

BAB II LANDASAN TEORI

2.1 Pengertian Aplikasi

Menurut **Jogiyanto** aplikasi merupakan penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*. Berdasarkan jenisnya, aplikasi komputer dapat dibagi menjadi beberapa kategori, yaitu :

1) *Enterprise*

Digunakan untuk organisasi yang cukup besar dengan maksud menghubungkan aliran data dan kebutuhan informasi antar bagian, contoh : *IT Helpdesk, Travel Management* dan lain-lain.

2) *Enterprise (SupPort)*

Sebagai aplikasi pendukung dari *Enterprise*, contohnya : *Database Management, Email Server dan Networking System*.

3) *Individual Worker*

Sebagai aplikasi yang biasa digunakan untuk mengolah/edit data oleh tiap individu. Contoh : *Ms.Office, Photoshop, Acrobat Reader* dan lain-lain.

4) Aplikasi Akses Konten

Aplikasi Akses Konten Adalah aplikasi yang digunakan oleh individu (hanya) untuk mengakses konten tanpa kemampuan untuk mengolah atau mengedit datanya melainkan hanya melakukan kustomisasi terbatas. Contoh : *Games, Media Player, Web Browser*.

5) Aplikasi Pendidikan

Biasanya berbentuk simulasi dan mengandung konten yang spesifik untuk pembelajaran.

6) Aplikasi Simulasi

Biasa digunakan untuk melakukan simulasi penelitian, pengembangan dan lain-lain. Contoh : Simulasi pengaturan lampu lalu lintas.

7) Aplikasi Pengembangan Media

Aplikasi ini berfungsi untuk mengolah/mengembangkan media biasanya untuk kepentingan komersial, hiburan dan pendidikan. Contoh : *Digital Animation Software*, *AudioVideo Converter* dan lain-lain.

8) Aplikasi Mekanika dan Produk

Aplikasi ini dibuat sebagai pelaksana/pengolah data yang spesifik untuk kebutuhan tertentu. Contoh : *Computer Aided Design (CAD)*, *Computer Aided Engineering (CAE)*, *SPSS* dan lain-lain.

2.2 QR Code

QR-Code(Quick Response code) merupakan salah satu bentuk enkripsi data yang dicetuskan pertama kali oleh Denso Wave yaitu sebuah perusahaan jepang dan dipublikasikan pada tahun 1994 (Bachtiar, 2012; Wave, 2015). *QR Code* berupa gambar dua dimensi (2D) awalnya diusulkan untuk digunakan pada industri *manufacture*, *mobile marketing* dan perusahaan *retailing*. Perkembangan teknologi *smartphone* membuat penggunaan *QR Code* semakin meluas dan digunakan dalam *enkripsi kode pasport*, *visa* dan *Id card* (Trujillo A.E., dkk. 2012), sosialisasi produk dan iklan *mobile*, termasuk *link Web*, *mask apaiboarding pass* (Wave, 2015) memfasilitasi *inventory control*, penataan saham, dan *checkout* (Mostafa, 2015) dan sebagai langkah-langkah yang efisien dan efektif menghubungkan perusahaan dengan pelanggan maka *QR Code* dapat discan oleh ponsel, kemudian terhubung dengan web dan tentunya menyediakan konten pengguna akhir (Huang, dkk, 2012). *QR Code* sebagai metode cepat dalam menyebarkan informasi dapat digunakan untuk menggantikan modul pembelajaran, menyimpan informasi film atau video dan ini merupakan langkah positif baik dan cerdas (Gummesson, 2015). Kemampuan *QR-code* untuk menampung informasi yang banyak walaupun dalam ukuran yang kecil. *QR Code* mampu menampung banyak informasi dengan 7.089 karakter numerik dan 4296 karakter *alfanumerik* (Narayanan, 2012). Pemindaian *QR Code* juga dapat digunakan sebagai sistem keamanan untuk tempat tertentu yang boleh dimasuki orang-orang tertentu seperti ruang riset ataupun ruang pusat (Supriyono dkk, 2013).

2.3 Teknik *Labelling*

Teknik *Labelling* adalah teknik penamaan barang menggunakan label QR Code. (JSIKA, 2016) menjelaskan bahwa “ *QR Code* adalah pengembangan kode bar yang biasanya terdiri atas garis-garis dengan ketebalan yang berbeda”.



Gambar 2.1 Contoh *QR Code*

QR Code memiliki beberapa kelebihan dibandingkan barcode traditional, (Ariska, 2016) kelebihan tersebut antara lain:

1. Mampu menyimpan data tersandi dalam kapasitas besar *QR Code* mampu menyandikan berbagai macam tipe data seperti *numeris*, karakter, *Kanji*, *Hiragana*, simbol, biner, bahkan mampu menyandikan 7089 karakter hanya dalam satu symbol.
2. *QR Code* mampu menyandikan data hanya dengan membutuhkan sepersepuluh ruangan yang dibutuhkan oleh *barcode* biasa.
3. Mampu membaca pada arah manapun (360 derajat) *QR Code* mampu dibaca dalam berbagai arah (*omni direksional*) secara cepat.

2.4 Manajemen Aset

Menurut Utomo, 2010 menjelaskan bahwa: Manajemen aset merupakan proses pengorganisasian, perencanaan, dan pengawasan terhadap pembelian, penggunaan, perawatan, perbaikan, dan atau penghapusan aset fisik untuk mengoptimalkan potensi *service delivery* dan meminimalkan resiko atau *cost* yang

berkaitan dengan usia hidup aset dengan menggunakan aset-aset *intangible* seperti aplikasi pengambilan keputusan berbasis *knowledge* dan proses bisnis. *Asset* (Aset) adalah barang, yang dalam pengertian hukum disebut benda, yang terdiri dari benda tidak bergerak dan benda bergerak, baik yang berwujud (*tangible*) maupun yang tidak berwujud (*Intangible*), yang tercakup dalam aktiva/kekayaan atau harta kekayaan dari suatu instansi, organisasi, badan usaha atau individu perorangan. Sedangkan Management artinya adalah pengelolaan, dan ini berasal dari kata kerja *to manage* yang artinya mengurus, mengatur, melaksanakan, memperlakukan, dan mengelola (Ariska & Jazman , 2016) . Untuk kebutuhan staf Biro Mall maka manajemen asset ini hanya dibatasi pada:

- 1) Penatausahaan
- 2) Pemanfaatan
- 3) Pengamanan dan pemeliharaan
- 4) Perencanaan kebutuhan dan penganggaran (pembiayaan); dan
- 5) *Monitoring* dan evaluasi.

2.5 Android

2.5.1 Pengertian Android

Satyaputra (2014.p.28) mendefinisikan bahwa, Android adalah sebuah sistem operasi untuk *smartphone* dan Tablet. Sistem operasi dapat diilustrasikan sebagai ‘jembatan’ antara piranti (*device*) dan penggunanya, sehingga pengguna bisa berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device*. Di dunia personal komputer, sistem operasi yang banyak dipakai adalah *Windows*, *Mac*, dan *Linux*.

Android adalah sistem operasi yang bersifat *open source* (sumber terbuka). Disebut *open source* karena *source code* (kode sumber) dari sitem operasi android dapat dilihat, di-*download*, dan dimodifikasi secara bebas. Paradigma *open source* ini memudahkan pengembangan teknologi android, karena semua pihak yang tertarik dapat memberikan kontribusi, baik pada pengembangan sistem operasi maupun aplikasi. Dikutif dari buku yang berjudul *Beging Android Programming with Adt Bubdle* yang ditulis oleh Alfa Satyaputra tahun 2014, Pada awal oktober 2013 tercatat ada lebih dari 850,000 aplikasi

android yang tersedia di google play (dulu bernama android market).Penjelasan Jenis Jenis Android pada tabel 2.1.

Table 2.1 Jenis Jenis Android

<i>Codename</i>	Tanggal	Versi
Tanpa nama	Februari 2008	1.0.1.1
Cupcake	Maret 2009	1.2 - 1.5
Donut	September 2009	1.6
Éclair	Oktober 2009	2.0 - 2.1
Froyo	Mei 2010	2.2 - 2.2.3
Gingerbread	desember 2010	2.3 - 2.4
Honeycomb	Februari 2011	3.0 - 3.2
Ice Cream Sandwich	19 October 2011	4.0
Jelly Bean	27 Juni 2012	4.1 - 4.3
KitKat	oktober 2013	4.4
Lollipop	November 2014	5.0
Marshmallow	Agustus 2015	6.0
Nougat	Agustus 2016	7.0
Oreo	Maret 2017	8.0

2.5.2 Fitur-fitur Android

Android akan memuat pada layar depan (*homescreen*), yakni navigasi utama dan pusat informasi pada perangkat, serupa dengan desktop pada komputer pribadi. Layar depan Android biasanya terdiri dari ikon aplikasi dan *widget*; ikon aplikasi berfungsi untuk menjalankan aplikasi terkait, sedangkan *widget* menampilkan konten secara langsung dan terbaru otomatis, misalnya prakiraan cuaca, kotak masuk surel pengguna, atau menampilkan stiker berita secara langsung dari layar depan. Layar depan bisa terdiri dari beberapa halaman, pengguna dapat menggeser bolak balik antara satu halaman ke

halaman lainnya, yang memungkinkan pengguna Android untuk mengatur tampilan perangkat sesuai dengan selera mereka. Beberapa aplikasi pihak ketiga yang tersedia di *Google Play* dan di toko aplikasi lainnya secara ekstensif mampu mengatur kembali tema layar depan Android, dan bahkan bisa meniru tampilan sistem operasi lain, misalnya *Windows Phone*. Kebanyakan produsen telepon seluler dan operator nirkabel menyesuaikan tampilan perangkat Android buatan mereka untuk membedakannya dari pesaing mereka. Di bagian atas layar terdapat status bar, yang menampilkan informasi tentang perangkat dan konektivitasnya. Status bar ini bisa "ditarik" ke bawah untuk membuka layar notifikasi yang menampilkan informasi penting atau pembaruan aplikasi, misalnya surel diterima atau *SMS* masuk, dengan cara tidak mengganggu kegiatan pengguna pada perangkat.

2.5.3 Kelebihan Android

Android memiliki beberapa kelebihan diantaranya adalah :

1) *User Friendly*

Kalimat ini sangat melekat pada sistem operasi Windows milik microsoft, ibaratnya kita dengan sangat mudah mengoperasikan komputer hanya dengan belajar beberapa hari bahkan beberapa jam saja, dan ini juga melekat pada *Android* yang berjalan pada *Smartphone*.

2) *Notifications*

Kita dengan sangat mudah mendapatkan notifikasi dari *smartphone android* dengan mengatur beberapa akun *Email*, *SMS*, *Voice Dial*, *Update* dan lain sebagainya.

3) Tampilan

Dari segi tampilan *Android* tidak kalah bagusya dari *iOs* milik *Apple*, karena memang dari awal *android* hampir mengusung teknologi *iOs* hanya saja ini versi murahya.

4) *Open Source*

Operating system ini memang dibuat *open source* oleh penciptanya, karena memang berbasis *kernel Linux*. Jadi jangan kaget jika diluar sana ada banyak sekali *Custom Rom* untuk masing-masing perangkat *android*.

5) Aplikasi

Untuk Aplikasi akan disajikan jutaan pilihan aplikasi yang menarik dari yang gratis hingga berbayar, dan kita bisa mendownloadnya di *Google Play*.

2.5.4 Kekurangan Android

Android memiliki beberapa kekurangan, diantaranya adalah

1) *Update System*

untuk melakukan *update system* tidaklah mudah kita harus menunggu dari masing-masing *vendor* untuk merilis *Update Versi* yang terbaru.

2) Baterai cepat habis

ini sering terjadi jika kita menyalakan paket data dan menggunakan *widget* serta aplikasi yang berjalan secara berlebihan, untuk mengatasinya kita harus mengurangi aktivitas aplikasi pada *smartphone*.

3) Lemot atau Lag

ini ada kaitannya dengan spesifikasi dari masing-masing perangkat, namun ada kalanya *Android* tidak bersahabat dengan beberapa *aplikasi* dikarenakan *Ram* ataupun prosesor yang kurang memadai, jadi sesuaikan aplikasi dengan perangkatnya.

2.6 Perangkat Lunak yang digunakan

Beberapa perangkat lunak yang digunakan dalam membangun aplikasi kamus tersebut, beberapa perangkat lunak yang digunakan adalah sebagai berikut:

2.6.1 *Android Software Development Kit*

SDK (*Software Development Kit*) adalah sebuah *set tools* untuk mengembangkan aplikasi untuk suatu software *package*, *framework* aplikasi, *platform hardware*, *console* permainan video, sistem operasi dan sejenisnya. Saat ini di sediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman java (Safaat, 2011.p.30).

Android SDK memiliki tools pengembangan yang cukup lengkap, yang memiliki antara lain debugger, libraries, *emulator* yang berbasis QEMU, dokumentasi, kode-kode contoh, dan tutorial. Platform pengembangan yang saat ini telah didukung antara lain komputer yang menggunakan Linux, Mac OS X, Windows XP hingga yang terbaru. IDE yang didukung secara resmi adalah Eclipse dengan menggunakan plugin *Android Development Tools* (ADT), meskipun demikian developer dapat menggunakan *editor teks* manapun untuk membuat *kode java dan XML* kemudian menggunakan *command line* untuk membuat, dan men-debug aplikasi android ataupun mengontrol perangkat android yang terhubung.

2.6.2 Android Studio

Android Studio adalah sebuah *Integrated Development Environment* (IDE) utama Google untuk mengembangkan pada platform Android. Karena Android Studio merupakan IDE dari Google, maka software ini dapat secara langsung terintegrasi dengan *Google Maps* menggunakan *API Key* yang dibuat di laman yang disediakan dari *Google Maps API* untuk mengintegrasikan peta dengan *software* sehingga peta akan secara otomatis ditampilkan di aplikasi yang dibuat. Selain terintegrasi dengan *Google Maps*, Android Studio juga dapat terintegrasi dengan database *SQLite Manager*, plugin untuk pengolahan dan penyimpanan informasi yang saling berkaitan untuk kemudian dibuat algoritma dari tiap data yang akan ditampilkan.

2.6.3 Firebase

Firebase Adalah *Backend as a Service* (BaaS) yang saat ini dimiliki oleh Google. Firebase merupakan solusi yang ditawarkan oleh Google untuk mempermudah pengembangan aplikasi mobile. Dua fitur menarik dari Firebase adalah *Firebase Remote Config* dan *Firebase Real Time Database*. Selain itu juga terdapat fitur pendukung untuk aplikasi yang memerlukan *push notification* yaitu *Firebase Notification Console*.

Firestore Database merupakan penyimpanan basis data *nonSQL* yang memungkinkan untuk menyimpan beberapa tipe data. Tipe data itu antara lain *String, Long, dan Boolean*. Data pada *Firestore Database* disimpan sebagai objek *JSON tree*. Tidak seperti basis data *SQL*, tidak ada tabel dan baris pada basis data *non-SQL*. Ketika ada penambahan data, data tersebut akan menjadi node pada struktur *JSON*. *Node* merupakan simpul yang berisi data dan bisa memiliki cabang-cabang berupa *node* lainnya yang berisi data pula. Proses pengisian suatu data ke *Firestore Database* dikenal dengan istilah *push*.

Selain *Firestore Database*, *Firestore* menyediakan beberapa layanan lainnya yang juga dimanfaatkan dalam pengembangan aplikasi ini. Layanan tersebut antara lain *Firestore Authentication, Storage, dan Cloud Messaging*. Pada pengembangan aplikasi, layanan lainnya yang digunakan pada pengembangan aplikasi adalah *Firestore Storage*. Layaknya sebuah penyimpanan awan, *Firestore Storage* memungkinkan pengembang untuk mengunggah atau mengunduh sebuah berkas. Pada pengembangan aplikasi.

2.6.4 Android SDK Manager dan AVD

Safaat dan Nazarudin (2011.p.40), Menguraikan bahwa *Android SDK Manager* adalah sebuah alat pengembangan perangkat lunak yang berguna untuk mengembangkan dan membuat aplikasi untuk *platform Android*. Didalam *Android SDK* terdapat *project sample* dengan kode sumber, alat-alat untuk membuat aplikasi, sebuah emulator *Android (AVD)*, dan *library* yang dibutuhkan untuk membangun aplikasi *Android*. Aplikasi yang ditulis pada bahasa *Java* dan berjalan di *Dalvik*, mesin virtual yang dirancang khusus untuk penggunaan *embedded* yang berjalan diatas *kernel Linux*.

2.6.5 Library Zxing (Zebra Crossing)

Library Zebra crossing atau sering juga disebut dengan *zxing* adalah sebuah *library open source* yang berfungsi untuk pemrosesan berbagai jenis *barcode 1 dimensi dan 2 dimensi*. *Library zxing* ini berfokus kepada penggunaan

kamera pada smartphone sebagai pengganti dari alat scanner, namun *library* ini juga dapat dipakai pada desktop dan server. *Zxing* dapat melakukan proses *encode* maupun *decode barcode*. *Library zxing* mendukung berbagai macam format *barcode* antara lain upc-a, upc-e, ean-8, ean-13, code 39, *data matrix*, *QR Code*, dan *code bar*.

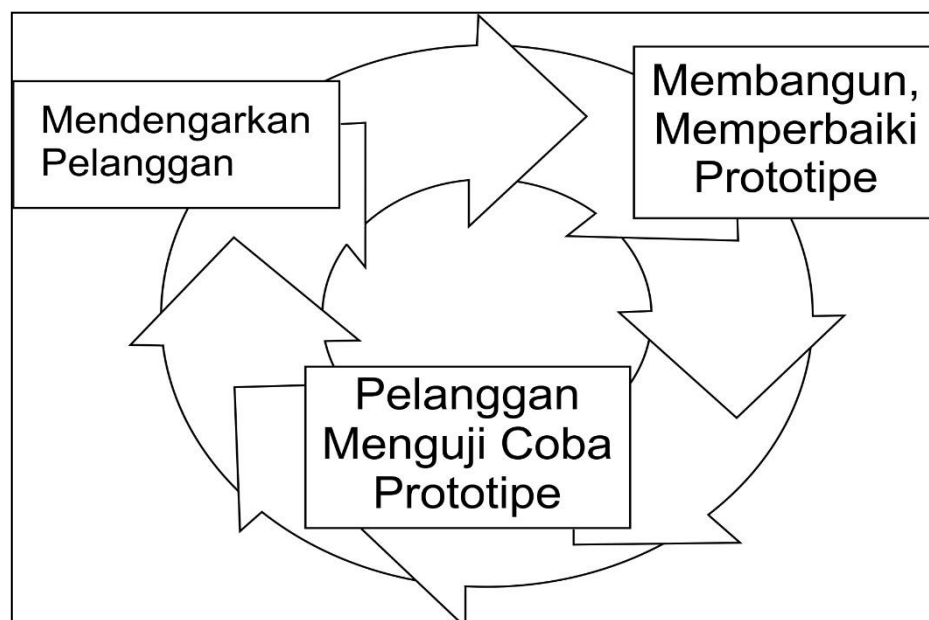
2.6.6 JSON (JavaScript Object Notation)

JSON merupakan sebuah bahasa pertukaran data yang dapat dengan mudah dibaca oleh manusia dan mudah untuk di-parsing dan digunakan oleh komputer. JSON sangat cocok digunakan dalam bahasa pemrograman java script. JSON memberikan performa yang lebih baik dibandingkan dengan XML. performa yang dimiliki JSON diyakini 100 kali lebih cepat dibandingkan dengan XML.

2.7 Metode Pengembangan Perangkat Lunak

2.7.1 Metode *Prototype*

Pressman (2012, p.40) menyatakan bahwa *Prototyping Paradigma* dimulai dengan pengumpulan kebutuhan, pengembang bertemu dengan pengguna dan mengidentifikasi objektif keseluruhan dari perangkat lunak, selanjutnya mengidentifikasi segala kebutuhan yang diketahui secara garis besar di mana definisi-definisi lebih jauh merupakan keharusan, kemudian dilakukan perancangan kilat, lalu diakhiri dengan evaluasi *prototyping*. Tahap-tahap rekayasa *software* dalam *prototype model* adalah seperti pada gambar 2.1 berikut.



r 2.2 Model Prototype.

Tahapan dalam metode prototype adalah sebagai berikut:

1) Pengumpulan kebutuhan

Developer dan *klien* bertemu untuk menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan.

Selanjutnya peneliti akan melakukan analisis terhadap data apa saja yang dibutuhkan, seperti analisis terhadap sistem yang berjalan, analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, dan analisis kebutuhan materi pembelajaran.

2) Perancangan

Perancangan dilakukan dengan cepat dan rancangan mewakili semua aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*. Dalam tahap ini peneliti akan membangun sebuah versi *prototype* yang dirancang kembali dimana masalah-masalah tersebut diselesaikan.

3) Evaluasi *prototype*

Pada tahap ini, calon pengguna mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan *software*. *Software* yang sudah jadi dijalankan dan akan dilakukan perbaikan apabila kurang memuaskan. Perbaikan termasuk dalam memperbaiki kesalahan atau kerusakan yang tidak ditemukan pada langkah sebelumnya.

Kelebihan *Prototype Model* adalah:

- 1) *End user* dapat berpartisipasi aktif.
- 2) Penentuan kebutuhan lebih mudah diwujudkan.
- 3) Mempersingkat waktu pengembangan *software*.

Kekurangan *Prototype Model* adalah:

- 1) Proses analisis dan perancangan terlalu singkat.
- 2) Mengesampingkan alternatif pemecahan masalah.

- 3) Biasanya kurang fleksibel dalam menghadapi perubahan.
- 4) *Prototype* yang dihasilkan tidak selamanya mudah dirubah.
- 5) *Prototype* terlalu cepat selesai.

2.7.2 Unified Modeling Language (UML)

Yasin (2012,p.194) mendefinisikan bahwa *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.

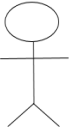
Yasin (2012,p.268) menguraikan bahwa tipe-tipe diagram UML adalah sebagai berikut :



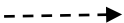
1) Use Case Diagram

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* merepresentasikan sebuah interaksi antara actor dan sistem.

Dalam *use case* diagram terdapat istilah seperti aktor, *use case* dan *case relationship*. Penjelasan simbol *Use Case* pada tabel 2.2.

Tabel 2.2 Simbol *Use Case*.




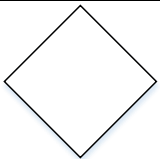
Simbol	Keterangan
	Aktor : Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan.

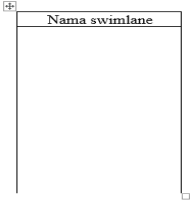
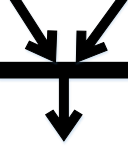
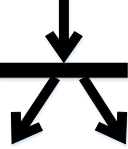
	<i>Use case</i> : perangkat tertinggi dari fungsionalitas yang dimiliki sistem.
	<i>Association</i> : adalah relasi antara aktor dan <i>use case</i> .
	<i>Generalisasi</i> : untuk memperlihatkan struktur pewaris yang terjadi.

2) 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 *flow chart* yang digunakan untuk memperlihatkan aliran kerja dari sistem. Notasi yang digunakan dalam *activity diagram* adalah sebagai berikut : Penjelasan simbol *Activity Diagram* pada tabel 2.3.

Tabel 2.3 Simbol *Activity Diagram*.

Simbol	Keterangan
	<i>Activity</i> : Menunjukkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Initial Node</i> : Bagaimana objek dibentuk atau diawali
	<i>Activity Final Node</i> : Bagaimana objek dibentuk dan diakhiri.
	<i>Decision</i> : Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.

	<p><i>Swimlane</i> : Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.</p>
	<p><i>Join</i> : Digunakan untuk menunjukkan kegiatan yang digabungkan.</p>
	<p><i>Fork</i> : Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel</p>

3) *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.

4) *Class Diagram*

Class diagram menggambarkan distruktur data dan desripsi *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 yaitu nama, atribut dan metode.

2.8 Penelitian Terdahulu

Tabel 2.4 berikut merupakan penelitian terdahulu yang telah dilakukan terkait dengan Teknik Labelling dan manajemen Aset.

Tabel 2.4 Penelitian Terdahulu

Nama	Judul	Terbit/Tahun	Penelitian Sebelumnya	Penelitian yang digunakan
Mita Pramihapsari dan Messa Prima Kaldera	Perancangan Labelling Pada Dokumen	Sekolah Tinggi Sandi Negara, Vol 20, 2012	Pada saat ini sistem persuratan yang masuk pada suatu instansi masih menggunakan sistem lama yaitu surat yang berklasifikasi biasa, rahasia dan sangat rahasia masuk dan di periksa terlebih dahulu oleh bagian tata usaha instansi tersebut.	Aplikasi yang sedang penulis buat yaitu membuat aplikasi yang lebih fleksibel atau lebih mudah digunakan oleh user dalam pendataan aset.
Akhmad Qashlim, Hasruddin	Implementasi Teknologi QR-Code Untuk Kartu Identitas	Universitas Al Asyariah Mandar, Vol 1, 2015	Id card yang menyimpan akses informasi yang sensitif dapat dimanfaatkan untuk berbagai hal, mengakses informasi rahasia, mendapatkan layanan keuangan, dan sebagainya. Beberapa mekanisme otentikasi id-card dapat dikembangkan untuk mengatasi masalah ini, seperti enkripsi informasi, image recognition, dan penggunaan barcode atau quick response code (QRCode).	Aplikasi yang penulis buat dapat menyimpan informasi data aset lebih aman karena menggunakan QR Code sehingga dapat meminimalisir kemungkinan redudansi data.
Julizal,Lukman, Imam Sunoto	Pengembangan QR Code Scanner Berbasis Android Dalam Penyampaian Informasi Absensi Siswa Menggunakan Short Message	Universitas Indraprasta PGRI, Jakarta Selatan	Sistem absensi ini menggunakan teknologi QR Code yang tersedia pada setiap handphone Android yang telah disediakan. Dalam prakteknya setiap siswa mempunyai kartu tanda siswa	Aplikasi yang dibuat menggunakan teknik labelling sehingga memudahkan user dalam mendata ataupun mengecek data aset yang

	Service		yang telah di tempeli QR Code.	sebelumnya sudah pernah diinputkan.
I Gusti Banjar Jawi, Heru Supriyono	Pemindaian QR Code Untuk Aplikasi Penampil Informasi Data Koleksi Di Museum Sangiran Sragen Berbasis Android	Jurusan Informatika Universitas Muhammadiyah Surakarta (UMS)	pada setiap ruangan koleksi pameran tidak memberikan informasi secara lengkap sedangkan permasalahan yang lain adalah sulitnya pengunjung yang ingin menyimpan informasi tersebut untuk dipelajari.	pada setiap ruangan koleksi pameran tidak memberikan informasi secara lengkap sedangkan permasalahan yang lain adalah sulitnya pengunjung yang ingin menyimpan informasi tersebut untuk dipelajari.
Joseph Dedy Irawan ,Emmalia Adriantantri	PEMANFAATAN QR-CODE SEGABAI MEDIA PROMOSI TOKO	Teknik Informatika, Institut Teknologi Nasional Malang	Selama ini dalam sistem penjualan yang sudah sangat umum dimanfaatkan oleh toko kecil bahkan toko besar adalah pemanfaatan Bar-Code sebagai ID dari suatu barang, dengan kode ini transaksi penjualan dapat berjalan dengan cepat dibandingkan dengan cara manual, akan tetapi selain Bar-Code ada lagi informasi yang dapat diperoleh dari QR-Code sebagai pemberi informasi alamat website. Akan tetapi banyak konsumen tidak bisa memanfaatkan informasi yang ada pada kode-kode tersebut.	aplikasi yang dapat membaca QR-Code agar pemilik toko dapat menyampaikan promo produk atau diskon dari toko dengan cepat dan mudah, dengan memanfaatkan QR-Code pelanggan hanya memerlukan kamera handphone untuk membaca informasi yang terdapat pada QR-Code dan dengan bantuan aplikasi yang harus diinstal sebelumnya .