

BAB II

LANDASAN TEORI

2.1 Sistem

Menurut Sutarman (2009), dalam bukunya yang berjudul *Pengantar Teknologi Informasi* : “ Sistem adalah kumpulan elemen yang saling berinteraksi dalam suatu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama”.

Menurut Azhar Susanto (2013) dalam bukunya yang berjudul *Sistem Informasi Akuntansi* : “Sistem adalah kumpulan/group dari sub sistem/bagian/komponen apapun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu”.

2.2 Presensi

Menurut Hanif Al Fatta (2009) presensi karyawan adalah suatu kegiatan pencatatan terhadap setiap kehadiran karyawan dengan tujuan untuk mengetahui data yang berkaitan dengan kehadiran karyawan secara periodik baik harian maupun bulanan.

2.3 Quick Response Code

Quick Response Code atau yang biasa disingkat *QR Code* adalah salah satu jenis kode batang dua dimensi yang dikembangkan *Denso Wave*. *Denso Wave* merupakan sebuah divisi di perusahaan Denso Corporation Jepang dan pertama kali dipublikasikan pada tahun 1994. *QR Code* merupakan bentuk evolusi kode batang. Tidak seperti *Barcode* yang hanya memiliki satu sisi yang berisi data, *QR Code* memiliki dua sisi yang dapat menampung data sehingga membuat *QR Code* dapat menampung data yang lebih besar ketimbang *Barcode*.

Menurut Rahmawati dan Rahman (2011) *QR Code* merupakan teknik yang mengubah data tertulis menjadi kode-kode 2 dimensi yang tercetak kedalam suatu media yang lebih ringkas. Dengan *QR Code*, informasi keaslian dokumen

tersebut dibuat menjadi lebih sederhana atau simpel tanpa menyetikkan informasi kode validasi pada dokumen tersebut

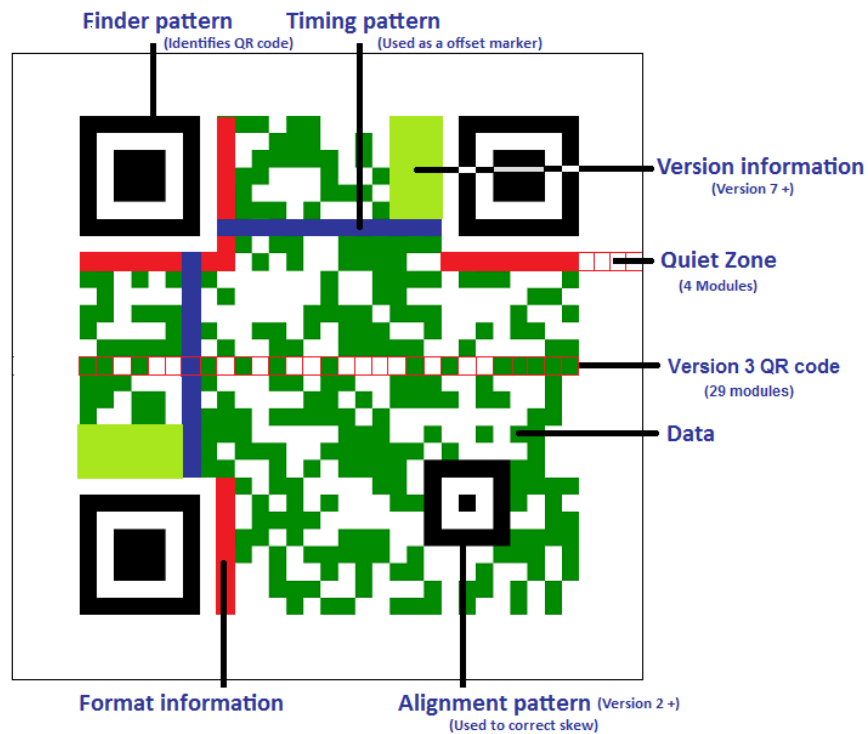
Untuk membaca suatu *QR Code* kita memerlukan alat yang disebut *QR Code Scanner*. Alat ini pada umumnya tersedia di dalam smartphone dalam bentuk aplikasi. Tujuan dari pembuatan *QR Code* itu sendiri adalah untuk mempermudah pengguna mendapatkan informasi yang ada pada *QR Code*, misalnya hanya dengan menscan *QR Code* suatu website yang berisikan URL web tersebut pengguna dapat mengaksesnya tanpa perlu menyetikkan alamat URL website.



Gambar 2.1 Contoh QR Code: “Rancang Bangun Sistem Informasi Presensi Mahasiswa Menggunakan QR Code Berbasis Android”

2.3.1 Struktur *QR Code*

1. *Finder pattern* sebagai identifikasi letak *QR Code*.
2. *Timing pattern* pola yang berfungsi untuk identifikasi koordinat pusat *QR Code*
3. *Version Information* menunjukkan versi dari sebuah *QR Code*.
4. *Quiet Zone* bagian terluar dari *QR code* berfungsi untuk mempermudah mengenali pengenalan *QR* oleh sensor *CCD*.
5. *QR Code version* versi *QR Code* yang digunakan.
6. *Data Area* adalah data dari *QR Code* yang di *encode* di dalam data area.
7. *Alignment pattern* pola yang berfungsi mengoreksi distorsi dari *QR Code*.
8. *Format Information* sebagai informasi tentang *error correction level* dan *mask pattern*.



Gambar 2.2 Struktur QR Code

2.3.2 Kelebihan QR Code

QR Code mampu menyimpan semua jenis data seperti data numerik, alfabitis, kanji, kana hiragana, symbol, dan kode biner. QR Code mampu menyimpan data jenis binary hingga 2.884 byte, numeric hingga 7.089 karakter, alphanumeric 4.296 karakter, dan huruf kanji hingga 1.817 karakter. QR Code juga dapat mengoreksi kesalahan hingga 30% dan juga QR Code dapat menampung jumlah data yang sama dengan Barcode ID dan tidak memerlukan ruang besar karena QR Code dapat menampung data secara horizontal dan vertical.

2.4 Android

Menurut Teguh Arifianto (2011:1), Android merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang berbasis linux. Menurut Hermawan (2011:1), Android merupakan OS (Operating System) mobile yang tumbuh di tengah OS lainnya yang berkembang dewasa ini.

Android memiliki empat karakteristik antara lain sebagai berikut:

1. Terbuka.

2. Pengembangan aplikasi yang cepat dan mudah.
3. Semua Aplikasi dibuat sama .
4. Memecahkan hambatan pada aplikasi.

2.4.1 Android Software Development Kit (SDK)

Android Software Development kit atau SDK adalah tools yang berfungsi untuk mengembangkan aplikasi berbasis Android. Android SDK ini terdiri dari library, debugger, emulator, dokumentasi, kode sumber, dan tutorial. Saat ini Android sudah mendukung arsitektur x86 pada Linux (distribusi Linux apapun untuk desktop modern), Mac OS X 10.4.8 atau lebih, Windows XP atau Vista. Persyaratan mencakup JDK, Apache Ant dan Python 2.2 atau yang lebih baru. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan plugin Android Development Tools (ADT), dengan ini pengembang dapat menggunakan teks editor untuk mengedit file Java dan XML serta menggunakan peralatan command line untuk menciptakan, membangun, melakukan debug aplikasi Android dan pengendalian perangkat Android (misalnya, reboot, menginstal paket perangkat lunak dengan jarak jauh).

2.4.2 Android Development Tools (ADT)

Android Development Tools atau ADT adalah plugin untuk Eclipse yang didesain untuk pengembangan aplikasi Android. ADT memungkinkan Eclipse untuk digunakan dalam membuat aplikasi Android baru, membuat User Interface, menambahkan komponen berdasarkan framework API Android, debug aplikasi, dan pemaketan aplikasi Android.

2.4.3 Android Virtual Device

Dalam pengembangan aplikasi Android kita memerlukan Android Virtual Device sebagai emulator perangkat Android yang berjalan pada komputer kita. Jadi dengan Android Virtual Device kita bisa menjalankan aplikasi yang kita kembangkan tanpa perlu menggunakan perangkat Android yang sebenarnya.

2.5 Java

Menurut Sun Microsystem, di dalam buku M. Shalahuddin dan Rosa A.S. (2010) Java adalah namasekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (standalone) ataupun pada lingkungan jaringan. Java adalah suatu bahasa pemrograman yang berorientasi objek dan dapat dijalankan di berbagai platform.

2.6 NodeJS

Node.js merupakan suatu perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis dalam sintaks bahasa pemrograman JavaScript. Selama ini JavaScript sebagai bahasa pemrograman yang berjalan di sisi client / browser saja, maka Node.js ada untuk melengkapi peran JavaScript sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi server, seperti halnya PHP, Ruby, Perl, dan sebagainya. Node.js dapat berjalan di sistem operasi Windows, Mac OS X dan Linux tanpa perlu ada perubahan kode program. Node.js memiliki pustaka server HTTP sendiri sehingga memungkinkan untuk menjalankan server web tanpa menggunakan program server web seperti Apache atau Nginx.

Untuk mengeksekusi Javascript sebagai bahasa server diperlukan engine yang cepat dan mempunyai performansi yang bagus. Engine Javascript dari Google bernama V8 yang dipakai oleh Node.js yang juga merupakan engine yang dipakai oleh browser Google Chrome.

2.7 MongoDB

MongoDB merupakan sistem basis data berorientasi dokumen lintas platform. *MongoDB* diklasifikasikan sebagai basis data *NoSQL*, *MongoDB* menghindari struktur basis data relasional tabel berbasis tradisional yang mendukung *JSON* seperti dokumen dengan skema dinamis (*MongoDB* menyebutnya sebagai format *BSON*), membuat integrasi data dalam beberapa jenis aplikasi lebih mudah dan lebih cepat. Dirilis di bawah kombinasi dari *GNU Affero General Public License* dan Lisensi *Apache*, *MongoDB* adalah perangkat lunak bebas dan sumber terbuka.

2.8 Basis Data

Menurut Abdul Kadir (2013) Database adalah suatu pengorganisasian data dengan tujuan agar data dapat dimanipulasi dan diperoleh dengan cepat dan mudah. Terkait dengan database, android menyediakan SQLite. Database tersebut bersifat relasional. Artinya data dalam database disusun dalam bentuk table, yang terdiri atas baris dan kolom. Suatu database hanya dapat digunakan oleh aplikasi yang menciptakannya. Secara umum, operasi yang dilakukan di database meliputi:

1. Pembuatan database
2. Pembukaan database
3. Penyisipan data di tabel
4. Pengubahan data di table
5. Penghapusan data di table
6. Penutupan database

2.9 API

API (Application Programming Interface) adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi lain.

2.10 Unified Modeling Language (UML)

Menurut Verdi Yasin, (2012) mendefinisikan *Unified Modelling Language* (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak, UML menawarkan sebuah standar untuk merancang model sebuah sistem. Tujuan Penggunaan UML yaitu untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin. Menurut Verdi Yasin (2012) tipe-tipe Diagram UML adalah sebagai berikut.

a. Use Case Diagram

Use Case Diagram adalah gambar dari beberapa atau seluruh aktor dan *use*

case dengan tujuan yang mengenali interaksi mereka dalam suatu sistem. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara *actor* dan sistem.

Komponen-komponen relasi dalam *use case* antara lain :

1. *Association*, menghubungkan *link* antar element.
2. *Generalization*, disebut juga *inheritance* (pewaris), sebuah elemen dapat merupakan spesialis dari elemen lainnya.
3. *Dependency*, sebuah element bergantung dalam beberapa cara ke elemen lainnya.
4. *Aggregation*, bentuk *association* dimana sebuah elemen berisi elemen lainnya.

Dalam *use case diagram* terdapat istilah seperti aktor, *use case* dan *case relationship*.

b. *Sequence Diagram*

Sequence diagram menggambarkan kolaborasi dinamis antara sejumlah dan untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. *Sequence diagram* menjelaskan interaksi objek yang disusun berdasarkan urutan waktu.

Dalam *sequence diagram* terdapat 2 simbol yaitu :

1. *Actor*, untuk menggambarkan pengguna sistem.
2. *Lifeline*, untuk menggambarkan kelas dan objek.

c. *Class Diagram*

Class diagram menggambarkan struktur data dan deskripsi *class*, *package*, dan objek beserta hubungan satu sama lain. *Class diagram* berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain.

Class memiliki 3 area pokok antara lain :

1. Nama
2. Atribut
3. Metode

Class menggambarkan keadaan (*attribute/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut.

d. *Activity Diagram*

Activity diagram menggambarkan rangkaian aliran dari aktifitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi. *Activity Diagram* berupa *flowchart* yang digunakan untuk memperlihatkan aliran kerja dari sistem. Notasi yang digunakan dalam *activity diagram* adalah sebagai berikut :

1. *Activity*

Notasi yang menggambarkan pelaksanaan dari beberapa proses dalam aliran pekerjaan.

2. *Transition*

Notasi yang digunakan untuk memperlihatkan jalan aliran kontrol dari *activity* ke *activity*.

3. *Decision*

Notasi yang menandakan kontrol cabang aliran berdasarkan *decision point*.

4. *Synchronization bars*

Aliran kerja notasi ini menandakan bahwa beberapa aktivitas dapat diselesaikan secara bersamaan.

2.11 Text Editor

Text editor adalah suatu software atau aplikasi perangkat lunak untuk membuat atau mengedit teks. Dalam hal ini kita menggunakan teks editor untuk membuat script PHP. Ada banyak text editor yang bisa digunakan untuk membuat script misalnya, Notepad ++. Adobe Dreamweaver, Sublime, ataupun aplikasi text bawaan windows yaitu notepad.