



BAB II
DASAR TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada penelitian sebelumnya dilakukan oleh Tumanggor dkk, mengenai Sistem Informasi Absensi Karyawan Menggunakan QR Code (*Quick Response*) Di PT. Redes Jaya Persada. Sistem ini bertujuan untuk penilaian karyawan agar disiplin dan untuk menghitung gaji karyawan berdasarkan kehadiran. Desain sistem ini menggunakan teknik pemodelan terstruktur dalam bentuk UML dan Menggunakan metode pengembangan perangkat lunak metode *waterfall*. Hasil pengujian Sistem Informasi yang dirancang bekerja dengan sesuai mekanisme perancangan sistem di awal. Penelitian sebelumnya menguraikan teori, temuan, dan bahan penelitian lain yang diperoleh dari acuan, yang dijadikan landasan untuk melakukan penelitian yang diusulkan[1].

Selanjutnya penelitian yang dilakukan oleh Sikumbang mengenai Sistem Informasi Absensi Pegawai Menggunakan Metode RAD dan Metode LBS Pada Koordinat Absensi. Penelitian ini menerapkan metode *Rapid Application Development* (RAD) yang bertujuan untuk mempersingkat waktu dalam perencanaan, perancangan, dan penerapan suatu sistem bila dibandingkan dengan metode tradisional. Lalu metode Location Based[2]. Service (LBS) digunakan untuk mengakses layanan informasi geografis yang digunakan oleh user dengan perangkat telepon seluler melalui koneksi jaringan seluler untuk memetakan lokasi untuk menentukan dimana lokasi keberadaannya. Dengan demikian, dengan adanya penelitian ini dapat membantu perusahaan dan pegawai dalam melakukan kinerjanya dan absensi ini digunakan untuk penggajian dan mengurangi beban kerja yang didapatkan oleh pegawai [2].

Selanjutnya penelitian yang dilakukan oleh Subiantoro, Sardiarinto Perancangan Sistem Absensi Pegawai Berbasis *Web*. Studi kasus: Kantor Kecamatan Purwodadi. Penelitian ini bertujuan untuk membangun pegawai menjadi lebih disiplin, bertanggung jawab dan mampu meningkatkan kinerjanya. Dari peningkatan kinerja ini disesuaikan pada hasil absensi yang telah dilakukan dan berpengaruh pada proses penggajian. Pada Kantor Kecamatan Purwodadi. Metode penelitian merupakan metode yang digunakan dalam pengumpulan data yang meliputi: metode riset, wawancara dan pustaka. Sedangkan pengembangan

perangkat lunak menggunakan *waterfall* yang meliputi: analisis kebutuhan, desain, pengkodean, pengujian dan implementasi[3].

Adapun perbedaan sistem dari penelitian sebelumnya yang telah dibuat dengan penelitian yang akan penulis kembangkan yaitu pada penelitian sebelumnya sistem lebih mengarah pada proses absensi yang digunakan untuk penggajian. Sedangkan penelitian yang dilakukan oleh penulis digunakan untuk proses absensi yang lebih cepat dan membantu bagian kepegawaian dalam rekap absensi. Metode pengembangannya yang dilakukan adalah pengembangan *prototype*. Menggunakan pengujian *Black-box testing*.

2.2 Landasan Teori

Dalam menunjang penelitian ini, maka diperlukannya teori-teori mendasar. Teori-teori yang digunakan dalam penelitian ini adalah:

2.2.1 Rekayasa Perangkat Lunak (RPL)

Rekayasa perangkat lunak (*software engineering*) adalah pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin.

Rekayasa perangkat lunak lebih fokus pada bagaimana membuat perangkat lunak yang memenuhi kriteria berikut

- a. Dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan (*maintainability*).
- b. Dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi (*dependability* dan *robust*).
- c. Efisien dari segi sumber daya dan penggunaan.
- d. Kemampuan untuk dipakai sesuai dengan kebutuhan (*usability*).

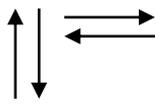
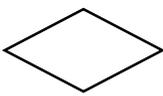
1. UML (*Unified Modelling Language*)

Unified Modelling Language adalah salah satu standar Bahasa untuk mendefinisikan requirement, membuat analisis & desain serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak [18].

A. Flowchart

Flowchart atau Bagan alir adalah bagan (*chart*) yang menunjukkan alir-alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) yang digunakan terutama untuk alat bantu di komunikasi dan untuk dokumentasi. Adapun simbol-simbol *Flowchart* sebagai berikut. [17]. Tabel 2.1 adalah simbol-simbol yang ada pada *flowchart* sebagai berikut:

Tabel 2. 1 Simbol - Simbol *Flowchart*

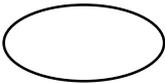
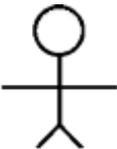
No	Simbol	Nama	Keterangan
1		<i>Terminal</i>	Memulai dan mengakhiri suatu program
2		<i>Input/output</i>	Memasukkan data maupun menunjukkan hasil dari suatu <i>process</i> tanpa tergantung dengan jenis peralatannya.
3		<i>Process</i>	Menunjukkan pengolahan yang akan dilakukan oleh komputer.
4		<i>Flow</i>	Menghubungkan antara simbol satu dengan simbol yang lain atau menyatakan jalannya arus dalam suatu proses. Simbol arus ini sering disebut juga dengan <i>connecting line</i> .
5		<i>Decision</i>	Memilih proses yang akan dilakukan berdasarkan kondisi tertentu.

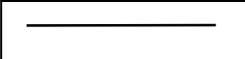
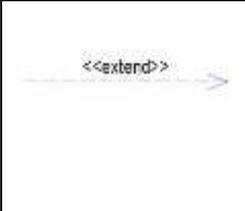
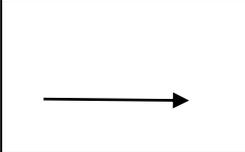
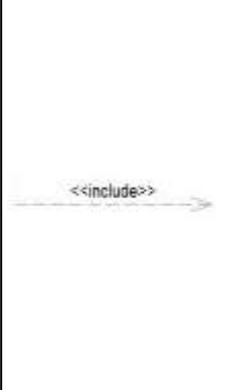
6		<i>Document</i>	Menyatakan masukan yang berasal dari dokumen dan keluaran yang berupa dokumen
7		<i>Manual Operation</i>	Menunjukkan proses yang tidak dilakukan oleh komputer.

B. Use Case Diagram

Use case atau diagram *use case* 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. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut pada Tabel 2.2 adalah simbol-simbol yang ada pada diagram *use case* [19]:

Tabel 2. 2 Simbol *Use Case Diagram*

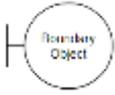
Simbol	Fungsi	Keterangan
	<i>Use case</i>	Sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i> .
	<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat. Walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
	<i>Association / Asosiasi</i>	Komunikasi antara aktor dan use case yang berpartisipasi pada use

		case atau use case memiliki interaksi dengan aktor.
	<i>Predefined Process</i>	Untuk mempersiapkan penyimpanan yang sedang atau akan digunakan dengan memberikan harga awal
	<i>Generalization / Generalisasi</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
	<i>Include / Uses (Menggunakan)</i>	Ada dua sudut pandang yang cukup besar mengenai include di use case yaitu : <ul style="list-style-type: none"> - <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan. - <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.

C. Sequence Diagram

Menurut (Sukamto & Shalahuddin, 2018:165) menjelaskan bahwa “diagram skuen menggambarkan kelakuan objek pada *use case* dengan menpendekatkan [18] . Oleh karena itu, untuk menggambar *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat *sequence diagram* juga dibutuhkan untuk melihat skenario yang pada *use case*. Berikut pada Tabel 2.3 adalah simbol-simbol yang ada pada *sequence diagram* :

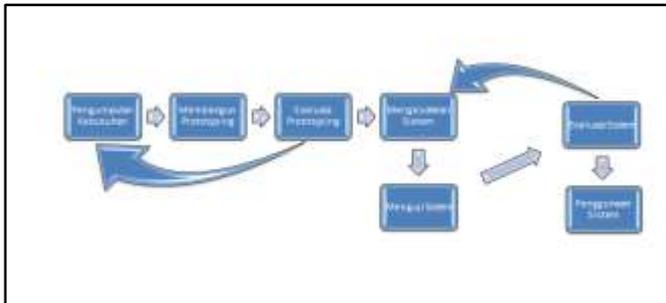
Tabel 2. 3 Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1	 Actor	<i>Actor</i>	Menggambarkan <i>user</i> atau pengguna.
2	 Entity Object	<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan.
3	 Boundary Object	<i>Boundary Class</i>	Menggambarkan sebuah form
4	 Control Object	<i>Control Class</i>	Menghubungkan antara <i>boundary</i> dengan tabel.
5		<i>Lifeline</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.

6		Activation bar	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya. Menggambar lamanya suatu pesan di proses
7		<i>Message</i>	Menggambarkan pengiriman pesan.

2. Metode Pengembangan Sistem

Metode pengembangan yang digunakan untuk membangun Aplikasi Kehadiran Pegawai Berbasis QR-Code adalah metode prototyping. Metode model *prototype* dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak. Metode model *prototype* (*prototyping* model) dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program *prototype* agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program *prototype* biasanya merupakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Program *prototype* ini dievaluasi oleh pelanggan atau user sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau user. Berikut adalah tahap pengembangan sistem dengan model *prototype*[11].



Gambar 2. 1 Metode *Prototyping*

Berikut adalah tahap-tahap dalam metode *prototyping* [12]:

1. Analisa kebutuhan
Pada tahap ini pelanggan dan pengembang bersama-sama mendefinisikan format dan kebutuhan perangkat lunak secara keseluruhan, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.
2. Membangun *prototyping*
Yang dilakukan pada tahap ini adalah membuat desain sementara yang berpusat pada penyajian kepada pelanggan, misalnya; dengan membuat contoh *input* dan *output*.
3. Evaluasi *prototyping*
Jika *prototype* yang sudah dibangun sesuai dengan keinginan pelanggan maka dilakukan evaluasi. Jika sudah sesuai akan dilanjutkan ke tahap pengkodean (*coding*), namun jika belum sesuai maka akan diperbaiki dengan mengulang kembali tahap 1, tahap 2, dan tahap 3.
4. Mengkodekan sistem
Tahap ini lakukan jika *prototype* yang dibuat sudah diterima dan disepakati yang dilanjutkan dengan pengkodean dengan bahasa pemrograman yang sesuai
5. Menguji sistem
Sistem yang telah selesai dibuat kemudian akan dilakukan pengujian dengan menggunakan metode *black-box*. Tujuannya adalah untuk mengetahui apakah sistem yang dibuat sudah dapat berfungsi dengan baik dan benar sesuai yang diharapkan.
6. Evaluasi sistem
Tahap ini dilakukan untuk mengetahui apakah sistem yang dibangun telah sesuai dengan yang diharapkan. Jika sudah memenuhi dapat

dilakukan tahap selanjutnya, namun jika tidak maka perlu dilakukan pengulangan pada tahap 4 dan tahap 5.

7. Menggunakan sistem

Sistem yang dibangun siap digunakan oleh pegawai setelah tahap pengujian berhasil dan diterima oleh pegawai.

3. Metode Pengujian Sistem

Pengujian sistem bertujuan untuk melihat apakah sistem yang telah dibuat sudah sesuai dengan tujuan awal pembuatan dan layak untuk dipergunakan. *Black Box Testing* adalah metode pengujian perangkat lunak yang digunakan untuk mengetahui apakah sekumpulan masukan (input) yang diuji menghasilkan keluaran (*output*) yang diharapkan oleh pengguna tanpa menyadari bagaimana masukan tersebut diproses menjadi keluaran oleh perangkat lunak. Pengujian perangkat lunak menggunakan *black-box testing* dilakukan untuk menguji apakah sistem yang dibuat telah memenuhi persyaratan perangkat lunak diminta oleh pengguna [16].

2.2.2 Basis Data

Basis data adalah kumpulan dari data yang saling berhubungan yang dikelompokkan dalam sebuah tabel atau beberapa tabel dan sebuah aplikasi program yang mengatur cara mengakses data tersebut.

Tujuan dari *DBMS (Database Management Systems)* yaitu untuk menyediakan sebuah cara untuk menyimpan dan mengambil informasi yang ada di dalam basis data.

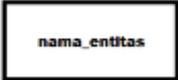
Sistem basis data menyediakan dua jenis bahasa yaitu *data definition language (DDL)* yang berfungsi untuk membuat suatu skema basis data dan *data manipulation language (DML)* yang berfungsi untuk melakukan pencarian dan pembaruan data pada basis data.

Basis data (*database*) juga dapat diartikan sebagai himpunan kelompok data yang saling terhubung dan diorganisasi sedemikian rupa supaya kelak dapat dimanfaatkan kembali secara cepat dan mudah. Database mempunyai delapan operasi dasar, seperti *create database, drop database, create table, drop table, insert, read, update, dan delete*. Sebuah database menyimpan data di dalam tabel, dimana setiap tabel memiliki baris dan kolom.

1. Entity Relationship Diagram (ERD)

Menurut (Sukamto & Shalahuddin, 2018:50) menyatakan bahwa “ERD digunakan untuk pemodelan basis data relasional”. Menurut Al-Bahra dalam (Rahmayu, 2016:34) menerangkan bahwa “*Entity Relationship Diagram* (ERD) adalah diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis :

Tabel 2. 4 Simbol *Entity Relationship Diagram*

No	Simbol	Nama	Keterangan
1.		Entitas	Merupakan suatu simbol untuk mewakili suatu objek dengan karakteristik sama yang dilengkapi oleh atribut.
2.		Relasi	Simbol yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan. Digambarkan dengan simbol belah ketupat.
3.		Atribut	Simbol yang menjelaskan karakteristik suatu entitas dan juga relasinya. Digambarkan dengan elips.

4		Key Atribut / Atribut Kunci Primer	Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan, biasanya berupa id. Kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa 23 kolom tersebut dapat bersifat unik.
5.		Connection	Simbol untuk merangkai keterkaitan antara notasi- notasi yang berupa garis penghubung yang digunakan seperti entitas, relasi dan atribut.
6		Association / Asosiasi	Penghubung antara relasi dan entitas diana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan yang lain disebut kardinalitas.

2. *QR-Code (Quick Respond 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[9].

QR-Code adalah suatu matriks dua dimensi yang didalamnya mampu menyimpan informasi hingga ribuan karakter alfanumerik. Untuk *QR-Code* sendiri sangat cocok untuk proses absensi. Sedangkan cara penggunaannya sebenarnya sangat mudah, karena proses pembuatannya yang tidak lama, maka seluruh informasi pun bisa terekam semua. Setiap penggunaannya bisa terhubung secara langsung ke konten digital dari website. *QR-Code* sendiri dinilai lebih praktis daripada pendahulunya, yaitu barcode. Karena *QR Code* mampu menyimpan data yang lebih banyak dari pada *barcode*[8].

2.2.3 Absensi

Absensi berarti “tidak hadir”, namun bisa dikatakan pula absensi merupakan ketidakhadiran atau kehadiran suatu objek dalam hal ini adalah orang, dimana orang tersebut terlibat dalam suatu organisasi yang mengharuskan adanya pemberitahuan tentang keadaan atau kehadiran atau ketidakhadirannya dalam ruang lingkup organisasi tersebut. Absensi sangat berpengaruh pada kinerja personal serta instansi 11 dimana ia bekerja, yang dapat dijadikan pertimbangan terhadap tindak lanjut serta pembuatan keputusan bagi kelangsungan perkembangan instansi tersebut[7].

Absensi dapat dikatakan suatu pendataan kehadiran yang merupakan bagian dari pelaporan yang ada dalam sebuah institusi. Absensi disusun dan diatur sehingga mudah mudah untuk dicari dan dipergunakan ketika diperlukan oleh pihak yang berkepentingan. Secara umum, jenis-jenis absensi menurut cara penggunaannya dapat dikelompokkan menjadi dua yaitu[8]:

- a. Absensi Manual, yang merupakan cara penulisan kehadiran dengan cara menggunakan pena berupa tanda tangan
- b. Absensi Non manual, yang merupakan cara penulisan kehadiran dengan menggunakan alat yang terkomputerisasi.

2.2.4 CodeIgniter (CI)

CodeIgniter merupakan *framework* yang diklaim memiliki eksekusi tercepat dibandingkan dengan *framework* lainnya (A A Wicaksono, Dkk, 2016). *CodeIgniter* merupakan sebuah web *application framework* yang digunakan untuk membangun aplikasi PHP dinamis yang dibangun menggunakan konsep Model View *Controller* development pattern[15]. *CodeIgniter* menyediakan berbagai macam *library* yang dapat mempermudah dalam pengembangan dan termasuk *framework* tercepat dibandingkan dengan *framework* lainnya. (E Rully , Yessica N, dan Yos Richard Beeh, 2012).