

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Hawari (2019) mengenai Sistem Informasi Pengajuan Cuti Karyawan Berbasis *Web* Menggunakan *Framework CodeIgniter* (Studi Kasus: Oakwood Premiere Cozmo), Hawari menyimpulkan bahwa sistem dapat menghemat waktu karyawan maupun bagian hr dalam proses pengajuan dan persetujuan cuti. *User/* karyawan dan bagian HR bisa memproses pengajuan cuti secara langsung, tanpa tatap muka.

Berdasarkan penelitian yang dilakukan oleh Winarni dan Handayani (2018) mengenai Sistem Informasi Pengajuan Cuti Berbasis *Website* Dan Penerapan SMS *Gatewaynotification* PT. Bank Negara Indonesia (Persero) Tbk Kantor Cabang Tanjung Pinang, Winarni dan Handayani menyimpulkan bahwa proses pengiriman surat permohonan dari masing-masing Kantor Cabang Pembantu ke Kantor Cabang Utama sudah terkomputerisasi.

Berdasarkan penelitian yang dilakukan oleh Sumarta, dkk (2019) mengenai Membangun Aplikasi Cuti Karyawan Dan Dosen Pada Universitas Bina Darma Palembang Berbasis *Android*, Sumarta, dkk menyimpulkan bahwa dengan adanya aplikasi cuti karyawan dan dosen pada Universitas Bina Darma Palembang ini membantu admin atau karyawan yang memiliki akses pada sistem cuti dalam memonitor waktu cuti karyawan dan dosen agar mereka dapat mendapatkan hak cutinya tepat waktu.

Berdasarkan penelitian yang dilakukan oleh Buana, dkk (2021) mengenai Penerapan Metode *Agile* Untuk Membangun Sistem Informasi Monitoring Santri Pondok Modern Asy-Syifa Balikpapan, Buana, dkk menyimpulkan bahwa sistem informasi yang dapat melakukan pencatatan prestasi, pelanggaran dan perizinan santri, serta dapat melakukan monitoring prestasi, pelanggaran dan perizinan santri yang berbasis *website* sesuai dengan kebutuhan Pondok Modern AsySyifa Balikpapan.

Berdasarkan penelitian yang dilakukan oleh Hakam, dkk (2021) mengenai Implementasi Metode *Agile* pada Sistem Manajemen Zakat Berbasis *Website* dengan Framework Laravel, Hakam, dkk menyimpulkan bahwa sistem yang dibangun telah memenuhi kebutuhan administrasi untuk menginputkan data-data seperti data Muzakki (pemberi zakat), data Mustahiq (penerima zakat), laporan transaksi harian.

Berdasarkan penelitian yang dilakukan oleh Zaef, dkk (2018) mengenai Sistem Informasi Penerimaan Peserta Didik Baru Berbasis *Android* Menggunakan Metode *Agile*, Zaef, dkk menyimpulkan bahwa Sistem Informasi Penerimaan Peserta Didik Baru Berbasis *Android* dibangun dengan menggunakan metode *agile* dimana sistem dapat menyesuaikan dengan terhadap perubahan yang ada pada penerimaan peserta didik baru dikarenakan tahapan yang ada pada *agile development* bersifat dinamis dengan melibatkan pengguna.

II.2. Landasan Teori

Berikut ini adalah landasan teori yang berkaitan dengan penelitian yang diambil dari beberapa jurnal terkait penelitian:

II.2.1. Rancang Bangun

Rancang bangun adalah menciptakan dan membuat suatu aplikasi ataupun sistem yang belum ada pada suatu instansi atau objek tersebut. Sistem adalah suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan. (Kurnianto dan Lukman, 2021: 717).

II.2.2. Aplikasi

Aplikasi merupakan sebuah subkelas dari perangkat lunak komputer dengan memanfaatkan kemampuan komputer sebagai penerapannya, sebagai penyimpan sesuatu hal, data, permasalahan, pekerjaan kedalam suatu sarana atau bahkan ke dalam sebuah wadah yang dapat diipakaisebagaipenerapan dan pengimplementasian terkait hal atau persoalan yang ada sampai beralih menjadi suatu bentuk yang baru tanpa menghapus poin dasar dari berbagai hal diatas, seperti data, persoalan dan keharusan itu sendiri. Dan aplikasi merupakan suatu perangkat komputer siap pakai bagi user, seperti mengerjakan suatu tugas yang di inputkan oleh pengguna. (Faris dan Wisaksono, 2021: 62).

Aplikasi Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus komputer eksekutif, aplikasi mempunyai arti

yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan. Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia, “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. (Turiana, dkk, 2021: 16).

II.2.3. Pengajuan

Pengajuan (permohonan) merupakan langkah awal dalam perizinan, dan permohonan adalah permintaan yang berkepentingan akan suatu keputusan, sehingga setiap penetapan izin harus didasarkan pada permohonan dari pihak yang berkepentingan atas dikeluarkannya keputusan (izin). (Lestari dan Djanggih, 2019: 154).

II.2.4. Penerimaan

Penerimaan (Persetujuan) adalah suatu perbuatan dengan mana satu orang atau lebih mengikatkan dirinya terhadap satu orang lain atau lebih. (Setiono dan Sulistyono, 2020: 65).

II.2.5. Cuti

Cuti merupakan salah satu hak pegawai atau karyawan diseluruh perusahaan yang ada, cuti yaitu tidak masuk kerja yang mendapat izin dari atasan dalam waktu tertentu, diizinkan dalam upaya untuk menjamin kesegaran rohani dan jasmani serta kepentingan karyawan. Beberapa jenis cuti yang dapat diambil oleh karyawan, dan persyaratan yang dipenuhi antara lain Cuti Tahunan, Cuti Besar, Cuti Sakit, Cuti

Bersalin. Cuti Alasan Penting. (Hawari, 2019: 178).

II.2.6. Metode Agile

Metode *agile* mulai diperkenalkan pada dekade 90an untuk menangani kelemahan yang tidak mampu untuk menangani penambahan atau perubahan kebutuhan pada saat proses pengembangan *software*. *Agile Software Development* merupakan salah satu metode pengembangan *software* yang memiliki sifat adaptif dan responsif terhadap perubahan. Adaptif karena mudah menyesuaikan diri dengan keadaan, responsif karena cepat dalam merespon (cepat tanggap). Hasil yang dikembangkan sampai saat ini sudah cukup banyak, diantaranya adalah *Extreme Programming*, *Scrum Methodology*, *Crystal Family*, *Dynamic System Development Method*, *Adaptive Software Development* dan *Feature Driven Development*. (Buana, dkk, 2021: 183).

Tahapan Metode Agile:

1. *Timebox Planning*, perencanaan dalam membangun sistem.
2. *Daily Stand-Up Meeting* (*Requirements Elicitation*, *Detail System Design*, *Coding Development & Testing*), dengan menganalisis terhadap sistem yang sedang berjalan, membuat desain dengan menggunakan *Unified Modelling Language*.
3. *Demonstration*, dengan menjelaskan mekanisme kerja dari sistem.
4. *Retrospective Meeting*, mengadakan pertemuan untuk membahas sistem. (Zaef, dkk, 2018: 228).

II.2.7. Web

Secara umum, *website (web)* dipahami sebagai sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk digital baik itu teks, gambar, animasi yang disediakan melalui jalur internet sehingga dapat diakses dari seluruh dunia yang memiliki koneksi internet. *Website* awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan *surfer* atau pengguna internet melakukan penelusuran informasi di internet. Informasi yang disajikan dengan *web* menggunakan konsep multimedia, informasi dapat disajikan dengan menggunakan banyak media, seperti teks, gambar, animasi, suara, atau film.

Website merupakan halaman-halaman yang berisi informasi yang ditampilkan oleh *browser* seperti *Mozilla Firefox*, *Google Chrome* atau yang lainnya. Dokumen pada *website* disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu halaman ke halaman yang lain, baik diantara halaman yang disimpan dalam *server* maupun *server* diseluruh dunia. (Manuhu dan Otniel, 2021: 306). (Arthalia dan Prasetyo, 2020: 96).

II.2.8. *Android*

Smartphone atau yang sering disebut dengan *android* adalah sebuah sistem operasi untuk *smartphone* dan *tablet*. Sistem operasi dapat diilustasikan sebagai jembatan antara peranti (*device*) dan penggunaannya, sehingga pengguna dapat berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device*. *Android* merupakan generasi baru *platform mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. (Kuswanto, 2020: 80).

II.2.9. HTML

Hypertext Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *web* Internet dan pemformatan *hipertext* sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman *web* dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web*. HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa). (Arthalia dan Prasetyo, 2020: 96).

II.2.10. *Hypertext Preprocessor (PHP)*

PHP merupakan kependekan dari *Hypertext Preprocessor*. PHP (*Hypertext Preprocessor*) adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs *web* dinamis. (Manuhu dan Otniel, 2021: 306).

II.2.11. MySQL

MySQL adalah salah satu aplikasi DBMS yang sudah banyak oleh para pemogram aplikasi *web*. Contoh DBMS lainnya adalah: PostgreSQL (*freeware*), SQL Server, MS Access dari Microsoft, DB2 dari IBM, Oracle dan Oracle Corp, Dbase, FoxPro, dan lain sebagainya. (Manuhu dan Otniel, 2021: 306).

II.2.12. *Unified Modelling Language (UML)*

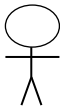
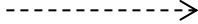
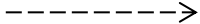
Unified Modelling Language (UML) yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019: 39).

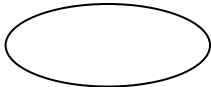
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. Use Case Diagram

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 aktor dengan sistem. *Use case diagram* dapat digambarkan dengan sumber-sumber pada Tabel II.1.

Tabel II.1. Simbol Use Case

Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang mnghubungkan antara objek satu dengan objek lainnya.

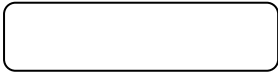
	<i>System</i>	Menspesifikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

(Sumber: Andikos, 2019: 39)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada tabel II.2.

Tabel II.2. Simbol Activity Diagram

Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final</i>	Bagaimana objek dibentuk dan dihancurkan
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

(Sumber: Andikos, 2019: 39)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

Tabel II.3. Simbol *Sequence Diagram*

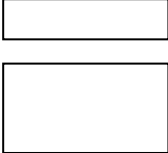
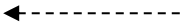
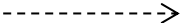
Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek entity, antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

(Sumber: Andikos, 2019: 39)

4. *Class Diagram* (Diagram Kelas)

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

Tabel II.4. Class Diagram

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Assocation</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber: Andikos, 2019: 39)