaan kode dengan cepat. *QR Code* juga memiliki kelebihan lain baik dalam hal unjuk kerja dan fungsi (Ariadi, 2011). Berikut ini merupakan kelebihan unjuk kerja dan fungsi yang dimiliki oleh *QR Code*.

1) Pembacaan Data dari Segala Arah (360 derajat)

Pembacaan kode matriks dengan menggunakan sensor kamera CCD *(Charge Coupled Device)* dimana data akan memindai baris per baris dari citra yang ditangkap dan kemudian disimpan dalam memori. Dengan menggunakan suatu perangkat lunak tertentu, detail citra akan dianalisa, *finding pattern* akan dikenali dan posisi simbol dideteksi. Setelah itu proses pembacaan kode akan diproses. Sedangkan pada simbol linier ataupun kode dua dimensi lain akan memakan lebih lama waktu untuk mendeteksi letak atau sudut ataupun besar dari simbol tersebut.

QR Code memiliki *finding pattern* yang terlihat pada gambar 2.3, untuk memberitahukan letak simbol matriks dua dimensi *QR Code* yang disusun pada ketiga sudutnya. Hal inilah yang membuat *QR Code* dapat dibaca dari segala arah atau 360 derajat. Rasio antara modul hitam dan modul putih pada *finding pattern*-nya selalu 1:1:3:1:1. Dengan rasio ini, *finding pattern* dapat mendeteksi keberadaan citra yang ditangkap sensor. Sebagai tambahan, dengan adanya ketiga *finding pattern* maka pengkodean akan lebih cepat dua puluh kali dibandingkan kode matriks lain.



Gambar 2.3. *Finding Pattern QR Code* (Ariadi, 2011).

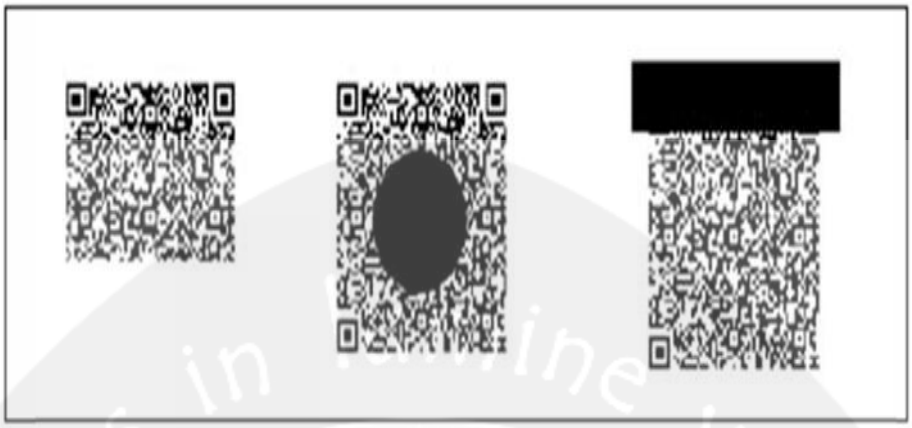
2) Ketahanan terhadap Penyimpangan Simbol

Simbol matriks 2 dimensi akan rentan terhadap penyimpangan bentuk ketika ditempatkan pada permukaan yang tidak rata (bergelombang), sehingga sensor pembaca menjadi miring karena sudut antara sensor CCD dan simbol matriks 2 dimensi ini telah berubah. Untuk memperbaiki penyimpangan ini, *QR Code* memiliki perata pola (*Alignment pattern*) yang menyusun dengan jarak yang teratur dalam satu daerah. *Alignment pattern*, akan memperhitungkan titik pusat dengan daerah terluar dari simbol matriks, sehingga dengan cara ini penyimpangan *linier* maupun *non-linier* masih dapat terbaca. Gambar 2.4 berikut ini merupakan jenis penyimpangan pada *QR Code*.

Gambar 2.4. Jenis Penyimpangan pada *QR Code* (Ariadi, 2011).

3) Fungsi Pemulihan Data (ketahanan terhadap kotor maupun kerusakan)

QR Code mempunyai empat tingkatan koreksi *error* (7%, 15%, 25% dan 30%) di dalam mengendalikan kerusakan yang diakibatkan kotor ataupun rusak. *QR Code* memanfaatkan algoritma *Reed-Solomon* yang tahan terhadap kerusakan tingkat tinggi. Jadi, ketika *QR Code* akan digunakan dalam lingkungan yang rawan kerusakan akibat dari lingkungan, disarankan untuk menggunakan koreksi *error* 30%. Gambar 2.5 berikut ini merupakan contoh dari kerusakan pada *QR Code*.



Gambar 2.5. Kerusakan pada *QR Code* (Ariadi, 2011).

4) Kemampuan *enkode* karakter kanji dan kana Jepang

QR Code berkembang pesat di negara jepang. Hal ini yang menyebabkan perkembangan *QR Code* untuk dapat menerima input data berupa karakter yang *non-alfabetis*. Ketika pembuatan *QR Code* dengan inputan berupa huruf jepang, maka data tersebut akan diubah ke dalam bentuk biner 16 bit (2 *byte*) untuk karakter tunggal, sedangkan untuk gabungan karakter akan di *enkode* dalam biner 13 bit. Hal ini memberikan keuntungan lain dimana proses enkode huruf jepang akan meningkatkan efisien 20% lebih banyak dari simbol kode 2 dimensi lain, dimana dengan volume data yang sama akan dapat dibuat pada area percetakan yang lebih kecil. (Ariadi, 2011)

5) Fungsi *Linking* pada Simbol

QR Code juga memiliki kemampuan dapat dipecah menjadi beberapa bagian dengan maksimum pembagiannya 16 bagian. Dengan fungsi *linking* ini, maka *QR Code* dicetak pada daerah yang tidak terlalu luas untuk sebuah *QR Code* tunggal. (Ariadi, 2011)

6) Proses *Masking*

Proses *Masking* pada *QR Code* berperan sangat penting dalam hal penyusunan modul hitam dan modul putih agar memiliki jumlah yang seimbang, untuk memungkinkan hal ini dapat digunakan pada operasi XOR yang diaplikasikan diantara area data dan daerah *mask pattern.* Ada sebanyak delapan *mask pattern* dalam *QR Code* yang kesemuanya itu dalam bentuk biner tiga bit. (Ariadi, 2011)

2.2.4. Spesifikasi Kode Matriks Dua Dimensi (*QR Code*)

QR Code memiliki kapasitas tinggi dalam hal data pengkodean, yaitu mampu menyimpan semua jenis data seperti numerik, alfanumerik, biner dan huruf kanji. Selain itu, *QR Code* juga memiliki empat tingkatan koreksi *error* yaitu 7%, 15%, 25% dan 30% di dalam mengendalikan kerusakan yang diakibatkan kotor ataupun rusak. Tabel 2.2 berikut ini menjelaskan tentang spesifikasi dari *QR Code*.

Tabel 2.2. Spesifikasi *QR Code* (Ariadi, 2011).

Jenis Simbol	Minimal 21 x 21 Modul dan maksimal 177 x 177 modul dengan peningkatan 1 versi = 4 modul	
Jenis Informasi dan Kapasitas	Numerik	Maksimum 7089 karakter
	Alfanumerik	Maksimum 4296 karakter
	Biner	Maksimum 2953 karakter
	Huruf Kanji	Maksimum 1817 karakter
Koreksi *Error*	Level L	Dapat mengembalikan data yang mengalami kerusakan 7%
	Level M	Dapat mengembalikan data yang mengalami kerusakan 15%
	Level Q	Dapat mengembalikan data yang mengalami kerusakan 25%
	Level H	Dapat mengembalikan data yang mengalami kerusakan 30%

2.2.5. *Mode* inputan Data

Mode inputan data yang dikenali oleh *QR Code* ada beberapa macam, diantaranya adalah sebagai berikut (Ariadi, 2011) :

1) *Mode* ECI (*Extended Channel Interpretation)*

Mode ini membolehkan kita untuk mengkodekan sekumpulan karakter, yang bukan termasuk karakter umum (*alfabet*), misalnya huruf arab, huruf sirilik serbia, yunani, dan ibrani.

2) *Mode* Numerik

Mode numerik akan mengkodekan data desimal dari angka 0 sampai 9 (ASCII : 30_{hex} - 39_{hex}) dengan kepadatan pengkodean 3 karakter, untuk setiap 10 bit biner.

3) *Mode* Alfanumerik

Mode alfanumerik memiliki jumlah 45 karakter, yaitu sebanyak 10 digit yang dimulai dari angka 0 sampai 9 (ASCII : 30_{hex} - 39_{hex}), karakter *alfabet* A sampai Z (ASCII : 41_{hex} – $5A_{hex}$) dan 9 karakter simbol (**spasi**, **$**, **%**, *, +, -, ., /, **:**) dengan pengkodean untuk ASCII (20_{hex}, 24_{hex}, $2A_{hex}$, $2B_{hex}$, $2D_{hex}$, $2E_{hex}$ $2F_{hex}$, dan $3A_{hex}$). Kepadatan pengkodean adalah 2 karakter untuk setiap 11 bit biner.

4) *Mode* 8 bit

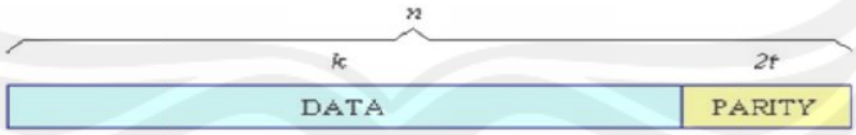
Mode ini menangani 8 bit bahasa latin dan karakter kana jepang, serta telah distandarisasi dalam bentuk JIS (*Japanese Industrial Standards)* X021, dalam ASCII dimulai dari 00_{hex} - FF_{hex}. Pada *mode* ini, kepadatan datanya adalah 8 bit untuk setiap karakter.

5) *Mode* huruf kanji

Mode ini menangani karakter kanji Jepang berdasarkan JIS X 0201 dan mode kepadatannya adalah setiap 2 *byte* karakter kanji, ditampung dalam 13 bit biner.

2.2.6. Fungsi koreksi *Error*

Reed Solomon code adalah kode siklik *non* biner yang terbuat dari 2^n bit biner dimana m lebih besar daripada 2. Untuk sebuah *codeword*, panjang kode adalah 8 bit, maka *Reed Solomon code* : $2^8 - 1 = 255$. Untuk mengkoreksi kesalahan pada *codeword*, maka ditambahkan *Reed Solomon code* agar dapat menahan dari kerusakan tanpa harus kehilangan data. Kemampuan koreksi *error*-nya bergantung pada jumlah data yang dikodekan. *Reed Solomon code* terdiri atas 2 bagian, yaitu bagian data dan bagian *parity*. *Reed-Solomon code* dinyatakan dengan kode (n,k) atau RS(n,k) dimana n adalah maksimum *codeword,* yaitu 255, sedangkan k adalah jumlah dari *codeword* data. Berikut ini merupakan penjelasan dari gambar 2.6 mengenai struktur *Reed Solomon code* :



Gambar 2.6. Struktur *Reed Solomon Code* (Ariadi, 2011).

2t atau simbol *parity* adalah *codeword* yang digunakan untuk koreksi *error* dengan nilai maksimun adalah t. Sebagai contoh, RS (255,223) artinya terdapat total 255 *codeword* yang terdiri atas 223 *codeword* data dan 32 *codeword parity.*

n = 255, k = 223

2t = (255-223) → t = 16

sehingga *error* yang dapat diperbaiki adalah sebanyak 16 *codeword.* Sebagai informasi 1 *codeword* adalah 8 bit, sehingga total koreksi yang dapat diperbaiki adalah 16 x 8 bit = 128 bit.

Ketika panjang *Reed-Solomon code* kurang dari $2^8 - 1$, maka *padding* 0 digunakan untuk membuat *Reed-Solomon code* tepat $2^8 - 1$. Sewaktu proses pembacaan kode, *padding* 0 akan dibuang. Hal ini disebut dengan penyingkatan *Reed-Solomon code*. Sebagai contoh, misalkan terdapat 100 *codeword* data untuk mengkoreksi 8 buah *error*, sehingga akan dibutuhkan tambahan 16 *codeword parity.* Jadi total keseluruhan dari *codeword* menjadi 116, yang mana masih kurang dari $2^8 - 1$, sehingga harus ditambahkan 139 *codeword padding* 0.

2.2.7. Konsep Kriptografi

Konsep kriptografi sendiri telah lama digunakan oleh manusia misalnya pada peradaban Mesir dan Romawi walau masih sangat sederhana. Prinsip-prinsip yang mendasari kriptografi yakni (Nababan, 2011) :

1) *Confidentiality*

Confidentiality (kerahasiaan) yaitu layanan yang ditujukan untuk menjaga agar isi pesan yang di kirimkan tidak dapat dibaca oleh pihak lain (kecuali pihak pengirim, pihak penerima atau pihak-pihak yang memiliki ijin). Umumnya hal ini dilakukan dengan cara menyandikan pesan menjadi *ciphertext* sehingga sulit dibaca dan dipahami.

2) *Data integrity*

Data integrity (keutuhan data) yaitu layanan yang mampu menjamin pesan masih asli atau utuh atau belum pernah dimanipulasi selama masa waktu

pengiriman. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi adanya manipulasi pesan tersebut oleh pihak-pihak yang tidak berhak antara lain penghapusan, pengubahan atau penambahan data yang tidak sah oleh pihak lain.

3) *Authentication*

Authentication (otentikasi) yaitu layanan yang berhubungan dengan identifikasi. Baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi maupun mengidentifikasi kebenaran sumber pesan. Dua pihak yang saling berkomunikasi harus dapat mengotentikasi satu sama lain sehingga ia dapat memastikan sumber pesan. Pesan yang di kirim melalui saluran komunikasi juga harus di otentikasi asalnya. Dengan kata lain, aspek keamanan ini dapat di ungkapkan sebagai pertanyaan : apakah pesan yang diterima benar-benar berasal dari pengirim yang benar.

4) *Non-repudiation*

Non-repudiation (anti penyangkalan) yaitu layanan yang dapat mencegah suatu pihak untuk menyangkal aksi yang dilakukan sebelumnya, misalnya pengirim pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan. Sebagai contoh, misalnya pengiriman pesan memberi otoritas kepada penerima pesan untuk melakukan pembelian, namun kemudian ia menyangkal telah memberikan otoritas tersebut.

Dalam bidang kriptografi terdapat istilah-istilah yang sering digunakan, diantaranya adalah sebagai berikut :

a) Pesan, *Plaintext* dan *Chiphertext*

Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah *plaintext* (*cleartext*). Pesan dapat berupa data atau informasi yang di kirim melalui kurir, saluran telekomunikasi maupun saluran lain. Pesan yang tersimpan tidak hanya berupa text, tetapi juga dapat berbentuk citra (*image*), suara atau bunyi (*audio*) dan video atau berkas biner lainnya. Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan perlu disandikan kebentuk lain yang tidak dapat dipahami. Bentuk pesan yang tersandi disebut c*iphertext* atau sering juga disebut *kriptogram*. C*iphertext* harus dapat ditransformasikan kembali menjadi *plaintext* semula agar pesan yang diterima bisa dibaca.

b) Pengirim dan Penerima

Komunikasi data melibatkan pertukaran pesan antara dua *entitas*. Pengirim (*sender*) adalah entitas yang mengirim pesan kepada *entitas* lainnya. Penerima (*receiver*) adalah *entitas* yang menerima pesan. *Entitas* yang dimaksud dapat berupa orang, mesin (komputer, kartu kredit) dan sebagainya. Jadi orang dapat bertukar pesan dengan orang lainnya (Alice berkomunikasi dengan Bob) sementara didalam jaringan komputer mesin (komputer) berkomunikasi dengan mesin. contoh : mesin *ATM* dengan komputer *server* di bank.

c) Enkripsi (E) dan Dekripsi (D)

Proses menyandikan *plainteks* menjadi *ciphertext* disebut enkripsi, sedangkan proses mengembalikan *ciphertext* menjadi *plainteks* kebentuk teks semula (pesan asli) disebut dekripsi. Enkripsi dan dekripsi dapat diterapkan baik pada pesan yang dikirim maupun pada pesan tersimpan.

d) *Chiper* dan Kunci

Algoritma kriptografi disebut juga *cipher* yaitu aturan untuk *enciphering* dan *deciphering*, atau fungsi matematik yang digunakan untuk enkripsi dan dekripsi. Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yaitu himpunan yang berisi elemen-elemen *plainteks* dan himpunan yang berisi *ciphertext*. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara kedua himpunan tersebut. Misalkan *P* menyatakan *plainteks* dan *C* menyatakan *ciphertext*, maka fungsi enkripsi *E* memetakan P ke C

E(P) = C

Dan fungsi dekripsi D memetakkan C ke P

D(C) = P

Karena proses *enkripsi* kemudian dekripsi mengembalikan pesan ke pesan asal, maka kesamaan berikut harus benar :

D (E(P) = P

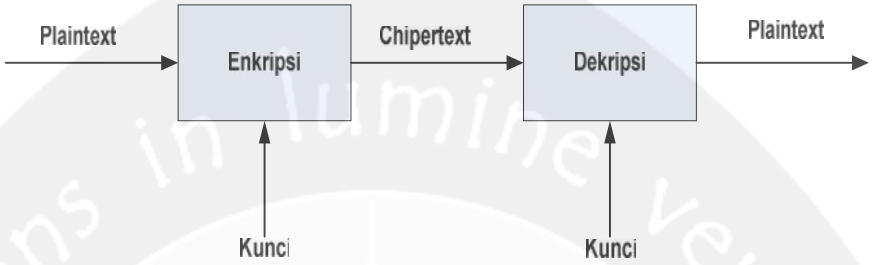
Kriptografi modern juga telah banyak mengatasi masalah dengan penggunaan kunci, yang dalam hal ini algoritma tidak lagi dirahasiakan, tetapi kunci harus dijaga kerahasiaannya. Kunci (*key*) adalah parameter yang digunakan untuk transformasi *enciphering* dan *deciphering*. Kunci biasanya berupa *string* atau deretan bilangan. Dengan menggunakan kunci K, maka fungsi enkripsi dan dekripsi dapat ditulis sebagai berikut :

$E_k(P)= C$ dan $D_k(C) = P$

dan kedua fungsi ini memenuhi :

$D_k(E_k(P))= P$

Gambar 2.7 berikut ini merupakan sebuah ilustrasi dari skema *enkripsi* dan *dekripsi* dengan menggunakan kunci terhadap sebuah pesan.



Gambar 2.7. Proses Enkripsi atau Dekripsi Sederhana (Nababan, 2011).

sedangkan untuk besar data yang akan diolah dalam satu kali proses, maka algoritma kriptografi dapat dibedakan menjadi dua jenis yaitu :

a. Algoritma *Block Chiper*

   Informasi atau data yang hendak dikirim dalam bentuk blok-blok besar (misalnya 64 bit) dimana blok-blok ini dioperasikan dengan fungsi enkripsi yang sama dan akan menghasilkan informasi rahasia dalam blok-blok yang berukuran sama.

b. Algoritma *Stream Chiper*

   Informasi atau data yang hendak dikirim dioperasikan dalam bentuk blok-blok yang lebih kecil (*byte* atau bit), biasanya satu karakter persatuan waktu proses, menggunakan tranformasi enkripsi yang berubah setiap waktu.

e) Penyadap

Penyadap (*eavesdropper*) adalah orang yang mencoba menangkap pesan selama ditransmisikan. Tujuan penyadap adalah untuk mendapatkan informasi sebanyak-banyaknya mengenai sistem kriptografi yang digunakan untuk berkomunikasi dengan maksud memecahkan *ciphertext*.

f) *Kriptanalisis* dan *Kriptologi*

Kriptanalisis adalah ilmu dan seni untuk memecahkan *ciphertext* menjadi *plainteks* tanpa mengetahui kunci yang digunakan dan pelakunya disebut Kriptanalis. Jika seorang kriptografer mentransformasikan *plainteks* menjadi *ciphertext* dengan suatu algoritma dan kunci maka sebaliknya seorang kriptanalis berusaha untuk memecahkan *ciphertext* tersebut untuk menemukan *plainteks* atau kunci.

Kriptologi adalah studi mengenai kriptografi dan kriptanalisis. Baik kriptografi maupun kriptanalisis keduanya saling berkaitan.

2.2.8. Kriptografi Kunci Simetris

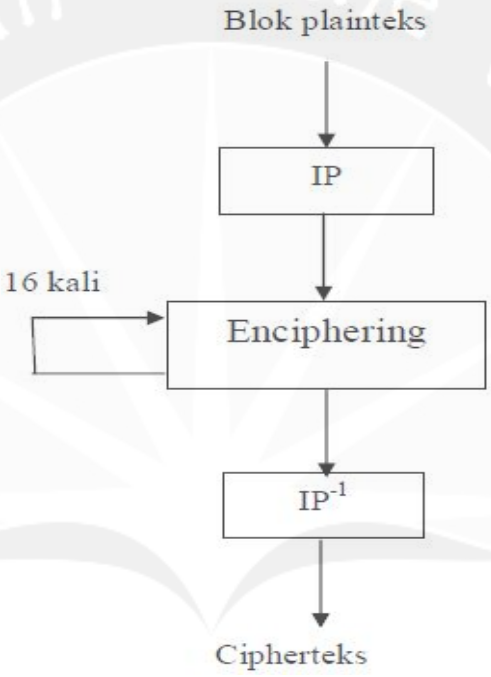
Algoritma ini mengasumsikan pengirim dan penerima pesan sudah berbagi kunci yang sama sebelum bertukar pesan. Algoritma simetris *(symmetric algorithm)* adalah suatu algoritma dimana kunci enkripsi yang digunakan sama dengan kunci dekripsi, sehingga algoritma ini disebut juga sebagai *single-key algorithm.* (Nababan, 2011). Algoritma yang memakai kunci simetris diantaranya adalah *Data Encryption Standard* (DES). Kelebihan dari algoritma simetris adalah sebagai berikut:

1) Kecepatan operasi lebih tinggi bila dibandingkan dengan algoritma asimetrik.
2) Karena kecepatannya yang cukup tinggi, maka dapat digunakan pada sistem *real-time.*

2.2.9. Algoritma *Data Encryption Standard* (DES).

Algoritma DES termasuk ke dalam sistem kriptografi kunci simetris dan tergolong ke dalam *cipher* blok, yang beroperasi pada ukuran blok sebesar 64 bit.

Algoritma DES mengenkripsi 64 bit plainteks menjadi 64 bit cipherteks dengan menggunakan 56 bit kunci internal (*internal key*). Kunci internal tersebut dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 bit. Gambar 2.8 berikut ini menjelaskan skema global dari proses enkripsi yang terjadi pada algoritma DES.



Gambar 2.8. Skema global enkripsi algoritma DES (Kromodimoeljo, 2009).

Keterangan skema global dari proses enkripsi yang terjadi pada algoritma DES di atas adalah sebagai berikut:

1) Blok plainteks dipermutasikan dengan matriks permutasi awal (*Initial Permutation*, IP).
2) Hasil permutasi awal kemudian dienkripsi sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.
3) Hasil enkripsi kemudian dipermutasikan dengan matriks permutasi balikan (*invers initial permutation*, IP^{-1}) menjadi blok *cipherteks*.

Di dalam proses enkripsi, blok *plainteks* dibagi menjadi dua bagian yaitu kiri (L) dan kanan (R), yang masing-masing panjangnya 32 bit. Kedua bagian ini masuk ke dalam 16 putaran algoritma DES. Pada setiap putaran i, blok R merupakan masukan untuk fungsi transformasi yang disebut *f*. Pada fungsi *f*, blok R dikombinasikan dengan kunci internal K_i. Keluaran dari fungsi *f*, akan di-XOR-kan dengan blok L untuk mendapatkan blok R yang baru. Sedangkan untuk blok L yang baru, langsung diambil dari blok R sebelumnya. Ini merupakan bentuk dari satu putaran algoritma DES. Secara matematis, satu putaran algoritma DES dinyatakan sebagai:

$L_i = R_{i-1}$ .....(1)

$R_i = L_{i-1} \oplus f\ (R_{i-1}, K_i)$ ..... (2)

Penjelasan dari rumus 1 dan 2 di atas adalah sebagai berikut :

R_{i-1} adalah blok yang sedang giliran dienkripsi.

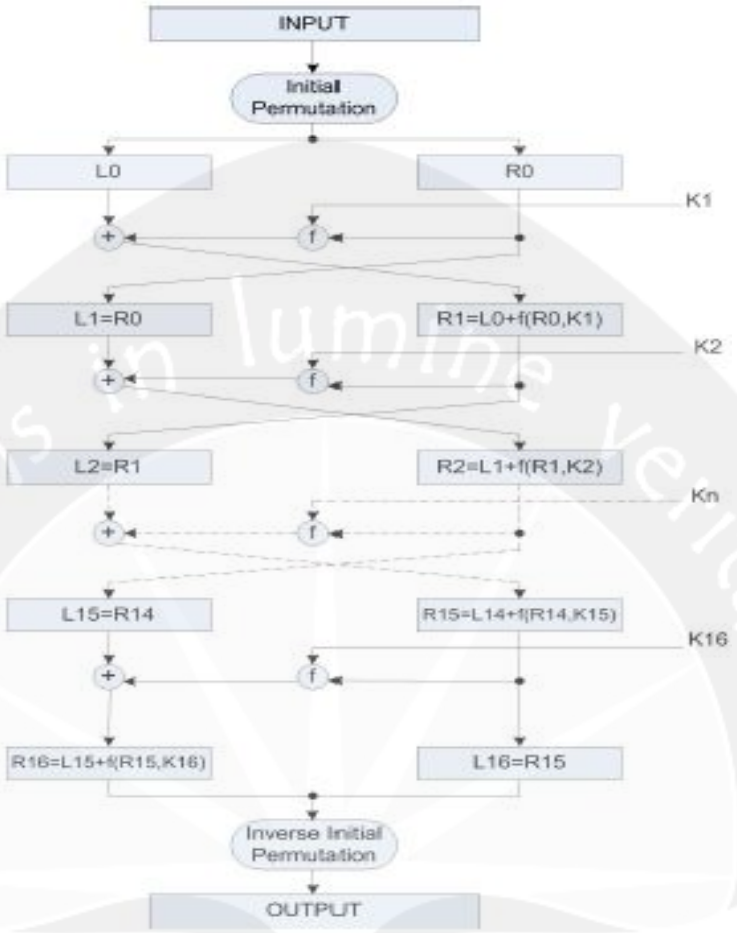
L_{i-1} adalah blok yang sedang giliran tidak dienkripsi.

$\oplus$ adalah operasi *exclusive or* secara *bitwise*.

f adalah fungsi *cipher*.

K_i adalah kunci untuk putaran *i*.

Gambar 2.9 memperlihatkan skema enkripsi algoritma DES yang lebih rinci, jika (L_{16},R_{16}) merupakan keluaran dari putaran ke 16, maka (L_{16},R_{16}) merupakan *pra-cipherteks* dari enkripsi ini. *Chiperteks* yang sebenarnya diperoleh dengan melakukan permutasi awal balikan IP^{-1}, terhadap blok *pra-cipherterks*.



Gambar 2.9. Proses Enkripsi DES (Kromodimoeljo, 2009).

Karena ada 16 putaran, maka dibutuhkan kunci internal sebanyak 16 buah juga, yaitu K_1, K_2,…, K_{16}. Kunci-kunci internal ini dapat dibangkitkan sebelum proses enkripsi atau bersamaan dengan proses enkripsi. Kunci internal dibangkitkan dari kunci eksternal yang diberikan oleh pengguna. Kunci eksternal panjangnya 64 bit atau 8 karakter, misalnya ada masukan kunci eksternal (K) yang tersusun dari 64 bit seperti yang terlihat pada tabel 2.3 berikut ini.

Tabel 2.3. Matriks kunci *eksternal* 64 bit (Kromodimoeljo, 2009).

Matriks kunci *eksternal* 64 bit									
	Bit ke:								
***Byte* ke:**		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
	1	1	2	3	4	5	6	7	8
	2	9	10	11	12	13	14	15	16
	3	17	18	18	20	21	22	23	24
	4	25	26	27	28	29	30	31	32
	5	33	34	35	36	37	38	39	40
	6	41	42	43	44	45	46	47	48
	7	49	50	51	52	53	54	55	56
	8	57	58	59	60	61	62	63	64

Tabel 2.3 matriks kunci eksternal ini, akan menjadi masukan untuk dapat dipermutasikan dengan menggunakan matriks permutasi kompresi (PC-1), seperti yang terlihat pada tabel 2.4.

Tabel 2.4. Matriks *Permutation Choice One* (PC-1) (Kromodimoeljo, 2009).

Matriks *Permutation Choice One* (PC-1)						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Dalam permutasi ini, tiap bit kedelapan (*parity* bit) dari delapan *byte* kunci diabaikan. Hasil permutasinya adalah sepanjang 56 bit, sehingga dapat dikatakan panjang kunci algoritma DES adalah 56 bit. Selanjutnya 56 bit dibagi menjadi 2 bagian, yaitu kiri dan kanan yang masing-masing panjangnya 28 bit, kemudian masing-masing disimpan di dalam C_0 dan D_0, seperti contoh berikut ini :

Co: berisi bit-bit dari K pada posisi:

57, 47, 41, 33, 25, 17, 9

1, 58, 50, 42, 34, 26, 18

10, 2, 59, 51, 43, 35, 27

19, 11, 3, 60, 52, 44, 36

D0: berisi bit-bit dari K pada posisi:

63, 55, 47, 39, 31, 23, 15

7, 62, 54, 46, 38, 30, 22

14, 6, 61, 53, 45, 37, 29

21, 13, 5, 28, 20, 12, 4

Selanjutnya, kedua bagian digeser ke kiri (*left shift*) sepanjang satu atau dua bit tergantung pada tiap putaran. Operasi pergeseran bersifat *round shift*. Jumlah pergeseran pada setiap putaran ditunjukkan pada tabel 2.5 sebagai berikut:

Tabel 2.5. Jumlah pergeseran bit pada setiap putaran (Kromodimoeljo, 2009).

Jumlah pergeseran bit pada setiap putaran																
Iterasi ke	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Pergeseran bit	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Misalnya (C_i, D_i) menyatakan penggabungan C_i dan D_i. (C_{i+1}, D_{i+1}) diperoleh dengan menggeser C_i dan D_i satu atau dua bit. Setelah pergeseran bit (C_i, D_i)

mengalami permutasi kompresi dengan menggunakan matriks PC-2 seperti pada tabel 2.6.

Tabel 2.6. Matriks *Permutation Choice Two* (*PC-2*) (Kromodimoeljo, 2009).

Matriks *Permutation Choice Two* (*PC-2*)					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Dengan permutasi ini, kunci internal K_1 diturunkan dari (C_i, D_i) yang dalam hal ini merupakan penggabungan bit-bit C_i pada posisi:

14, 17, 11, 24, 1, 5

3, 28, 15, 6, 21, 10

23, 19, 12, 4, 26, 8

16, 7, 27, 20, 13, 2

Dengan bit-bit D_i pada posisi:

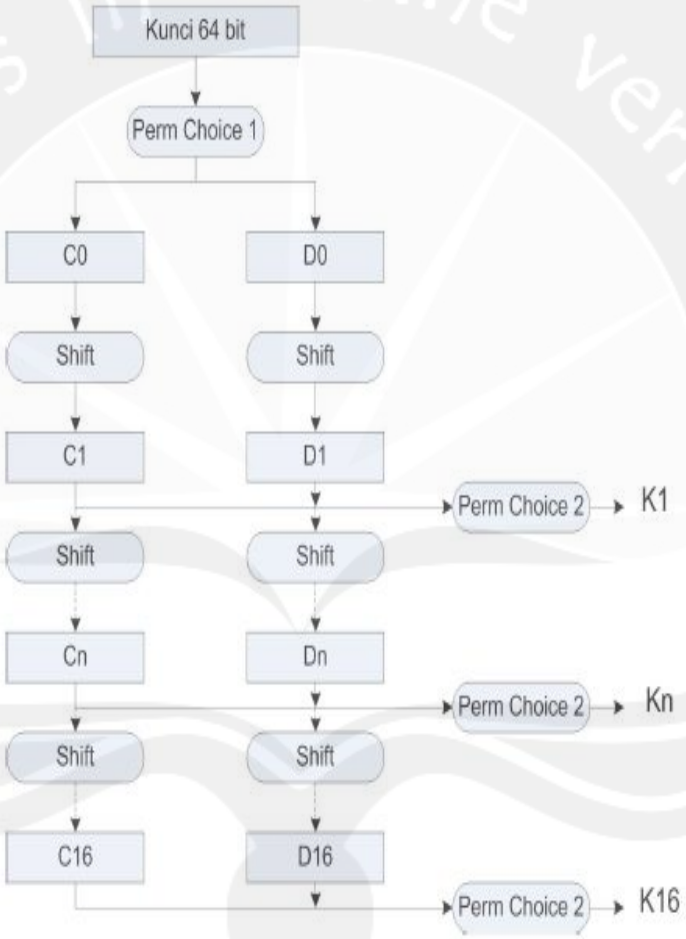
41, 52, 31, 37, 47, 55

30, 40, 51, 45, 33, 48

44, 49, 39, 56, 34, 53

46, 42, 50, 36, 29, 32

Jadi, setiap kunci internal K_i mempunyai panjang 48 bit. Proses pembangkitan kunci-kunci internal ditunjukkan pada gambar 2.10. Bila jumlah pergeseran bit-bit pada tabel 2.5 dijumlahkan semuanya, maka jumlah seluruhnya sama dengan 28, yang sama dengan jumlah bit pada C_i dan D_i. Karena itu, setelah putaran ke-16 akan didapatkan kembali $C_{16} = C_0$ dan $D_{16} = D_0$.



Gambar 2.10. Algoritma *Key Schedule* DES (Kromodimoeljo, 2009).

Untuk proses enkripsi data pada algoritma DES, sebelum putaran pertama terhadap blok *plainteks* pada tabel 2.7, akan dilakukan permutasi awal (*initial permutation*, IP) terlebih dahulu. Tujuannya adalah mengacak *plainteks* yang ada, sehingga urutan bit-bit di dalamnya berubah.

Tabel 2.7. Matriks *Plainteks* 64 bit (Kromodimoeljo, 2009).

Matriks *Plainteks* 64 bit									
	Bit ke:								
Byte ke:		1	2	3	4	5	6	7	8
	1	1	2	3	4	5	6	7	8
	2	9	10	11	12	13	14	15	16
	3	17	18	18	20	21	22	23	24
	4	25	26	27	28	29	30	31	32
	5	33	34	35	36	37	38	39	40
	6	41	42	43	44	45	46	47	48
	7	49	50	51	52	53	54	55	56
	8	57	58	59	60	61	62	63	64

Pengacakan dilakukan dengan menggunakan matriks *initial permutation* awal seperti pada tabel 2.8.

Tabel 2.8. Matriks *Initial Permutation* (IP) (Kromodimoeljo, 2009).

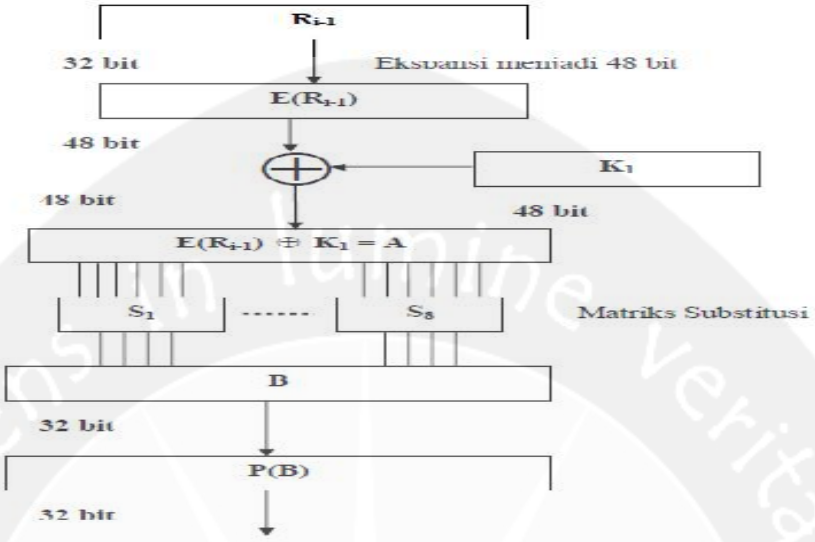
Matriks *Initial Permutation* (IP)									
	Bit ke:								
Byte ke:		1	2	3	4	5	6	7	8
	1	58	50	42	34	26	18	10	2
	2	60	52	44	36	28	20	12	4
	3	62	54	46	38	30	22	14	6
	4	64	56	48	40	32	24	16	8
	5	57	49	41	33	25	17	9	1
	6	59	51	43	35	27	19	11	3
	7	61	53	45	37	29	21	13	5
	8	63	55	47	39	31	23	15	7

Proses enkripsi terhadap blok *plainteks* dilakukan setelah permutasi awal. Setiap blok plainteks mengalami 16 kali perputaran enkripsi dan secara matematis, satu putaran algoritma DES dinyatakan sebagai:

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus f\,(R_{i-1}, K_i)$

untuk diagram komputasi fungsi f, diperlihatkan seperti pada gambar 2.11.



Gambar 2.11. Diagram komputasi fungsi f (Kromodimoeljo, 2009).

E adalah ekspansi yang memperluas blok R_{i-1} yang panjangnya 32 bit menjadi blok 48 bit. Fungsi ekspansi direalisasikan dengan matriks permutasi ekspansi seperti pada tabel 2.9 berikut ini.

Tabel 2.9. Matriks Permutasi Ekspansi (Kromodimoeljo, 2009).

Matriks Permutasi Ekspansi					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Selanjutnya hasil ekspansi, yaitu $E(R_{i-1})$ yang panjangnya 48 bit di-XOR-kan dengan K_1 yang panjangnya 48 bit dan menghasilkan vektor A yang panjangnya 48 bit:

$E(R_{i-1}) \oplus K_1 = A$

Vektor A dikelompokkan menjadi 8 kelompok, masing-masing 6 bit dan menjadi masukan bagi proses substitusi. Proses subtitusi dilakukan dengan menggunakan delapan kotak subtitusi (*S-Box*), yaitu dari kotak subtitusi (*S-Box* 1 sampai *S-Box* 8). Setiap kotak subtitusi akan menerima masukan 6 bit dan menghasilkan keluaran sebesar 4 bit. Kelompok 6 bit pertama menggunakan kotak subtitusi 1(*S-Box* 1), kelompok 6 bit kedua menggunakan kotak subtitusi 2 (*S-Box* 2) dan seterusnya. Tabel 2.10 berikut ini merupakan contoh dari kotak subtitusi 1(*S-Box* 1):

Tabel 2.10. Kotak Subtitusi 1(*S-Box* 1) (Kromodimoeljo, 2009).

Kotak Subtitusi 1 (*S-Box* 1)																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Tabel 2.11 berikut ini merupakan contoh dari kotak subtitusi 2(*S-Box* 2):

Tabel 2.11. Kotak Subtitusi 2(*S-Box* 2) (Kromodimoeljo, 2009).

Kotak Subtitusi 2 (*S-Box* 2)																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Tabel 2.12 berikut ini merupakan contoh dari kotak subtitusi 3(*S-Box* 3):

Tabel 2.12. Kotak Subtitusi 3(*S-Box* 3) (Kromodimoeljo, 2009).

Kotak Subtitusi 3 (*S-Box* 3)																
	0	**1**	**2**	**3**	**4**	**5**	**6**	7	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

Tabel 2.13 berikut ini merupakan contoh dari kotak subtitusi 4(*S-Box* 4):

Tabel 2.13. Kotak Subtitusi 4(*S-Box* 4) (Kromodimoeljo, 2009).

Kotak Subtitusi 4 (*S-Box* 4)																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Tabel 2.14 berikut ini merupakan contoh dari kotak subtitusi 5(*S-Box* 5):

Tabel 2.14. Kotak Subtitusi 5(*S-Box* 5) (Kromodimoeljo, 2009).

Kotak Subtitusi 5 (*S-Box* 5)																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

Tabel 2.15 berikut ini merupakan contoh dari kotak subtitusi 6(*S-Box* 6):

Tabel 2.15. Kotak Subtitusi 6(*S-Box* 6) (Kromodimoeljo, 2009).

Kotak Subtitusi 6 (*S-Box* 6)																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

Tabel 2.16 berikut ini merupakan contoh dari kotak subtitusi 7(*S-Box* 7):

Tabel 2.16. Kotak Subtitusi 7(*S-Box* 7) (Kromodimoeljo, 2009).

Kotak Subtitusi 7 (*S-Box* 7)																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

Tabel 2.17 berikut ini merupakan contoh dari kotak subtitusi 8(*S-Box* 8):

Tabel 2.17. Kotak Subtitusi 8(*S-Box* 8) (Kromodimoeljo, 2009).

Kotak Subtitusi 8 (*S-Box* 8)																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Keluaran proses substitusi adalah vektor B yang panjangnya 48 bit, Vektor B menjadi masukan untuk proses permutasi. Tujuan permutasi adalah untuk mengacak hasil proses substitusi kotak-S (*S-Box*). Permutasi dilakukan dengan menggunakan matriks permutasi P (P-*box*). Tabel 2.18 berikut ini merupakan tabel matriks permutasi P.

Tabel 2.18. Matriks Permutasi P (Kromodimoeljo, 2009).

Matriks Permutasi P			
16	17	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Bit-bit P(B) merupakan keluaran dari fungsi *f*. Akhirnya, bit-bit P(B) di-XOR-kan dengan L_{i-1} untuk mendapatkan R_i.

$Ri = L_{i-1} \oplus P(B)$ jadi,

$(L_i, R_i) = (R_{i-1}, L_{i-1} \oplus P(B))$

Permutasi terakhir dilakukan setelah 16 kali putaran terhadap gabungan blok kiri dan blok kanan, proses permutasi menggunakan matriks permutasi awal

balikan (*inverse initial permutation*, IP^{-1}). Tabel 2.19 berikut ini merupakan tabel matriks *inverse initial permutation* (IP^{-1}).

Tabel 2.19. Matriks *invers initial permutation* (IP^{-1}) (Kromodimoeljo, 2009).

Matriks *invers initial permutation* (IP^{-1})							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Untuk proses dekripsi terhadap blok *cipherteks* merupakan kebalikan dari proses enkripsi. DES menggunakan algoritma yang sama untuk proses enkripsi dan dekripsi. Jika pada proses enkripsi urutan kunci internal yang digunakan adalah K_1, K_2,…,K_{16}, maka pada proses dekripsi urutan kunci yang digunakan adalah K_{16}, K_{15}, …, K_1. Untuk tiap putaran 16, 15, …, 1, keluaran pada setiap putaran dekripsi adalah :

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus f(R_{i-1}, K_1)$

yang dalam hal ini (R_{16},L_{16}) adalah blok masukan awal untuk dekripsi. Blok (R_{16},L_{16}) diperoleh dengan mempermutasikan *cipherteks* dengan matriks permutasi (IP^{-1}). Pra-keluaran dari dekripsi adalah (L_0,R_0). Dengan permutasi awal (IP) akan didapatkan kembali blok *plainteks* semula.

2.2.10. *Mobile Ticketing*

Mobile ticketing atau tiket berbasis *mobile* merupakan sebuah mekanisme penyediaan tiket dalam bentuk elektronik yang tersimpan di dalam *handphone*. Dimana saat ini suatu perusahaan penyedia tiket akan lebih diuntungkan dengan adanya penerapan tiket dalam bentuk elektronik, karena dapat menghilangkan biaya produksi yang berkaitan dengan tiket konvensional. Disamping itu, perusahaan juga bisa meningkatkan kenyamanan pelanggannya dengan menawarkan cara-cara baru dan sederhana dalam membeli tiket.

Untuk proses pengiriman tiket elektronik via *handphone*, bisa dilakukan dengan menggunakan layanan seperti *Electronic Mail* (E-Mail). Alasan alternatif ini dipilih karena selain gratis, fitur seperti E-Mail juga telah banyak mendukung layanan dalam berbagai format dokumen seperti file program, gambar, grafik dan sebagainya, serta paling sering digunakan oleh berbagai individu, organisasi, maupun perusahaan.

2.2.11. *Electronic Mail* (E-Mail)

Electronic Mail (E-Mail) merupakan sebuah alat komunikasi yang digunakan untuk mengirim pesan dalam berbagai format dokumen seperti file-file berupa program, gambar, grafik, audio, dan video dalam suatu jaringan komputer intranet maupun internet.

2.2.12. Stadion Utama Gelora Bung Karno (GBK).

Bermula dari Asean Games III Tahun 1958 di Tokyo dimana oleh Asian Games Federation, Indonesia ditunjuk untuk menjadi penyelenggara *Asian Games* ke IV Tahun 1962. Maka pada saat itu Presiden R.I. Pertama Ir. Soekarno segera

menjawab tantangan dengan menentukan lokasi yang tepat untuk perhelatan akbar tersebut, dengan membangun Sarana dan Prasarana Olahraga. Melihat letak geografis dan pengembangan kota Jakarta di kemudian hari, maka pilihan jatuh ke arah selatan yaitu daerah Senayan, yang merupakan batas antara Jakarta Kota dan Satelit Kebayoran Baru.

Upacara pembukaan Asian Games ke IV tahun 1962 dilaksanakan di Stadion Utama Gelora Bung Karno yang dihadiri oleh lebih dari 110.000 orang. Pada Pidatonya Presiden R.I. Pertama Ir. Soekarno (Bung Karno) mengatakan bahwa peristiwa ini merupakan tonggak sejarah bagi Bangsa Indonesia khususnya dibidang olahraga yang merupakan bagian dari *Nation* and *Character Building*, maupun dalam rangka pergaulan dengan bangsa-bangsa lain di dunia.

Setahun kemudian dilaksanakan *GANEFO (Games of The New Emergencing Forces*) ke 1 tahun 1963. Dengan selesainya pembangunan Gelanggang Olahraga Bung Karno pada saat itu membuktikan bahwa bangsa Indonesia mampu melaksanakan pembangunan sebuah komplek olahraga bertaraf international yang pada masa itu belum banyak dimiliki oleh Negara maju sekalipun. Seiring dengan perkembangan jaman maka dikomplek Gelora Bung Karno dilaksanakan berbagai pembangunan fasilitas olahraga maupun fasilitas pendukung lainnya.Dukungan kepada dunia olahraga menjadi fokus dan perhatian kami dimana Gelora Bung Karno telah menanamkan dan tidak kurang Rp. 1 Triliun dalam bentuk berbagai Prasarana dan Sarana serta fasilitas lainnya sebagai bentuk sumbangsih kepada dunia olahraga.

Saat ini Kawasan Gelora Bung Karno berdiri berbagai macam fasilitas untuk kegiatan olahraga sebanyak 36 Venues, Politik, Bisnis, Rekreasi dan Pariwisata. Fungsi lain Kawasan Gelora Bung Karno adalah memiliki 84% Kawasan Terbuka Hijau yang merupakan daerah resapan air dengan lingkungan hijau seluas 67,5% yang masih terdapat kelestarian aneka pepohonan langka yang besar dan rindang yang merupakan hutan kota juga sebagai tempat bermukimnya 22 jenis burung liar yang senantiasa berkicau sepanjang hari menambah suasana asri di kawasan ini.

Selain itu juga telah dilakukan penataan secara terpadu dan menyeluruh pada Kawasan Gelora Bung Karno yaitu dengan dibangunnya plaza, gerbang, air mancur dan pedestrian yang tidak lain adalah untuk meningkatkan penampilan serta kenyamanan bagi masyarakat pengguna yang berkunjung di Kawasan Gelora Bung Karno.

Latar Belakang Pembentukan Gelora Bung Karno

1) KEPPRES 318 Tahun 1962 : Pembentukan Yayasan Gelora Bung Karno.
2) KEPPRES 4 Tahun 1984 : Badan Pengelola Gelanggang Olahraga Senayan Sebagaimana Telah Beberapa Kali Diubah, Terakhir Diubah Dengan KEPPRES 94 Tahun 2004.
3) KEPPRES 7 Tahun 2001 : Perubahan Nama Gelanggang Olahraga Senayan Menjadi Gelanggang Olahraga Bung Karno.
4) Keputusan Menteri Keuangan Nomor 233 Tahun 2008 : Tentang Penetapan Gelora Bung Karno Sebagai BLU (Badan Layanan Umum).


serviens in lumine veritatis