

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Terdapat beberapa penelitian sebelumnya yang meneliti tentang kasus serupa mengenai *Human Resource Information System (HRIS)*. Hal ini bisa menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperbanyak teori yang digunakan untuk mengkaji penelitian yang dilakukan penulis. Berikut ini penelitian terdahulu berupa jurnal terkait dengan penelitian yang dilakukan penulis.

Penelitian pertama yang dikaji oleh penulis adalah penelitian mengenai “Sistem Informasi Penyedia Lowongan Kerja Berbasis Web” yang dilakukan oleh (Masnur & Iklila, 2021). Penelitian tersebut bertujuan untuk membantu pencari kerja dengan mudah mengakses informasi mengenai lowongan kerja yang sesuai dengan keahlian. Dari hasil penelitian tersebut, penulis menyimpulkan bahwa Sistem Informasi Penyedia Lowongan Kerja Berbasis Web memudahkan seseorang dalam mencari lowongan pekerjaan dan dapat diakses di mana pun dan kapan pun (Masnur & Iklila, 2021).

Pada penelitian selanjutnya yaitu mengenai “Perancangan Sistem Informasi Kepegawaian Di Bouw Atelier Contractors Dengan Metode Software Development Life Cycle Waterfall” yang dilakukan oleh (Ardianti & Chazar, 2022). Penelitian tersebut bertujuan untuk mengelola data kepegawaian. Dari hasil penelitian tersebut, penulis menyimpulkan bahwa sistem yang dikembangkan meningkatkan akurasi dan efisiensi dalam mengelola data pegawai (Ardianti & Chazar, 2022).

Penelitian terakhir yang dikaji oleh penulis yaitu mengenai “Sistem Informasi Kepegawaian Menggunakan Metode Waterfall pada PT. Tirta Osmosis Sampurna Palembang” yang dilakukan oleh (Ardiansyah & Sriwidiya, 2022). Penelitian tersebut bertujuan untuk mempermudah mengolah data perekrutan pegawai dan penyimpanan data pegawai. Dari hasil penelitian tersebut, penulis menyimpulkan bahwa sistem yang dikembangkan dapat membantu proses penerimaan calon pegawai, pegawai dapat melakukan absensi, ijin, dan resign, admin dapat melakukan pengolahan data pegawai dan memberi informasi melalui sistem, pemimpin hanya dapat melihat absensi dan data pegawai.

Dari ketiga penelitian di atas, dapat disimpulkan bahwa sistem informasi yang akan dibangun oleh penulis memiliki fitur yang hampir sama. Persamaan fitur yang paling menonjol ditunjukkan pada penelitian ketiga yang ditulis oleh (Ardiansyah & Sriwidiya, 2022) dengan judul penelitian “Sistem Informasi Kepegawaian Menggunakan Metode Waterfall pada PT. Tirta Osmosis Sampurna Palembang” di mana karyawan dapat melakukan absensi, mengajukan ijin, sakit maupun cuti, serta pengunduran diri. Akan tetapi, yang membedakan antara penelitian di atas dengan penelitian yang dilakukan oleh penulis mempunyai perbedaan fitur yaitu karyawan dapat melihat *Key Performance Indeks (KPI)* untuk mengetahui seberapa besar kontribusi karyawan pada perusahaan serta adanya kepala cabang yang akan ikut serta dalam berjalannya aplikasi yang akan dibangun dengan *job description* melihat absensi pegawai, melakukan persetujuan cuti yang diajukan karyawan, dan mengajukan mutasi karyawan.

2.2 Teori Terkait

2.2.1 Sistem Informasi

Menurut (Prehanto, 2020) sistem informasi merupakan proses pengumpulan, penyimpanan, analisis sebuah informasi dengan tujuan tertentu. Sistem informasi terdiri dari data (input) dan menghasilkan laporan (output) sehingga diterima oleh sistem lainnya serta kegiatan strategi dalam suatu organisasi dalam melakukan tindakan atau keputusan.

2.2.2 Human Resource Information System

Human Resource Information System (HRIS) merupakan sebuah sistem informasi yang mengangani permasalahan sumber daya manusia pada organisasi atau perusahaan (Kambang, 2021).

Human Resource Information System (HRIS) adalah sistem yang digunakan untuk memperoleh, menyimpan, manipulasi, analisis, mengambil, dan mendistribusikan informasi mengenai sumber daya manusia sebuah organisasi atau perusahaan untuk mendukung keputusan manajerial (Kavanagh & Johnson, 2018).

2.2.3 Database

Secara teori, *database* memiliki dua komponen utama yaitu data dan informasi. Tujuan dari *database* adalah mengelola data yang dapat diubah menjadi informasi yang sesuai serta proses pengambilan, penghapusan, dan pengeditan data dapat dilakukan secara efektif, efisien, dan akurat (Rawat & Purnama, 2021).

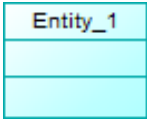
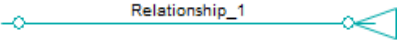
2.2.3.1 Database Management System (DBMS)

Menurut (Rawat & Purnama, 2021) *Database Management System (DBMS)* merupakan aplikasi yang mengelola dan mengeksekusi *database query*. *Database Management System (DBMS)* digunakan untuk mengelola *database* secara efektif dan efisien mulai dari pembuatan awal *database* hingga operasi yang diterapkan seperti memasukkan, mengedit, menghapus *query* terhadap data, membuat laporan, dan sebagainya. *Database Management System (DBMS)* yang paling terkenal saat ini adalah *Relational Database Management System (RDBMS)* dimana data dalam bentuk tabel saling berhubungan. Salah satu aplikasi *Relational Database Management System (RDBMS)* yang paling banyak digunakan adalah *MySQL*. *MySQL* adalah aplikasi *Relational Database Management System (RDBMS) open source* yang paling banyak digunakan dengan pengguna mencapai lebih dari seratus juta pengguna di dunia. *MySQL* menjadi pilihan utama dalam pengembangan aplikasi dikarenakan kecepatan dan kemudahannya.

2.2.3.3 Conceptual Data Model (CDM)

Conceptual Data Model (CDM) merupakan konsep yang berkaitan dengan pandangan pemakai terhadap data yang di simpan dalam basis data. *Conceptual Data Model (CDM)* sudah di buat dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi ke dalam *data base*. *Conceptual Data Model (CDM)* merupakan hasil penjabaran lebih lanjut dari *Entity Relationship Diagram (ERD)*. Ada aturan-aturan yang harus diikuti dalam melakukan konversi *Entity Relationship Diagram (ERD)* menjadi *Conceptual Data Model (CDM)* (Sugiarti, 2013).

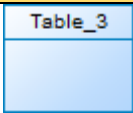

Tabel 2.1 *Copceptual Data Model (CDM)*

Simbol	Nama	Keterangan
	Entitas	Entitas atau tabel menyimpan data dalam data base.
	Relasi	Relasi antar tabel yang terdiri dari nama relasi.

2.2.3.4 *Physical Data Model (PDM)*

Physical Data Model (PDM) adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data-data tersebut. Setiap tabel memiliki jumlah kolom di mana setiap kolom memiliki nama yang unik beserta tipe datanya. *Physical Data Model (PDM)* merupakan konsep yang menerangkan detail dari bagaimana data di simpan di dalam *data base*.

Tabel 2.2 *Physical Data Model (PDM)*

Simbol	Nama	Keterangan
	Tabel	Tabel yang menyimpan data dalam data base.
	Relasi	Relasi antar tabel yang terdiri dari persamaan antara primary key tabel yang mengacu pada kunci yang menjadi referensi acuan pada tabel lainnya.

2.2.4 Website

Website merupakan nama halaman web yang dapat diakses jika terkoneksi jaringan internet. Selain jaringan internet, website harus dibuka melalui *web browser* (Oktarini et al., 2019).

2.2.5 Hypertext Preprocessor (PHP)

Menurut (Enterprise, 2017) PHP merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis website yang memiliki sifat dinamis dan interaktif. Jenis kategori website (Oktarini et al., 2019):

a. Web Dinamis

Web yang terstruktur diperuntukkan untuk diupdate sesering mungkin. Biasanya disediakan halaman backend untuk melakukan perubahan konten dari website tersebut.

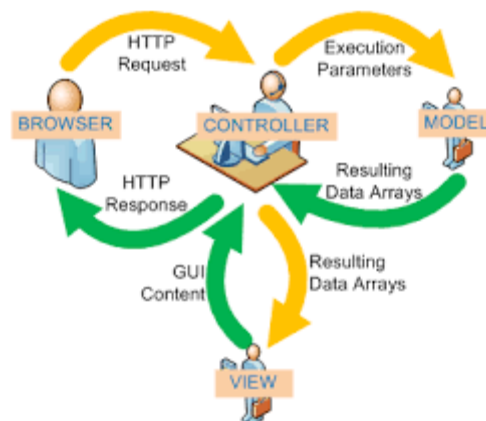
Contoh: web media sosial, web toko online, dan lain-lain

b. Web Interaktif

Web yang penggunanya saling berinteraksi. Biasanya berupa forum diskusi ataupun blog di mana adanya moderator sebagai pengatur alur diskusi.

2.2.6 Codeigniter

Codeigniter merupakan sebuah *framework* aplikasi yang berbasis *website* yang menggunakan konsep MVC (*Model, View, Controller*). *Framework* PHP ini bisa dijadikan *tools* bagi seorang *web developer* (Sulistiani & Hendra Saputra, 2020).



Gambar 2.1 Konsep MVC

2.2.7 Agile Development

Agile Development adalah proses yang mendukung filosofi *Agile* misalnya, *Extreme Programming (XP)* dan *Scrum*. Praktiknya menggunakan *version control system (VCS)* seperti *GitHub*, menetapkan standar pengkodean, dan memberikan demo mingguan kepada *stakeholder*. *Agile Development* menggabungkan bagian-bagian yang mendukung filosofi *Agile* salah satunya mencampurkan beberapa ide baru saat melakukan pengembangan. Hasilnya berupa produk yang berkualitas dan efektif (Shore & Warden, 2008).

2.2.8 Unified Modeling Language (UML)


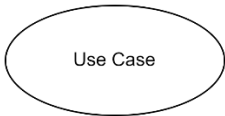

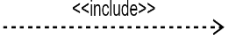
Unified Modeling Language (UML) merupakan metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah software yang berorientasi pada objek dengan standar penulisan yang didalamnya terdapat sebuah proses bisnis dan penulisan kelas-kelas dalam sebuah bahasa yang spesifik.

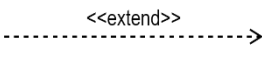
Beberapa diagram *Unified Modeling Language (UML)* yang digunakan dalam pengembangan sistem antara lain:

2.2.9.1 Use Case Diagram

Use Case Diagram merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem dan merepresentasikan sebuah interaksi antara aktor dan sistem (Prihandoyo, 2018).

Tabel 2.3 *Use Case Diagram*





Simbol	Nama	Keterangan
 <p>Actor</p>	<i>Actor</i>	<i>User</i> yang berinteraksi dengan sistem dan memperoleh manfaat dari sistem. Digambarkan dengan figur seperti orang dan diberi nama sesuai dengan perannya.
 <p>Use Case</p>	<i>Use Case</i>	Merupakan bagian utama dari fungsionalitas sistem yang diberi label kata kerja.
	<i>Association</i>	Penghubung antara aktor dengan <i>use case</i> yang saling berinteraksi dengannya.
 <p><<include>></p>	<i>Include</i>	Penyertaan fungsi dari satu <i>use case</i> di dalam yang lain. Digambarkan dengan


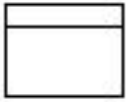
		yang ditarik dari <i>use case</i> ke pangkalan <i>use case</i> yang digunakan.
	<i>Extend</i>	Perluasan dari <i>use case</i> untuk memasukkan perilaku opsional.

2.2.9.2 Activity Diagram

Di dalam *Activity Diagram* terdapat aktor yang merupakan entitas dari manusia yang melakukan aktifitas pada sistem. (Prihandoyo, 2018).

Tabel 2.4 *Activity Diagram*


Simbol	Nama	Keterangan
	Status Awal	Status awal aktivitas sistem, bagaimana objek dibentuk atau diawali.
	Status Akhir	Status akhir aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.




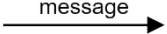
	Penggabungan / <i>Join</i>	Asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu.
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.2.9.3 Sequence Diagram

Menurut (Dennis et al., 2015), *Sequence Diagram* adalah model dinamis yang menunjukkan urutan pesan yang disampaikan antar objek yang berada pada *Use Case Diagram*. Hal ini dikarenakan *Sequence Diagram* menekankan urutan berdasarkan waktu dari aktivitas yang terjadi di antara sekumpulan objek yang dapat membantu dan memahami *Use Case Diagram* yang kompleks.

Tabel 2.5 *Sequence Diagram*

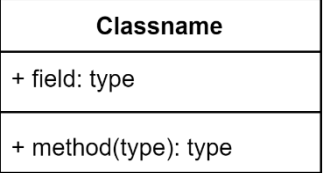
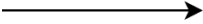
Simbol	Nama	Keterangan
	Aktor	Aktor merupakan orang diluar sistem yang melakukan interaksi dengan sistem serta dapat mengirim ataupun mendapatkan pesan dari sistem.

	Objek	Objek digambarkan persegi panjang yang berisi nama objek. Objek digunakan untuk merekam perilaku objek pada sistem.
	<i>Lifeline</i>	<i>Lifeline</i> digambarkan dengan garis putus-putus. <i>Lifeline</i> memiliki fungsi untuk menunjukkan aktifitas dari sebuah objek.
	<i>Activation box</i>	<i>Activation Box</i> digambarkan dengan persegi panjang vertikal. <i>Activation Box</i> menggambarkan lama waktu yang diperlukan untuk objek menyelesaikan tugasnya. Semakin lama waktunya, <i>Activation Box</i> akan semakin panjang.
	<i>Message</i>	<i>Message</i> berfungsi untuk menyampaikan informasi dari satu objek ke objek lain. <i>Message</i> digambarkan dengan anak panah.

2.2.9.4 Class Diagram

Class Diagram adalah model statis yang menunjukkan hubungan antar kelas yang tetap konstan dalam sistem dari waktu ke waktu (Dennis et al., 2015).

Tabel 2.6 *Class Diagram*

Simbol	Nama	Keterangan
	<i>Class</i>	<i>Class</i> merepresentasikan orang, tempat, atau sesuatu yang perlu disimpan di dalam sistem.
+ field: type	Atribut	Atribut merupakan data dari sebuah <i>Class</i> . Atribut dapat ditulis lebih jelas dengan menuliskan tipe datanya.
+ method(type): type	Operasi	Operasi menggambarkan bagaimana sebuah <i>Class</i> dapat berinteraksi dengan data.
	<i>Inheritance</i> atau Generalisasi	<i>Inheritance</i> atau Generalisasi merepresentasikan hubungan antar <i>Class</i> untuk mewarisi seluruh atribut dan metode dari <i>Class</i> .