

## **BAB II**

### **DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian sebelumnya yang dilakukan oleh Akhmad Rizwan Anwari, Agus Salim Muin, Wagino, dengan judul “Aplikasi Sistem Informasi Pensiun Pendidik dan Tenaga Kependidikan Pegawai Negeri Sipil Pada Dinas Pendidikan dan Kebudayaan Provinsi Kalimantan Selatan” [4]. Sistem dibuat untuk dapat membantu mempermudah proses pengarsipan dokumen dan manajemen inventaris dapat dilakukan secara cepat. Dalam penelitian ini penulis menggunakan software Sublime Text 3 dengan bahasa pemrograman HTML, PHP dan database MySQL.

Penelitian yang lain yang dilakukan oleh Zuli Astari, Desi Pibriana, Rusbandi telah melakukan penelitian yang berjudul “Perancangan Sistem Informasi Kepegawaian Pada PT. Bakri Karya Sarana Batam” [5]. Tujuan dilakukannya penelitian ini adalah untuk merancang dan membuat sistem informasi kepegawaian berbasis web dalam mengelola data kepegawaian seperti proses perekrutan pegawai baru, absensi pegawai, cuti dan pengunduran diri. Metodologi yang digunakan dalam penelitian ini adalah metodologi iterasi yang terdiri dari tahap perencanaan, analisis masalah, perancangan, implementasi dan pemeliharaan.

Penelitian yang sejenis yang dilakukan oleh Mohamad Reza Fachlevi dan R. Fenny Syafariani telah melakukan penelitian yang berjudul “Perancangan Sistem Informas Kepegawaian Berbasis Website Di Bagian Kepegawaian SDN Binakarya I Kabupaten Garut” [6]. Tujuan dilakukannya penelitian ini adalah untuk membantu proses pengajuan pensiun, KGB, kenaikan pangkat, penilaian pegawai sehingga dapat mempercepat pekerjaan. Dalam penelitian ini metode pengembangan menggunakan metode *prototype*, serta alat bantu yang digunakan adalah *usecase*, *activity diagram*, *class diagram*, *object diagram* dan *deployment diagram*.

Penelitian lain dilakukan oleh Susi Susilowati dan Rahmi Widiana pada tahun 2019 dengan judul “Penerapan Website Sistem Pengajuan Cuti Pegawai Pada Kantor Kecamatan Ciawi Bogor” [7]. Tujuan penelitian menghasilkan program pengajuan cuti berbasis web sehingga memberikan kemudahan kepada pengguna dalam mengajukan dan memproses cuti. Dalam penelitian ini metode penelitian yang digunakan ialah metode

deskriptif dan tindakan (aksi) serta untuk merancang sistem informasi kepegawaian digunakan pendekatan berorientasi objek (OOP) dan untuk metode pengembangan menggunakan metode prototipe, serta alat bantu yang digunakan adalah *usecase*, *activity diagram*, *class diagram*, *object diagram* dan *deployment diagram*.

Penelitian sebelumnya juga dilakukan oleh Ayu Alivia Hidayah Safitri dan Maulana Ardhiansyah yang berjudul “Perancangan Sistem Informasi Cuti Kepegawaian Berbasis Web Dengan Metode *Prototype* (Studi Kasus : PT. Jakarta Realty” [8]. Tujuan dibuatnya sistem ini adalah memudahkan admin dan pegawai di PT. Jakarta Realty memperoleh data pengajuan cuti yang lebih cepat dan akurat. Juga dalam pengajuan cuti ini kekeliruan dan keterlambatan laporan data pengajuan cuti akan berkurang karena sudah tersimpan rapi didalam sistem dan database dengan aman. Pada penelitian sistem informasi cuti berbasis web ini menggunakan bahasa pemrograman PHP dan Basis Data MySQL.

Penelitian kali ini, bermaksud mengembangkan sistem informasi pengajuan cuti dan pensiun berbasis website. Sistem ini dapat membantu admin dalam mengelola data pengajuan serta membantu para pegawai dalam proses pengajuan cuti dan pensiun. Selain itu, di dalam sistem ini menginformasikan syarat-syarat pengajuan cuti dan pensiun pegawai, form pengajuan, serta terdapat fitur notifikasi email. Sistem ini memiliki perbedaan dengan penelitian-penelitian yang pernah dilakukan sebelumnya antara lain yaitu menyediakan pengajuan pensiun yang menghubungkan pegawai dengan kantor pusat melalui sistem.

## **2.2 Landasan Teori**

### **2.2.1 Rekayasa Perangkat Lunak**

Perangkat Lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*) [9]. Dalam perangkat lunak tidak hanya berisi program, tetapi juga semua dokumentasi dan konfigurasi data yang saling terhubung sehingga program beroperasi dengan benar. Perangkat lunak cukup sering dibuat dan akhirnya tidak dipakai karena tidak memenuhi kebutuhan pelanggan atau bahkan karena masalah non-teknis seperti ketidakmauan pengguna untuk mengubah cara kerja dari manual ke otomatis, atau ketidakmampuan pengguna menggunakan komputer. Oleh karena itu, rekayasa perangkat lunak dibutuhkan agar perangkat lunak yang dibuat tidak hanya menjadi perangkat lunak yang tidak dipakai.

Rekayasa Perangkat Lunak (RPL) adalah disiplin ilmu yang membahas semua spek produksi perangkat lunak, mulai dari tahap awal spesifikasi sampai pemeliharaan sistem. Rekayasa perangkat lunak berhubungan dengan masalah praktis dalam memproduksi perangkat lunak. Rekayasa perangkat lunak lebih fokus pada praktik pengembangan perangkat lunak yang bermanfaat kepada pelanggan (*customer*).

Menurut Al Fatta, Pengujian sistem merupakan proses mengeksekusi sistem perangkat lunak untuk menentukan apakah sistem perangkat lunak tersebut cocok dengan spesifikasi sistem dan berjalan sesuai dengan lingkungan yang diinginkan [10]. Pengujian sistem sering diasosiasikan dengan pencarian *bug*, ketidaksempurnaan program, kesalahan pada baris program yang menyebabkan kegagalan pada eksekusi sistem perangkat lunak.

Ada dua metode untuk melakukan unit testing :

a) *Black Box Testing*

Terfokus pada apakah unit program memenuhi kebutuhan (*requirement*) yang disebutkan dalam spesifikasi. Cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses bisnis yang diinginkan.

b) *White Box Testing*

Merupakan cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak. Metode pengujian yang digunakan pada website ini adalah *Black Box Testing*.

Menurut Rizky dalam Wijayanto “*Black Box Testing* adalah tipe testing yang memerlukan perangkat lunak, dimana perangkat tersebut tidak diketahui kinerja internalnya”.Perangkat lunak tersebut akan dieksekusi kemudian berusaha dites apakah telah memenuhi kebutuhan pengguna yang didefinisikan pada awal tanpa harus membongkar listing programnya.

Menurut Pressman, dalam melakukan perancangan sistem yang akan dikembangkan dapat menggunakan *prototype* [11]. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat yang akan dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna, dalam hal ini pengguna dari perangkat yang dikembangkan adalah pegawai. Kemudian membuat sebuah rancangan yang selanjutnya akan dievaluasi kembali sebelum diproduksi secara benar.

*Prototype* bukanlah sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat *prototype* dibuat untuk memenuhi kebutuhan pengguna dan pada saat yang sama memungkinkan pengembang untuk lebih memahami kebutuhan pengguna secara lebih baik lagi.

Berikut merupakan manfaat yang diperoleh dari penggunaan *prototyping* adalah :

- a) Metode *prototype* memungkinkan pengguna untuk mendapatkan sistem yang lebih baik.
- b) Penggunaan metode *prototype* dengan fleksible dapat menyesuaikan kebutuhan user karena *prototype* menerima masukan pengguna sampai akhir.
- c) User dengan perancang memiliki waktu yang lebih banyak dalam interaksi seputar kebutuhan yang diinginkan.

Metode *prototype* ini memiliki beberapa tahapan yang memiliki perannya masing - masing selama proses perancangan perangkat lunak yang bisa dijelaskan masing - masing tahapan tersebut pada penjelasan dibawah ini :

#### 6) *Communication*

Dimulai dengan tahap *communication*, tahapan ini bertujuan untuk mengidentifikasi berbagai kebutuhan aplikasi yang akan dirancang nantinya dengan melibatkan para client yang bersangkutan agar selama proses perancangan bisa memberikan hasil yang tepat sesuai keinginan *client* yang bersangkutan.

#### 7) *Quick Plan*

Pada tahap *quick plan* ini perancang perangkat lunak akan melakukan perencanaan cepat sesuai dengan spesifikasi kebutuhan *user* berdasarkan data yang telah dikumpulkan pada tahap *communication* dengan merancang desain antarmuka yang dibutuhkan dan kebutuhan pendukung pada proses ini.

#### 8) *Modeling Quick Design*

Pada tahap ini tim perancang akan membuat model design UML ataupun pemodelan yang dibutuhkan lainnya dengan waktu perancangan yang efektif untuk mendeskripsikan kebutuhan *client* berdasarkan analisis yang telah dilakukan sebelumnya.

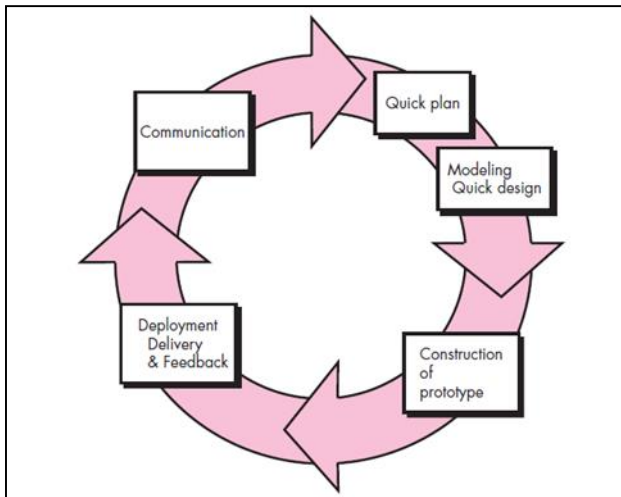
#### 9) *Construction of Prototype*

Selanjutnya pada tahap ini perancang akan memulai membangun perangkat lunak berdasarkan data yang telah dikumpulkan sebelumnya, proses pembangunan ini lebih berfokus terhadap aspek utama perangkat

lunak dengan maksud pada proses selanjutnya perancang bisa dengan cepat mendapatkan *feedback* dari *client* tentang perangkat lunak yang dibuat.

#### 10) *Deployment Delivery & Feedback*

Dalam tahap ini *prototype* akan diserahkan kepada *client* untuk mendapatkan *feedback* dari hasil *prototype* tersebut, *feedback* tersebut akan digunakan sebagai landasan untuk memperbaiki *prototype* agar sesuai dengan spesifikasi kebutuhan *client*.



**Gambar 2. 1** Metode *Prototype*

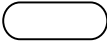
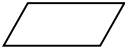

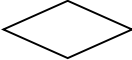




Alat bantu atau *tools-tools* perangkat lunak yang memberikan topangan otomatis ataupun semi-otomatis pada prosesnya dengan metode yang ada seperti *Computer Aided Software Engineering (CASE)*.


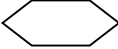
#### a. *Flowchart*

*Flowchart* atau Diagram Alir merupakan bagan (*chart*) yang mengarahkan alir (*flow*) di dalam prosedur atau program sistem secara logika [12]. *Flowchart* adalah cara untuk menjelaskan tahap-tahap pemecahan masalah dengan merepresentasikan simbol-simbol tertentu yang mudah dipahami, mudah digunakan dan standar. Tujuan penggunaan *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai dan rapi dengan menggunakan simbol-simbol yang standar yang dapat dimengerti oleh *programmer*.

Berikut ini merupakan simbol-simbol yang terdapat pada *flowchart* seperti pada tabel dibawah ini:

**Tabel 2. 1** *Simbol Flowchart*

No	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Memulai dan mengakhiri suatu program
2.		<i>Input/Output</i>	Memasukan data maupun menunjukkan hasil dari suatu process tanpa tergantung dengan jenis peralatannya
3.		<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 connecting line.
4.		<i>Decision</i>	Memilih proses berdasarkan kondisi yang ada.
5.		<i>Document</i>	Merupakan simbol untuk data yang terbentuk informasi.
6.		<i>Manual Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh komputer/pc.
7.		<i>Punched Card</i>	Menyatakan input berasal dari kartu atau output ditulis ke kartu.
8.		<i>Connector</i>	Menyatakan sambungan dari suatu proses ke proses lainnya dalam halaman / lembar yang sama.

No	Simbol	Nama	Keterangan
9.		<i>Offline Connector</i>	Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman atau lembar yang berbeda.
10.		<i>Predefined Process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.


### b. UML (*Unified Modeling Language*)






UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan reqUIrement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [9]. UML muncul dikarenakan adanya kebutuhan pemodelan visual untuk menguraikan, memvisualkan, membentuk dan dokumentasi dari sistem perangkat lunak.

#### a. Use Case Diagram

*Use case diagram* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Secara kasar *use case* digunakan untuk mengetahUI fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi–fungsi itu. Berikut Tabel simbol–simbol yang ada pada diagram *use case* :

**Tabel 2. 2** Simbol Use Case


No.	Simbol	Nama	Keterangan
1.		<i>Use case</i>	Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .

No.	Simbol	Nama	Keterangan
2.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
3.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case target</i> memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas






### b. Sequence Diagram

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek oleh karena itu untuk menggambarkan *Sequence diagram* maka harus diketahui objek yang terlibat dalam sebuah *use case* beserta metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Berikut adalah simbol yang ada pada diagram *Sequence diagram* :

**Tabel 2.3** Simbol Sequence Diagram

No.	Simbol	Nama	Keterangan
1.		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.



No.	Simbol	Nama	Keterangan
2.		<i>Actor</i>	Menggambarkan <i>user</i> atau pengguna.
3.		<i>Message</i>	Spesifikasi dari komunikasi antar <i>objek</i> yang memuat informasi – informasi tentang aktivitas yang terjadi.
4.		<i>Boundary</i>	Menggambarkan sebuah <i>form</i> .
5.		<i>Control Class</i>	Menghubungkan <i>boundary</i> dengan Tabel.
6.		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.

### 2.2.2 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek atau (*Object Oriented Programming/OOP*) merupakan suatu pendekatan pemrograman yang semua data dan fungsinya dibungkus menggunakan *object* dan *class* [13]. Pemrograman berorientasi objek berbeda dengan pemrograman prosedural yang hanya menggunakan satu halaman kebawah untuk mengerjakan banyak perintah atau statement. Penggunaan pemrograman berorientasi objek sangat banyak sekali, contoh : java, php, perl, c#, cobol, dan lainnya.

Objek dalam OOP adalah unit terkecil pemrograman yang masih memiliki data (sifat karakteristik) dan fungsi. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan, dan memanipulasi data. Class adalah wadah berisi pemodelan suatu objek, mendeskripsikan karakteristik dan fungsi objek tersebut. Karena class merupakan wadah yang akan digunakan untuk menciptakan objek tersebut, maka Class harus diciptakan terlebih dahulu.

OOP memberikan kemudahan dalam pembuatan sebuah program, keuntungan yang didapat apabila membuat program berorientasi objek atau *Object Oriented Programming* (OOP) antara lain :

- 1) Reusability, kode yang dibuat dapat digunakan kembali
- 2) Extensibility , pemrogram dapat membuat metode baru atau mengubah yang sudah ada sesuai yang diinginkan tanpa harus membuat kode dari awal
- 3) Maintainability, kode yang sudah dibuat lebih mudah untuk dikelola apabila aplikasi yang dibuat berskala besar yang memungkinkan adanya error dalam pengembangannya hal tersebut dapat diatasi dengan OOP karena pemrograman OOP sudah menggunakan konsep modularitas.

Terdapat beberapa cara untuk menentukan karakteristik dalam pendekatan berorientasi objek, tetapi secara umum mencakup empat hal, yaitu identifikasi, klasifikasi, polymorphism (polimorfisme) dan inheritance (pewarisan). Metodologi pengembangan sistem berorientasi objek mempunyai tiga karakteristik utama, yaitu :

- 1) Encapsulation  
Encapsulation (pengkapsulan) merupakan dasar untuk pembahasan ruang lingkup program terhadap data yang diproses.
- 2) Inheritance  
Inheritance (Pewarisan) adalah tehnik yang menyatakan bahwa anak dari objek akan mewarisi data / atribut dan metode dari induknya langsung. Sifat yang dimiliki oleh kelas induknya tidak perlu diulang dalam setiap subkelasnya.
- 3) Polymorphism  
Polymorphism (polimorfisme) yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda.

### **2.2.3 Basis Data**

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya untuk memelihara data yang sudah diolah dan membuat informasi tersedia saat dibutuhkan. Basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat [14]. Kebutuhan basis data dalam sistem informasi meliputi memasukkan, menyimpan, mengambil kembali data dan membuat laporan berdasarkan data yang telah disimpan. Oleh sebab itu, untuk merancang tabel yang akan

dibuat maka diperlukan pola pikir penyimpanan data nantinya jika dalam bentuk baris data (record) dimana setiap baris terdiri beberapa kolom .

Kegunaan utama sistem basis data adalah agar pengguna mampu menyusun suatu pandangan (view) abstraksi data. Hal ini bertujuan untuk menyederhanakan intraksi antara pengguna dengan sistemnya dan basis data dapat mempresentasikan pandangan yang berbeda kepada para pengguna, programmer, dan administratornya. Sebab tidak semua pengguna basis data terlatih dengan baik dan penggunaanya terbagi dalam berbagai tingkatan, maka kompleksitas basis data akan tersembunyi dari para pengguna melalui beberapa level abstraksi data. Ketika memandang basis data, pemakai dapat dikelompokkan menjadi 3 tingkatan yaitu :

1. Level Fisik (physical view/internal view).

Merupakan tingkatan terendah dalam abstraksi data yang menunjukkan bagaimana data disimpan dalam kondisi sebenarnya. Level ini merupakan bentuk paling kompleks, dimana struktur data level terendah digambarkan pada level ini.

2. Level Konseptual.

Merupakan level yang menggambarkan data apa yang sebenarnya (secara fungsional) disimpan dalam basis data, beserta relasi yang terjadi antara data. Level ini menggambarkan keseluruhan database, dimana administrator basis data (DBA) membangun dan mengolah basis data, sedangkan pemakai tidak memperdulikan kerumitan dalam struktur level fisik lagi.

3. Level Pandangan Pemakai.

Merupakan level dengan tingkatan tertinggi, yang menggambarkan hanya satu bagian dari keseluruhan database. Beberapa pengguna basis data tidak membutuhkan semua isi basis data misalkan bagian personalia hanya membutuhkan data file karyawan dan gaji, tidak membutuhkan data file gudang, transaksi barang masuk.

MySQL merupakan software database open source yang paling populer di dunia. MySQL menjadi pilihan utama bagi banyak pengembang software dan aplikasi hal ini dikarenakan kelebihan MySQL, diantaranya sintaksnya yang mudah dipahami, didukung program-program umum seperti C, C++, Java, PHP, Pyton. MySQL sebenarnya merupakan turunan salah satu konsep utama database sejak lama, yaitu SQL (*Structured Query Language*) [18].

SQL adalah sebuah konsep pengoperasian database terutama untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keadaan suatu sistem database (DBMS) dapat diketahui dari kerja optimizernya dalam melakukan proses perintah-perintah SQL yang dibuat oleh user maupun program-program aplikasinya. SQL dibagi menjadi dua bentuk query yaitu :

#### 1. *Data Definition Language (DDL)*

DDL adalah sebuah *Metode Query* SQL yang berguna untuk mendefinisikan data sebuah *database*, adapun *Query* yang dimiliki adalah:

- a) Create : Digunakan untuk pembuatan table dan *database*.
- b) Drop : Digunakan untuk penghapusan table dan *database*.
- c) Alter : Digunakan untuk perubahan struktur table yang dibuat, baik menambah *field* (add), mengganti nama *field* (*change*), ataupun menamakannya kembali (*rename*) serta menghapus (drop).

#### 2. *Data Manipulation Language (DML)*

DML adalah sebuah metode *Query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *Query* ini adalah untuk melakukan pemanipulasian database yang telah ada atau telah dibuat sebelumnya. Adapun *Query* yang termasuk didalamnya adalah :

- a) Insert : Digunakan untuk penginputan data pada table *database*.
- b) Updat : Digunakan untuk perubahan data pada table *database*.
- c) Delete : Digunakan untuk penghapusan data pada table *database*.