

## **BAB II**

### **LANDASAN TEORI**

Bab ini akan menjelaskan tentang landasan teori yang menjadi dasar dalam pelaksanaan kegiatan tugas akhir. Landasan teori yang akan dijelaskan meliputi gambaran sistem, gambaran sistem informasi, serta gambaran *QR Code*.

#### **2.1 Proses Belajar Mengajar**

Pengertian proses belajar mengajar atau bisa disebut pembelajaran pembelajaran yaitu suatu proses interaksi antara siswa dengan pengajar dan sumber belajar dalam suatu lingkungan. Pembelajaran merupakan bentuk bantuan yang diberikan pengajar supaya bisa terjadi proses mendapatkan ilmu dan pengetahuan, penguasaan kemahiran, pembentukan sikap dan kepercayaan pada murid. Dapat dikatakan bahwa pembelajaran adalah proses untuk membantu murid supaya bisa belajar secara baik.<sup>[9]</sup>

Pembelajaran mempunyai arti yang mirip dengan pengajaran, meskipun memiliki konotasi yang tidak sama.<sup>[10]</sup> Pada konteks pendidikan, seorang dosen memberikan materi perkuliahan agar mahasiswa bisa memahami dan menguasai isi matakuliah tersebut. Sehingga mahasiswa tersebut memperoleh suatu pengetahuan dan meningkatkan bakatnya untuk bersaing di dunia pekerjaan.

##### **2.1.1 Daftar Kehadiran**

Daftar kehadiran adalah naskah dinas yang dipergunakan untuk mencatat dan mengetahui kehadiran seseorang. Naskah dinas ini akan menjadi bukti kehadiran seseorang untuk keperluan atau tujuan tertentu. Biasanya daftar kehadiran ini digunakan pada setiap pertemuan atau kegiatan penting seperti perkuliahan, pekerjaan, acara pernikahan, dan yang lainnya.<sup>[11]</sup> Pada proses belajar mengajar daftar kehadiran ini sangat penting, karena daftar kehadiran tersebut

sering kali dijadikan sebagai bukti apakah siswa atau pengajar yang mengikuti proses belajar mengajar tersebut menghadiri kegiatan tersebut atau tidak.

## 2.2 Sistem

Definisi sistem berkembang sesuai dengan konteks dimana pengertian sistem itu digunakan. Berikut akan diberikan beberapa definisi sistem secara umum:

1. Kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan yang sama, misalnya sistem tatasurya, sistem pencernaan, sistem transportasi umum, sistem otomotif, sistem komputer dan sebagainya.
2. Sekumpulan objek-objek yang saling berelasi dan berinteraksi serta berhubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai suatu tujuan.

Dengan demikian, secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi dan saling bergantung satu sama lain. <sup>[1]</sup>

### 2.2.1 Karakteristik Sistem

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. <sup>[2]</sup> Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya:

1. Batasan (*boundary*), penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. Lingkungan (*environment*), segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.
3. Masukan (*input*), sumber daya (data, bahan baku, peralatan atau energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*output*), sumber daya atau produk (informasi, laporan, dokumen, tampilan layar computer, atau barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.

5. Komponen (*component*), kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*).
6. Penghubung (*interface*), tempat dimana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*storage*), area yang dikuasai dan digunakan untuk menyimpan sementara atau tetap dari informasi, *energy*, bahan baku, dan sebagainya

### 2.2.2 Sistem yang Buruk

Untuk menghindari pengembangan suatu sistem yang buruk, perlu diketahui beberapa ciri dari sistem yang buruk:

- a. Tidak memenuhi kebutuhan pengguna
- b. Performa sistem yang buruk
- c. Reabilitas rendah
- d. Kegunaan rendah
- e. Contoh-contoh kesulitan:
  - Tidak terjadwal
  - Tidak ada rencana anggaran
  - Bisa berjalan = 100% *over budget*

### 2.2.3 Beberapa Konsep Sistem yang Penting

Untuk lebih mudah memahami pengertian sistem dan sistem informasi lebih jauh maka perlu diingat beberapa konsep yang penting dalam pengembangan sistem, yaitu:

#### 1. Dekomposisi

Untuk menganalisa dan memahami secara menyeluruh sebuah sistem yang besar, biasanya dibutuhkan waktu yang cukup lama. Untuk mempermudah pekerjaan ini digunakan konsep dekomposisi.<sup>[2]</sup> Dekomposisi adalah pembagian

sistem ke dalam komponen-komponen yang lebih kecil (subsistem). Dekomposisi memiliki beberapa keuntungan, diantaranya:

- a. Analis menjadi lebih mudah mengatur dan menganalisa setiap subsistem secara lebih detail.
- b. Pada pengembangan sistem, sistem bisa didekomposisi menjadi beberapa modul. Pengembangan beberapa modul bisa dilakukan secara paralel dengan syarat tidak ada ketergantungan antar modul yang dibangun.

## 2. Modularitas

Konsep modularitas berhubungan dengan dekomposisi. Pada saat melakukan dekomposisi, diharapkan sistem yang besar terbagi menjadi subsistem-subsistem yang relatif ukurannya. Dengan modul-modul ini maka beban kerja mengembangkan sistem bisa didistribusikan secara merata pada semua sumber daya yang ada.<sup>[2]</sup> Pengembangan sistem jadi lebih sederhana karena hanya terfokus pada satu modul terlebih dahulu, baru dilakukan integrasi antar modul.

## 3. *Coupling*

Dari modul-modul yang kita peroleh, kadang-kadang ditemukan beberapa modul yang memiliki ketergantungan dengan modul yang lain. Pada kasus seperti ini, modul-modul yang saling bergantung harus dipasangkan (*couple*).<sup>[2]</sup> Dengan cara ini bisa diketahui modul yang bisa bekerja secara independen dan modul-modul yang harus diselesaikan terlebih dahulu sebelum modul yang lain bisa bekerja.

## 4. Kohesi

Dari proses *coupling* antar modul, kita bisa dapatkan kelompok-kelompok modul dengan karakteristik yang hampir sama. Disini muncul konsep kohesi dimana kelompok modul itu harus dianalisis bersama-sama dengan kelompok modul yang saling berkohesi.<sup>[2]</sup>

## 2.3 Sistem Informasi

Data merupakan nilai, keadaan, atau suatu sifat yang berdiri sendiri lepas dari konteks apapun. Sementara informasi adalah data yang telah diproses atau diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang (*Davis, 1995*). *McLeod (1995)* mengatakan bahwa informasi adalah data yang telah diproses atau data yang memiliki arti.<sup>[2]</sup>



Gambar 2.1 Konsep Sistem Informasi<sup>[1]</sup>

Sehingga sistem informasi merupakan sebuah sistem yang terdiri atas rangkaian subsistem informasi terhadap pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan.<sup>[2]</sup>

### 2.3.1 Komponen Sistem Informasi

Dalam suatu sistem informasi terdapat komponen-komponen sebagai berikut:

1. Perangkat keras (*hardware*), mencakup berbagai peranti fisik seperti komputer dan printer.
2. Perangkat lunak (*software*) atau program, yaitu sekumpulan instruksi yang memungkinkan perangkat keras memproses data.
3. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
4. Orang, yaitu semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.

5. Basis data (*database*), yaitu sekumpulan tabel, hubungan dan lain-lain yang berkaitan dengan penyimpanan data.
6. Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

## 2.4 Siklus Hidup Pengembangan Perangkat Lunak

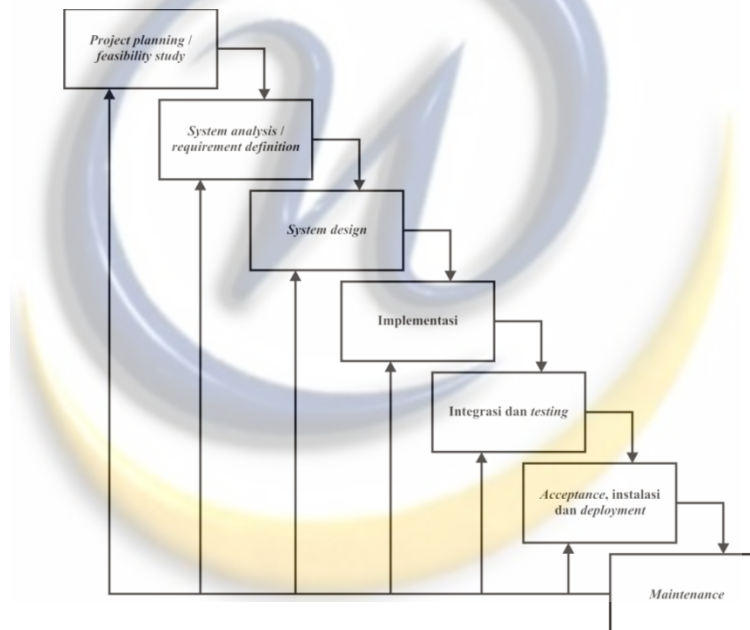
Siklus hidup pengembangan perangkat lunak atau *software development life cycle* (SDLC) adalah tahapan yang umum dilakukan dalam pembuatan atau perbaikan / penggantian sebuah sistem / *software*, dan mengacu juga pada metodologi yang digunakan orang untuk mengembangkan *software*.<sup>[3]</sup>

Konsep SDLC menjelaskan urutan-urutan fase, aktivitas dan keluaran setiap fase terkait dengan pengembangan perangkat lunak. Setiap fase pada SDLC menggunakan hasil dari fase sebelumnya. Hingga saat ini terdapat, terdapat banyak model SDLC yang sudah dipublikasikan oleh peneliti, misalnya *waterfall*, *fountain*, *spiral*, *build and fix*, *rapid prototyping*, *incremental* dan sebagainya.

*Waterfall* merupakan model SDLC yang tertua dan paling populer. Pada SDLC *waterfall*, terdapat beberapa urutan aktivitas sebagai berikut :

1. *Project planning / feasibility study*: menentukan tujuan atau target utama projek.
2. *System analysis / requirement definition*: mendefinisikan tujuan project menjadi fungsi-fungsi dan operasi yang lebih terinci, yang harus disediakan oleh aplikasi, termasuk menganalisis kebutuhan informasi *end-user*.
3. *System design*: mengembangkan fitur dan operasi yang diharapkan dapat disediakan oleh aplikasi secara lebih terinci, termasuk rancangan tampilan antarmuka, aturan bisnis, diagram proses, *pseudocode* dan dokumentasi lainnya.

4. Implementasi: mewujudkan rancangan sistem dalam bentuk kode program / aplikasi.
5. Integrasi dan *testing*: menggabungkan semua elemen dalam satu lingkungan uji yang terpadu, untuk memeriksa *error*, *bugs* dan *interoperability*.
6. *Acceptance*, instalasi dan *deployment*: tahap terakhir pada satu siklus awal pengembangan *software*, yaitu *software* dipasang dan digunakan langsung pada operasional bisnis yang sebenarnya.
7. *Maintenance*: berupa pemeliharaan, koreksi, perbaikan atau perubahan yang terjadi selama *software* digunakan.



**Gambar 2.2 Waterfall Model**

## 2.5 Pengujian Sistem

Pengujian sistem merupakan proses mengeksekusi sistem perangkat lunak untuk menentukan apakah perangkat lunak tersebut cocok dan berjalan sesuai dengan spesifikasi sistem yang telah dirancang sebelumnya. Menemukan dan menghilangkan ketidaksempurnaan ini disebut dengan *debugging*. Kesalahan sistem perangkat lunak sangat bervariasi, dimulai dari *output* yang tidak sesuai, sistem yang *crash*, sampai dengan sistem yang menggunakan memori terlalu



banyak atau sistem yang berjalan terlalu lambat. *Bug* seperti ini terjadi karena kode program yang dieksekusi belum diuji, urutan program yang dieksekusi belum diuji, kombinasi dari nilai input yang belum diuji, atau lingkungan pemrograman yang belum diuji.<sup>[2]</sup>

### 2.5.1 Metode Pengujian

Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi transaksi, *query*, atau jalur navigasi yang mewakili penggunaan sistem yang tipikal, kritis, atau abnormal. Isu kunci pada pengembangan sistem adalah pemilihan sekelompok *test-case* yang cocok, sekecil dan secepat mungkin, untuk meyakinkan perilaku sistem secara detail. Pengujian harus mencakup unit testing, yang memeriksa validasi dari prosedur dan fungsi-fungsi secara independen dari komponen sistem lain. Kemudian modul testing harus menyusul dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai dengan karakteristik sistem yang diinginkan.<sup>[2]</sup>

### 2.5.2 Unit Testing

Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan *unit testing*, yaitu:

#### 1. Black-Box Testing

Terfokus pada apakah unit program memenuhi kebutuhan (*requirement*) yang disebutkan dalam spesifikasi. Pada *black box testing*, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil unit tersebut sesuai dengan proses bisnis yang diinginkan atau tidak. Jika ada unit yang tidak sesuai outputnya maka untuk menyelesaikannya, diteruskan pada pengujian kedua yaitu *white box testing*.<sup>[2]</sup>



## 2. White-Box Testing

*White box testing* adalah cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak. Jika ada modul yang menghasilkan output tidak sesuai dengan proses bisnis, maka baris-baris program, variabel, dan parameter yang terlibat pada unit tersebut akan diperiksa satu per satu dan diperbaiki, kemudian di-compile ulang. <sup>[2]</sup>

### 2.6 Smartphone

Secara umum *smartphone* adalah sebuah ponsel multifungsi yang menggabungkan beberapa fungsi dari sebuah PDA, seperti *personal scheduler*, kalender dan *phone book*. Sebuah *smartphone* dilengkapi dengan kemampuannya untuk mengakses internet, memeriksa *e-mail*, memainkan *game online* sampai menulis dan mengedit dokumen *spreadsheet* seperti file *Microsoft Word* dan *Excel* layaknya sebuah komputer mini. Oleh karena itu, seperti halnya pada komputer, Anda juga dimungkinkan untuk membuat sebuah aplikasi yang selanjutnya dapat dijalankan pada *smartphone*. <sup>[12]</sup>

#### 2.6.1 Android

*Android* adalah 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 mereka sendiri untuk digunakan oleh bermacam peranti bergerak. <sup>[13]</sup> Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile*, dan *Nvidia*. <sup>[14]</sup>

### 2.7 Barcode (Kode Baris)

Di awal perkembangannya, penggunaan kode baris dilakukan untuk membantu proses pemeriksaan barang-barang secara otomatis pada supermarket.

Tetapi saat ini kode baris sudah banyak digunakan dalam berbagai aplikasi seperti misalnya digunakan sebagai kartu identitas, kartu kredit dan untuk pemeriksaan secara otomatis pada perpustakaan.<sup>[7]</sup>

Kode baris digambarkan dalam bentuk baris hitam tebal dan tipis yang disusun berderet sejajar horizontal. Untuk membantu pembacaan secara manual dicantumkan juga angka-angka di bawah kode baris tersebut. Angka-angka tersebut tidak mendasari pola kode baris yang tercantum. Ukuran dari kode baris tersebut dapat diperbesar maupun diperkecil dari ukuran nominalnya tanpa tergantung dari mesin yang membaca.

*Barcode* terdiri dari 2 jenis. Pertama yaitu *barcode* satu dimensi, biasanya dinamakan *linear bar codes* (kode berbentuk baris). Berikut contoh dari *linear barcode*:

1. *Code 39 (code 3 of 9)* adalah sebuah *barcode alphanumerik (full ASCII)* yang memiliki panjang baris yang bervariasi. Aplikasi *barcode* jenis *code 39* adalah untuk *inventory*, *asset tracking* dan digunakan pada tanda pengenalan identitas.<sup>[7]</sup>



**Gambar 2.3 Code 39<sup>[15]</sup>**

2. *Code 128* adalah suatu *barcode alphanumerik (full ASCII)* yang memiliki kerapatan (*density*) yang sangat tinggi dan panjang baris yang bervariasi. *Barcode code 128* ideal untuk aplikasi seperti *shipping and warehouse management* (pengaturan maskapai pelayaran dan pengelolaan gudang).<sup>[7]</sup>
3. *Interleaved 2 of 5* adalah sebuah *barcode* yang berbentuk numerik dan memiliki panjang baris yang bervariasi. *Barcode interleaved 2 of 5* dapat dipergunakan untuk aplikasi industri dan laboratorium.<sup>[7]</sup>

4. UPC (*Universal Product Code*) adalah sebuah *barcode* yang berbentuk numerik dan memiliki panjang baris yang tetap (*fixed*). UPC digunakan untuk pelabelan pada produk-produk kecil atau eceran (*retail product labeling*). Simbol ini dibuat untuk kemudahan pemeriksaan keaslian suatu produk. Bilangan-bilangan UPC harus diregistrasikan atau terdaftar di *Uniform Code Council*.<sup>[7]</sup>



**Gambar 2.4 Universal Product Code<sup>[16]</sup>**

Sedangkan *barcode* dua dimensi adalah *barcode* yang dikembangkan lebih dari sepuluh tahun lalu, tetapi baru sekarang ini mulai semakin populer. *Barcode* dua dimensi ini memiliki beberapa keuntungan dibandingkan *linear bar codes* (*barcode* satu dimensi) yaitu, dengan menggunakan *barcode* dua dimensi, informasi atau data yang besar dapat disimpan di dalam suatu ruang (*space*) yang lebih kecil. Contoh *barcode* dua dimensi adalah “*symbolology PDF417*” yang dapat menyimpan lebih dari 2000 karakter di dalam sebuah ruang (*space*) yang berukuran 4 *inch* persegi (*in*<sup>2</sup>). Ada juga kode QR (*Quick Response*) atau biasa dikenal dengan istilah *QR Code*. Kode QR adalah suatu jenis kode matriks atau kode batang dua dimensi dengan fungsionalitas utama yaitu dapat dengan mudah dibaca dan untuk menyampaikan informasi dengan cepat dan mendapatkan respons yang cepat pula. Berbeda dengan kode batang, yang hanya menyimpan informasi secara horizontal, kode QR mampu menyimpan informasi secara horizontal dan vertikal, oleh karena itu secara otomatis Kode QR dapat menampung informasi yang lebih banyak daripada kode batang.<sup>[7]</sup>



**Gambar 2.5 PDF417<sup>[17]</sup>**

## 2.8 QR Code

Kode QR adalah suatu jenis kode matriks atau kode batang dua dimensi yang dikembangkan oleh *Denso Wave*, sebuah divisi *Denso Corporation* yang merupakan sebuah perusahaan Jepang dan dipublikasikan pada tahun 1994 dengan fungsionalitas utama yaitu dapat dengan mudah dibaca oleh pemindai QR merupakan singkatan dari quick response atau respons cepat, yang sesuai dengan tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respons yang cepat pula. Berbeda dengan kode batang, yang hanya menyimpan informasi secara horizontal, kode QR mampu menyimpan informasi secara horizontal dan vertikal, oleh karena itu secara otomatis Kode QR dapat menampung informasi yang lebih banyak daripada kode batang.<sup>[5]</sup>

### 2.6.1 Fungsi QR Code

Kode QR berfungsi bagaikan hipertaut fisik yang dapat menyimpan alamat dan URL, nomer telepon, teks dan sms yang dapat digunakan pada majalah, surat harian, iklan, pada tanda-tanda bus, kartu nama ataupun media lainnya. Atau dengan kata lain sebagai penghubung secara cepat konten daring dan konten luring.<sup>[5]</sup> Kehadiran kode ini memungkinkan audiens berinteraksi dengan media yang ditempelinya melalui ponsel secara efektif dan efisien. Pengguna juga dapat menghasilkan dan mencetak sendiri kode QR untuk orang lain dengan mengunjungi salah satu dari beberapa ensiklopedia kode QR.<sup>[4]</sup>

#### 1. Kepentingan Komersial

Selain itu, desain QR memungkinkan pengguna untuk memasukkan logo perusahaan, klip video ataupun foto ke kode QR, tanpa menghilangkan substansi informasi apapun dari sumber yang dimasukkan. Contoh penggunaan kode QR yang di dalamnya memuat konten klip video adalah kode QR yang digunakan oleh kelompok penyanyi dari Inggris bernama *Pet Shop Boys* pada tahun 2007. Ketika kode dipindai dengan benar, maka pengguna akan diarahkan ke situs *Pet Shop Boys*. Selain itu pada tahun 2009 kode QR digunakan untuk kampanye pemasaran *Movie 9* di *San Diego Comic Con*. Pada saat itu, pelanggan diberikan

kartu yang menampilkan kode QR yang telah terintegrasi dengan karya seni yang bersangkutan. Jadi, pelanggan dapat mengakses cuplikan film melalui kode QR tersebut.<sup>[4]</sup>

## 2. Kepentingan Umum

Kode QR dapat dimanfaatkan sebagai keamanan makanan dengan cara menambahkan kode QR yang berisikan data-data mengenai kandungan nutrisi dan masa kadaluarsa pada tiap label makanan sehingga pelanggan dapat merasa lebih aman dalam memilih makanan yang dibeli sebab mereka dapat mengetahui informasi-informasi tentang makanan tersebut. Di Jepang, hal ini telah diterapkan oleh *McDonald*. Terdapat 19 jenis "*sandwich*" yang diberi kode QR yang mengandung informasi alergi, jumlah kalori dan nutrisi yang terkandung dalam *sandwich* tersebut. Selain itu kode QR juga dapat diberikan di halte bus, sehingga penumpang dapat mengetahui keberadaan bus yang sedang ditunggu. Cara kerjanya adalah dengan memberikan hipertaut ke kamera CCTV di setiap jalan melalui koneksi internet pada ponsel. Lebih lanjut lagi, kode QR dapat dipasang pada kartu pelajar, sehingga akan mempermudah proses absensi siswa, dan mempermudah akses bagi para siswa, guru, dan orang tua murid kepada informasi proses belajar mengajar.<sup>[4]</sup>

### 2.6.2 Penggunaan *QR Code*

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

pengguna menggunakan teknologi kode QR sebagai penyembunyi pesan. Berikut adalah beberapa teknologi yang dapat digabung dengan teknologi kode QR:

1. *Tracking Number* (Memeriksa Nomor Resi / *Tracking* Suatu Barang)
2. *Transport Ticketing* (Nomor Id Penumpang)
3. Pesan Singkat
4. Nomor Telepon
5. *Email*
6. Sistem Penjualan atau Sistem Diskon
7. Alamat Website
8. Kartu Nama
9. *Product Number*
10. Dan lain-lain

### 2.6.3 Desain *QR Code*

Desain dari kode QR berbentuk kotak persegi dengan warna mendasar putih sebagai tampilan belakang layar atau background dan diisi dengan gabungan beberapa data berbentuk kotak persegi kecil atau berbentuk pixel berwarna hitam. Rumit atau tidaknya posisi pixel pada kode QR bergantung pada jumlah serta kombinasi data yang terkandung di dalamnya. Adapun data yang dapat disimpan dalam kode QR berupa nomor, huruf, *alphanumeric*, *byte*, *symbol* dan bahkan kanji dapat dimasukkan kedalam *QR Code*.<sup>[5]</sup>

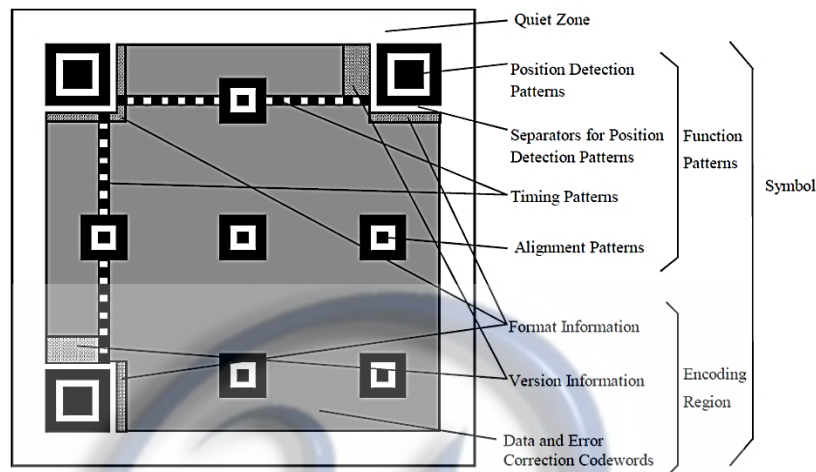


**Gambar 2.6 Contoh Simbol *QR Code***<sup>[18]</sup>

Seperti pada Gambar 2.6, terdapat pola struktur tersendiri ketika membuat kode QR. Dalam pembuatannya ukuran kode QR harus dibuat seimbang atau sama sehingga membentuk persegi. Di dalam *QR Code* terdapat beberapa pola



dimana pola tersebut tidak boleh dilakukan *encoding* data. Selain itu symbol harus dikelilingi oleh zona garis putih atau yang biasa disebut *Quiet Zone*.<sup>[6]</sup> Berikut struktur dari kode QR:



**Gambar 2.7 Struktur QR Code<sup>[6]</sup>**

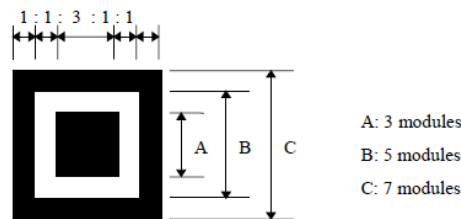
### 1. *Symbol Version and Size*

Terdapat 40 ukuran dari kode QR yang disebut sebagai versi 1, versi 2, dst. Dari setiap versi tersebut memiliki kapasitas penyimpanan data yang berbeda. Semakin besar data yang disimpan maka semakin besar pula gambar kode QR yang terbentuk.<sup>[6]</sup>

### 2. *Finder Pattern*

*Finder Pattern* diletakan pada 3 buah titik khusus yang terletak di kanan atas, kiri atas dan kiri bawah gambar kode QR. Pola gambar dari *finder pattern* berbentuk persegi berukuran 7 x 7 warna gelap, ditambah 5 x 5 warna terang, dan ditambah 3 x 3 warna gelap. Sehingga perbandingan pola *finder pattern* yaitu 1:1:3:1:1 seperti yang diilustrasikan pada Gambar 2.8. Simbol ini sangat penting, dikarenakan simbol ini mendefinisikan orientasi serta lokasi pola dalam melakukan proses encoding.<sup>[6]</sup>





**Gambar 2.8 Pola Position Detection<sup>[6]</sup>**

### 3. *Separator*

Sebuah *separator* atau pembatas ditempatkan diantara masing-masing pola *position detection*. Format pola mengelilingi *position detection* dengan menggunakan warna terang.<sup>[6]</sup>

### 4. *Timing Pattern*

*Timing pattern* berada pada setiap ujung pola *position detection* yang memotong dari ujung pola *detection* satu ke ujung pola *detection* lainnya. Format pola dari pembatas membentuk sebuah garis yang berisikan nilai 1 dan 0 yang nilainya dimasukan secara bergantian.<sup>[6]</sup>

### 5. *Alignment Patterns*

Merupakan sebuah pola untuk menyelaraskan posisi dengan *finder pattern*. Ini bertujuan agar membuat beberapa blok bagian apabila data yang disimpan sangatlah banyak. *Alignment patterns* biasanya digunakan pada kode QR versi 2 ke atas.<sup>[6]</sup>

### 6. *Encoding Region*

Merupakan wilayah dari data yang dimasukan dan di-*encode* kedalam bentuk kode QR. Pada *encoding region* juga terdapat *Error Correction* yaitu sebuah proses pelacakan kesalahan selama transmisi data berlangsung.<sup>[6]</sup>

### 7. *Quiet Zone*

Merupakan sebuah zona dimana tidak ada pola atau *marking* yang terdapat pada zona tersebut. Zona ini berada pada 4 sisi gambar kode QR. Fungsi dari

*Quiet Zone* ini yaitu agar dapat menentukan zona kode pada kode QR. Apabila kode QR tidak memiliki *Quiet Zone* maka akan ada kemungkinan ketika pembacaan kode QR akan sulit terbaca bahkan tidak terbaca sedikitpun. <sup>[6]</sup>

#### 2.6.4 Kelebihan *QR Code*

Teknologi *QR Code* merupakan pengembangan dari teknologi *barcode*. Teknologi *barcode* banyak sekali kekurangan terutama dalam penyimpanan data sehingga dibuatlah teknologi *QR Code*.<sup>[5]</sup> Berikut merupakan beberapa *kelebihan* dari teknologi *QR Code*:

1. Kode QR memiliki kapasitas tinggi dalam data pengkodean, yaitu mampu menyimpan semua jenis data, seperti data numerik, data alfabitis, kanji, kana, hiragana, simbol, dan kode biner. Secara spesifik
2. Kode QR mampu menyimpan data jenis numerik sampai dengan 7.089 karakter, data alphanumerik sampai dengan 4.296 karakter, kode binari sampai dengan 2.844 byte, dan huruf kanji sampai dengan 1.817 karakter.
3. Kode QR memiliki tampilan yang lebih kecil daripada kode batang. Hal ini dikarenakan kode QR mampu menampung data secara horizontal dan vertikal, oleh karena itu secara otomatis ukuran dari tampilannya gambar kode QR bisa hanya sepersepuluh dari ukuran sebuah kode batang.
4. Kode QR juga tahan terhadap kerusakan, sebab kode QR mampu memperbaiki kesalahan sampai dengan 30%. Oleh karena itu, walaupun sebagian simbol kode QR kotor ataupun rusak, data tetap dapat disimpan dan dibaca. Tiga tanda berbentuk persegi di tiga sudut memiliki fungsi agar simbol dapat dibaca dengan hasil yang sama dari sudut manapun sepanjang 360 derajat