

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Refrensi jurnal terkait yang digunakan dalam mendukung penelitian publikasi ilmiah dalam jurnal lokal.

Penelitian yang dilakukan oleh Ermatita, Rahmat Izwan Heroza, Miftahul Jannah (2017) dengan judul “Pengembangan Sistem Absensi Menggunakan QR *Code Reader* Berbasis *Android* (Studi Kasus : Fakultas Ilmu Komputer Jurusan Sistem Informasi)”. Salah satu teknologi telekomunikasi yang sekarang berkembang sangat pesat adalah ponsel pintar. Tren ini memberikan perspektif baru untuk pengembangan teknologi informasi dan komunikasi dalam beraktivitas. Salah satunya adalah istilah BYOD atau "*Bring Your Own Device* (Bawa Perangkat Anda Sendiri)". BYOD memunculkan ide penelitian, bagaimana konsep ini dapat diterapkan dan menciptakan sebuah sistem yang dapat membantu orang di universitas. QR singkatan dari *Quick Response* bertujuan untuk menyampaikan informasi dengan cepat dan mendapatkan respon yang cepat pula. Berbeda dengan *barcode*, yang hanya menyimpan informasi secara horizontal, QR *Code* dapat menyimpan informasi secara horizontal dan vertikal, dan oleh karena itu secara otomatis QR *Code* dapat menyimpan informasi lebih dari *barcode*.

Penelitian yang dilakukan oleh Nugraha Saputra (2016) dengan judul “Perancangan Aplikasi *Barcode* Scanner Pada Sistem Absensi CV. Pemindo Menggunakan *Android ZXING Library*”. Aplikasi yang semakin berkembang

pesat telah memberi berbagai manfaat diberbagai instansi. CV. Pemindo merupakan sebuah instansi swasta yang ada di kota Medan yang sistem absensi pegawainya masih dilakukan secara manual menggunakan sebuah catatan. Pada studi kali ini, penulis akan merancang sebuah aplikasi yang akan diterapkan di CV. Pemindo guna menggantikan sistem absensi manual ke arah sistem digital. Aplikasi ini dibangun dengan tingkat mobilitas tinggi yang memanfaatkan sebuah perangkat *mobile* sebagai media absensi. Aplikasi akan berfungsi sebagai *barcode* scanner atau pembaca *barcode* yang merupakan id card setiap pegawai. Dengan adanya aplikasi nantinya proses absensi pegawai akan dilakukan menggunakan perangkat *mobile* sebagai medianya. Hal ini akan lebih memudahkan proses absensi serta rekapitulasi absensi tersebut. Dalam implementasinya aplikasi juga akan memanfaatkan *Zxing Library* sebagai fungsi untuk membuat *scanner* pada perangkat *mobile*.

Penelitian yang dilakukan oleh Nurul Mawaddah (2017) dengan judul “Aplikasi *Mobile Scan* Parkir Berbasis *QR Code*”. Tidak tersedianya layanan parkir terkomputerisasi berdampak pada sulitnya membuat laporan keluar masuk kendaraan di kawasan Universitas Harapan Medan. Sistem parkir yang masih mengadopsi prosedur manual itu terkadang dapat merugikan pengendara dimana sangat mungkin sekali terjadi sebuah tindak pencurian. Untuk itu pada penelitian kali ini penulis memiliki ide untuk menghasilkan sebuah sistem parkir terkomputerisasi dengan memanfaatkan aplikasi *mobile* sebagai *scanner* dan teknologi *QR Code*. Penggunaan *QR Code* sebagai media parkir ini diharapkan dapat meningkatkan keamanan dalam pengawasan dan validasi sistem parkir.

Adapun sistem dibangun dengan bahasa pemrograman Java dan PHP serta MySQL sebagai databasenya.

Penelitian yang dilakukan oleh Bedi Prasetyo (2016) dengan judul “Analisa dan Perancangan Perangkat Lunak Absensi Karyawan Menggunakan QR Code Pada PT. Inti Buana”. Banyak cara yang digunakan untuk pengolahan absensi karyawan, salah satunya yaitu dengan menggunakan mesin *barcode*. Tujuan dari teknologi QR Code (*Quick Response Code*) di PT.Inti Buana Adalah sebagai alat bantu dalam melakukan kegiatan pengolahan data karyawan, kartu identitas karyawan, dan data absensi karyawan ,selain itu teknologi ini dapat bermanfaat bagi karyawan karena dapat melakukan kegiatan absensi secara terkomputerisasi..

Penelitian yang dilakukan oleh Norhikmah, Azizah Rahma Safitri, Laili Annas Solikhan (2016) dengan judul “Penggunaan QR Code Dalam Presensi Berbasis Android”. Presensi dalam kegiatan pelatihan menjadi suatu keharusan, yang dimana akan menjadi bagian dari laporan kegiatan pelatihan. Pelatihan di AMCC masih menggunakan presensi secara manual yaitu peserta/anggota membubuhkan tanda tangan pada suatu daftar hadir yang telah disediakan menggunakan kertas,pada saat pelatihan, dan diedarkan satu persatu keanggota/peserta sehingga dapat mengganggu konsentrasi peserta saat mendengarkan penjelasan si pameri/tentor. Maka dari itu dibutuhkan Qr Code dalam sistem presensi berbasis *android*. Hasil penelitian ini adalah dengan menggunakan QR Code presensi menjadi lebih cepat dan efisien dibanding presensi secara manual, sehingga rekapitulasi laporan pun menjadi lebih cepat.

II.2. Landasan Teori

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa tinjauan pustaka pada penelitian ini yaitu:

II.2.1. Aplikasi

Perangkat lunak aplikasi adalah suatu perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer. Tapi secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah data, lembar kerja dan pemutar media.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*Application Suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org*. yang menggabungkan suatu aplikasi pengolah data, lembar kerja serta beberapa aplikasi lainnya. Aplikasi - aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Seringkali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna, contohnya suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata, walaupun dibuat pada aplikasi lembar kerja yang terpisah (Rizki Ahmad Fauzi ; 2017 : 55).

II.2.2. Sistem dan Absensi

Sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lainnya, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul dan bersama-sama untuk melakukan kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Suatu sistem mempunyai karakteristik tertentu, yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environment*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*), dan sasaran (*objective*) atau tujuan (*goal*). Untuk mencapai tujuan dari suatu sistem yang dibuat, dibutuhkan 3 perangkat atau alat bantu yang dapat meningkatkan kinerja dari sebuah sistem, sehingga tujuan dari suatu sistem tersebut dapat dicapai. Tiga perangkat tersebut meliputi: perangkat keras (*hardware*), perangkat lunak (*software*), dan manusia (*brainware*). Perangkat keras dapat berupa komputer, sedangkan perangkat lunak dapat berupa program dan manusia dapat berupa manajer. Analisis sistem, programmer, ataupun siapa saja yang bertindak sebagai user. Sistem memiliki beberapa syarat, yaitu diantaranya:

1. Sistem harus dibentuk untuk menyelesaikan tujuan.
2. Elemen sistem harus mempunyai rencana yang ditetapkan.
3. Adanya hubungan diantara elemen sistem.
4. Unsur dasar dari proses (arus informasi, energi dan material) lebih penting dari pada elemen sistem.

5. Tujuan organisasi lebih penting dari pada tujuan elemen.(Nugraha Saputra ; 2016 : 1).

II.2.3. QR Code

QR *Code* adalah sebuah kode batang dua dimensi yang ditemukan oleh sebuah perusahaan Jepang bernama *Denso Wave* pada tahun 1994. QR *Code* ini adalah pengembangan dari kode batang sebelumnya. Pada model *barcode* lama, data disimpan secara horizontal saja sedangkan pada QR *Code*, data disimpan baik secara vertikal maupun horizontal (Nugraha Saputra ; 2016 : 1).

QR-*Code* merupakan teknik yang mengubah data tertulis menjadi kode-kode 2dimensi yang tercetak kedalam suatu media yang lebih ringkas. QR-*Code* adalah *barcode* 2-dimensi yang diperkenalkan pertama kali oleh perusahaan Jepang *Denso-Wave* pada tahun 1994. *Barcode* ini pertama kali digunakan untuk pendataan inventaris produksi suku cadang kendaraan dan sekarang sudah digunakan dalam berbagai bidang. QR adalah singkatan dari Quick Response karena ditujukan untuk diterjemahkan isinya dengan cepat. QR-*Code* merupakan pengembangan dari *barcode* satu dimensi, QR-*Code* salah satu tipe dari *barcode* yang dapat dibaca menggunakan kamera handphone (Norhikmah, Azizah, Laili ; 2016 : 2)

II.2.4. Android Studio

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada *event Google I/O Conference* untuk tahun 2013. Sejak saat itu, *Android Studio* menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi *Android*.

Android Studio sendiri dikembangkan berdasarkan *IntelliJ IDEA* yang mirip dengan *Eclipse* disertai dengan *ADT Plugin (Android Development Tools)*. *Android Studio* memiliki fitur :

- a. Projek berbasis pada *Gradle Build*
- b. *Refactory* dan pembenahan bug yang cepat
- c. *Tools* baru yang bernama “*Lint*” diklaim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d. Mendukung *Proguard And App-signing* untuk keamanan.
- e. Memiliki GUI aplikasi *android* lebih mudah
- f. Didukung oleh *Google Cloud Platfrom* untuk setiap aplikasi yang dikembangkan (Andi Juansyah ; 2015 : 3)

Android SDK merupakan tools bagi para programmer yang ingin mengembangkan aplikasi berbasis *google android*. *Android SDK* mencakup seperangkat alat pengembangan yang komprehensif. *Android SDK* terdiri dari *debugger, libraries, handset emulator*, dokumentasi, contoh kode, dan tutorial.

ADT adalah sebuah *Plugin* untuk *Eclipse* yang menyediakan alat yang terintergrasi dengan IDE *Eclipse*. Sedangkan *Android SDK (Software*

Development Kit) menyediakan *Tools* dan *API* yang diperlukan untuk mengembangkan aplikasi pada *Platform Android* dengan menggunakan bahasa pemrograman *Java*. *ADT* menyediakan *fitur* yang membantu untuk mengembangkan aplikasi *Android* dengan cepat. *ADT* menyediakan *GUI* untuk mengakses banyak *command line tools* SD dengan desain *UI* untuk perancangan, desain, dan pembuatan *User Interface* aplikasi. Karena *ADT* adalah sebuah *Plugin* untuk *Eclipse*, maka didapatkan fungsi *IDE* yang stabil dengan fitur *Android* yang spesifik yang digabung dengan *ADT* (Wandy Damarullah ; 2013 : 81).

Android adalah sistem operasi untuk telepon seluler yang berbasis *Linux*. *Android* menyediakan *Platform* terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh *Android Inc*, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh *Google Inc*. Untuk pengembangannya, dibentuklah *Open Handset Alliance* (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia* (M. Ichwan ; 2011 : 15).

II.2.5. UML (*Unified Modeling Language*)

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada *OOAD* terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

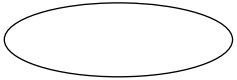
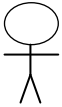
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.



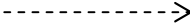
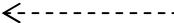
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

- *Use Case Diagram*

Use Case diagram merupakan pemodelan untuk kelakuan (*Behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case* diagram, yaitu :

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan</p>




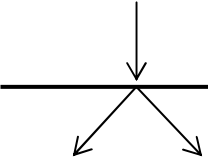
	<i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

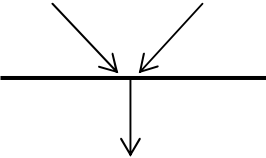
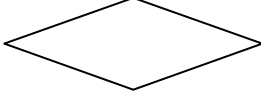

(Sumber : Windu Gata ; 2013 : 4)

- Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

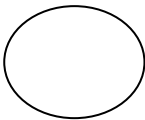
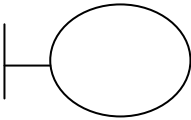
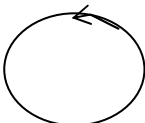
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.


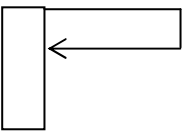


(Sumber : Windu Gata ; 2013 : 6)

- Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.

	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata ; 2013 : 7)

- *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class Diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.4. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gata ; 2013 : 8)