

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan.

Athin Pratiwi, 2018, “Implementasi sistem informasi manajemen kepegawaian (SIMPEG) sebagai dasar pengambilan keputusan bidang sumber daya manusia (SDM) di BKDIKLATDA Kota Salatiga”. Hasil dari Penelitian Pelaksanaan Sistem Informasi Manajemen Kepegawaian (SIMPEG) di BKDiklatda Kota Salatiga sudah cukup baik dari teknologi yang digunakan maupun kualitas data. Dengan implementasi SIMPEG dapat membantu BKDiklatda untuk mengelola data kepegawaian di Pemerintah Kota Salatiga secara cepat, tepat, dan akurat. Data yang ada dalam SIMPEG sudah valid, sehingga dapat digunakan dalam proses pengambilan keputusan dalam bidang kepegawaian. Dalam memastikan data tersebut valid BKDiklatda memiliki e file atau elektronik file.

Hilman Abdurrahman Achmad, 2016, “Sistem Informasi Manajemen Kepegawaian di Dinas Komunikasi dan Informatika Bandung Barat”. Hasil dari penelitian berupa sistem informasi manajemen kepegawaian di Diskominfo Bandung Barat yang ditekankan kepada interaksi yang lebih interaktif dimana kegiatan administratif kepegawaian dapat dilakukan didalamsistem seperti melakukan peremajaan data (DUK), kenaikan pangkat, kenaikan gaji berkala,

pengajuan cuti dan mutasi pemberhentian dalam melakukan pengisian data dan melampirkan persyaratan kegiatan administratif.

Muhdar Abdurahman (2018) melakukan penelitian yaitu Sistem Informasi Data Pegawai Berbasis *Web* Pada Kementerian Kelautan Dan Perikanan Kota Ternate. Aplikasi ini dapat mempermudah admin dalam mengelola data pegawai secara efisien dan media informasi kepada masyarakat pada Kelautan dan Perikanan.

Riana Isti Muslikhah, 2019, "Dukungan sistem informasi manajemen kepegawaian terhadap pengambilan keputusan di BKPP Kabupaten Sukoharjo". Hasil dari penelitian Adanya kekurangan yang dihadapi oleh BKPP Kabupaten Sukoharjo dalam penerapan SIMPEG dapat dijadikan sebagai bahan evaluasi untuk memperbaiki Sistem Informasi Manajemen Kepegawaian dengan melihat setiap komponennya, sehingga diharapkan dengan perbaikan tersebut akan memudahkan dalam pengelolaan data kepegawaian sehingga SIMPEG dapat berjalan optimal sebagai pendukung pelaksanaan manajemen kepegawaian.

Jadi menurut peneliti sendiri, hasil dari penelitian yang akan dikerjakan. Data-data manajemen kepegawaian pada PT. Jangkar Mas diolah lebih terorganisis agar tidak terjadi kerusakan dan kehilangan data serta pengolahan pegawai menjadi lebih efektif dan efisien untuk kedepannya. Pada tahap hasil program akan diterapkan pengolahan data kepegawaian secara otomatis menampilkan hasil berupa laporan data kepegawaian.

II.2. Landasan Teori

II.2.1. Sistem Informasi

Menurut Abdhulghani T., Solehudin T., di dalam jurnal Leowaldy Pawan (2021), secara umum sistem informasi dapat diartikan sebagai aplikasi komputer untuk mendukung operasi dari suatu organisasi: operasi, instalasi, dan perawatan komputer, perangkat lunak, dan data.

II.2.2. Manajemen

Menurut Khairrun Nisa (2021), manajemen merupakan salah satu proses interaksi yang membantu orang sebagai fitur asosiasi yang sering digunakan untuk mengatur, menyusun, menyelesaikan, dan mengendalikan latihan, yang semuanya ditujukan untuk tujuan yang telah ditetapkan sebelumnya dan dilanjutkan dalam jangka panjang.

II.2.3. Sistem Informasi Manajemen

Menurut Neni Nuraini dan Imam Ahmad (2021), sistem informasi manajemen dapat didefinisikan sebagai kumpulan dari interaksi sistem-sistem informasi yang bertanggung jawab mengumpulkan dan mengolah data untuk menyediakan informasi yang berguna untuk semua tingkatan manajemen di dalam kegiatan perencanaan dan pengendalian.

II.2.4. Pegawai

Menurut Mardiasmo di dalam jurnal Ruslan (2021), pegawai adalah orang pribadi yang bekerja pada pemberi kerja baik sebagai pegawai tetap atau pegawai tidak tetap/tenaga kerja lepas berdasarkan perjanjian atau kesepakatan kerja baik secara tertulis maupun tidak tertulis.

II.2.5. Sistem Informasi Manajemen Kepegawaian

Menurut Musanef di dalam jurnal Hamzah, dkk (2021), sistem informasi manajemen kepegawaian (SIMPEG) adalah suatu tatanan bagi proses pengumpulan, pengolahan, penganalisaan, penyajian data dan informasi yang diperlukan untuk menunjang administrasi dan manajemen yang berkaitan dengan pegawai.

II.2.6. Website

Menurut M. Hamdan Romadhon, dkk, (2021) *Website* adalah kumpulan informasi/kumpulan *page* yang biasa diakses lewat jalur internet. Setiap orang di berbagai tempat dan segala waktu bisa menggunakannya selama terhubung secara *online* di jaringan internet. Secara teknis, *website* adalah kumpulan dari *page*, yang tergabung kedalam suatu domain atau subdomain tertentu.

II.2.7. PHP

Menurut Agus Saputra di dalam jurnal Noni Rahmawati dan Herry Mulyono (2016) PHP atau yang memiliki kepanjangan PHP *Hypertext Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu *website* dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi. HTML digunakan sebagai pembangun atau pondasi dari kerangka *layout web*, sedangkan PHP difungsikan sebagai prosesnya sehingga dengan adanya PHP tersebut, *web* akan sangat mudah di-*maintenance*.

II.2.8. MySQL

Menurut Priyo, S. dkk, (2016) MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan

SQL sebagai bahasa dasar untuk mengakses databasenya. MySQL termasuk jenis RDBMS (*Relational Database Management System*). Pada MySQL, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

II.2.9. *Unified Modeling Language (UML)*

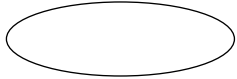
Menurut Dharwiyanti, S. di dalam jurnal Ucu Nugraha, dkk, (2017) *Unified Modeling Language (UML)* merupakan salah satu alat bantu yang dapat digunakan dalam bahasa pemrograman yang berorientasi objek, saai ini UML akan mulai menjadi standar masa depan bagi industri pengembangan sistem/perangkat lunak yang berorientasi objek sebab pada dasarnya UML digunakan oleh banyak perusahaan rekayasa seperti IBM, *Microsoft*, dan sebagainya.

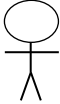


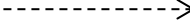
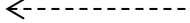
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *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. Simbol-simbol yang digunakan dalam *use case Diagram* dapat dilihat pada tabel II.1 dibawah ini :

Tabel II.1. Simbol *Use Case*

| Gambar | Keterangan |
|---|---|
|  | <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>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 <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p> |
|  | <p>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 mengindikasikan aliran data.</p> |
|  | <p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p> |
|  | <p><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.</p> |
|  | <p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p> |

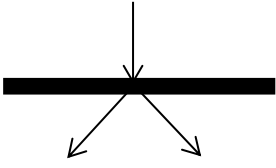
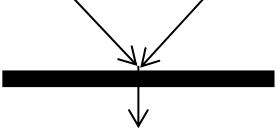
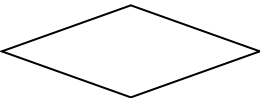

(Sumber : Adi Fitra Andikos, M.Kom, 2019)

2. 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* dapat dilihat pada tabel II.2 :

Tabel II.2. Simbol Activity Diagram

| Gambar | Keterangan |
|---|--|
|  | <p><i>Start point</i>, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.</p> |
|  | <p><i>End point</i>, akhir aktifitas.</p> |
|  | <p><i>Activites</i>, menggambarkan suatu proses/kegiatan bisnis.</p> |

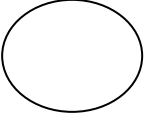
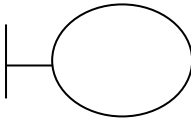
| | |
|---|--|
|  | <p><i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.</p> |
|  | <p><i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.</p> |
|  | <p><i>Decision Points</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i>, <i>false</i>.</p> |
|  | <p><i>Swimlane</i>, pembagian <i>activity Diagram</i> untuk menunjukkan siapa melakukan apa.</p> |

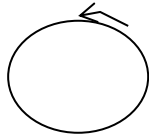
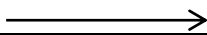
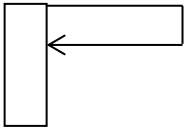


(Sumber : Adi Fitra Andikos, M.Kom, 2019)

3. 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* dapat dilihat pada tabel II.3 :

Tabel II.3. Simbol *Sequence Diagram*

| Gambar | Keterangan |
|---|--|
|  | <p><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.</p> |
|  | <p><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.</p> |

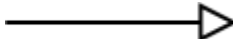
| | |
|--|--|
|  | <i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek. |
|  | <i>Message</i> , simbol mengirim pesan antar <i>class</i> . |
|  | <i>Recursive</i> , 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> . |



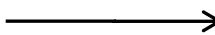
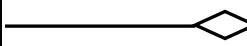

(Sumber : Adi Fitra Andikos, M.Kom, 2019)

4. Diagram Kelas (*Class Diagram*)

Class Diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. Simbol *class Diagram* dan *multiplicity class Diagram* dapat dilihat pada Tabel II.4 dan Tabel II.5. dibawah ini :

Tabel II.4. Simbol *Class Diagram*

| Gambar | Keterangan |
|---|---|
|  | <i>Generalization</i> , untuk menghubungkan antar kelas dengan arti umum-khusus. Jadi jika ada kelas dengan arti umum-khusus. Jadi jika ada kelas bermakna umum dan kelas bermakna khusus dapat menggunakan simbol ini. |

| | | | | |
|--|--|----------|----------|--|
| <table border="1"> <tr><td>Nama_kelas</td></tr> <tr><td>+atribut</td></tr> <tr><td>+operasi</td></tr> </table> | Nama_kelas | +atribut | +operasi | <p><i>Class</i>, untuk sebuah kelas pada struktur sistem. Penulisan tidak boleh menggunakan spasi. Simbol ini memiliki 3 susunan, yaitu kotak pertama adalah nama kelas, kedua atribut dan ketiga operasi.</p> |
| Nama_kelas | | | | |
| +atribut | | | | |
| +operasi | | | | |
|  | <p><i>Interface</i>, untuk simbol <i>interface</i> atau dalam bahasa indonesianya antar muka. Konsep yang digunakan pun sama dengan pemrograman berorientasi object (OOP).</p> | | | |
|  | <p><i>Association</i>, digunakan untuk menghubungkan atau merelasikan kelas satu dengan kelas yang lainnya dengan makna umum.</p> | | | |
|  | <p><i>Directed Association</i>, adalah relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.</p> | | | |
|  | <p><i>Aggregation</i>, adalah relasi antar kelas dengan makna semua bagian.</p> | | | |
|  | <p><i>Dependency</i>, adalah relasi antar kelas dengan makna kebergantungan antar kelas.</p> | | | |

(Sumber : Adi Fitra Andikos, M.Kom, 2019)