BAB II

TINJAUAN PUSTAKA

2.1 JDIH

Jaringan Dokumentasi dan Informasi Hukum menurut Peraturan Presiden Nomor 33 Tahun 2012 merupakan wadah pendayagunaan bersama atas dokumen hukum secara tertib, terpadu, dan berkesinambungan serta merupakan sarana pemberian layanan informasi hukum secara lengkap, akurat, mudah, dan cepat, dalam Peraturan Presiden tersebut juga dijelaskan salah satu anggota JDIH adalah biro hukum atau unit kerja pada instansi pemerintah/pemerintah daerah yang tugas dan fungsinya menyelenggarakan kegiatan yang berkaitan dengan Dokumen Hukum pada :

1. Kementrian Negara
2. Sekretariat Lembaga Negara
3. Lembaga Pemerintahan Non Kementrian
4. Pemerintahan Provinsi
5. Pemerintah Kabupaten/ Kota
6. Sekretariat Dewan Perwakilan Rakyat Daerah Tingkat Provinsi dan Kabupaten/Kota

2.2 Web

Menurut Rohi A., 2015 dalam jurnal (Sistem Informasi Penjadwalan Dokter Berbassis Web Dengan Menggunakan *Framework Codeigniter*, 2017) web adalah sekumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet.

2.3 PHP (*Hypertext Preproccessor*)

Secara khusus PHP adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis (Umar, 2015). PHP termasuk dalam *Open Source product*, sehingga *source code* PHP dapat diubah dan didistribusikan secara

bebas. Salah satu keunggulan yang dimiliki PHP adalah kemampuannya untuk melakukan koneksi ke berbagai macam *software* sistem manajemen basis data atau *Database Management System* (DBMS), sehingga dapat menciptakan suatu halaman *web* yang dinamis.

2.4 MySQL

MySQL adalah sebuah bentuk *database* yang berjalan sebagai *server*, tidak meletakkan *database* tersebut dalam satu mesin dengan aplikasi yang digunakan, sehingga dapat meletakkan sebuah *database* pada sebuah mesin khusus dan dapat diletakkan ditempat yang jauh komputer pengaksesannya. MySQL merupakan *database* yang sangat kuat dan cukup stabil digunakan sebagai media penyimpanan data. MySQL adalah salah satu *database manajement system* (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, Postgre SQL, dan lainnya (Umar, 2015).

2.5 XAMPP

XAMPP adalah sebuah perangkat lunak yang biasa digunakan untuk membuat server sendiri pada PC atau Laptop. Xampp menggabungkan tiga aplikasi dalam satu paket yaitu Apache, MySQL, dan PHP. XAMPP bekerja secara offline, sehingga pengembang dapat menampilkan isi konten kepada orang lain, tanpa harus terhubung dengan internet.

2.6 Microsoft Visual Studio

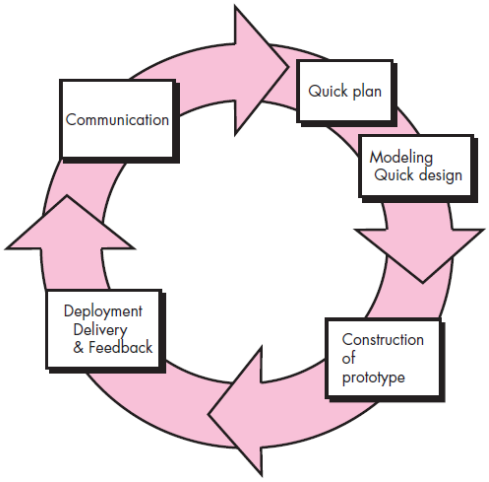
Microsoft Visual Studio adalah sebuah perangkat lunak yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console* aplikasi windows, ataupun aplikasi web. Visual Studio mencakup *compiler*, SDK, Integrated Development Environtment (IDE), dan dokumentasi umumnya berupa MSDN *Library*. Kompiler yang dimasukan kedalam paket Visual Studio antara lain : Visual C++, Visual Basic.NET, Visual Interdev, Visual J++, Visual J#, Visual Fox Pro dan Visual Source Safe.

2.7 Metode Pengembangan Prototyping

Menurut (Pressman 2012:50), dalam melakukan perancangan system yang akan dikembangkan dapat mengunakan metode prototype. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat yang akan dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna, kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelum diproduksi secara benar. Prototype bukanlah merupakan 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.

Berikut adalah tahapan dalam metode prototype:

a) Komunikasi dan pengumpulan data awal, yaitu analisis terhadap kebutuhan pengguna (dalam hal ini adalah pelanggan)
b) *Quick design* (desain cepat), yaitu pembuatan desain secara umum untuk selanjutnya dikembangkan kembali.
c) Pembentukan prototype, yaitu pembuatan perangkat prototype termasuk pengujian dan penyempurnaan.
d) Evaluasi terhadap prototype, yaitu mengevaluasi prototype dan memperhalus analisis terhadap kebutuhan pengguna.
e) Perbaikan prototype, yaitu pembuatan tipe yang sebenarnya berdasarkan hasil dari evaluasi prototype.
f) Produksi akhir, yaitu memproduksi perangkat secara benar sehingga dapat digunakan oleh pengguna.



Gambar 2.1 Metode *Prototype*

(Sumber Pressman, 2012:51)

2.8 Konsep MVC *(Model, View, Controller)*

Menurut Betha Sidik, 2015 dalam jurnal (Sistem Informasi Penjadwalan Dokter Berbassis Web Dengan Menggunakan *Framework Codeigniter*, 2017) framework adalah kumpulan intruksi-intruksi yang digunakan dalam classs dan function-function dengan fungsi masing-masing untuk memudahkan developer dalam memanggilnya tanpa harus menuliskan syntax program yang sama berulang-ulang, serta dapat menghemat waktu. Sedangkan codeigniter adalah salah satu jenis *framework* yang bersifat open source digunakan untuk membangun aplikasi php dinamis. Tujuan utama pengembangan codeigniter adalah untuk membantu *developer* untuk mengerjakan aplikasi lebih cepat dari pada menulis semua code dari awal. Codeigniter menyediakan berbagai macam *library* yang dapat mempermudah dalam pengembangan. Codelgniter diperkenalkan kepada publik pada tanggal 28 februari 2006. Codelgniter sendiri dibangun menggunakan konsep Model-View-Controller *development pattern* yang membagi aplikasi menjadi tiga bagian, yaitu model yang mengurusi interaksi antara aplikasi dan *database*, *view* yang mengurusi urusan logika pemrograman dan *controller* yang mengatur interaksi antara *view* dan model. Dalam konsep MVC ada tiga komponen yang saling berinteraksi didalamnya yaitu :

1. *View*, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh *controlle*r. *View* berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. Model, biasanya berhubungan langsung dengan database untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.
3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, *controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

2.9 Sistem Pemodelan

2.9.1*Unified Modelling Language* (UML)

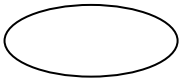
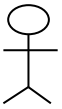
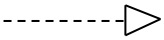
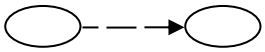
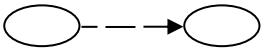
Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan gambar untuk memvisualisasikan, menspesifikasikan, membangun dan mendokumentasian sebuah sistem pengembangan perangkat lunak berbasis Objek. *Unified Modeling Language* (UML) bukanlah bahasa pemprograman tetapi model-model yang tercipta berhubungan langsung dengan berbagai macam bahasa pemprograman berorientasi obyek, seperti Java (Urva 2015). UML tersusun atas sejumlah elemen grafis membentuk diagram-diagram. Dalam penelitian ini melakukan desain hanya 2 diagram yaitu *Use Case* Diagram dan *Activity* Diagram.

2.9.1.1 *Use Case*

Use Case adalah diagram yang mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem yang akan dibuat. Diagaram *Use Case* digunakan untuk mengetahui fugsi apa saja yang ada di dalam sebuah sistem, dan siapa saja yang berhak menggunakan fungsi tersebut .

Berikut ini adalah simbol-simbol yang ada pada *Use Case* Diagram terdapat pada tabel 2.1

Tabel 2.1 Simbol *Use Case* Diagram

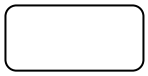
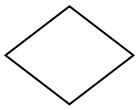
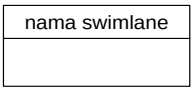
Simbol	**Keterangan**
Use Case 	Menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem.
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.
Asosiasi	Komunikasi antara *use case* dan aktor yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.
Generalisasi 	Sebagai penghubung antara aktor-*use case* atau *use case*-*use case*.
<<Include>> 	*Include Relationship* (relasi cakupan) : Memungkinkan suatu *use case* untuk menggunakan fungsionalitas yang disediakan oleh *use case* yang lainnya.
<<Extend>> 	*Extend Relationship* : Memungkinkan relasi *use case* memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh *use case* yang lainnya.

2.9.1.2 *Activity Diagram*

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana suatu aktivitas berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa kegiatan. Aktivitas

menggambarkan proses yang akan berjalan, sedangkan *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas (Syafitri 2016). Berikut ini adalah simbol-simbol yang ada pada *Activity* Diagram terdapat pada tabel 2.2

Tabel 2.2 Simbol *Activity* Diagram

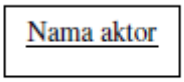
Simbol	**Keterangan**
Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan 	Asosiasi percabangan dimana ada pilihan aktivitas lebih dari satu.
Penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.

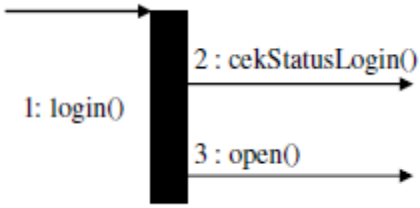
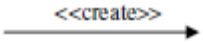
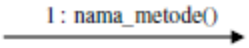
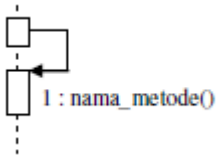
2..9.1.3 *Sequence Diagram*

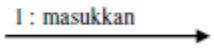
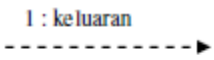
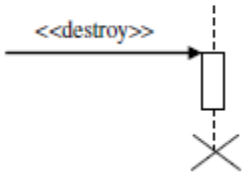
Sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirim dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui

objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuan sehingga semakin banyak usecase yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. (Sukamto dan Shalahuddin. 2019) Berikut adalah simbol-simbol yang ada pada diagram sekuen :

Tabel 2.1 *Sequence Diagram*

No	Simbol	Deskripsi
1	Aktor  Atau  Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.
2	Garis hidup/*lifeline* 	Menyatakan kehidupan suatu objek

3	Objek Nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan
4	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubungdengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya  Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif
5	Pesan tipe *create* 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6	Pesan tipe *call* 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka

		operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7	Pesan tipe *send* 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8	Pesan tipe *return* 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9	Pesan tipe *destroy* 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada *create* maka ada *destroy*

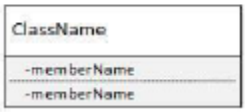
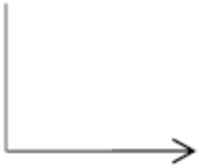
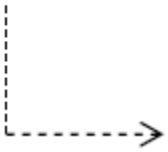
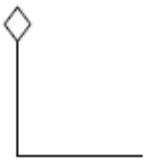
2.9.1.4 *Class Diagram*

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan method atau operasi. (Sukamto dan Shalahuddin. 2019) Berikut penjelasan atribut dan method :

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau method adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas :

Tabel 2.3 Simbol Class Diagram

No.	Simbol	Deskripsi
1	Kelas 	Kelas pada struktur sistem.
2	Antarmuka/*interface* 	Sama dengan konsep *interface* dalam pemrograman berorientasi objek.
3	Asosiasi/*association* 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*.
4	Asosiasi berarah/ *directed association* 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*.
5	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umumkhusus).
6	Kebergantungan/*dependensi* 	Relasi antar kelas dengan makna kebergantungan antar kelas
7	Agrgasi/*aggregation* 	Relasi antar kelas dengan makna semua-bagian (*whole-part*)

2.10 Penelitian Terkait

Berikut ini merupakan beberapa penelitian yang berkaitan dengan proposal skripsi ini :

Tabel 2.1 Penelitian Terkait

No.	Nama	Judul	Keterangan	Sumber
1.	Irawan Rio, Sulistyowati.	Implementasi Framework CodeIgniter Untuk Pengembangan Website Pada Dinas Perkebunan Provinsi Kalimantan Tengah	Penelitian ini membahas penggunaan Framework Codeigniter dalam mengembangkan aplikasi web pada Dinas Perkebunan	JURNAL SAINTEKOM STMIK PALANGKARAYA 07 (1) : 68-80 (2017)
2.	Wijaya K, Christian A.	IMPLEMENTASI METODE MODEL VIEW CONTROLLER DALAM RANCANG BANGUN WEBSITE SMK YAYASAN BAKTI PRABUMULIH	Penelitian ini membahas penerapan MVC untuk memudahkan rancang bangun website pada SMK Yayasan Bakti Prabumulih	JURNAL KOMPUTER DAN INFORMATIKA UNIVERSITAS BINA SARANA INFORMATIKA 21 (1) : 95-102 2019
3.	Pindoyono, Patan	PENGEMBANGAN SISTEM INFORMMASI	Penelitian ini membahas penggunaan	JURNAL PENDIDIKAN TEKNIK

		BERBASIS WEB MENGUNAKAN FRAMEWORK CODEIGNETER DI SMK NEGERI 1 JOGONALAN KLATEN	Framework Codeigniter dalam pengembangan web di SMKN 1 Jogonalan, Klaten.	ELEKTRONIKA (2017)